Receiver-based Protocol Enhancements for Wireless Ad-Hoc and Sensor Networks

Yuqun Zhang

Supervised by
Professor Wendi Heinzelman

A thesis
submitted in partial fulfillment
of the
Requirements for the Degree
Master of Science
in
Electrical Engineering

Department of Electrical and Computer Engineering
Arts, Sciences and Engineering
Edmund A. Hajim School of Engineering and Applied Sciences
University of Rochester
Rochester, NY

2010

Curriculum Vitae

The author was born in Jiangxi province, P. R. China in 1986. He joined Tianjin University in 2004, where he received a B.S. degree in Electronic and Information Engineering in 2008. Then, he began his graduate studies at the Department of Electrical and Computer Engineering at the University of Rochester in 2008. He is currently working towards his M.S. degree in the area of wireless communications and networking. His research interests include mobile ad hoc networks and wireless sensor networks.

Acknowledgements

I would like to start with thanking Professor Wendi Heinzelman, my research and thesis advisor throughout my Master's studies. Professor Heinzelman offered me valuable opportunities to get involved with research on wireless networks. She enlightened me to propose interesting research topics and encouraged me to solve research challenges, and her erudition, vision, as well as rigorous attitude in research impress and inspire me all along. Her amiability and optimistic life attitude sets a perfect example for my personality as well. Additionally, I especially thank her for strongly recommending me that I apply for PhD programs such that I can obtain the opportunity to continue my academic career. It is my great honor to work with her. In addition, I would like to thank Professor Alireza Seyedi for patiently helping me solve my confusions in and out of his class, and being willing to be my recommender for my PhD applications. I would also like to thank Professor Kai Shen for being my committee member and providing valuable suggestions on my research.

I would like to thank Dr. Ilker Demirkol for guiding me throughout my studies and research. His profound insight into research topics and expertise in solving problems reveal the standards of being a competent researcher and will inspire me for my future PhD studies. I would also like to thank Chen-Hsiang Feng, who is my teammate working in the same projects with me, my tutor offering endless help for my research and studies, and my brother carrying me all the time. I am very grateful to all my labmates who create a wonderfully ambient atmosphere where I feel friendship and harmony.

During the two years I have spent in Rochester, I am very much obliged to Yuhong Yao, Zhe Gao, Xiuyin He and all the other friends I have made here. Their unselfish support has helped me through all the difficulties and setbacks.

My final deep gratitude is for my parents, Mingzhan Zhang and Qianyi Mao. They always support my decisions on my personal journey, even when it means I would spend less time accompanying them. I could not have gone this far without their support. I am so lucky and proud to have such a wonderful family.

Abstract

Receiver-based protocols have been proposed as a means of allowing communication when nodes do not maintain any state information. In receiver-based protocols, receivers contend to be the next-hop router of a packet, and the transmitter selects the "best" receiver under a given optimality criteria to become the next hop for transmission. Receiver-based protocols are unique in wireless ad hoc and sensor networks in that they remove the need for costly state maintenance, and they do not require any synchronization of nodes' sleep-awake schedules for duty cycling.

In this thesis, I investigate the advantages of receiver-based protocols in multicast and convergecast communication, which are two important communication patterns in wireless ad hoc and sensor networks. Furthermore, I develop several enhancements to existing receiver-based protocols to improve performance. For multicast communication, I use the RBMulticast protocol, which is a stateless receiver-based multicast protocol that simply uses a list of the multicast members' (e.g., sinks') addresses, embedded in packet headers, to enable receivers to decide the best way to forward the multicast traffic. Using simulations, I investigate the performance of RBMulticast in terms of packet delivery ratio, average latency and average overhead, and I develop a new retransmission scheme to enhance the performance of RBMulticast.

For receiver-based protocols using convergecast communication, I develop novel duty cycle assignment algorithms. I model the expected energy consumption of nodes utilizing receiver-based protocols as a function of their duty cycle and expected traffic load, which can be derived from their distance to the sink node. Using this analysis, I determine the duty cycle that minimizes the expected energy dissipation for a given node distance to the sink. Moreover, I propose an adaptation for the derived distance-based duty cycle based on local observed traffic. Simulation results indicate that both methods have advantages over a traditional fixed duty cycle approach.

Contents

List of Figures

Chapter 1

Introduction

Energy efficiency is one of the main design goals for ad hoc and sensor networks due to the limited battery capacity of the wireless nodes. Communication and idle listening are major components of energy consumption in such networks. It is, therefore, crucial to minimize the number of packet transmissions performed for the execution of the application, and to reduce as much as possible the amount of idle listening without sacrificing the packet delivery ratio.

The most common forms of communication in wireless ad hoc networks are broadcast, where a source sends data to the entire network, multicast, where a source sends data to a subset of the nodes in the network called the multicast members, and unicast, where a source sends data to a single destination. Wireless sensor networks utilize these methods of communication as well as convergecast, where all the sensors (sources) send data to a single sink. In this thesis, I focus on multicast and convergecast traffic patterns.

To reduce the number of packets transmitted in multicast, a good protocol should provide short routing paths from the source to the multicast members, as well as efficiency in terms of the total number of links the packets traverse to get to all the multicast members, i.e., the packet should be split off to different routing branches only when necessary. Shorter routing paths lead to reduced packet delay, and improved efficiency leads to a reduction in the energy consumption from transmitting fewer packets. These two properties are usually contradictory to each other, and algorithms must make a trade-off to best fit their requirements.

In convergecast traffic patterns, data from all nodes are collected at a common sink. This creates a hot-spot in the network, where nodes closest to the sink must relay much more data than nodes far from the sink. This makes the traffic patterns in the network non-uniform, and this must be considered in convergecast routing protocols.

Receiver-based routing is a relatively new method of routing whereby receivers contend to be the next-hop node. This enables stateless routing and is ideal for dynamic networks due to either node mobility or node duty cycling, since the transmitter does not need to know which receivers are currently available. Receiver-based routing can be used for both multicast and convergecast communication.

Duty cycling, where a node is periodically placed into the sleep mode, is an effective method of lowering the time for idle listening and reducing energy dissipation in WSNs. The lower the duty cycle, the more nodes can sleep and the more energy they will save, but the fewer nodes are available to participate in data routing at any given time, which will increase transmission latency and decrease throughput. Thus, there is a trade-off between energy efficiency, transmission latency, and throughput, and hence it is important to set duty cycle appropriately.

1.1 Receiver-based Protocols

In a typical stateless receiver-based routing protocol, all receivers contend to be the next-hop router when they hear a packet route request, and the transmitter selects the receiver that is optimum according to some set criteria (e.g., the node that is closest to the sink) as the next hop. Specifically, one approach, considered in this thesis, is for the transmitter to initiate communication by sending an RTS packet that indicates the transmitter's location and the location of the sink. Nodes that hear the RTS packet first determine whether they make forward progress to the sink, and, if so, they calculate their distance to the sink. After a delay proportional to their distance to the sink, nodes send a CTS packet back to the transmitter. The first node that sends a CTS packet is selected as the next hop by the transmitter, and the transmitter forwards the data packet to that node.

1.2 RBMulticast

Building on existing unicast receiver-based protocols, RBMulticast [11] is a stateless cross-layer multicast protocol where no information except the individual multicast destinations' location information is required for successful packet delivery. No tree creation or maintenance or neighbor table maintenance is required, making it ideally suited for both static and dynamic networks. Packet routing, splitting packets into multiple routes, and the medium access of individual nodes rely solely on the location information of stationary destination nodes. The medium access method employed does not require any state information such as neighbor wake-up time or any a-priori operations such as time synchronization.

In RBMulticast, receivers contend for the channel based on their potential contribution to forwarding a packet, which is inspired by the cross-layer protocol XLM [1], a receiver-based unicast protocol designed for wireless sensor networks (WSNs). Nodes that make the most forward progress to the destination will contend earlier and hence have a higher chance to become the next-hop node. The multicast routing uses the concepts of "virtual nodes and "multicast regions for forwarding packets closer to the destination multicast members and determining when packets should be split into separate routes to finally reach the multicast members.

A new retransmission scheme based on [11] is proposed to improve the performance of packet delivery ratio, where retransmissions are spread to a duration larger than the maximum sleep duration of the nodes. In this thesis, I compare the performance of RBMulticast to that of the XLM [1] unicast protocol to show the performance gain achieved by RBMulticast. The results show that RBMulticast achieves much better performance in terms of latency and network traffic. It is also shown that RBMulticast achieves high packet delivery success rates even in highly dynamic networks, e.g., over 90% even when relay nodes move at speeds up to 30 m/s. Such high performance is not realizable for highly dynamic networks using other multicast approaches, since nodes must keep updated information about the network. RBMulticast is lightweight and robust, making it ideally suited for multicast applications in ad hoc networks such as Wireless Sensor Networks (WSNs) and Mobile Ad hoc Networks (MANETs).

1.3 Energy-Efficient Duty Cycle Assignment

Many sensor network applications require convergecast communication, where data from sensors are transmitted to a sink in the network. In this type of communication pattern, nodes close to the sink must transmit much more data than nodes far from the sink, and hence the duty cycles of the nodes should be adjusted appropriately to ensure energy efficiency while meeting traffic demands and keeping latency low.

In this thesis, I derive a mathematical model to determine the energy dissipation of a node as a function of its duty cycle and its distance to the sink for convergecast data patterns and receiver-based routing. Using this model, I find the optimal duty cycle as a function of node distance to the sink to minimize energy dissipation. Additionally, in order to balance energy efficiency and latency, I develop a traffic adaptive duty cycle approach that begins with the distance-based duty cycle and adapts the duty cycle based on current local traffic patterns observed by the node. In receiver-based protocols, the number of retransmitted RTS packets provides a direct indication of the traffic. Under heavy traffic, nodes must generate many retransmitted RTS packets. If the number of retransmitted RTS packets outnumbers the number of original RTS packets, nodes should increase their duty cycle in order to alleviate the traffic congestion; otherwise, they should decrease their duty cycle to save energy. This approach allows the duty cycle to be tuned to trade-off energy and latency for observed local traffic patterns.

Two duty cycle assignment methods are proposed in this thesis: Distance-based Duty Cycle Assignment (DDCA), where the optimal duty cycle is assigned to each node based on the node to sink distance, and Traffic-Adaptive Distance-based Duty Cycle Assignment (TDDCA), where the duty cycle is initialized to the one given by the DDCA method and adapted to the traffic as explained above. Simulation results show that DDCA and TDDCA reduce energy consumption compared with a network-wide constant duty cycle method. Additionally, TDDCA reduces latency at the expense of an increase in energy consumption compared with DDCA, which indicates that it is able to trade-off the lower latency of the network-wide constant duty cycle method and the energy efficiency of the distant-based duty cycle method (DDCA).

1.4 Thesis Organization

This thesis is organized as follows. In Chapter 2 I discuss the related work in this field. In Chapter 3, I describe the RBMulticast protocol, and I detail the updates I proposed to RBMulticast and provide discussion of the performance results comparing RBMulticast to the XLM unicast protocol. In Chapter 4, I describe the two methods for energy-efficient duty cycle assignment. Chapter 5 provides conclusions and future work that can be done in this area.

Chapter 2

Related Work

2.1 Receiver-based Routing

In receiver-based routing, receivers contend to be the next-hop router of a packet, and the transmitter selects the “best” receiver under a given optimality criteria to become the next hop for transmission. For example, in the receiver-based protocol Implicit Geographic Forwarding (IGF) [3], all receivers contend to be the next-hop router when they hear a packet route request, and the transmitter selects the receiver that is closest to the sink as the next hop. Specifically, the transmitter initiates communication by sending an RTS packet that indicates the transmitter’s location and the location of the sink. Nodes that hear the RTS packet first determine whether they make forward progress to the sink, and, if so, they calculate their distance to the sink. After a delay proportional to their distance to the sink, nodes send a CTS packet back to the transmitter. The first node that sends a CTS packet is selected as the next hop by the transmitter, and the transmitter forwards the data packet to that node.

Researchers have analyzed the performance of receiver-based routing through mathematical models [33] [32] and shown that receiver-based routing protocols perform well in terms of hop distance, energy and latency. Extensions to traditional receiver-based routing have included providing information about link quality for making routing decisions [1], and supporting multiple paths by strategically selecting relay nodes [26].

