Stony Brook University

Efficient Medium Access Protocols for Wireless and RFID Networks

A Dissertation Presented

by

Shweta Jain

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2007

Stony Brook University

The Graduate School

Shweta Jain

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Dr. Samir R. Das – Dissertation Advisor
Associate Professor of Computer Science

Dr. Jennifer L. Wong – Chairperson of Defense
Assistant Professor of Computer Science

Dr. Himanshu Gupta
Assistant Professor of Computer Science

Dr. Vijay Raghunathan
Assistant Professor of Electrical and Computer Engineering
Purdue University

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Efficient Medium Access Protocols for Wireless and RFID Networks

by

Shweta Jain

Doctor of Philosophy

in

Computer Science

Stony Brook University

2007

Wireless multihop networks of various forms – such as ad hoc, mesh or RFID networks – are getting popular as a means of creating a pervasive wireless networking mechanism. The central concept in such networks is use of multihop relaying. Multihop wireless links give rise to new challenges in medium access control (MAC) protocols. The challenges include interference, fading, improving network throughput and guaranteeing fairness. We have used various cross layer design techniques to combat these challenges.

In our first work, we develop a cross-layer solution called MAC-layer anycast that combats link loss due to interference or fading by exploiting path diversity available from the routing layer. We develop an 802.11-like protocol to implement anycast. We show via both simulations and testbed experiments that it is superior to 802.11-like protocols. We also show that anycast is very useful when used in conjunction with directional antenna or multiple channels, as well as for improving reliability and efficiency of MAC-layer multicast.

In our second work, we have demonstrated the benefits of using the physical layer signal level information to improve the accuracy of scheduling algorithms. To this end, we use the TelosB motes platform to model the relationship between the packet capture probability and SINR based on measurements. We show how this model can be used to develop a realistic interference model for a given testbed using only O(n) measurements on the testbed. We provide validation results for the accuracy of this approach for predicting whether a set of links are schedulable concurrently.

In our third work, we develop protocols for provisioning max-min fair bandwidth for multihop flows. Here, we develop a two-part solution that combines queueing/scheduling and MAC protocol for guaranteeing max-min fairness for multihop flows.

Finally, we focus our attention to RFID networks where new forms of interference are possible due to presence of two different entities, RFID tags and readers. We demonstrate via a testbed how interferences can be resolved in a RFID networks via simple carrier-sensing mechanism that can be implemented using commodity hardware.

Dedicated to
My husband Senthil for his patience and to my parents.

Contents

List of Figures

List of Tables

Acknowledgements

It is a pleasure writing this acknowledgment to all the people who have been of great help and support in my path to writing this thesis. First I would extend gratitude to my doctoral advisor Dr. Samir Das for his guidance and critique without which it would be impossible to complete this dissertation. I am also grateful to Dr Jennifer Wong for her useful comments and motivation toward this effort.

I would like to thank Dr Himanshu Gupta, Dr Jie Gao and Dr I.V Ramakrishnan for their encouragement and support. Also a special thank to Dr. Vijay Raghunathan for being my committee member and for being an excellent advisor during my internship at NEC.

In addition I thank colleagues, alumni and members of the WINGS lab Xianjin, Ritesh, Vishnu, Bin, Anand, Anand Prabhu, Zongheng, Prahllad for their endless support, help and parties that made the experience of being in the lab truly enjoyable. I thank my friends Sandra, Rupa, Devaki, Ruchi and Ruby for being excellent and patient listeners and my parents for being present whenever I needed. Last but not the least, I thank my husband Senthil, for his patience, support, help and comfort which helped me get through those lower moments of my life.

Chapter 1

Introduction

Advancements in nanotechnology have given birth to new generation of ubiquitous mobile devices with high processing speed. Recent development of a breakthrough chip stacking technology by scientists at IBM, has paved the way for three-dimensional chips[5]. This new development has the promise to extend Moore's law beyond its expected limits which will lead to smaller, faster and low power devices. Advancements in wireless network technology has made it possible to connect these devices together over the wireless link. These developments together have spurred the proliferation of a large family of network enabled mobile handheld embedded devices such as laptops, PDAs, mP3 players, gaming consoles and wearable computing solutions, which can communicate with each other to share music, video and data. These devices have a great impact on society and the economy. The market for wireless devices is expected to grow at an annual rate of 15.5% in terms of number of units sold. It is expected that in the year 2007 about 880 million units would be sold. The demand for mobile Internet access both in and out of office has accelerated the growth of the wireless Internet related industry. In the fourth quarter of the year 2006, the worldwide revenue from Wi-Fi had risen to over US $1.0 billion up from US$845.7 million in the previous year [37].

While users are always aware of the wireless Internet access as it is used quite explicitly, RFID technology forms a part of our daily life in a more discrete fashion. RFID tags are often embedded in objects in retail stores, security cards, automatic toll payment tags, transit cards and even credit cards. These tags provide a simple contactless method of identifying objects and information related to the object can be written and read on the tags. RFID has become a part of everyday life and has a much larger impact on the economy than other wireless technologies[71]. The prediction for worldwide revenues from RFID tags is expected to rise by $2.5 billion between 2004 and 2009. The maximum impact of this technology is expected to be in improving business processes. The second-largest market for RFID is forecasted to be in consumer products, despite the privacy issues in RFID that has held back

initial growth of applications in the consumer sector[36].

A key challenge in wireless networks is the problem of sharing the common broadcast medium between multiple users of the network. Wireless communication between a pair of devices is affected by communication between another pair if the devices are close enough for their signals to overlap. Coexistence of various wireless devices and systems in a small area requires coordination among the devices to enable conflict free communication. A medium access control (MAC) protocol povides this coordination and enables interference free communication. MAC protocol also provides quality of service and fair medium access to all contending devices. Therefore, MAC is an important feature in all types of wireless devices and it is the key focus of this thesis. A brief overview of existing technologies, key challenges and contributions of this thesis is explained in the following sections.

1.1 Wireless Networks

The most widely used technology for medium access in wireless local area networks, multihop mesh and ad-hoc networks is the IEEE 802.11 standard MAC protocol. The 802.11 standard was first ratified in 1997. The standard at that time offered a 2Mbps data rate which has now increased to upto 600Mbps in the upcoming 802.11n standard[16]. Three main 802.11 standards (802.11a/b/g) are already in the market while 802.11n pre-releases have been seen. While the 802.11b and 802.11g standards operate in the same 2.4GHz frequency band, the 802.11a standard operates on the 5GHz band[7][8][9] and the 802.11n standard allows communication in both 2GHz and 5GHz bands[16].

Some key challenges that 802.11 protocol solves are collision and hidden terminal[98] and the PHY layer design in 802.11n has the provision to improve the performance in multipath fading scenarios using multiple input multiple output (MIMO) antennas. There are some core issues and challenges that remain unsolved. Some of these challenges are time varying channel conditions, spatial reuse by exploiting channel and antenna diversity, accurate scheduling techniques to improve network utilization, fair medium access in multihop networks and quality of service in multimedia transmission[107]. We attempt to solve some of these challenges in this thesis and our key contributions are (a) cross layer design to overcome losses due to time varying channel conditions, improve performance of networks that use channel and antenna diversity and provide reliable multicast in multihop networks, (b)fair medium access to multihop flows in multihop mesh networks and (c) modeling the impact of physical layer signal to noise and interference ratio on successful packet reception to make accurate scheduling decisions.

One of the key challenges that is addressed in this thesis is transient link

losses in a wireless network. It is well known that in a wireless network that the quality of link between two nodes varies with time. This temporal variation in link quality depends upon the signal to noise and interference ratio as well as multipath fading. Multipath fading occurs due to different components of the transmitted signal being reflected by surrounding objects and combining constructively or destructively at the receiver. Both interference and fading are time varying and may make certain links unavailable for some periods of time. This transient unavailability of links may be sufficient for a routing layer to start a route repair and for TCP to bring down the offered load. Such upper layer reaction to lower layer issues is harmful as it reduces the network utilization. Due to this harmful interlayer interaction, there is an increasing interest in breaking the protocol layer structure (OSI model) in favor of cross layer design techniques[88][29]. Thus, instead of an upper layer reaction to transient link losses, it may be possible for MAC protocols to detect and cope with the short term variations in link quality. The 802.11 protocol deals with such transient variations by retransmissions along the same link until the transmission is successful or a certain number of retries have been performed. Retransmissions cause packet delays and increase the overhead and if the time for which the link is unavailable is large enough, these retransmissions may even be futile.