2.2 Multicast Routing

Existing multicast protocols for WSNs and mobile ad hoc networks (MANETs) generally use a tree to connect the multicast members [13, 15, 20, 25, 28, 30]. For example, the Takahashi-Matsuyama heuristic can be used to incrementally build a Steiner tree for multicast routing [6, 29]. Additionally, multicast algorithms rely on routing tables maintained at intermediate nodes for building and maintaining the multicast tree [7, 23]. ODMRP [20], CAMP [13] and PUMA [28] are suitable for high mobility rates when a large number of nodes are needed to forward data messages. MAODV [25], ADMR [15] and AMRIS [30] requires fewer nodes but more reconstructing trees for forwarding data messages.

In location-based approaches to multicast routing [2,9,19], nodes obtain location information by default as an application requirement (e.g., a home fire detection sensor would know where it is located) or as provided by a system module (e.g., GPS or a location-finding service). If location information is known, multicast routing is possible based solely on location information without building any external tree structure. For example, PBM [21] weights the number of next hop neighbor nodes and total geographic distance from the current node to all destination nodes and compares this to a predefined threshold to decide whether or not the packet should be split. PBM is a generalization of GFG (Greedy-Face-Greedy) [4] routing to operate over multiple destinations. GMR [27] selects neighbors based on a cost over progress framework integrated with greedy neighbor selection. Geocast [18] delivers multicast packets by restricted flooding. Nodes forward multicast packets only if they are in the Forwarding Zone calculated at run time from global knowledge of location information.

2.3 Duty Cycling

Many protocols, including receiver-based routing and multicast routing, use duty-cycling to save energy. Using duty-cycling, nodes follow a sleep-wake cycle, making them unavailable for routing when they are in the sleep cycle. Typical duty-cycling approaches consider a fixed network-wide duty cycle for all nodes. However, newer approaches have looked into adapting the duty cycle. For example, adapting the duty cycle to the local traffic was proposed in PMAC [31], where the sleep-wakeup schedule is represented by a string of bits that are updated each period using local traffic information available at the node. These schedules are exchanged at the end of each period so neighboring nodes are aware of each others' schedules. Another adaptive duty cycle approach, ALPL, adjusts a node's duty cycle according to the node's neighbors' duty cycles in order to support the data flows it receives [16]. [22] dynamically controls the duty cycle so that the target rate of transmitted packets is reached, while the consumed energy is minimized. However, none of these approaches provide optimize the duty cycle for convergecast data patterns and receiver-based routing.

Chapter 3

RBMulticast Enhancements and Performance

3.1 RBMulticast Protocol Description

RBMulticast is a receiver-based cross-layer protocol that performs multicast routing based on multicast members' location information. In RBMulticast, the medium access probabilities of nodes are decided based on their potential contribution to forward the packet. The receiver-based MAC only needs the sender node's location and the destination node's location, which are provided in the MAC packet, to decide the next hop along the route. It is assumed that the "void" (hole) problem in geographic routing is solved implicitly, for example, using the right-handed rule as in GPSR [17].

3.1.1 RBMulticast Overview

Nodes in RBMulticast create "multicast regions" centered around themselves. There are several ways to create these regions (see Section 3.1.2), but for simplicity it can be assumed that each multicast region corresponds to one quadrant of the network, for a grid centered at the node.

When a user initiates a request to send a packet to a multicast group, data is passed down to the RBMulticast module in the protocol stack. Once the RBMulticast module gets this packet, it retrieves the group list from its group table, assigns the group nodes to the multicast regions based on their locations, and using these locations, calculates a "virtual node" location for each multicast region. RBMulticast replicates the packet for each multicast region that contains one or more multicast members and appends a header consisting of a list of destination nodes (multicast members) in that region. The destination of a replicated packet is the "virtual node" of the corresponding multicast region, which can be determined in several ways (see Section 3.1.4), e.g., as the geometric mean of the locations of all the multicast members in that multicast region. In the end, all packets for all multicast regions are inserted in the MAC queue, and are then broadcasted to the neighborhood. The node closest to the virtual node (within the available relay nodes as determined by receiver-based contention at the MAC layer) will take responsibility for forwarding the packet. The procedure for transmitting packets is summarized in pseudo code in Algorithm 3.1.1.

When a node receives a multicast packet, it drops the packet if it is not in the forwarding zone. The forwarding zone is the area within the radio range of the sender that has a smaller distance to the destination than the sender-destination distance.

If the node is in the forwarding zone, it then retrieves the destination node list from the RBMulticast packet header. If this node is inside the destination list, it removes itself from the list and passes a copy of the packet to the upper layers in the protocol stack. Finally, if there still remain nodes in the destination list, multicast regions and virtual nodes are recalculated, and new packets are generated if required. The packets (one per multicast region that contains multicast members)

Algorithm 3.1.1 RBMulticast Send

Require: Packet output from upper layer
Ensure: Packets inserted to MAC queue

```
Get group list N from group table
for node n in group list N do
    for multicast region r in 4 quadrants regions R do
        if n ∈ r then
            Add n into r.list
        end if
    end for
end for
for r ∈ R do
    if r.list is non-empty then
        Duplicate a new packet p
        Add RBMulticast header (r.list) to p
        Insert p to MAC queue
    end if
end for
```

are then inserted in the MAC queue for transmission. The procedure executed after receiving packets is summarized in pseudo code in Algorithm 3.1.2.

Fig. 3.1 gives an example of how RBMulticast is employed. The two multicast regions, the south-west and north-west quadrants, contain only one multicast member each, and thus a packet is sent directly to these multicast destinations. The north-east multicast region has three multicast members, and thus a single packet is sent to the virtual node located at the geometric mean of the locations of the multicast members (dotted circle with label 3 in the figure). The south-east multicast region has no multicast members, and hence no packet is transmitted into this region. Once a packet sent towards a virtual node reaches an intermediate node for which the multicast members are no longer in the same multicast region, the node will split off packets to each of the multicast regions accordingly.

Algorithm 3.1.2 RBMulticast Receive

Require: Packet input from lower layer
Ensure: Forwarded packets inserted to MAC queue

```
Drop packet if not in Forwarding zone
Get destination list D from packet header
for node d in destination list D do
    if I am d then
        Duplicate the packet and input to upper layer
        Remove d from list D
    end if
end for
for d ∈ D do
    for multicast region r in 4 quadrants regions R do
        if d ∈ r then
            Add d into r.list
        end if
    end for
end for
for r ∈ R do
    if r.list is non-empty then
        Duplicate a new packet p
        Add RBMulticast header (r.list) to p
        Insert p to MAC queue
    end if
end for
```

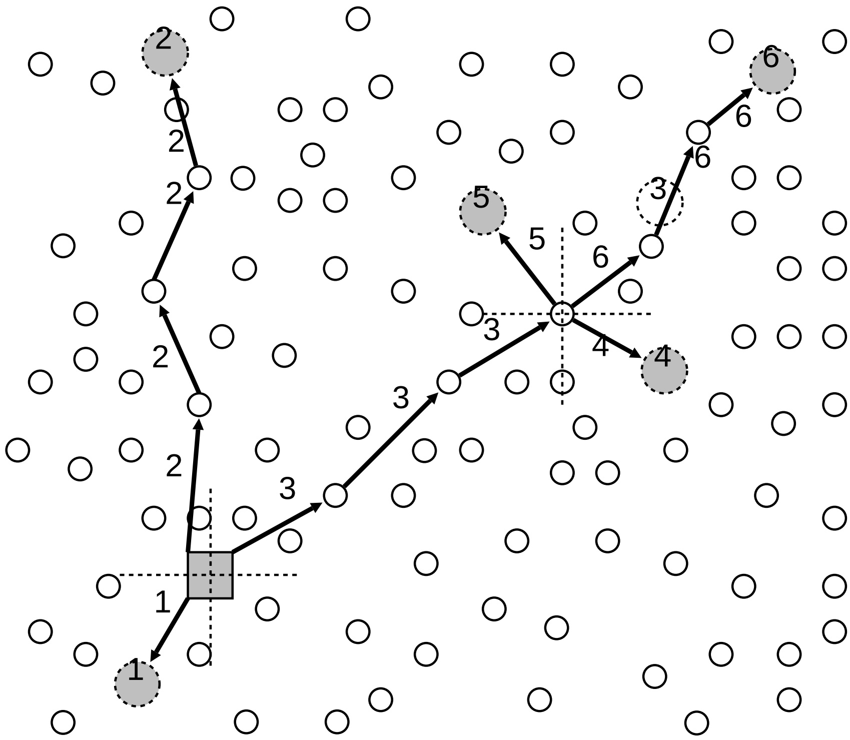

Figure 3.1: Example showing how RBMulticast delivers multicast packets. The source node is the square node. Multicast members are shaded circles, and virtual nodes are dotted circles. Because every destination node will become a virtual node at the end, they are all shown with dotted circles. The number on the side of the lines indicate the destination of that packet.

3.1.2 Multicast Regions

Once a node receives a multicast packet (from the application layer or from a previous hop node), it divides the network into multicast regions, and it will split off a copy of the packet to each region that contains one or more multicast members. The division of the network into multicast regions considered in this thesis is shown in Fig. 3.2.

3.1.3 Packet Splitting

Algorithms 3.1.1 and 3.1.2 describe RBMulticast method that splits packets at relay nodes for which the multicast destinations reside in different regions. This method is used in the protocol description due to its simplicity.

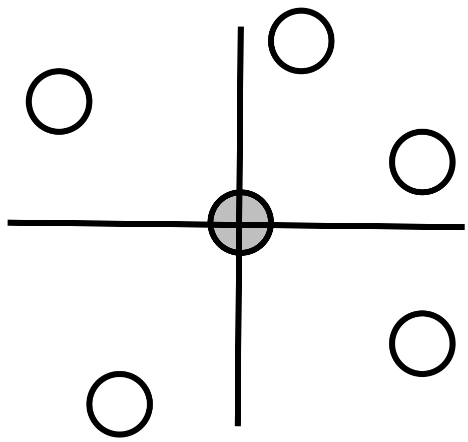

Figure 3.2: Dividing the space into four quadrants.

3.1.4 Virtual Node

In RBMulticast, because there is no assumption of knowledge of neighbor nodes and no routing tables are assumed, the node determines a "virtual node" located at the geographic mean of the multicast members for each multicast region. This virtual node is used as an imaginary destination for the multicast packet in that region. The virtual nodes are not necessarily reachable or even physically exist, as illustrated in Fig. 3.1. The idea behind that is even if a virtual node does not exist, the nodes can still find a route using the assumed receiver-based MAC protocol to get the packet closer to the location of the virtual node. On the other hand, when using the nearest multicast node as the destination, all node addresses physically exist and virtual nodes are not necessary.

3.2 RBMulticast Performance Enhancements

I implemented RBMulticast in the OPNET Simulator [24] to investigate its network performance with the underlying MAC protocol in both static and mobile scenarios. Results of detailed packet-level OPNET simulations with high densities showed that RBMulticast suffers from packet collisions when packets are split. The reason for these collisions and a MAC-level improvement I proposed for reducing the effect of collisions is described in this section.

3.2.1 Split Packet Contention Problem in RBMulticast and the Proposed Solution

RBMulticast requires splitting the packet at a node, if the locations of the multicast members, which are listed in the header of the packet, reside in different regions for that node. The packet splitting creates replicas of a packet, updating the destination node locations in each packet accordingly. After replication, all the packets generated are immediately inserted into the node's buffer to be transmitted. However, this creates a burst of packet traffic and congestion within the transmission range of the splitting node, since the relay nodes receiving the packets will contend with the splitting node with the remaining split packets. The problem is more severe for large interference-to-transmission range ratios, because of the higher number of relay nodes contending with the splitting nodes and with each other.

To decrease the contentions and the possibility of collisions after packet replications, I first define a transmission order for the replicated packets based on the region. For example, the packets destined to the northeast region are transmitted first, the ones destined for the northwest region are transmitted second, etc. Then, a certain splitting packet time interval (SPTI) is used between transmissions destined to different regions. One disadvantage of delaying the packet transmissions by the duration SPTI is the increase in the end-to-end delay, i.e., the latency. To determine a reasonable value for SPTI that achieves a high packet delivery ratio with an acceptable latency, I conducted simulations with 5 sinks uniformly distributed along the edge of the scenario and 200 nodes with 100% duty cycle. The effect of SPTI is shown in Table 3.1. It is clearly seen that the packet delivery ratio increases when SPTI is used compared to not using it. An SPTI value of 80 ms can

SPTI (sec)	0	0.05	0.08	0.125	0.25
PDR	0.75	0.88	0.94	0.95	0.95

Table 3.1: Packet delivery ratio for different SPTI.

achieve almost as high packet delivery ratio as that with high SPTI. That means a low latency can still be achieved using SPTI without sacrificing the packet delivery ratio performance. The threshold for the performance gain (80 ms in this case) depends on the transmission range, interference range, the time for completing one hop transmission, and the number of replicated packets, which is a function of the distribution of the multicast members. One alternative is to introduce delay not between transmissions of the replicated packets, but before an intermediate node starts forwarding packets.

3.2.2 MAC-level Improvements

To achieve a high packet delivery ratio, at least one relay candidate should be awake and listening to the channel during an RTS packet transmission or any of its retransmissions. Each RTS transmission requires a preceding random backoff and, for the case of no CTS reply, the CTS timeout duration. To guarantee reaching a relay candidate if one exists, all retransmissions should be spread to a duration larger than the maximum sleep duration of the nodes, i.e.,

$$t_{sleep} \leq Retx_{max} \times (t_{backoff} + timeout_{CTS}) \tag{3.1}$$

where t_{sleep} is the sleep duration, $Retx_{max}$ is the maximum number of RTS retransmissions, $t_{backoff}$ is the expected backoff time for contention, and $timeout_{CTS}$ is the timeout duration a node will wait to receive a CTS. $t_{backoff}$ can be calculated as

$$t_{backoff} = E[SlotNumber] \times t_{slot} \tag{3.2}$$

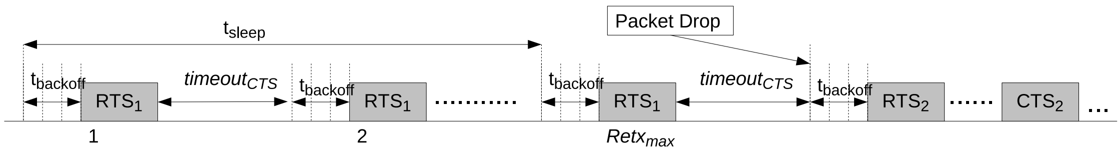

Figure 3.3: Adjustment of $timeout_{CTS}$ for MAC improvement.

where $SlotNumber$ is the selected backoff slot number, and t_{slot} is the duration of a slot. The idea of spreading RTS retransmissions to a duration larger than t_{sleep} is illustrated in Fig. 3.3.

Simulations with 200 nodes randomly distributed and 5 sinks uniformly distributed along the edge of the scenario are conducted to evaluate the MAC-level improvements, compared with the retransmission method where a maximum retransmission time is equal to 10% of the listening interval. Fig. 3.4 shows that a duty cycle as low as around 20% in the MAC-level improvements can achieve almost the same high packet delivery ratio as high duty cycles, and it performs better than the fixed retransmission time method when the duty cycle is low.

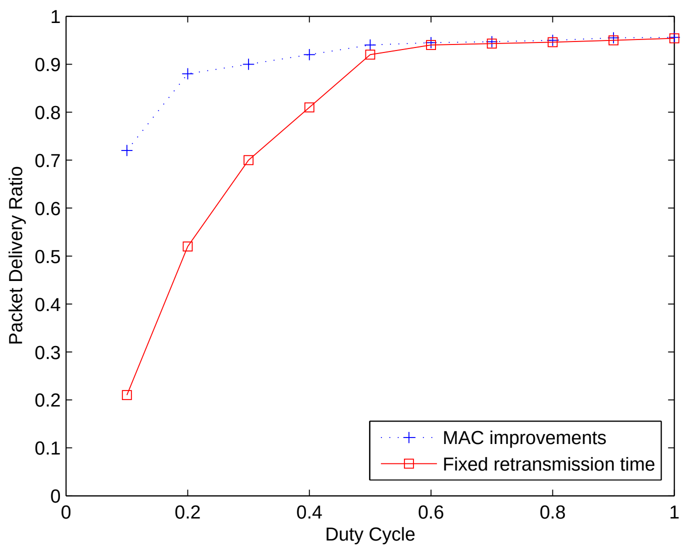

Figure 3.4: Packet delivery ratio under MAC improvements.

The parameter $timeout_{CTS}$, which is enlarged by 10 times compared with the default value of the implementation of the original RBMulticast, is used as the duration between the end of an RTS transmission and its retransmission when no CTS reply is received for the previous RTS. Thus, the backoff duration for CTS contention of the relay candidates is enlarged by 10 times correspondingly, which allows relay candidates to have more time listening to the channel and to have their schedule canceled if they hear an ongoing CTS transmission.

3.3 RBMulticast Performance Evaluation

I define multiple scenarios for RBMulticast simulations to evaluate the three performance metrics: packet delivery ratio, latency and the average traffic generated to transmit one data packet to all multicast members. In all scenarios, the area is a $150m \times 150m$ square. The transmission range is $30m$ and the interference range is approximately $80m$. The channel data rate is $220Kbps$. The length of RTS, CTS, and ACK packets is 78 bits and of raw data packets is 400 bits. The SPTI is set to 0.1 s, and the maximum number of retransmissions is set to 25. Each parameter set is evaluated with 10 simulation runs whose averages are displayed in the figures.

It is difficult to compare RBMulticast with existing multicast routing protocols, as any protocol that requires state maintenance will not be able to function in the dynamic environments tested here, such as with node duty cycles as low as 20% and node mobility as high as 30 m/s. Hence, I compared RBMulticast to using stateless unicast protocols to send the packets individually to each multicast member.

Two unicast protocols are compared to illustrate the advantage of RBMulticast. The first protocol is Unicast based on the original XLM MAC, which is denoted as UOX, and the second protocol is Unicast based on the improved XLM MAC, denoted by UIX, where the MAC-level improvements proposed in Section 3.2.2 are applied. Both unicast protocols are run with 100% duty cycle and are compared to

RBMulticast with 20%, 60% and 100% duty cycle.

3.3.1 Static Nodes, Five Sinks

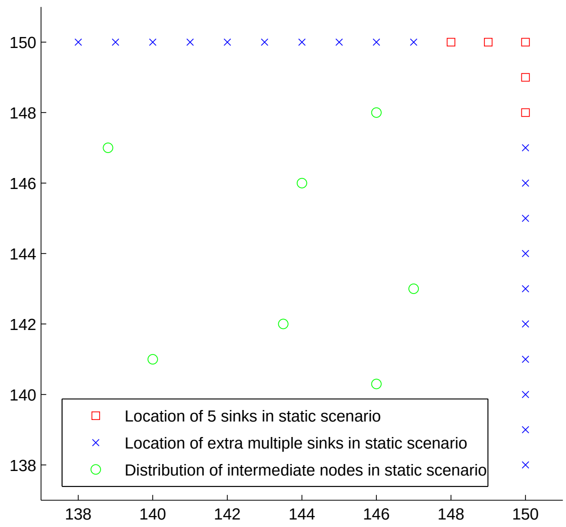

Figure 3.5: Multicast member (sink) locations in the simulation. The additional members align along the boundaries for different simulations. Note that this figure shows a blow-up of the north-east quadrant of the simulation area.

The first set of simulations are investigated to evaluate the performance of RBMulticast using static nodes with the source located at (0, 0) and 5 sinks located at the edge of the target area, as shown by the rectangular nodes in Fig 3.5. The average packet delivery ratios observed for varying numbers of nodes are shown in Fig. 3.6. As seen in the figure, the packet delivery ratio is very low for a small number of nodes, which is due to the high probability of holes in the network. When there are no holes in the area, which is achieved with high density, the packet delivery ratio is close to 100% for RBMulticast, independent of the duty cycle value. This interesting result is due to the improvements proposed for the MAC: SPTI efficiently reduces the contention of multiple splitting packets, and the extended CTS timeout enables finding a relay node, even when the nodes spend much time sleeping. It is also shown that the packet delivery ratio is not reduced as density increases, which is usually the case due to multiple CTS replies causing congestion in the network.

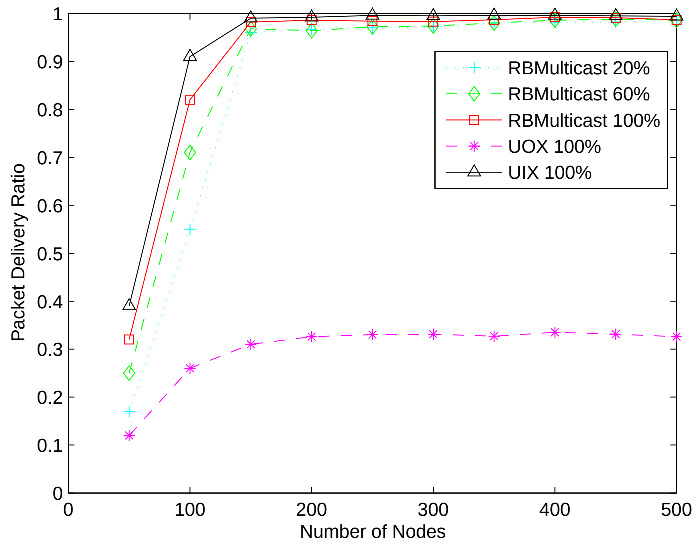

Figure 3.6: Packet delivery ratio vs. number of nodes. (static nodes, 5 sinks)

Fig.3.6 shows that UIX performs slightly better than RBMulticast in terms of packet delivery ratios, and both perform much better than UOX. For UIX, packets are deferred before their transmission due to the timer on the buffer in XLM. Therefore, before the transmitter node's timer reaches its timeout to send the next packet to a different multicast member, it is quite possible that the previously transmitted packet has reached the destination node and thus will not create interference for the new packet. With a reasonable node density such that no holes exist in the network, the packet delivery ratio is expected to be high as shown in Fig. 3.6. Although the same improved MAC guarantees the packet delivery ratio performance of RBMulticast to be high, due to the fixed value of SPTI, multiple replicated packets still exist in the network and successively contend for the channel within a short period of time. As a result, more back-offs or collisions occur than that of UIX and contribute to the similar but slightly worse performance. For UOX, once the burst of packets is inserted into the buffer of the MAC layer with no time interval before the transmission attempts, the relay nodes receiving packets would contend with the remaining packets in the buffer of the source node and lead to a low packet delivery ratio.

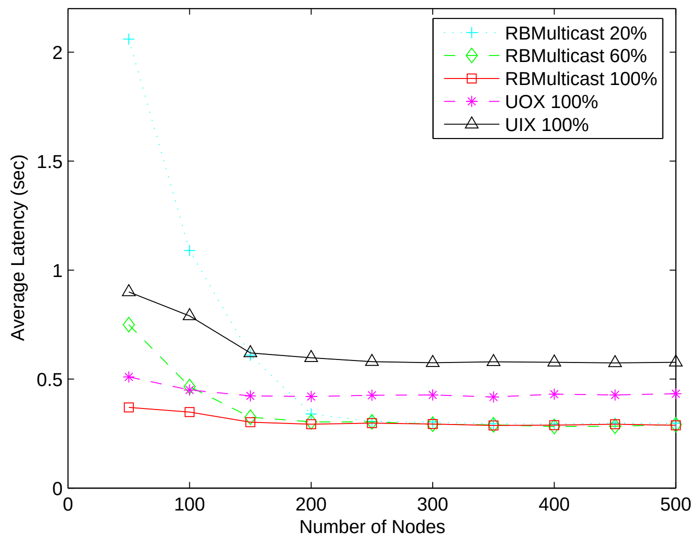

Figure 3.7: Average latency vs. number of nodes. (static nodes, 5 sinks)

Fig. 3.7 shows the latency as a function of the number of nodes. Under low duty cycle and low node density of RBMulticast, since the sleeping times are not synchronized, it is very possible that no relay node candidate can be found in the first attempt, and multiple retransmissions are needed to find a relay node. As the duty cycle and the density increase, more relay node candidates are available and fewer retransmissions are needed, which leads to a decrease in the latency. Fig. 3.7 also shows that for low density values, the average latency is high for all three protocols. With an increase in the density, the average latency becomes constant. Since RBMulticast reduces the total number of transmissions to reach all multicast members, the average latency is lower than the other two protocols. Having more time for retransmissions in the improved MAC layer, UIX has a higher average latency than UOX.