Our first and one of our main contributions in this thesis is an anycast mechanism at the data link layer which interacts with the routing and physical layers to benefit from path diversity available due to multipath routing to make an instantaneous decision about which link should be selected for transmission. The goal in this cross layer design is to choose the best next hop to forward packets when multiple next hop choices are available. Given a sufficient amount of available path diversity, using the anycast mechanism can significantly reduce the number of transmission retries as well as packet drop probabilities. We have also explored similar anycasting techniques to reduce problems of deafness in networks that use directional antenna and multiple channels to achieve spatial reuse for better network utilization and to provide reliable multicast in multihop networks.

Another challenge that we address in this thesis is the lack of accurate modeling of wireless interference. Wireless communication range is often modeled as a unit disk wherein, the transmitted signal strength is assumed to be an inverse power of distance from the transmitter. Using these simplified assumptions researchers often construct a conflict graph based upon distance between sender-receiver pairs in the network. While these assumption make the algorithm design more tractable but in reality these models make inaccurate estimate of the network throughput and therefore there is an increasing interest in designing better estimates of the transmission channel[76]. Accurate interference models are important to improve the accuracy and throughput achieved by a scheduling algorithms in wireless networks. For this reason, recently the focus has shifted to more realistic, SINR-based models

such as the physical interference model[109][90][39]. This model is based upon the classic theory that relates signal to interference and noise ratio to the probability of successful reception of the transmission or simply the packet reception rate (PRR). The PRR is high ($\sim$1) if the SINR is above a certain threshold and less than 1 otherwise. Thus the knowledge of the physical channel conditions can be useful in making scheduling decisions in the upper layer. In order to design accurate scheduling algorithms for wireless multihop networks, it is worthwhile to experimentally model this SINR vs PRR relation. Our second contribution in this thesis is an experimental model of channel quality in terms of signal, interference and noise levels at the receiver. We also demonstrate that knowing the SINR on a link one can quite accurately predict the PRR on a link given a set of simultaneously active links in the network. This model may be adopted in an upper layer scheduling algorithm or in a central controller to make better informed scheduling decisions.

The broadcast nature of wireless transmission poses many challenges for medium access techniques in multihop networks. One such problem is fair medium access among multihop flows in the network. It is easy to build scenarios in which the medium sharing is biased against a set of links simply because of their relative positions in the network with respect to other links[51]. The concept of fairness in ad-hoc and mesh networks may be different from that in wireless LANS. This is simply because nodes in wireless LANs only carry traffic that they have generated while nodes in multihop networks relay packets for other nodes in the network. Thus, fairness in multihop networks must be flow based instead of being node based. In the context of fairness, an appropriate and viable solution is maxmin fair allocation to each multihop flow. In a maxmin fair allocation resources are allocated in order of increasing demand such that no user gets a resource share larger than its demand and sources with unsatisfied demands get an equal share of the resource. Also a user with unsatisfied demands cannot increase its resource usage without reducing the resource usage of other users that already have an equal or lesser resource usage than itself.

Our third contribution in this thesis is distributed protocol suite that consists of an upper layer queuing mechanism and a first in first out MAC protocol which together form a complete solution toward providing maxmin fair medium access. We achieve this fairness goal by introducing a scheduling layer between data link and routing layers to perform maxmin fair rate computation and scheduling in wireless mesh networks. This layer interacts with the network to exchange the transmission schedules of neighboring nodes, which in turn helps in computing fair schedules in the network. First, the scheduling layer estimates the maxmin fair rate of all multihop flows in the network using a distributed protocol. This estimation uses the knowledge of the flow contention graph that the network nodes learn by exchanging local information. Second, this layer enforces the computed rate by controlling

the rate at which a flow is scheduled at the link layer. Third, a back pressure flow control is used to reduce the transmission rate of a flow if it has been exceeding its fair rate.

In the context of fairness, we also argue that the fair rate estimation can at best be approximated in an 802.11 based MAC protocol. The random access mechanism with exponential backoff has been shown to be unfair in prior work [55],[59]. Thus, to complement our fair rate estimation and scheduling procedures, we develop a virtual time based MAC protocol. This virtual time based protocol, schedules transmissions based upon the arrival times of the packets to be transmitted. This technique ensures that in a contention region, packets will be transmitted in a first in first out order.

1.2 Radio Frequency Identification (RFID)

While MAC protocols for wireless ad-hoc and mesh networks have undergone much research, similar research in radio frequency identification (RFID) networks is still in early stages, especially so in multi-reader RFID networks. RFID tags and readers share the same RF medium for communicating with each other. Thus, a RFID system comprising of multiple tags and readers in close vicinity also suffer from wireless interference when transmissions from multiple readers and/or tags overlap. Collisions in RFID systems can be classified into three categories (a) tag-tag collision which results from interference between signals from multiple tags that may simultaneously start transmission in response to a reader's command (b) reader-reader collision where simultaneous transmission from two readers interfere at the tag and (c) reader-tag collision where a reader's transmission interferers with a tag response on another reader. There are well known MAC protocols to solve the tag-tag collision problem, and most readers implement some form of an anticollision protocol to resolve conflicts[86][12]. The reader-reader and reader-tag collisions that occur in dense reader deployment is a key challenge in RFID research. This problem has been studied in the EPCGlobal Class1 Gen1 and Gen2 standards for UHF readers [14] [15]. In Gen 1 standard, the reader-tag collision problem is mitigated by allowing frequency hopping in the UHF band or by time division multiple access. In Gen 2 the readers and tags operate on different frequencies so that the tag response does not interfere or collide with reader signals. Either solution requires fairly sophisticated technology. We contribute to the RFID technology through a collision avoidance protocol for dense deployment of RFID readers. We have designed solution for both reader-reader and reader-tag collisions in a dense RFID network. We experiment with a network that is implemented using mote-based RFID readers. To implement the MAC protocol, we have developed an

appropriate carrier sensing circuit using an RFID tag as an antenna and the mote as an apparatus to sample received signal strength. We have augmented a commercially available OEM RFID reader module with such carrier sensing capability and interfaced it with motes. Our main contribution in this work is the development of a carrier sensing capability in an RFID reader and the implementation of the CSMA MAC protocol that takes advantage of this new capability.

In the following chapters, we will provide detailed description and analysis of our contributions. We start with the anycast mechanism for combating multipath fading in Chapter 2 followed by application of anycast in directional antenna and multichannel networks in Chapter 3 and reliable multicast MAC in Chapter 4. In Chapter 5 we discuss the details of physical interference modeling in wireless networks and present the accuracy of the model. We discuss our maxmin fair scheduling algorithm in Chapter 6 and in Chapter 7, we discuss a medium access protocol for collision avoidance in RFID networks.

Chapter 2

Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks

2.1 Introduction

It is well-known that in wireless ad hoc networks, the "link" between two nodes is a "soft" entity [32]. From basic communication theory, its existence is governed by whether the signal to interference plus noise power ratio (SINR) at the receiver exceeds a given threshold (called the *receive threshold* γ). γ is determined by the data rate, the modulation technique, receiver design, and the target bit error rate (BER) the receiver is able to withstand (i.e., able to correct using coding techniques). SINR is again influenced by transient factors such as transmit power, distance between the transmitter and receiver, multipath fading, and interference and noise powers reaching the receiver. Multipath fading [80] is caused by different components of the transmitted signal being reflected by the surrounding objects, and reaching the receiver via paths of different lengths, and combining either constructively or destructively. Interference is caused by signals for other, unintended nearby transmitters. Both fading and interference could be time varying. Significant changes in fading and interference levels (beyond that can be masked by changes in sending data rate [17, 41])[1] may lead to transient "loss" of a link. This loss is often sufficient for many common routing and transport protocols to react – either to repair routes or to bring down the offered load. This leads to various operational inefficiencies, given that this loss is transient. Thus, there is a need to incorporate mechanisms that can "withstand" this loss of link at shorter time-scales.

While fundamentally new approaches are necessary to incorporate this soft abstraction for a link in the upper layer protocol design, it is often possible to take

[1]Note that while physical layer techniques can mask effect of fading and interference, this work does not target physical layer techniques. Here, the interest is working on beyond physical layer capabilities, by exploring alternative paths.

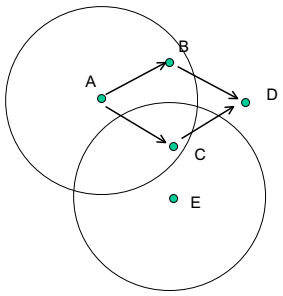


Figure 2.1: Example scenario motivating anycast. Node A can forward packets to D either via B or C. But an ongoing transmission at E may interfere at C. If A chooses to forward via C, the transmission will defer until E's transmission is complete. Such instantaneous channel conditions are unknown to the routing layer that discovers the routes.

an "ad hoc" approach that we pursue in this chapter. Here, a "hard" (stable, on or off) abstraction is still used for the link from the viewpoint of the upper layer – something it is designed to handle comfortably. However, now multiple link options are provided to the link layer, and the link layer is given the responsibility to make an instantaneous decision on which link to forward the packet on. We design a MAC-layer *anycasting*[27] scheme to perform this decision making and to forward the packet.

To implement anycasting, the link layer must take advantage of a multipath routing protocol [61, 106, 68, 73]. Assume that multiple routing paths have been computed from the source and also from the intermediate nodes to the destination. Typically, the routing layer decides which of the several paths should be used for data forwarding and then the MAC layer is responsible to deliver the packet to the next hop along the chosen path. Now, predominant channel conditions (e.g., because of multipath fading and interference) may cause data transmission to defer or even fail causing the network layer to attempt using an alternate next hop. See a simple example in Figure 2.1. This leads to multiple transmission retries, wasting bandwidth and increasing delay. A better, alternative approach would be, for the link layer, to choose the next hop by observing the channel conditions on all possible next hop links. This "channel state-based" anycasting should improve performance, requiring very little operational coordination between the routing and MAC layers.

The goal of this work is to develop an anycast MAC layer protocol to do this "channel state-based" next hop selection. While such a MAC layer protocol can be designed in many ways, a reasonable step is to do this design as an extension/variation of the commonly used IEEE Standard 802.11 [13] MAC layer. This makes performance easy to analyze and compare.

The rest of the chapter is organized as follows. In Section II, we The rest of the chapter is organized as follows. In Section II, we provide an overview of the 802.11 MAC protocol operation and describe the properties of a fading channel. In Section 2.3 , we describe our extension of 802.11 that implements anycasting to do the channel state based next hop link selection. We also describe the essentials of the multipath routing layer followed by section 2.4 in which we present performance evaluation of anycast. We have analyzed the performance of anycast in a grid network via analytical modeling, and an experimental testbed using Berkeley motes. We have also performed detailed simulation-based evaluations using the popular ns-2 simulator. We describe the related work in Section 2.5 and conclude in Section 2.6.

2.2 Background and Motivation

We start by briefly reviewing the impact of channel model in the IEEE 802.11 standard distributed coordination function (DCF) [13]. This is the MAC layer functionality that we will later extend in this chapter.

2.2.1 Impact of Channel Model

Note that even though RTS retries are allowed in 802.11, it usually takes care of problems due to RTS collision or NAV being set at the receiver. These are indicative of high interference at the receiver. However, the protocol has little option to overcome the effect of time-varying multipath fading – something that cannot be easily removed by simple changes in the protocol. To understand things better, in this subsection we present a well-known radio propagation model, and then analyze how this may influence 802.11 behavior.

Assume that the signal power transmitted by the transmitter is P_T. The signal power P_R received at the receiver at a distance d from the transmitter at time instant t is explained by a combination of large-scale and small-scale propagation models [80]. The large-scale model explains variations in P_R for large changes in d, while the small-scale model explains the same for small changes in d or t. It is well-recognized that in the large-scale, P_R drops with distance following an inverse-power law:

$$P_R \propto \frac{P_T}{d^\alpha},$$

where α is a constant dependent on the exact nature of the model used and is usually between 2–5 depending on the environment. The constant factor governing the above proportionality is a function of parameters not of direct concern to us

here, such as antenna parameters, transmit carrier frequency etc. The small-scale model influences this received power with a multiplicative, time-varying factor with known statistical characteristics. When there is a dominant signal component present (say, the line-of-sight or LOS component) among various signal components reflected at various objects and being superimposed at the receiver, this factor follows the *Ricean* probability distribution [80] given by,

$$p(r) = \frac{r}{\sigma^2} e^{-\frac{(r^2+A^2)}{2\sigma^2}} I_0\Big(\frac{Ar}{\sigma^2}\Big),$$

where A is the peak amplitude of the dominant signal, σ^2 is the variance of the multipath, and $I_0(.)$ is the modified Bessel function of the first kind and zero-order. The Ricean distribution is typically described in terms of a parameter K, given by

$$K = \frac{A^2}{2\sigma^2}.$$

As A increases (i.e., the dominant path increases in amplitude), K also increases.

When the transmitter, receiver or objects in the surrounding environments are moving, there is a *Doppler shift* in the frequency of the received signal. Let us denote the maximum Doppler shift by f_m, where $f_m = vf_c/c$, v being the maximum perceived relative velocity between the transmitter and receiver (which could be caused by the motion of surrounding objects reflecting transmitted signal), f_c is the carrier frequency and c is the speed of light. The Doppler shift causes the signal power to fluctuate in time but with certain temporal correlation property. This fluctuation is usually described by the *level crossing rate* (N_R) which is the rate at which the signal envelop, normalized to the RMS (root mean square) value, crosses a given level R in the positive going direction. N_R depends on the given level R, the parameter K and the maximum Doppler shift f_m [80]. Knowing N_R, the *average fade duration* (average duration for which the signal level is below a given level R) can be computed as,

$$\bar{\tau} = \frac{Pr(r \leq R)}{N_R},$$

where $Pr(r \leq R)$ is the cumulative distribution function of the Ricean distribution.

Data presented in [83] for Doppler frequencies that can be encountered in practice[2] show that the average fade duration can be in the order of tens of milliseconds. As a specific example, for the 2.4 GHz carrier frequency (f_c) and 2

[2]While data for only $f_m = 20$ Hz is presented in [83], the average fade duration for any f_m can be easily computed, given that the relationship between N_R and f_m is linear.

m/sec relative speed (v), the Doppler frequency f_m is 16 Hz. For this Doppler frequency, for 10 dB or more power loss due to fading, the average fade duration is approximately 10 ms; for 5 dB or more it is approximately 20 ms; increasing to approximately 30 ms for 1 dB.

Common routing protocols in ad hoc networks focus on optimizing the number of hops between source and destination. This tends to increase the physical distance of each hop, so that the number of hops is minimum. This lowers the received power P_R as modeled by the large-scale propagation model. Thus, even a small reduction in received signal power due to fading may make the SINR fall below the receive threshold γ causing a transient loss of link that may persist for several tens of milliseconds.[3]

Compare these average fade durations with the fact that it takes approximately 30ms for the RTS retries to fail 7 times causing the MAC to drop the frame. This is computed by using the interframe spacings and slot times from the standard specifications [13], assuming each random backoff lasts for its average duration, and the NAV is never set. Setting of NAV during the time a node is on backoff will extend the backoff time by the NAV period. This analysis shows that it is quite possible that a link is in fade long enough that data transmission will fail in spite of multiple retries. It is also conceivable from the above analysis that it is very likely that 802.11 will need to make a few RTS retries to complete the entire exchange. This fact will later be verified via simulation experiments.

2.3 Channel State-Based Link Selection

Assume now that multiple possible next hop options are presented to the transmitter, and its responsibility is to transmit to *any one* of these receivers successfully. Since fading on different links is expected to be uncorrelated, it is unlikely that all links are in deep enough fade at the same time with SINR $< \gamma$. Thus, it is likely that transmission on at least one link is possible without any significant number of retries in the average case. In the next sub-section, we describe an extension of 802.11 that uses this idea.

[3]Note that physical layer techniques such as transmit power control and rate control can be used to tackle such link loss to some extent. In general, the design of an anycast MAC should subsume the transmit power and rate control approaches in the physical layer. However, with a given physical layer design, loss of link will still be a reality, and anycasting can always play an important role in the design space.

2.3.1 Anycast Extension for 802.11

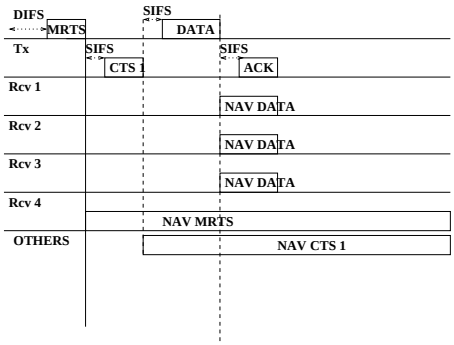
The anycast extension uses a similar handshaking protocol as in 802.11 DCF, but takes advantage of multiple receivers with the goal to transmit the frame to any one of them successfully. It can be thought of an anycasting scheme in the link layer. The routing layer computes multiple routes between the source and destination. We will describe this mechanism in the next subsection. At each hop, the routing layer passes on the multiple next hop information to the MAC layer. The transmitter now "multicasts" the RTS to these multiple next hops (it is actually a broadcast control packet as before). We will refer to the multicast RTS as MRTS; it contains all the next hop receiver addresses. Because of practical considerations (such as RTS packet size), we limit the number of next hops to use to a maximum of four.

The four next hops are assigned a priority order, which can be determined by the respective positions of their addresses in the MRTS packet. The priority can come from the routing or any lower layer. As an example for routing layer, the next hop leading to a shorter path to the destination gets higher priority, or the next hop that has fewer number of packets waiting in the interface queue gets higher priority. As an example for the MAC/physical layer, relevant statistics related to the amount of error correction can be used as an indicator for the quality of the link and hence to determine its priority. A combination of the above can also be used.