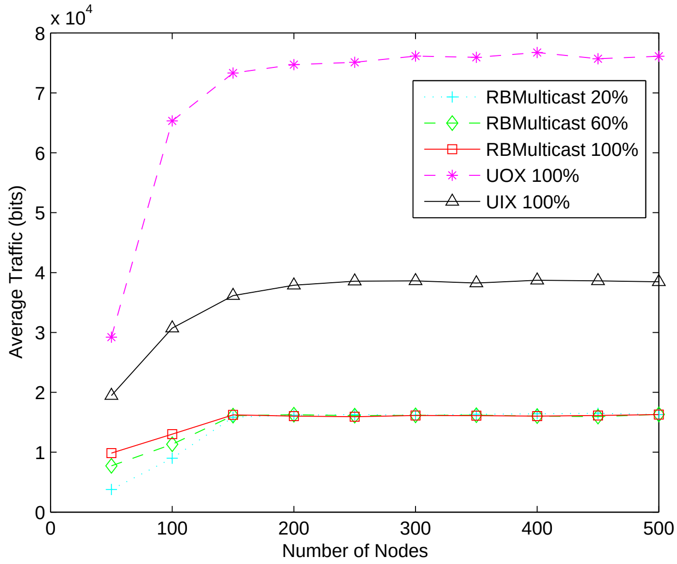

Figure 3.8: Average traffic for transmitting one data packet vs. number of nodes. (static nodes, 5 sinks)

The average traffic generated to transmit one data packet to all multicast members is shown in Fig.3.8. The value is calculated by dividing the total traffic generated to transmit one data packet (RTS/CTS/DATA/ACK) by the packet delivery ratio. Since RBMulticast requires fewer packet transmissions, it generates the least traffic for the delivery of a data packet among the three methods (under 100% duty cycle, UIX generates average traffic roughly 2.3 times compared with RBMulticast). By having fewer retransmissions due to the advantage of the improved MAC, UIX generates less traffic than UOX. In low densities, more retransmissions occur and more packets are dropped because no relay node is found for forwarding. Hence, the average traffic for successfully transmitting one packet to all multicast members is higher than that of higher densities. Also the fact that the average traffic for the three different duty cycles does not differ significantly is due to the improved MAC where not many retransmissions are needed to accomplish the delivery of a data packet, even with low duty cycle.

3.3.2 Mobile Nodes, Five Sinks

The second set of simulations are performed to investigate the performance of RB-Multicast in mobile scenarios. All intermediate nodes move according to the Random Waypoint mobility model with a certain speed. The source and multicast members are moved inward 25m as compared to Fig. 3.5 to avoid the issues with the "cluster into the middle" effect of the Random Waypoint model [12,14]. A duty cycle of 100% is investigated for three different numbers of nodes: 100, 200 and 300.

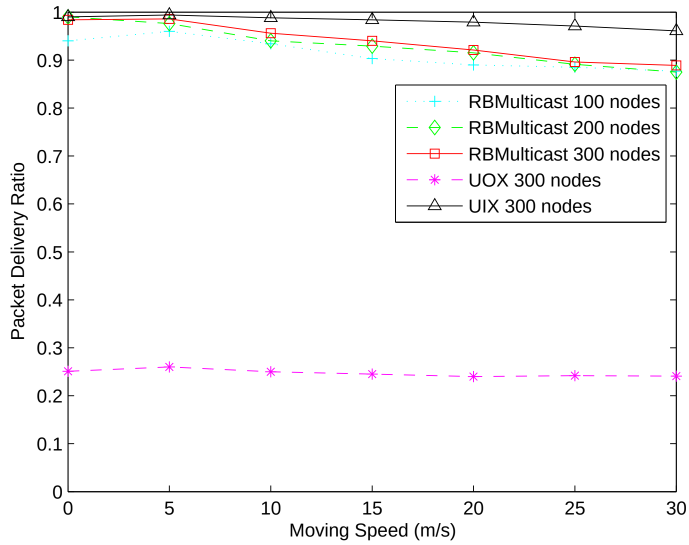

Figure 3.9: Packet delivery ratio vs. number of nodes. (mobile nodes, 5 sinks)

Fig. 3.9 shows the packet delivery ratio as a function of mobile speed. Note that the data points corresponding to 0 m/s show the performance of static networks. All three curves indicate that when the intermediate nodes are moving at low speeds and the node density is low, the performance is slightly better than that when they are static. The reason is that the "empty holes" that exist in the static scenario when the density is low, can be eased when the nodes move into the "empty holes" and become relay candidates. When nodes move fast, more link breaks can occur because a receiver moves out of the transmission range of the transmitter. Fig.

3.9 shows that UIX performs the best among the three protocols and RBMulticast performs better than UOX due to the same reason as that of the static scenario.

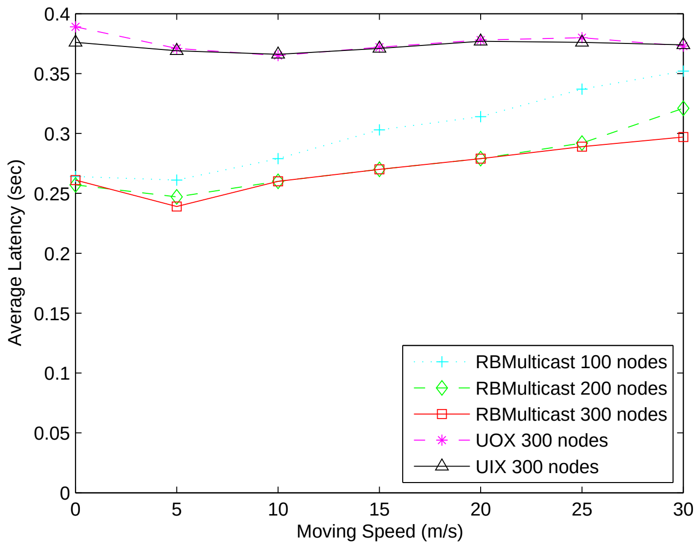

Figure 3.10: Average latency vs. number of nodes. (mobile nodes, 5 sinks)

Fig. 3.10 shows the average latency as a function of mobile speed. When density is increased, less time is required to finish the transmission. As seen in the figure, RBMulticast has the least latency among the three protocols, for the same reason as in the static scenario.

Fig. 3.11 shows the average traffic generated to transmit one data packet as a function of mobile speed. When the speed of mobile nodes increases, the average traffic generated per transmission becomes higher due to the increase in the number of retransmissions caused by more link breaks. Note that RBMulticast has the least traffic since it requires the smallest number of hops among the three protocols (under the same number of nodes, UIX generates average traffic of 1.7 to 2.2 times that of RBMulticast).

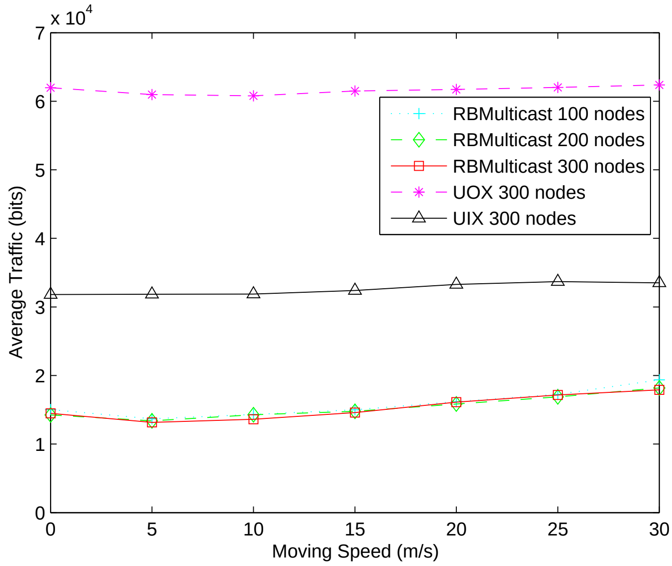

Figure 3.11: Average traffic for transmitting one date packet vs. number of nodes. (mobile nodes, 5 sinks)

3.3.3 Static Nodes, Varying Number of Sinks

To further test the robustness of RBMulticast, the third simulation scenario evaluates the performance in terms of the number of sinks when all nodes are static. 300 nodes are deployed in all scenarios. Sinks are located around the upper-right corner as shown in Fig. 3.5.

Fig. 3.12 shows the packet delivery ratio as a function of the number of sinks. With an increase in the number of sinks, all three protocols have lower packet delivery ratios. In RBMulticast, because more sinks require more replicas of packets, fiercer contention occurs for the channel. Therefore it is expected that the packet delivery ratio will decrease with an increase in the number of multicast members. As seen in Fig. 3.12, RBMulticast is more robust to an increase in the number of multicast members. Above 20 members, RBMulticast gives better packet delivery ratio compared to UIX. It is expected that with an increase in the number of sinks, the advantage of RBMulticast over UIX will be larger.

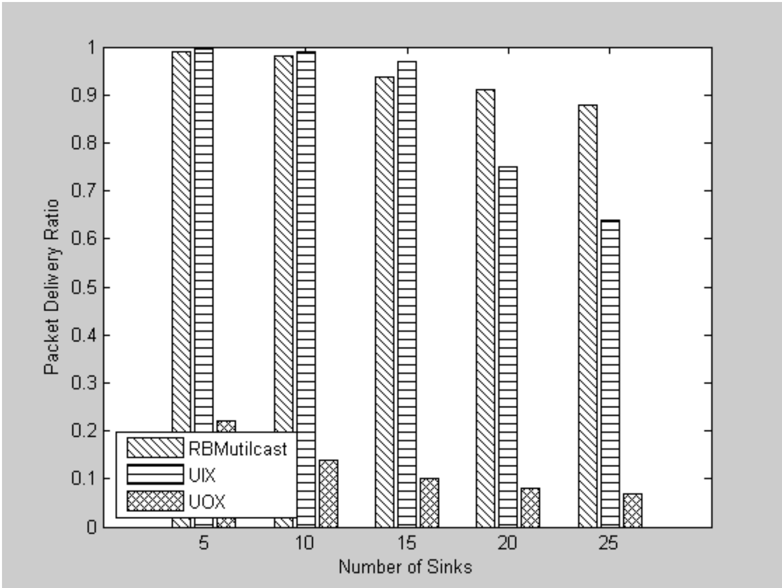

Figure 3.12: Packet delivery ratio vs. number of sinks. (static scenario, 300 nodes)

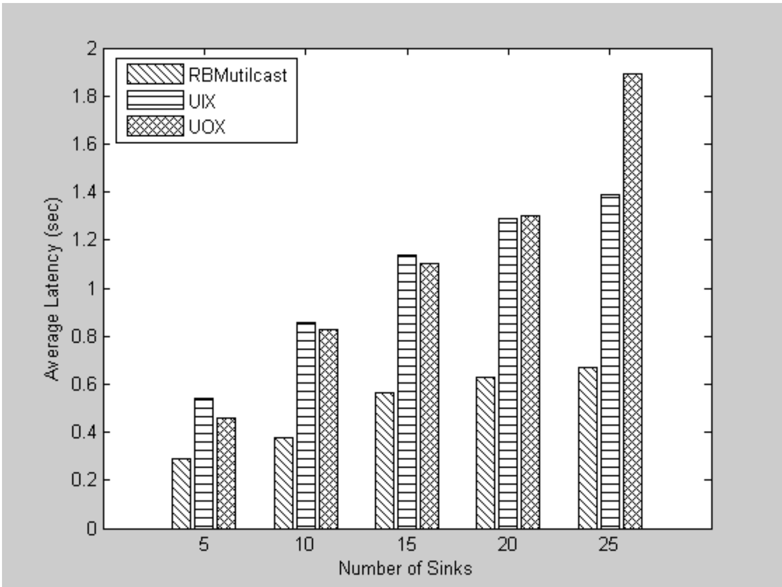

Figure 3.13: Average latency vs. number of sinks. (static scenario, 300 nodes)

Fig. 3.13 shows the average latency as a function of the number of sinks. More replicated packets lead to more contention and retransmissions, which results in higher latency when the number of sinks increases in RBMutlicast. However, since packets in RBMulticast travel through fewer hops, RBMulticast latency is much lower than the other two protocols. Note that when the number of sinks is small, UOX performs better than UIX due to a lower maximum retransmission count, while when the number of sinks increases, the MAC improvement reveals its advantage by reducing the latency through avoiding contentions.