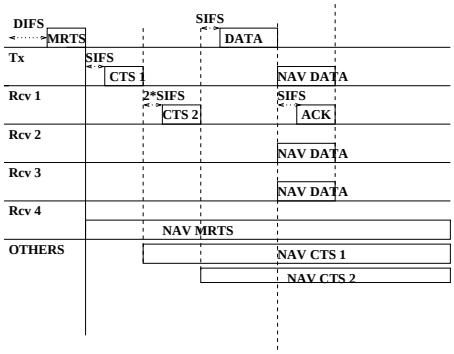
When an intended receiver receives the MRTS packet, it responds by a CTS. These CTS transmissions are staggered in time in order of their priorities. The first receiver in the order transmits the CTS after an SIFS, the second after a period equal to the time to transmit a CTS and $3\times$ SIFS, and so on. See Figures 2.2(a), 2.2(b), 2.2(c) for an illustration. Note that the staggering ensures that the CTSs are separated by at least $2\times$ SIFS period; thus they do not collide.

When the transmitter receives a CTS (which may or may not be the first CTS transmitted), it transmits the DATA frame to the sender of this CTS (which would be the highest priority receiver that responded) after an SIFS interval. This ensures that other, lower priority receivers hear the DATA *before* they send CTS — as the next one in priority will not send a CTS until another SIFS interval — and suppress any further CTS transmission. All such receivers then set their NAV until the end of the ACK packet. (The DATA packet carries this period in the header just in case these receivers missed the MRTS). See Figure 2.2(a) for an illustration when the very first CTS transmitted has been successfully received. We provide two other illustrations demonstrating the scenarios when the first CTS was not received, but the second was received (Figure 2.2(b)); and when all but the fourth CTS were not received (Figure 2.2(c)).

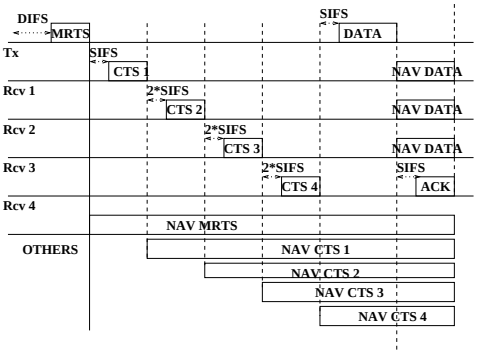
Any other node that hears the MRTS (*exposed* node), sets its NAV for the entire duration mentioned in the MRTS packet. This duration depends upon the



(a) 1st CTS is received.



(b) 2nd CTS is received.



(c) 4th CTS is received.

Figure 2.2: Time line showing the anycast extension of 802.11.

number of receivers (which can be a maximum of four) to which MRTS is being sent. For instance, if the number of receivers is k, the NAV is set to $k\times$ CTS + $(2k+1)\times$ SIFS + DATA + ACK time. This time is the maximum time needed for the data transfer to complete. Similarly, any node that hears any of the CTSs (*hidden* node) sets its NAV until the ACK period. For example, such a node upon receiving the i-th CTS, will set its NAV for the period $(2(k-i)+1)\times$ SIFS + $(k-i)\times$ CTS + DATA + ACK. See Figures 2.2(a), 2.2(b), 2.2(c).

If none of the CTSs are received successfully, the transmitter goes into a random backoff and then retries again with the same receivers. The random backoff procedure is exactly as in 802.11 except that in the experiments we have allowed a lower number of maximum retries – six instead of seven. This is because the possibility of failure is much less with multiple choices of the next hop.

Note that the protocol reduces to 802.11 when there is only one next hop receiver. This gives us an opportunity for a fair performance comparison. Also, note that when multiple next hops are indeed available and the CTS from the highest priority receiver is received successfully, this would be the same receiver sending CTS in an equivalent 802.11-based scenario. In this case again, the protocol behaves similar to 802.11, but it sets a longer NAV period for the hidden and exposed terminals. In this context, also note that in situations when multiple CTS's come back, all nodes in the vicinity of the receivers sending CTS's set up their NAV, while only the last one is involved in communication. The anycast mechanism in this manner increases the number of nodes that are exposed terminals and should therefore refrain from any communication. This can potentially reduce the network throughput. One way to cancel this NAV setup if the receiver is not involved in the communication is if the receiver sends explicit NAV cancelation messages. But, while the data is being sent to the last receiver, each of the other receivers would sense a busy channel and therefore they cannot engage in any transmission themselves. Thus, there is no easy way to resolve this problem. However, our simulation studies do show that even with large traffic diversity, anycast performs very well relative to 802.11. Thus, the harmful effect of silencing this nodes is not high enough to mask the benefit of the protocol.

It is possible that the fade state of the channel can change from the point when CTS is transmitted to when DATA or ACK is transmitted, causing the exchange to fail. But we claim that it is unlikely. The *coherence* period (T_c) of a fading channel defines the approximate interval the channel state remains very correlated or, in other words, does not change significantly [80]. T_c is approximately equal to the inverse of the Doppler frequency (f_m). From the values we have used in the previous section, it is easy to see that the coherence period is expected to be large enough for the DATA transmission to succeed if a CTS indeed has succeeded. As an example, for $f_m = 16$ Hz, T_c = 62.5 ms. Compare this with the time to transmit

a 1000 byte DATA frame. At 2 Mbps the transmission time would be 4 ms; at 11 Mbps it would be 0.73 ms.

It is obvious that the protocol benefits the most when a fair number of choices for the next hop is available. This increases the probability that the data transmission takes place successfully. Thus the effective operation of the protocol is dependent on a routing layer being able to compute enough redundant routing paths. The next subsection discusses the design choices we make in the routing layer that plays a significant role in the performance.

2.3.2 Design of Multipath Routing Layer

Multipath routing protocols have been explored in mobile ad hoc networks to maintain multiple redundant routes to provide fault tolerance and also for load balancing [73, 67, 61]. Availability of multiple routes reduces route maintenance overhead as routes need to be recomputed only when all available routes fail. Also, it is possible to forward data packets over multiple routes simultaneously (dispersity routing [62]) to provide more traffic diversity and to reduce load on each individual route [73].

We will use an on-demand multipath routing protocol to provide the MAC layer with multiple next hop links. Specifically, we will use AOMDV [61], a multipath extension of a popular on-demand single path routing protocol AODV [74, 26] that is based on the distance vector concept. In AODV, when a traffic source needs a route to the destination, it initiates a route discovery by flooding a route request (RREQ) for the destination in the network, and then waits for the route reply (RREP). When an intermediate node receives the first copy of a RREQ packet, it sets up a reverse path to the source using the previous hop of the RREQ as the next hop on the reverse path. In addition, if there is a valid route available to the destination, it unicasts a RREP back to the source via the reverse path; otherwise it rebroadcasts the RREQ packet. Duplicate copies of the RREQ are discarded. The destination, on receiving the first copy of a RREQ packet, behaves the same way. As a RREP proceeds to the source it builds a forward path to the destination at each hop.

In AOMDV, a node can form multiple reverse routes to the source using the duplicates of the RREQ packet; but it still rebroadcasts only one RREQ. Additionally, the destination or any node having a path to the destination may choose to respond to multiple RREQs it receives via multiple reverse paths already formed. As presented in [61], AOMDV uses mechanisms to ensure link disjointness of the multiple paths; however, in this work we have turned off these mechanisms to allow overlapped routes. The benefit is that removal of the disjointness constraint automatically provides many more paths. We will see later that more paths are

beneficial for performance.

Note that this is a significant departure from multipath routing techniques that try to guarantee some form of disjointness [61] to ensure independence of path failures. However, this is important only when link failures are viewed as a more "stable" event, i.e., links change state (from off to on, for example) in the time scale of route changes in the routing protocol. In the model we are interested in, link failures are transient, and links are expected to change state within a much shorter time scale. This may not be true, however, when link failures may be caused by mobility. In the simulation experiments we report later, we still see significant improvement with overlapped paths even in mobile scenarios, making it a sensible design choice.

Note that in our model, the routing packets also face the same fading channel as the data packets. Thus, transient link failures impact the route discovery process, which is unavoidable. Routing may also form next hop links that could be too weak normally, but just had been strong enough during route discovery. We have made simple optimizations to AOMDV to make routing more efficient. As an example, the RREPs are broadcast instead of unicast. This gives an opportunity to at least some of the next-hop neighbors on the reverse path to receive the packet successfully, and form the forward paths. Here again, we rely on the assumption of lack of correlation between the channel state of different links on the same node. The traditional timer-based route expiry in AODV or AOMDV is not used, because this may delete unused, but possibly valid routes. Other key techniques in AOMDV, such as use of sequence numbers for loop prevention and determining freshness of routes, and the route error-based route erasure process are not altered.

AOMDV uses a sequence number-based method (similar to AODV) to prevent looping and also to eliminate stale routing information. AOMDV is flexible enough to provide disjoint (link- or node-disjoint) or overlapped routes. Naturally, allowing overlapped routes gives a large number of routes providing the protocol as many forwarding choices as possible at each hop. In prior work [61], however, we have explored disjoint path routing as the impact of fading was not analyzed, and links failed primarily because of mobility. This ensures that link failures most often are caused by mobility and thus they are not very transient. Thus, overlapped routes were not useful, as a single link failure may cause failure of many routes at the same time. In the following section simulation results will show that use of disjoint paths (i) bring down the overall performance of either protocol and (ii) the relative advantage of the multiple next hop extension almost vanishes. One other design choice we need to make, is whether to allow paths that are too long relative to the shortest paths. This issue presents a trade-off that must be carefully orchestrated. To understand this, take an example where 802.11 fails to transmit on a next hop link because of fading, causing it to retry. Assume that we are using the shortest

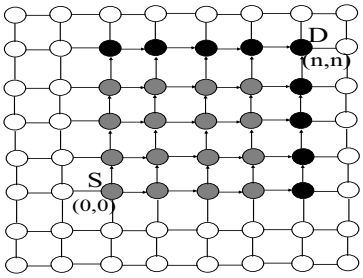


Figure 2.3: Grid network for analyzing packet delivery probability.

path routing and the data packet is still k hops away from the destination needing at least k more transmission attempts for the packet to reach the destination. If we use anycast instead, under an identical scenario, the protocol will choose an alternate next hop. Assume that the current node is $k + l$ hops away from the destination via this alternate next hop. This means that even though this transmission is successful, the packet still needs at least $k + l$ transmission attempts to reach the destination. Thus, the 802.11 transmission must fail at least l times for the multipath extension to be of any value. Of course, $l = 0$ is an ideal possibility; but this may reduce the number of alternate paths drastically. We empirically evaluated various possibilities for l, and found that $l = 1$ to be a reasonable choice. Thus, we allow only those paths to be formed in the routing table that are at most one hop larger than the shortest path. The value of l can be a parameter of the protocol. It is worthwhile to mention here that in [67] the authors also have noted that limiting the path length difference (l) is a useful optimization in multipath routing.

2.4 Performance Evaluation

We present three sets of performance results. The first set builds a simple model to analytically evaluate packet delivery probability in a grid network when single or multiple next hop links are available. The second set presents experimental evaluation on the Berkeley motes platform in a similar grid network. Both these networks provides valuable insights, even though they are restricted in some form — because of tractability reasons for the analytical model and logistical reasons in the experimental motes testbed. The third set of results use *ns-2* [34] based simulations, that do not have any of these restrictions and can use elaborate scenarios.

2.4.1 Analysis for a Grid Network

Consider a two dimensional grid network as in Figure 2.3 with 4-nearest neighbor connectivity. This model is representative of networks with a rich set multipaths such that many forwarding options are available. This network model is simple enough to study closed form expressions for packet loss probabilities for multihop routing with unicast or anycast forwarding. Suppose, nodes S and D are the source and destination nodes respectively. Without loss of generality assume that the coordinate of S is $(0, 0)$ and that of D is (n, n). The shortest path length between S and D is $2n$. The nodes falling on the shortest paths are shaded. 2 next hops are possible on all hops on all shortest paths *except* on the boundary nodes on the $n \times n$ rectangle beyond n hops from S. These nodes are shaded in dark in Figure 2.3. On these boundary nodes, only 1 next hop is possible.

Now, assume that the probability of a link loss is p and the probabilities are independent. If only a single next hop is used for packet forwarding and their is no retry, the packet drop probability at each hop is p. Thus, the probability P that a packet from S will reach D is given by,

$$P = (1 - p)^{2n}.$$

If multiple next hops are available (in this case the maximum is a modest 2), the packet drop probability at each hop is either p (if there is only one next hop) or p^2 (if there are 2 next hops). Note that 2 next hops are available for each of the first n hops; beyond this, the boundary nodes can provide only 1 next hop, but the rest of the nodes can still provide 2. Thus, in the last n hops, each hop can undergo a packet loss with probability p or p^2. To determine the combined probability, we need to evaluate the proportion of paths that go through boundary and non-boundary nodes for each hop beyond the first n hops.

If a node (i, j) is at a distance l from S (i.e., the node is at the l-th hop), $i + j = l$. Simple combinatorics can determine that the number of (shortest) paths of length l from S to node (i, j) is

$$\frac{(i + j)!}{i!j!}.$$

A node could be a boundary node only if $l \geq n$. A boundary node on a shortest path must satisfy i or $j = n$, and a non-boundary node on a shortest path must satisfy i or $j = (n - 1), (n - 2), \ldots, (l - n + 1)$. This determines that the number of such paths going through *all* boundary nodes at hop $n \leq l < 2n$ is given by

$$B(l) = \frac{2(l!)}{n!(l - n)!},$$

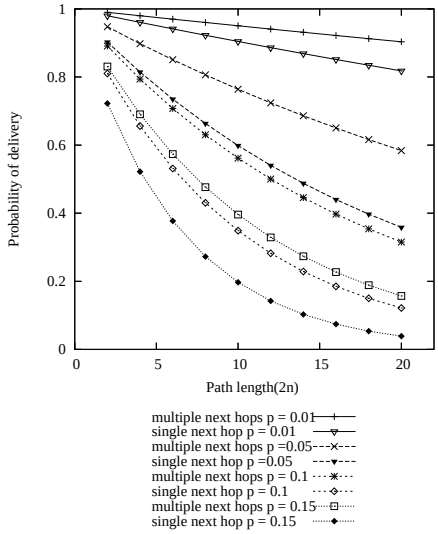


Figure 2.4: Packet delivery probabilities for the grid network of Figure 2.3 with single (unicast) and multiple next hop forwarding (anycast).

the factor 2 coming from the fact there are two boundary nodes at each hop. Also, the number of paths going through all non-boundary nodes at hop $n \leq l < 2n$ is given by,

$$NB(l) = \sum_{k=1}^{2n-l-1} \frac{l!}{(n-k)!(l-n+k)!}.$$

Since all paths are equally likely in our model, at hop l a boundary or a non-boundary node will be used simply in proportion to the number of paths going through them. Accordingly the packet drop probability at hop l will be either p or p^2, respectively. Combining all these, the probability P that a packet from S will reach D is given by

$$P \;=\; (1-p^2)^n \times \prod_{l=n}^{2n-1} \left\{ 1 - \frac{B(l)p + NB(l)p^2}{B(l) + NB(l)} \right\}.$$

The first term is for the first n hops and the second term is for the following n hops.

Figure 2.4 plots the packet delivery probability P versus the path length ($2n$) for different link loss probabilities (p) for both single (unicast) and multiple next hop forwardings (anycast). Note that even though only a maximum of 2 next hops are used, there is a significant relative improvement in delivery probability with multiple next hops, particularly as the path length increases. Larger number of next hop possibilities should improve the probability further.

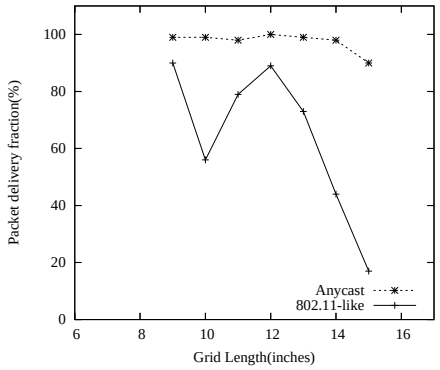


Figure 2.5: Packet delivery fraction in the 4×4 Berkeley motes testbed with S-MAC protocol stack.

2.4.2 Evaluation on Experimental Testbed

We implemented the anycast protocol on Berkeley motes platform, manufactured by Crossbow Technology [1, 4]. While our original intention is to use anycast as a replacement for 802.11-based MAC layer protocol, implementing anycast on 802.11-based hardware requires modification of the firmware in the network interface card. This requires working with the chipset and/or card manufacturers.However, a proof-of-concept implementation is possible on the Berkeley motes platform, where link layer protocols are implemented in software as a part of the protocol stack in the TinyOS operating system [4, 40]. We did a proof of concept implementation in software using the TinyOS [4, 40] platform on Mica motes. We used the Mica platform for our experiments that uses an Atmel ATMEGA series microcontroller (4MHz, 8-bit) as the processor and an RFM TR1000 transceiver operating at 916MHz as the radio interface. In the Mica platform the radio bit rate limited to about 50 Kbps. This speed is CPU limited, as the protocol processing happens at the sole processor on the mote.