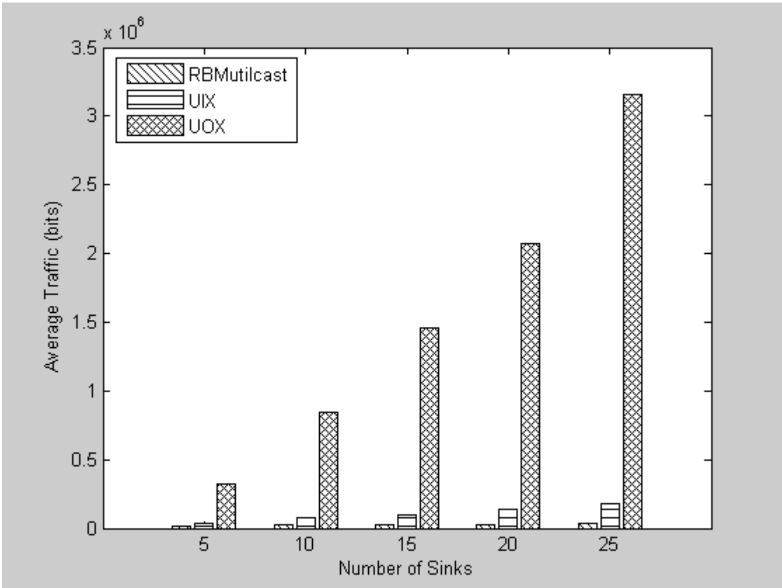

Figure 3.14: Average traffic for transmitting one data packet vs. number of sinks. (static scenario, 300 nodes)

Fig. 3.14 shows the average traffic generated to transmit one data packet as a function of the number of sinks. Because unicast needs to send separate packets to each sink, many paths are repeated and redundant. Hence, with the increase of the number of sinks, RBMulticast possesses a greater advantage in terms of average traffic. Note that due to the large number of retransmissions with the original XLM MAC, UOX always has a much larger traffic generation than the other two protocols, which verifies the necessity of the MAC improvement.

3.3.4 Mobile Nodes, Varying Number of Sinks

The fourth set of simulations are applied to test the robustness of RBMulticast for varying numbers of sinks under mobile scenarios. The source node is located at (25,

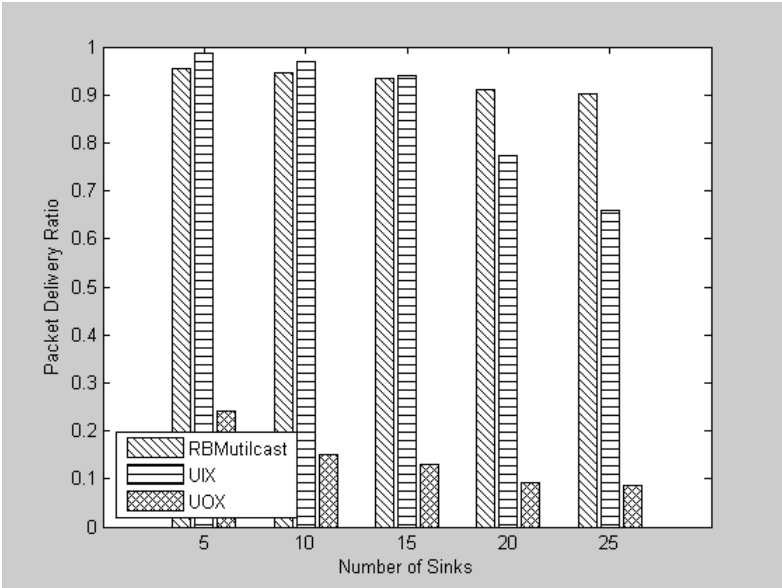

Figure 3.15: Packet delivery ratio vs. number of sinks. (mobile scenario, 300 nodes)

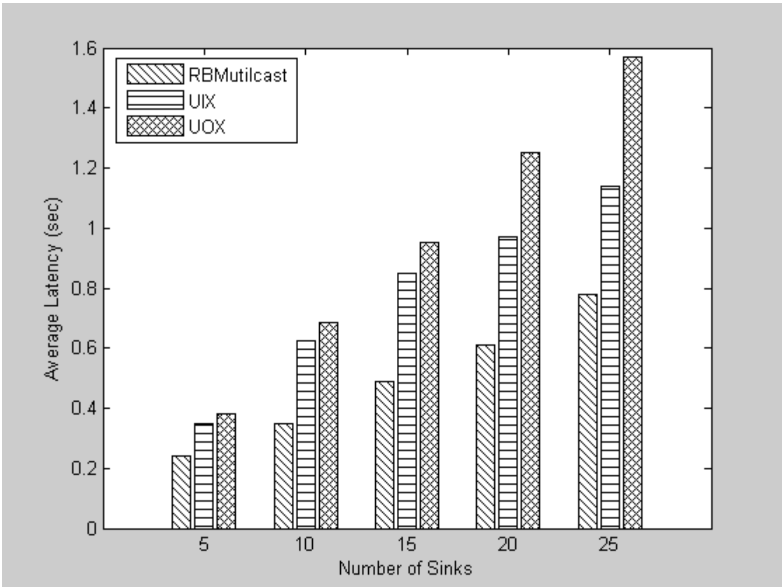

Figure 3.16: Average latency vs. number of sinks. (mobile scenario, 300 nodes)

25). Sinks are located around the upper-right corner of the inner $100m \times 100m$ area with the distance of $1m$ between the adjacent sinks as the rectangular and cross nodes seen in Fig. 3.5. 300 intermediate nodes move with a speed of 20 m/s inside the $150m \times 150m$ area. The Random Waypoint model is applied. The duty cycle

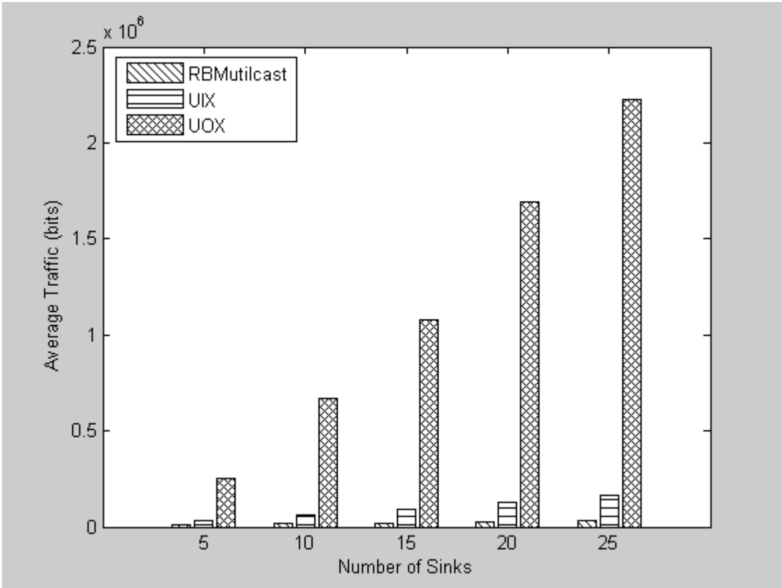

Figure 3.17: Average traffic for transmitting one data packet vs. number of sinks. (mobile scenario, 300 nodes)

investigated is 100%. Fig. 3.15, Fig. 3.16 and Fig. 3.17 show a similar performance as that of the static scenario.

3.3.5 Uniformly Distributed Sinks, Mobile Networks

A new scenario is developed to further illustrate the performance of RBMulticast. 300 mobile intermediate nodes are randomly deployed in a $150m \times 150m$ scenario with a moving speed of $10m/s$. Sinks are uniformly distributed along the upper edge and right edge (e.g., as in [5]) of the $100m \times 100m$ inner area with the source located in $(25, 25)$, i.e., at the lower-left corner of the inner area.

Fig. 3.18 shows that when the number of sinks are 10 and 15, RBMulticast performs worse compared with the previous scenario, and UIX possesses a larger advantage in packet delivery ratio. This is because when multicast members are sparsely distributed, RBMulticast requires splitting more packets, leading to more contentions.

Fig. 3.19 and 3.20 show that RBMulticast's advantage over UIX in average la-

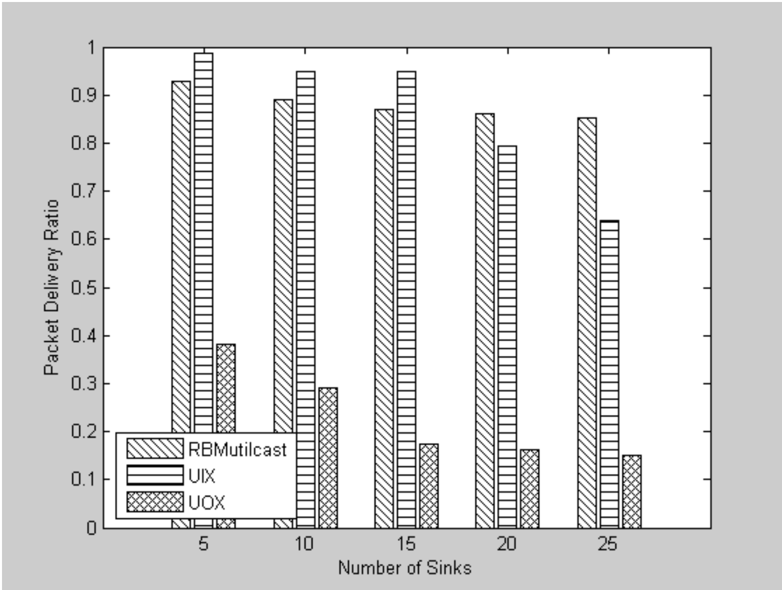

Figure 3.18: Packet delivery ratio vs. number of sinks. (mobile scenario, 300 nodes)

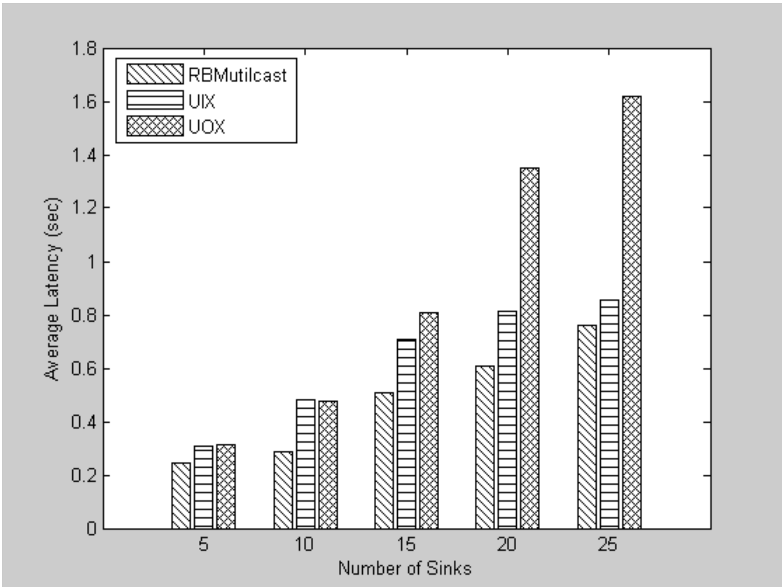

Figure 3.19: Average latency vs. number of sinks. (mobile scenario, 300 nodes)

tency and average traffic is not that evident compared with the previous scenario because more contention leads to a larger delay, and the decrease of the packet delivery ratio directly increases the average traffic to successfully transmit one packet to all multicast members.

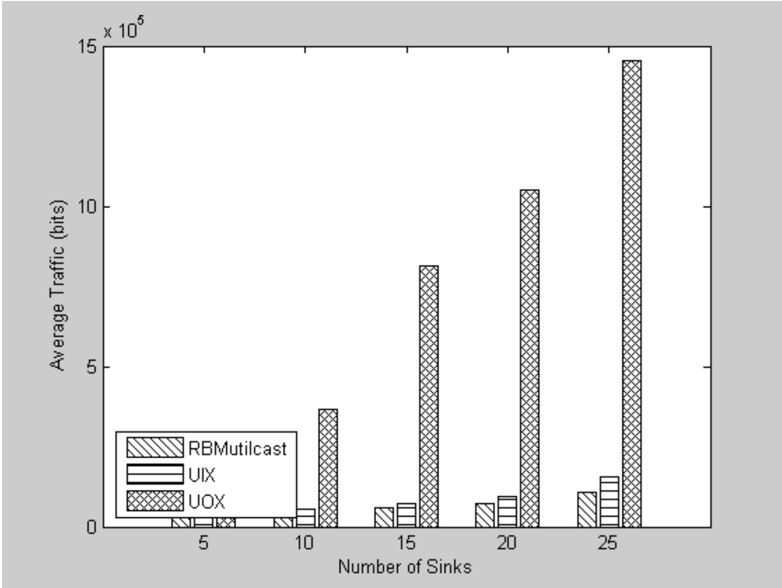

Figure 3.20: Average traffic for transmitting one data packet vs. number of sinks. (mobile scenario, 300 nodes)

These simulation results shows that the performance of RBMulticast is tightly connected with the location of the sinks. Generally, RBMulticast has a larger advantage compared with unicast when sinks cluster than when they are sparsely distributed.