For a meaningful implementation, we used the S-MAC protocol stack [104, 105] developed in USC/ISI. S-MAC replaces the MAC and PHY layer implementations in the original TinyOS network protocol stack and provides a flexible architecture to develop new MAC protocols by providing a flexible packet format and clear separation between the MAC and PHY layers. The original S-MAC implementation [104, 105] uses a protocol very similar to the IEEE 802.11 DCF for channel access operating in the ad hoc mode, including implementations of inter-frame spacings, physical and virtual carrier sensing, backoffs and retries, RTS/CTS/DATA/ACK based handshake, and network allocation vectors. It also uses several innovations for energy management, which we turned off to make the protocol very similar to

regular 802.11.Since the entire implementation is in software, this provides an excellent platform to experiment with new MAC protocols albiet with low data rate radios.

We modified the S-MAC protocol stack to implement anycast by modifying the base 802.11-like implementation. In the test scenario we placed 16 motes in a square 4×4 grid configuration as in Figure 2.3. Back-to-back data packets are transmitted from one corner of the 4×4 grid to the opposite corner. Routes are manually set up exploring all possible paths (similar to the analysis in Section 2.4.1). Figure 2.5 shows the relative packet delivery performance of the 802.11-like protocol and our anycast implementation in the S-MAC protocol stack. The length of a side of the unit grid is varied to provide an independent means to control the radio performance. Increasing the length beyond a threshold makes the signal strength fairly weak and radio performance very much prone to multipath fading and other noise. The experiments were performed in a small laboratory room in a computer science department in its natural state, i.e., with usual furniture, people moving around and possible sources of radio noise; but no noise was intentionally created to influence the experiments.[4] An average of a large number of experiments is reported in Figure 2.5. The positions (including pose) of the motes were kept unaltered across experiments with the same grid size. Note the poor packet delivery performance for the 802.11-like protocol as the grid size is increased.[5] Anycast provides an excellent performance over the entire range.

2.4.3 Simulation Model

We used the *ns-2* [34] simulator with the AOMDV protocol [61] in the routing layer and the anycast protocol in the MAC layer. As mentioned before, the AOMDV model used here allows overlapped paths; and only those paths are used that are at most one hop larger than the shortest path the protocol is able to find. With 802.11, the traditional forwarding model is followed. The next hop link on the shortest path is attempted first. Upon failure (i.e., when maximum retry count is exceeded), this link is marked down and the next shortest alternative is used. A route error is

[4] We indeed have seen significant improvements in performance of the 802.11-like implementation in remote, quiet and open outdoor environments, where not much link diversity could be obtained to make anycast significantly meaningful. Such environment also provided a much larger radio range.

[5] We also noticed some amount of unstable performance for the 802.11-like protocol for lack of diversity. For example, at certain grid lengths (10 and 11 inches) the performance was relatively poor, possibly due to some multipath effects created at these lengths.

generated only when all alternatives are exhausted. In the anycast protocol, the next hop priorities are generated based on path lengths alone.

The traffic model uses CBR (constant bit rate) traffic with randomly chosen source-destination pairs. A traffic rate of 1 packet/sec (512 byte packet) per flow was used in the experiments. Load is varied by varying the number of flows (number of sources). For each packet delivered to the destination the number of hops it traveled is logged, and its average statistics is used as a parameter in the performance plots. For mobile experiments, the popular *random waypoint* mobility model [23] is used. Here, a mobile node alternately pauses and moves to a randomly chosen location with a constant but randomly chosen speed. The pause times and the average speed are parameters of this model.

The radio propagation model uses the two-ray ground reflection path loss model [80] for the large-scale propagation model (as in the *ns-2* distribution), augmented by a small-scale model modeling Ricean fading as presented in Subsection 2.2.1. The ns-2 extension provided by the authors of [17] has been used for the fading model. Here, the Ricean fading is modeled using an efficient simulation technique that also captures the time correlation of the signal envelop depending on the Doppler spread created by the relative motion of the transmitter and/or receiver (could also be caused by the motion of reflecting objects). The technique employs a lookup operation in a pre-generated dataset containing the components of the time-sequenced fading envelop.

The original implementation in [83] uses the simulation time instant to index into a channel table that causes all next hop links from a node to undergo exactly similar fading which is unrealistic. In order to make them uncorrelated, the index uses both simulation time (to provide time correlation) and the next hop node id (to prevent correlation between channel conditions on all next hops links). Similar "corrections" for the same the code base has also been reported in [41] in the context of multi-rate MAC implementations. A value of 5 dB for the Ricean K factor has been used unless otherwise stated. For stationary networks, a max relative velocity v of 1 m/sec has been used to compute the Doppler shift f_m.

Three different network models have been used for evaluation each with 200 nodes and various number of traffic flows: The first model is a stationary grid network similar to Figure 2.3. Here, the grid is, however, rectangular 40×5 with the distance between adjacent nodes in the grid being 100m. Note that the nominal radio range (without fading) being about 250m, it gives a fair number of routing paths between random pairs of source and destination. We ran several simulations with various numbers of sources. Since the distance between the source-destination pairs is a sensitive parameter (as we have seen in the model developed in the previous subsection), we have controlled the random selection of source and destinations in a way to give us specific values for the"shortest" path lengths (in hops).

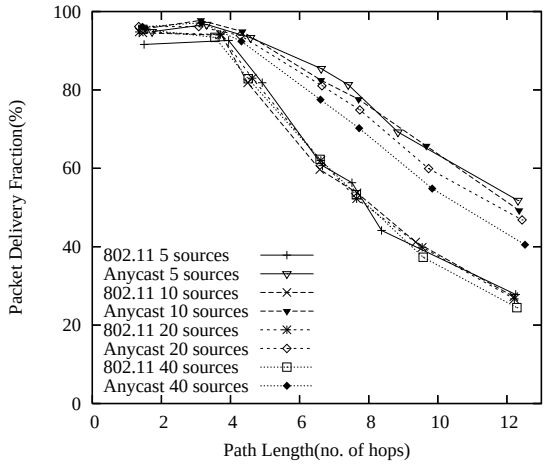
The second model uses a network of 200 randomly positioned stationary nodes in the same area (4000m $\times$ 500m). Similar experiments were run by controlling the random choices of source destination pairs so that their shortest path lengths fall close to pre-selected specific values. The third model uses the same number of nodes in the same area; but now the are mobile and follow the random waypoint mobility model. The pause times and speed are varied to control the mobility. Because of mobility, it was not possible to control the hop-wise distance. All simulations are run for 900 simulated seconds. Each data point represents the average of 5 runs.

2.4.4 Simulation Results in Grid, Random and Mobile Networks

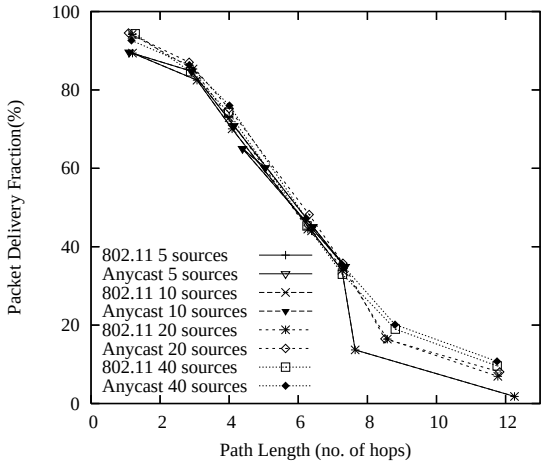
Figure 2.6(a) plots the average packet delivery fraction for the stationary grid network model for the two link layer models. As expected, the delivery fraction goes down with increase in path lengths with anycast performing better – with the performance differential increasing with the path length. A performance gain of up to a factor of 2 is observed for large path lengths.