3.4 Conclusion

In this chapter, I overview a stateless cross-layer multicast protocol: RBMulticast. After proposing a new retransmission scheme, I evaluate the performance of RB-Multicast in terms of packet delivery ratio, average latency and average traffic. Compared with the XLM multiple unicast, simulation results show that RBMulticast achieves much better performance in terms of latency and network traffic. It is also shown that RBMulticast achieves high packet delivery success rates even in highly dynamic networks.

Chapter 4

Energy-efficient Duty Cycle Assignment Methods

In this chapter, two duty cycle assignment methods are proposed: Distance-based Duty Cycle Assignment (DDCA), where the optimal duty cycle is assigned to each node based on the node to sink distance, and Traffic-Adaptive Distance-based Duty Cycle Assignment (TDDCA), where the duty cycle is initialized to the one given by the DDCA method and adapted to the traffic.

4.1 Distance-based Duty Cycle Assignment Methods

The traffic relayed at a node is related to its distance to the sink, the number of source nodes, the packet traffic generated by each source node in the network, and the node density. In this section, I present this relationship analytically, then, given the average traffic observed at a node, I derive the duty cycle for minimizing energy.

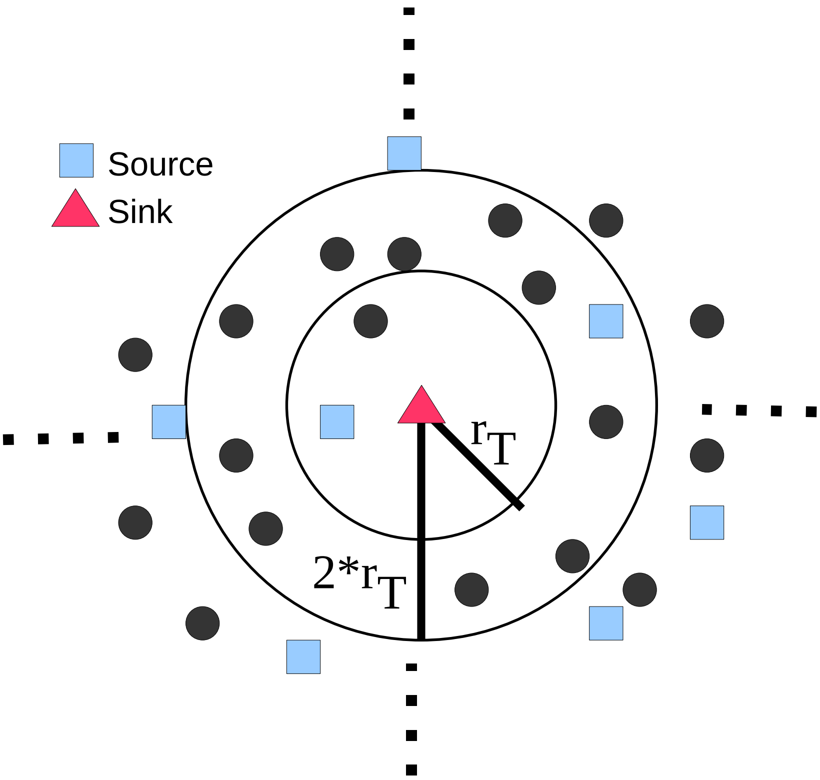

Figure 4.1: Sample network topology.

4.1.1 Traffic Rate Analysis

For the analysis, I assume a circle area with the sink located in the center and the nodes including the sources uniformly randomly allocated as illustrated in Fig. 4.1, where r_T is the transmission range. I define the n^{th} ring to be the ring whose inner circle is $(n-1) * r_T$ away from the sink with width r_T. Hence, the n^{th} ring contains the nodes that are n-hops away from the sink. Let there be N_n nodes in this ring. Assuming that each one-hop transmission advances the packet by the transmission range of a node, the traffic that must be relayed by all of the nodes located in the n^{th} ring per unit time, Γ_n, is the summation of the traffic generated by the source nodes in the n^{th} ring and within the rings outside of the n^{th} ring per unit time, i.e.,

$$\Gamma_n = \lambda_g \rho_s \pi (R^2 - [(n-1) r_T]^2), \tag{4.1}$$

where λ_g is the average traffic generation rate of the source nodes, ρ_s is the density of source nodes, and R is the radius of the network area.

Since Γ_n is the total traffic relayed per unit time in the n^{th} ring, a node within that ring relays a traffic with a mean Γ_n/N_n packets per unit time A node with a distance r to the sink resides in the $n = \lceil \frac{r}{r_T} \rceil$ ring The number of nodes in the n^{th} ring is

$$N_n = \rho_r \pi \left\{ (nr_T)^2 - [(n-1)r_T]^2 \right\}, \tag{4.2}$$

where ρ_r is the density of nodes. Hence, the average traffic rate of a node at distance r, λ_r, is Γ_n/N_n, i.e.,

$$\lambda_r = \frac{\lambda_g \rho_s \pi \left\{ R^2 - [(\lceil \frac{r}{r_T} \rceil - 1)r_T]^2 \right\}}{\rho_r \pi \left\{ [(\lceil \frac{r}{r_T} \rceil)r_T]^2 - [(\lceil \frac{r}{r_T} \rceil - 1)r_T]^2 \right\}}. \tag{4.3}$$

4.1.2 Duty Cycle for a Given Expected Traffic Rate

The time required for a transmission and the energy efficiency of the network is closely related to the duty cycle values used. Higher duty cycle values provide more nodes available for data routing, such that the possibility to have no relay nodes is decreased and a lower latency is achieved, yet they consume more energy. In this section, I derive the duty cycle that minimizes the energy consumption for a given traffic rate.

In [32], a similar derivation is done for unicast traffic, where every node can be a source or a destination. I adapt the analysis presented in [32] for convergecast traffic and the MAC protocol modifications described below. Although a receiver-based MAC protocol is analyzed in [32], the simulation results showed a high number of collisions and high CTS traffic load for the MAC protocol investigated. To reduce the number of collisions and the CTS traffic load, I propose the following MAC modification. In [32], the relay region (locations with geographic advancement to the sink) is divided into N_p priority regions, and each region is assigned a contention slot such that priority region i is assigned the ith slot in the contention window. I assign each priority region N_r CTS contention slots, such that priority region i is assigned the slots $((N_i - 1) \times N_r, N_i \times N_r - 1)$. This reduces CTS collisions, as all nodes in priority region i can select one of the N_r CTS contention slots to send its CTS packet.

N_p and N_r should be functions of the node density. They can be optimized to reduce CTS collisions while keeping the latency low. In reality, since each priority region has a different area, the expected number of the nodes located in different priority regions is different. Therefore, N_r can be adjusted corresponding to the specific priority region instead of being constant across the entire relay region.

The duty cycle analysis is based on the idea that the expected energy consumption of a sensor node is proportional to the expected total awake time, t_l, of the node. This is because the radio idle listening power is approximately the same as the transmission and reception power in WSNs [8]. A constant power value P is assumed for idle listening, transmission, and reception.

In the following analysis, N denotes the average number of nodes within a node's transmission range, d denotes duty cycle, and λ_r denotes the average traffic rate of a node located at distance r to the sink node given in (4.3). Assuming a Poisson or uniform packet generation rate, the average traffic rate of a node follows the Poisson distribution. The probability that a node detects no traffic can be calculated to be $e^{-\lambda_r N T_L}$ where T_L is its listen period at each cycle and $\lambda_r N$ is the average packet arrival rate within its transmission range. Thus, the probability a node detects any ongoing traffic is $p_0 = 1 - e^{-\lambda_r N T_L}$. If ξ is the ratio of the relay region (i.e., the region in which nodes make forward progress to the sink) to the transmission area, $p_0\xi$ is the probability of a node detecting ongoing traffic and residing in the relay region of that traffic.

When a node has a packet to send, it sends an RTS packet and keeps retransmitting the RTS packet until receiving a CTS packet. The expected number of RTS transmissions needed before the first successful RTS/CTS handshake is

$$\begin{aligned} \sum_{i=1}^{\infty} i(1-p_1)^i p_1 &= \tfrac{1-p_1}{p_1} \\ &= (e^{\xi dN}-1)^{-1} \end{aligned} \tag{4.4}$$

where $p_1 = 1 - e^{-\xi dN}$ is the probability that at least one node replies to the RTS packet, since number of nodes residing in an area can be approximated by Poisson distribution for uniformly random deployment [10]. For each retransmission, the node sends out an RTS packet and waits for $N_p \times N_r$ CTS slot durations. The expected time needed before the first successful RTS/CTS handshake, t_H, is then

$$t_H = (e^{\xi dN}-1)^{-1}(T_{RTS} + N_p N_r T_{CTS}), \tag{4.5}$$

where T_{RTS} and T_{CTS} are the transmission delays for RTS and CTS packets, respectively.

The expected total time for a complete RTS, CTS, DATA and ACK packet communication is

$$t_C = T_{RTS} + xT_{CTS} + T_{DATA} + T_{ACK}, \tag{4.6}$$

where x represents the number of CTS contention slots up to and including the first successful CTS packet, T_{DATA} and T_{ACK} are the required times for DATA and ACK packets, respectively. The formula for x can be calculated from a standard CSMA model, and I omit it here for the sake of brevity.

Therefore, the expected total time for a node to transmit a packet, including all the failed RTS packets and the successful data exchange is $t_t = t_H + t_C$, as shown in Fig. 4.2.

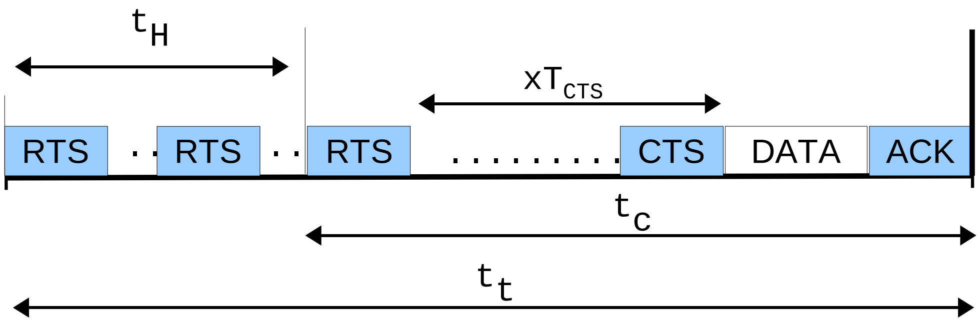

Figure 4.2: Representation of packet exchange durations.

The expected time for a node to receive an RTS packet during a listening period is $\frac{T_L}{2}$. An approximation for the probability that a node wins the contention and is selected as the relay node is given in [32] as $\dfrac{1-e^{-\xi dN}}{\xi dN}$. Then, the average active time of a node that receives traffic and that resides in the relay region of the sender node is

$$t_1 \;= \tfrac{T_L}{2} + \frac{1-e^{-\xi dN}}{\xi dN} t_C + (1 - \frac{1-e^{-\xi dN}}{\xi dN})\tfrac{T_L}{2}. \tag{4.7}$$

Finally, the expected time a node is awake during one listen period is $t_l = (1-p_0\xi)T_L + p_0\xi t_1$, where $(1-p_0\xi)$ represents the probability that either a node hears no traffic or hears some traffic but is not in the relay region, in which case the node is awake for T_L time.

The expression for the expected energy consumption $\bar{P}$, then, can be derived as

$$\begin{aligned} \bar{P} &\simeq P \times \tfrac{t_l}{T_L/d} + \lambda_r P t_t \\ &\simeq P\{d + \lambda_r[(e^{\xi dN}-1)^{-1}(T_{RTS} + N_p N_r T_{CTS}) \\ &\quad +(2-e^{-\xi dN})(T_{RTS} + xT_{CTS} + T_{DATA} + T_{ACK})]\} \\ &\simeq P\big\{d + \lambda_r\{[(e^{\xi dN}-1)^{-1}(1+N_pN_r) + 2+ \\ &\quad x]T_{CTS} + 2T_{DATA}\}\big\} \\ &\simeq P\{d + \lambda_r[(e^{\xi dN}-1)^{-1} N_p N_r T_{CTL} + 2T_{DATA}]\}, \end{aligned} \tag{4.8}$$

where $T_{RTS} \simeq T_{CTS} \simeq T_{ACK} = T_{CTL}$, and $1-e^{-\lambda_r N T_L} \simeq \lambda_r N T_L$ when $\lambda_r N T_L << 1$.

Since xT_{CTS} is dominated by the other components in the formula, it is eliminated as a simplification.

I take the derivative of the expected energy consumption function with respect to d and set it to zero to find the duty cycle that minimizes the expected energy consumption. The duty cycle resulting in E_{min} is $d_{opt} = \frac{\log[\frac{\alpha+2+\sqrt{\alpha(\alpha+4)}}{2}]}{\xi N}$ where $\alpha = \lambda_r \xi N N_p N_r T_{CTS}$. Finally, the mathematical relation between duty cycle and average traffic rate is derived. The value of λ_r for a node is found with the analysis presented in Section 4.1.1.

4.1.3 Duty Cycle Assignment Methods Proposed

The Distance-based Duty Cycle Assignment (DDCA) method defines the duty cycle of a node to be the duty cycle based on the analysis presented in Sections 4.1.1 and 4.1.2. Since those analysis do not take packet contention and collision into consideration, I round up the duty cycle found by DDCA to be $\frac{\lceil 100d \rceil}{100}$. Although the analysis presented considers expected traffic observed by a node at a given distance, in practice the actual traffic loads vary per node and over time. Moreover, the entire analysis focuses on minimizing energy consumption while leaving the end-to-end delay performance as a later concern. Aiming to solve these problems, I also propose a distance-based duty cycle assignment scheme combined with the actual traffic pattern observed. In general, the receiver-based protocols do not exchange any traffic information between nodes to achieve stateless communication. However, RTS packets can be used to observe the traffic load. The number of retransmitted RTS packets increases either when a node's duty cycle is too low and no relay candidates can be found, or when the traffic load is too high and the high contention of nodes causes collisions of the RTS packets from different transmitters. For either case, increasing the duty cycle would increase the probability of successful communication.

I introduce a piggyback flag to the original packet header of the RTS packet to indicate whether this packet is being retransmitted or not. A counter is also set in every node to record the numbers of the initial and retransmitted RTS packets. If the total number of the received retransmitted RTS packets in the current cycle outweighs the total number of the received initial RTS packets, it indicates severe contention in the neighborhood, and the duty cycle of the node is increased to mitigate the traffic load. Otherwise, the duty cycle is decreased every cycle down to a minimal 1% to minimize the energy consumption. This method is called Traffic-Adaptive Distance-based Duty Cycle Assignment (TDDCA). TDDCA is expected to improve the latency performance, since it takes into account not only the distance-based duty cycle assignment, but also the spatiotemporal traffic information in a particular network deployment.

4.2 Performance Evaluation of Duty Cycle Assignment

Simulations are performed using the OPNET simulator to compare the two methods proposed, namely DDCA and TDDCA, with the network-wide constant duty cycle assignment method. In the network-wide constant duty cycle method, the duty cycle is set to the duty cycle found by the DDCA method for the nodes one hop away from the sink, such that a high packet delivery ratio is guaranteed. In this way, the common constant duty cycle is obtained to minimize the energy consumption across the network.

The performance metrics evaluated are packet delivery ratio, average energy consumption, and average latency. The radius of the target area R is set to be $90\,m$ and the transmission range r_T for all nodes is set to be $30\,m$. For simplicity, in the simulations I assume the relay region ratio ξ is constant and set to 0.4, and the power for transmission, reception and idle listening is set to 1 unit. The sink is located in the center of the area, where 400 nodes are uniformly randomly deployed. In TDDCA, the duty cycle is changed by 0.01 every listening interval based on the observed RTS retransmissions. Fig. 4.3 shows the duty cycle in DDCA and the constant method with 20 sources.

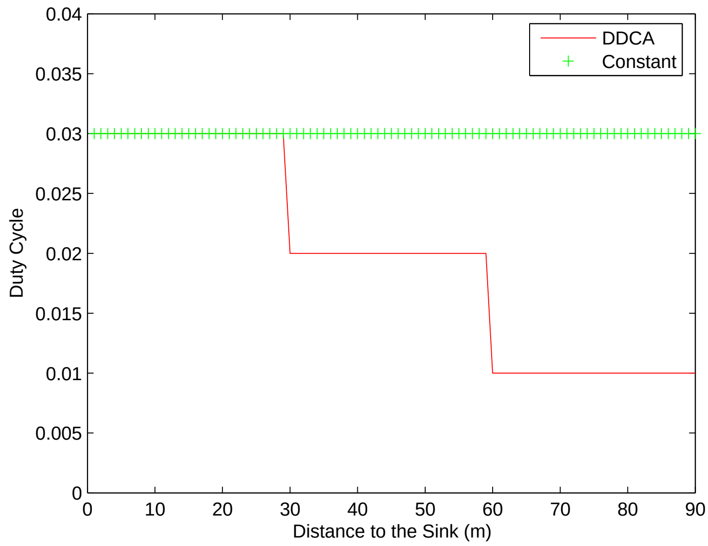

Figure 4.3: Duty cycle in DDCA and Constant method.

Two sets of simulations are performed to investigate the performance of the presented duty cycle assignment methods for a varying number of sources and a varying packet generation rate, λ_g. The effect of the number of sources is investigated for a packet generation rate of 0.5 packet/sec and the effect of the packet generation rate is investigated for 40 sources.

The packet delivery ratio (PDR) values achieved by the three methods are presented in Fig. 4.4 and Fig. 4.5. In all three methods, the PDR results are very close and higher than 97% for light traffic loads. With an increase in traffic load, the constant duty cycle method performs the best because its higher duty cycle can provide more awake nodes to participate in data routing. The slightly worse performance of TDDCA compared to the constant duty cycle method indicates that

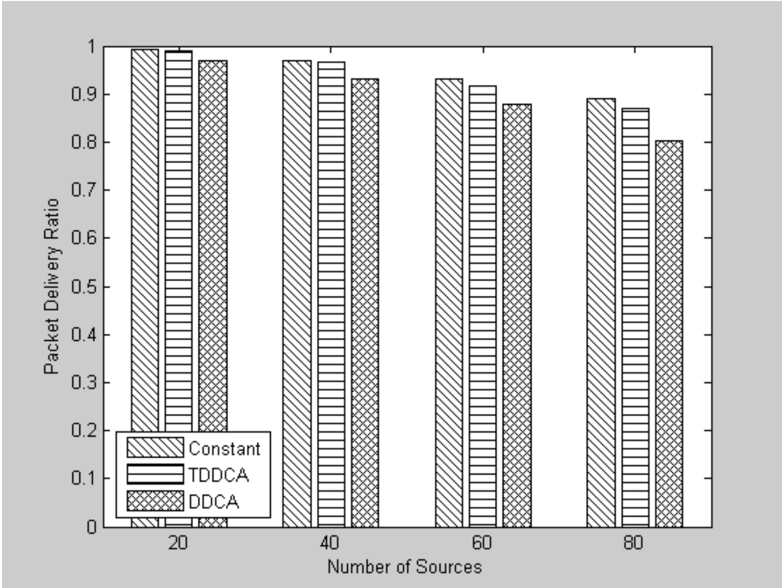

Figure 4.4: Packet delivery ratio vs. the number of sources

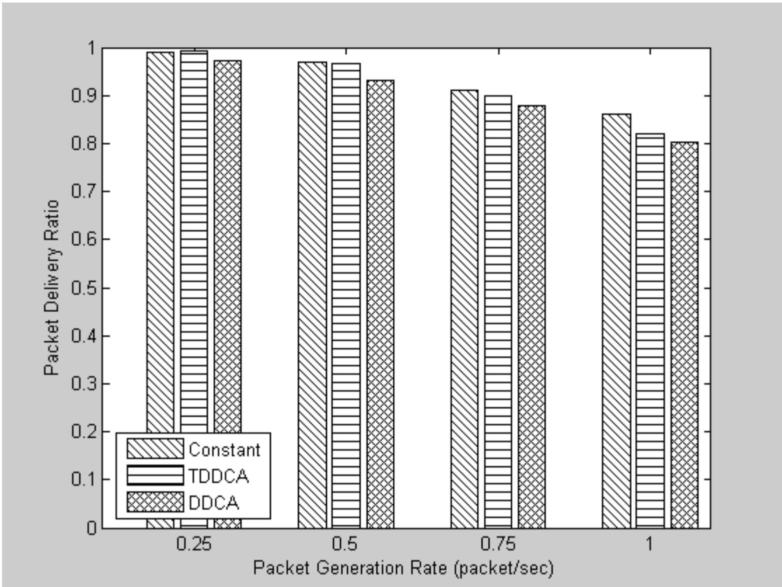

Figure 4.5: Packet delivery ratio vs. the source packet generation rate λ_g

the fixed increments and decrements in duty cycle is not efficient in terms of PDR. One alternative is to use varying duty cycle increments and decrements as proposed in [22].

While PDRs are approximately the same using all three methods, Figs. 4.6 and 4.7 both show that TDDCA and DDCA are more energy-efficient than the constant

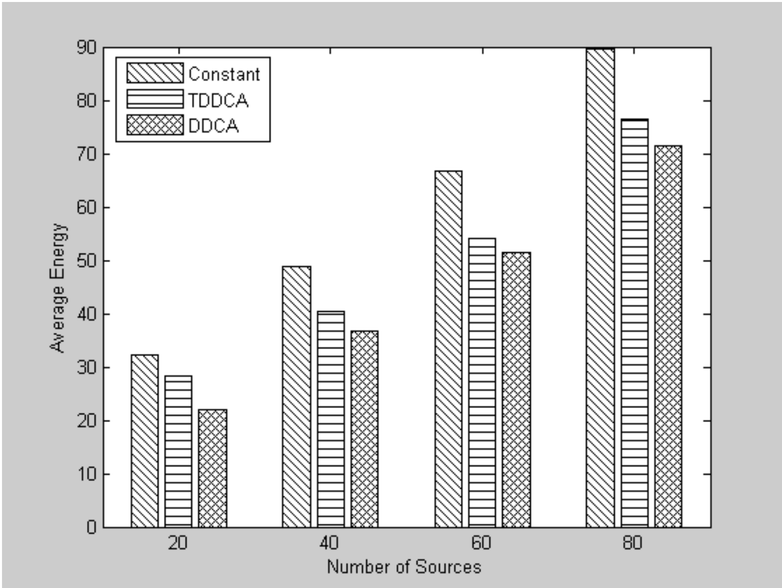

Figure 4.6: Average energy consumption vs. the number of sources

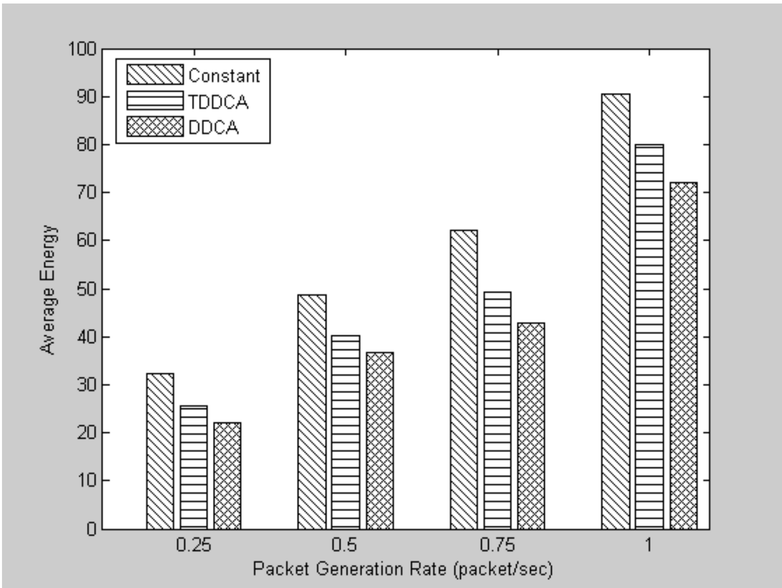

Figure 4.7: Average energy consumption vs. the source packet generation rate λ_g

duty cycle method, and that DDCA performs better than TDDCA. DDCA reduces energy dissipation between 21% and 32% compared to the constant duty cycle method, while TDDCA reduces energy dissipation between 12% and 19% compared to the constant duty cycle method. Because the entire network is likely to generate more retransmitted RTS packets than original RTS packets, TDDCA increases duty cycle more often than decreasing it. The reason is as follows: in the area near the sink where traffic is heavy, available nodes that receive the first RTS packet turn to a busy state until they win the contention or receive a CTS packet from another node for the same RTS packet. In this busy state, receivers do not reply RTS packets from other transmitters, which results in retransmitted RTS packets even when there are awake nodes within nodes' transmission ranges. On the other hand, in the area far from the sink where traffic is light, the duty cycles of nodes are low such that it is possible that there are no awake nodes that can hear an RTS packet when it is broadcasted. Thus, retransmitted RTS packets are generated in this case as well. Generally the fact that TDDCA increases the duty cycle more often than decreasing it leads to its larger average energy consumption than DDCA.