Note also that the anycast performance is going down with increase in number of traffic sources, while for 802.11, the performance is almost independent of this parameter. It turns out that with more traffic diversity the route discovery is unable to provide a large number of routes because of loss of route request packets due to increased interference. Note that route request packets are broadcast packets and thus they are more susceptible to fading and interference as they cannot be retransmitted. Figure 2.9 demonstrates this effect, where the percentage of MRTSs that have 1,2,3 or 4 next hops are plotted against number of sources. Note the increase in unicast MRTS (i.e., MRTS with only one next hop receiver) with traffic, and corresponding decrease in MRTSs with 3 or 4 next hops. When routing is modified to restrict the routing to discover only link-disjoint paths, the performance improvement with anycast is almost non-existent. Figure 2.6(b) demonstrates this. This figure uses the same simulation runs as before, only with a change in routing. We investigated the reason for the lack of performance gain with disjoint path routing. As alluded to before in Subsection 2.3.2, the major cause is lack of sufficient number of next hops. Figure 2.10 confirms this hypothesis by comparing the fraction of unicast MRTSs (MRTSs with only 1 next hop) for these two variations. Note the large number of unicast MRTS for disjoint path routing relative to the overlapped paths case, showing that multiple next hops are not often available for disjoint path routing.[6] From this point onward, only overlapped path was used

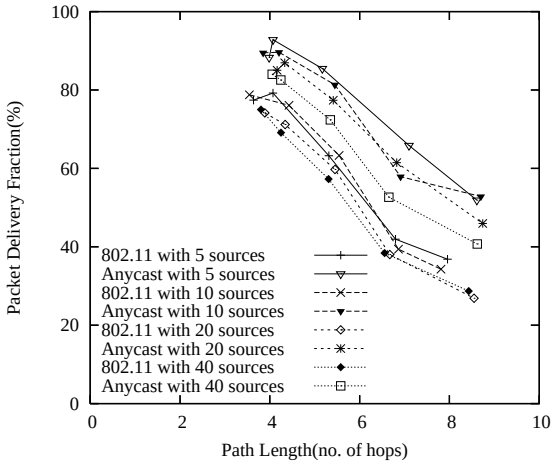
[6]It may appear that disjoint path routing means that only the source has more than one next hop and not any of the intermediate nodes. However, the protocol used here follows the disjoint path definition in [61] where a node I on the path



(a) Stationary grid network with overlapping path routing.



(b) Stationary grid network with disjoint path routing.



(c) Stationary random network with overlapping path routing.

Figure 2.6: Packet delivery fraction with 802.11 and anycast in static networks.

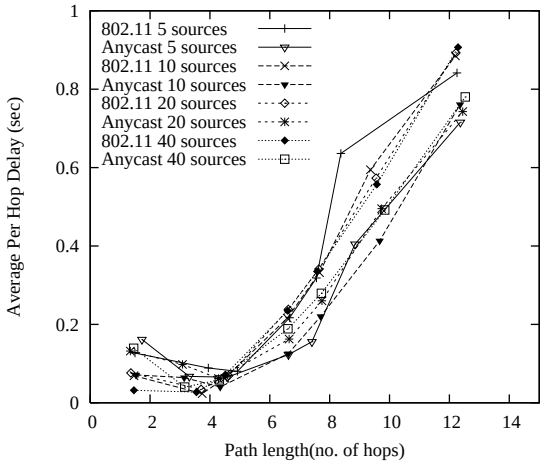


Figure 2.7: Average per hop delay with 802.11 and anycast in static grid network with overlapping path routing.

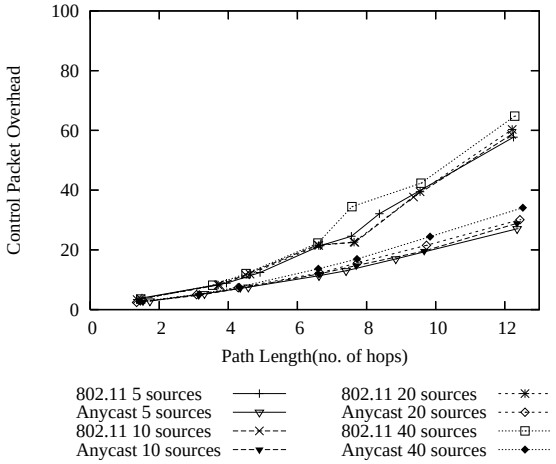


Figure 2.8: Control packet overhead in 802.11 and anycast in static grid network with overlapping path routing.

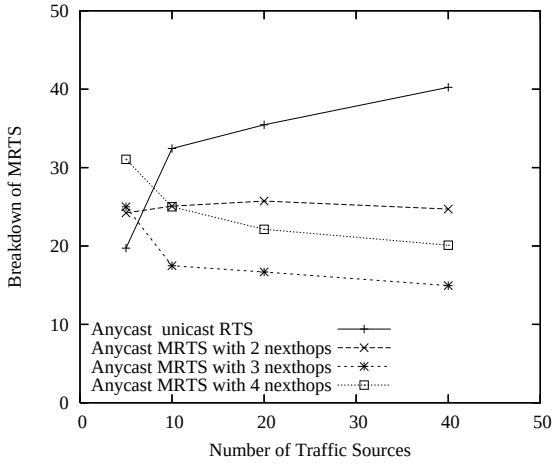


Figure 2.9: Percentage of MRTS packets with different numbers of next hops in stationary grid network (average path length is approx 6).

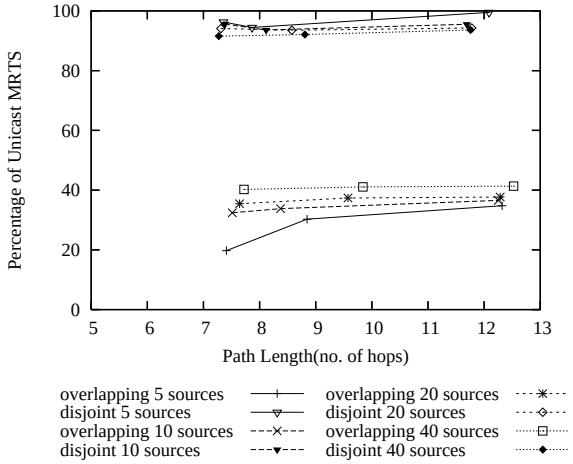


Figure 2.10: Percentage of unicast MRTS packets in the stationary grid network for disjoint path and overlapping path routing.

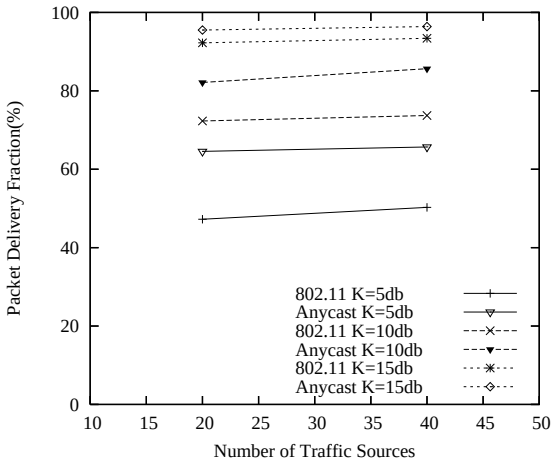
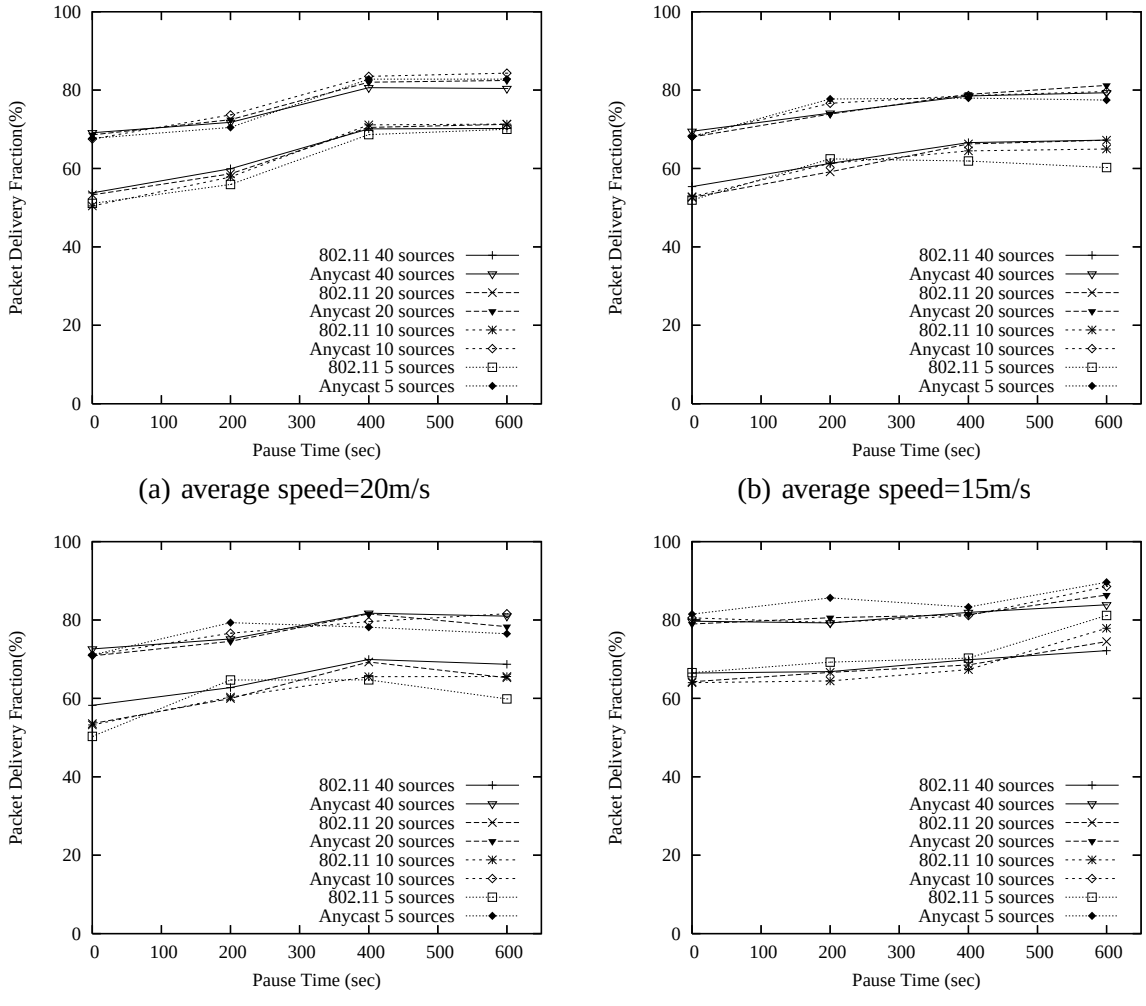