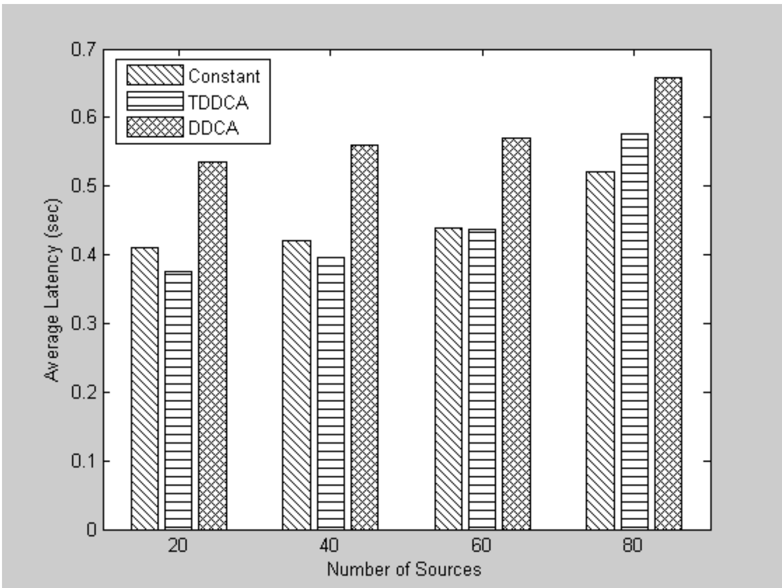

Figure 4.8: Average latency vs. the number of sources

Figs. 4.8 and 4.9 show that TDDCA performs the best in terms of latency. In light traffic, TDDCA achieves better latency values compared with DDCA, e.g., latency using TDDCA is 30% less than latency using DDCA when the number of sources is 20. Since nodes are likely to increase their duty cycle rather than to decrease it, in TDDCA there are more nodes available to contend for the channel

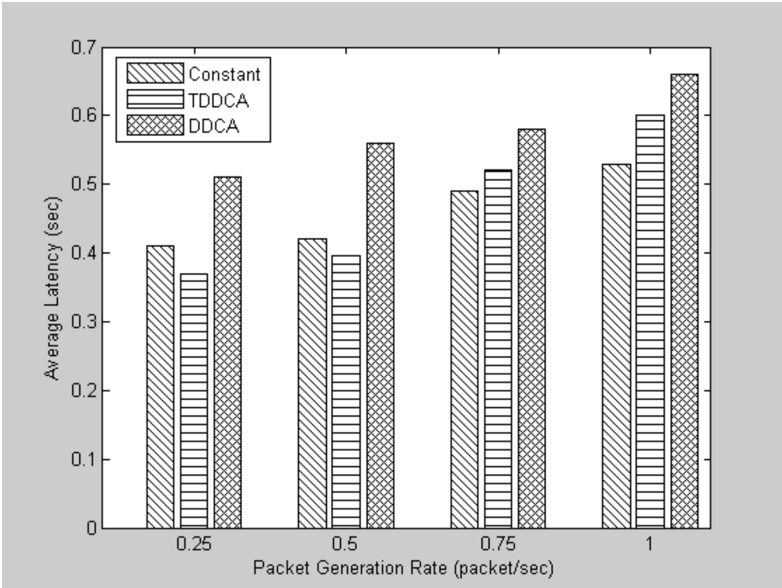

Figure 4.9: Average latency vs. the source packet generation rate λ_g

and latency is reduced compared with DDCA. It is also shown that in heavy traffic, TDDCA performs worse in terms of latency compared with the constant duty cycle method. This is because under the severe impact of packet collisions and contention, traffic patterns vary between every listening interval such that a simple comparison between the number of original RTS packets and retransmitted RTS packets cannot reflect the current level of traffic accurately enough. Hence, the method of adjusting duty cycles by 0.01 in each listening interval is not effective to achieve a low latency in high traffic conditions.

In summary, both DDCA and TDDCA are more energy-efficient than the constant duty cycle method, while achieving similar packet delivery ratio performance. Compared with DDCA, TDDCA has an advantage in terms of latency.

4.3 Conclusion

In this chapter, two duty cycle assignment algorithms are proposed. Simulation results show that both methods decrease energy consumption compared with the constant duty cycle method for the scenarios investigated. The traffic-adaptive distance-based duty cycle assignment method achieves energy improvements without sacrificing from the latency and throughput significantly.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, I evaluate a new stateless multicast protocol for ad-hoc networks called Receiver-BasedMulticast (RBMulticast). RBMulticast uses geographic location information to route multicast packets, where nodes divide the network into geographic multicast regions and split off packets depending on the locations of the multicast members. RBMulticast stores a destination list inside the packet header; this destination list provides information on all multicast members to which this packet is targeted. Thus, there is no need for a multicast tree and therefore no tree state is stored at the intermediate nodes. RBMulticast also utilizes a receiver-based MAC layer to further reduce the complexity of routing packets. Because the receiver-based MAC protocol can determine the next hop node in a distributed manner, the sender node does not need a routing table or a neighbor table to send packets but instead uses a virtual node as the packet destination. Thus RBMulticast requires the least amount of state of any existing multicast protocol. After being developed with a new retransmission scheme that I proposed, simulations of RBMulticast show that it can achieve high success rates, low latency and low overhead in terms of the amount of generated traffic for both static and dynamic scenarios, making RBMulticast well suited for both mobile and stationary ad hoc network environments.

I also derive the duty cycle for a node as a function of its distance to the sink to minimize expected energy consumption for convergecast traffic patterns and receiver-based routing. Based on my analysis, I develope two duty cycle assignment algorithms: Distance-based duty cycle assignment (DDCA) method and Traffic-adaptive distance-based duty cycle (TDDCA) method. The DDCA analysis in one hop distance to the sink offers the duty cycle which minimizes the energy consumption while keeping the high throughput for the constant duty cycle assignments across the entire networks as well. Simulation results show that both methods decrease energy consumption compared with the constant duty cycle method by up to 32% for the scenarios investigated. The Traffic-adaptive distance-based duty cycle assignment method achieves energy improvements without sacrificing from the latency and throughput significantly.

5.2 Future Work

Simulation results show that the performance of RBMulticast is closely related to the location of the sinks. Generally, RBMulticast performs better when sinks cluster than when they are sparsely distributed. Therefore, RBMulticast can be extended to improve performance when sinks are sparsely distributed. In the duty cycle assignment algorithms, both the Distance- based duty cycle assignment method and the Traffic-adaptive distance-based duty cycle method show a better performance in low and medium traffic than in high traffic. My analysis can be extended to improve the performance of distance-based duty cycle assignment in heavy traffic scenarios, by taking packet collisions and contention into account.

In both proposed duty cycle assignment methods, nodes closer to the sink always consume more energy than nodes further from the sink. Given the same battery life for each node, nodes closer to the sink would use up their energy early. Instead of calculating the duty cycle for each node to minimize energy consumption with a constant node density, another method to realize energy efficiency is to use my analysis to derive the optimum node density given a fixed duty cycle for each node.

Bibliography

[1] I.F. Akyildiz, M.C. Vuran, and O.B. Akan. A cross-layer protocol for wireless sensor networks. In *Proc. of CISS 2006*, March 2006.

[2] S. Basagni, I. Chlamtac, and V. R. Syrotiuk. Location aware, dependable multicast for mobile ad hoc networks. *Computer Networks*, 36(5/6):659–670, 2001.

[3] B. Blum, T. He, S. Son, and J. Stankovic. IGF: A state-free robust communication protocol for wireless sensor networks. Technical report, In: Proceedings of the 3rd IEEE Workshop on Applications and Services in Wireless Networks, 2003.

[4] P. Bose, P. Morin, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Wireless Networks*, pages 48–55, 1999.

[5] Q. Cao, T. He, and T. Abdelzaher. uCast: Unified connectionless multicast for energy efficient content distribution in sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 18(2):240–250, Feb. 2007.

[6] K. Chen and K. Nahrstedt. Effective location-guided tree construction algorithms for small group multicast in manet. *INFOCOM. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1180–1189, 2002.

[7] C. Chiang, M. Gerla, and L. Zhang. Shared tree wireless network multicast. In *Computer Communications and Networks, 1997. Proceedings., Sixth International Conference on*, pages 28–33, Sep 1997.

[8] Crossbow Technology. http://www.xbow.com/.

[9] S.M. Das, H. Pucha, and Y.C. Hu. Distributed hashing for scalable multicast in wireless ad hoc networks. *Parallel and Distributed Systems, IEEE Transactions on*, 19(3):347–362, March 2008.

[10] I. Demirkol, C. Ersoy, F. Alagoz, and H. Delic. The impact of a realistic packet traffic model on the performance of surveillance wireless sensor networks. *Computer Networks*, 53(3):382 – 399, Aug. 2009.

[11] C. Feng and W. B. Heinzelman. RBMulticast: Receiver based multicast for wireless sensor networks. *IEEE Wireless Communications and Networking Conference (WCNC '09), April 2009.*

[12] D. W. Gage. Many-Robot MCM search systems. In *Proceedings of Autonomous Vehicles in Mine Countermeasures Symposium*, pages 9–55, 1995.

[13] J.J. Garcia-Luna-Aceves and E.L. Madruga. A multicast routing protocol for ad-hoc networks. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 784–792 vol.2, Mar 1999.

[14] C. Hsin and M. Liu. Hitting time analysis for a class of random packet forwarding schemes in ad hoc networks. *Ad Hoc Netw.*, 7(3):500–513, 2009.

[15] J. G. Jetcheva and D. B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *MobiHoc '01: Proceedings of the*

2nd ACM international symposium on Mobile ad hoc networking & computing, pages 33–44, New York, NY, USA, 2001. ACM.

[16] R. Jurdak, P. Baldi, and C. Videira Lopez. Adaptive low power listening for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 6(8):988–1004, 2007.

[17] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM.

[18] Y. Ko and N. H. Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. *wmcsa*, 0:101, 1999.

[19] D. Koutsonikolas, S. Das, H.Y. Charlie, and I. Stojmenovic. Hierarchical geographic multicast routing for wireless sensor networks. In *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*, pages 347–354, Oct. 2007.

[20] S. Lee, M. Gerla, and C. Chiang. On-demand multicast routing protocol. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 1298–1302 vol.3, 1999.

[21] M. Mauve, H. Fuler, J. Widmer, and T. Lang. Position-based multicast routing for mobile ad-hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):53–55, 2003.

[22] C. J. Merlin and W. B. Heinzelman. Duty cycle control for low power listening MAC protocols. In *In Proc. MASS08*, 2008.

[23] A. Okura, T. Ihara, and A. Miura. Bam: branch aggregation multicast for wireless sensor networks. *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 10 pp.–, Nov. 2005.

[24] OPNET. http://www.opnet.com/.

[25] E. M. Royer and C. E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 207–218, New York, NY, USA, 1999. ACM.

[26] E. Rozner, J. Seshadri, Y. A. Mehta, and L. Qiu. SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE Transactions on Mobile Computing*, 8(12):1622–1635, 2009.

[27] J.A. Sanchez, P.M. Ruiz, and I. Stojmnenovic. GMR: Geographic multicast routing for wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 1, pages 20–29, Sept. 2006.

[28] R. Vaishampayan and J.J. Garcia-Luna-Aceves. Efficient and robust multicast routing in mobile ad hoc networks. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 304–313, Oct. 2004.

[29] W. Wang, X. Li, and Y. Wang. Truthful multicast routing in selfish wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 245–259, New York, NY, USA, 2004. ACM.

[30] C.W. Wu and Y.C. Tay. AMRIS: a multicast protocol for ad hoc wireless networks. In *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*, volume 1, pages 25–29 vol.1, 1999.

[31] T. Zheng, S. Radhakrishnan, and V. Sarangan. PMAC: An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proc. IPDPS '05*, page 237.1, Washington, DC, USA, 2005. IEEE Computer Society.

[32] M. Zorzi and R. R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, 2003.

[33] M. Zorzi and R.R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance. *IEEE Transactions on Mobile Computing*, 2(4):337–348, Oct.-Dec. 2003.