Figure 2.11: Affect of Ricean K factor on packet delivery fraction.

for routing. Figure 2.6(c) shows the packet delivery performance in the stationary random network. Note again that performance improvement varies from about 20% to upto about a factor of 2 for large path lengths. Because of the randomness involved the hop-wise distances could not be varied over as wide a value as in the grid network. We also analyzed the impact of the changes in fading in this set up. Figure 2.11 shows packet delivery fraction for a specific set of scenarios with 20 and 40 sources when the hop-wise distance is about 4. Here,the Ricean K parameter is varied which influences the relative amplitude of the dominant signal component. Note that the dominant component is relatively stronger (larger K value) the impact of fading is less. Thus, with smaller K, the absolute performance degrades, but the performance differential between multiple and single next hops increases. Finally, we will look at mobile scenarios with different mobility. Figure 2.12(a) presents the packet delivery performance in a mobile scenario with average speed of 20 m/s. Note that anycast is performing about 25–40% relative to the unicast performance. In these set of experiments the impact of increasing load (number of sources) is minimal. This is because of relatively small average path lengths (about 3.5) realized in these experiments. Figure 2.12(b), Figure 2.12(c) and 2.12(d) show a scenarios in which average speed of each node is 15 m/s,10 m/s and 5 m/s respectively. 802.11 delivers less than 60% of the packets at high mobility while anycast is able to deliver upto 75% of the packets. At 5 m/s, anycast delivers 80% of the packets while 802.11 is barely able to cross the 60% mark.

P_1 from S and D is allowed to form an independent path P_2 to D which is link-disjoint from P_1.



(a) average speed=20m/s (b) average speed=15m/s

(c) average speed=10m/s (d) average speed=5m/s

Figure 2.12: Packet delivery fraction for 802.11 and anycast in mobile scenarios

2.4.5 Comparison of Overheads in Anycast and 802.11

In this section we have presented results that compare overheads in the anycast and 802.11 protocols. We have observed from the analysis in Section 2.4.1 as well as the packet delivery fraction graphs in the previous section, that the benefits of anycasting is more prominent when the path length between the source and destination is longer as opposed to when the paths are less than four hops in length. We can obtain a larger range of path lengths in the grid networks than in random or mobile networks where path lengths are difficult to control due to randomness and mobility. In order to show the overheads of the two protocols over a large range of path lengths as well as for the sake of brevity, we will present the overhead results for static grid networks only. We have seen that the other scenarios also follow similar trends.

We have compared average per hop delays incurred by packets that were successfully received at the destination. This is computed as the ratio of the average delay incurred by the packets and the average number of hops traversed from the source to the destination. We observe in Figure 2.7 that this delay in the anycast scheme is higher than in 802.11 when the paths are on an average less than four hops long. We observe here that simultaneous transmission to reach any nexthop in anycast incurs more delay than retrying the same path as in 802.11. This may be due to the lack of path diversity when the distance between source and destination is less. However, as path lengths increase, packets in the anycast mechanism show lower delay than in 802.11. At path length of approximately 12 hops, anycast shows upto 12% lower delay than 802.11.

In both anycast and 802.11 protocols, the traffic due to control packet exchange is a source of overhead and in anycast, the additional CTS packets might cause even more overhead. In order to understand the effect of additional control packets exchanged in anycast, we will analyze the control overhead of the two protocols. We compute the control overhead as the ratio of the total number of RTS and CTS packets sent along the entire path from the source to the destination and the total number of data packets that are successfully received at the destination. We present the result in Figure 2.8. As expected, the control overhead is low when the path length is small but it increases as the data packets have to be routed through more nodes to reach the destination. It is interesting to note that the control overhead in anycast is actually lower than that in 802.11 and as the path length increases, the difference becomes wider. In 12 hop paths, 802.11 sends more than 60 control packets for every data packet that reaches the destination, while anycast sends only around 30 control packets per data packet. Note that in an ideal scenario, for 12 hop paths, the number of control packets per data packet would be 24, two packets for each hop in the path. This result clearly shows that the multiple CTS transmissions

in the anycast protocol presents a much lower overhead than the multiple RTS/CTS sent in 802.11 as it retries several times before succeeding in sending packets to the next hop node.

Our experimental results establish the benefits of anycast in practical wireless networks that have far from ideal channel conditions. In wireless networks where the path lengths are larger than four hops, the anycast mechanism not only provides a higher packet delivery fraction but does so with lower packet delays and exchanges less number of control packets as compared to the 802.11 protocol.

2.5 Related Work

In [57], a combination of forwarding and MAC layer protocol called *selection diversity forwarding* has been proposed. Here, the data frame is multicast to a set of candidate nodes, each of which send back ACK control packets. Then only one node is chosen from this set by the forwarding node and issued a *forwarding order* control packet, which is again acknowledged. This is the node that will forward the data packet further; and others will discard the packet. Note that there is no channel reservation such as 802.11 or our anycast extension. Data packets can easily collide, and the overall exchange takes longer as the forwarding order has to wait to for all ACKs. The criterion to choose the forwarding node depends on the upper layer protocol. For example, the forwarding node could be the one that provides the maximum forward progress in geographic forwarding. Selection diversity forwarding has been shown to perform better than fixed forwarding mechanisms, such as NFP (nearest with forward progress) or MFR (most forward with fixed radius) for Rayleigh fading channels.

Several recent articles build on the 802.11 standard to estimate the channel condition and automatically adapt the sending bit rate to match the channel conditions. However, they still use single next hop, and use the unicast forwarding model in 802.11. In the RBAR protocol [41], the receiver estimates the channel condition by the physical layer analysis of the RTS packet and determines the best rate to send the data frame. The control packets are sent using the base (lowest) rate so that they are always successfully delivered. The OAR protocol [17] extends this idea to send multiple back-to-back packets when the channel condition is determined to be good. OAR also takes care to ensure fairness, as there is a chance in this protocol that links with better channel conditions can get more share of the channel bandwidth.

In [77] an adaptive transmission protocol is used that adjusts the power and code rate of the transmitted signal to adapt to the channel conditions. But this scheme does not work when a poor quality link has not been used by the routing

protocol for some time. The work suggests an alternate forwarding technique dependent on multipath routing that alters routing paths to discover links that may have improved recently.

Three recent papers also motivate use of anycasting in the MAC layer. In [27] authors motivate anycast as a general-purpose MAC layer method to take decisions on packet forwarding in short time scales. They describe potential use of anycast from the point of view of improving spatial reuse and reducing interference. They describe applications with power-controlled multiple access and directional antenna. However, since this is a position paper, no performance evaluation is reported. In the same forum, an "opportunistic" routing mechanism is presented [20, 21], which is very similar in spirit to the selection diversity forwarding work described earlier. Another protocol called GeRaF [108] also contains similar ideas, but has been specifically applied for geographic forwarding. Here, the interest is more on modeling, rather than a practical implementation.

Two recent studies [45, 100] used a protocol similar to ours in spirit, however, for a different goal. These protocols exploit multiuser diversity in the context of an access point-based system. Similar exploitation of multiuser diversity was also explored earlier in channel state based scheduling [19] protocols. In contrast, we exploit path diversity.

2.6 Conclusions

We have proposed an anycast mechanism at the link layer that forwards packets to the best suitable next hop link to enable efficient packet forwarding on a multihop route. This mechanism is dependent on the availability of multiple next hops, which could be computed by a multipath routing protocol. We have designed the link layer protocol as an extension of the popular IEEE Standard 802.11 and carried out an extensive performance evaluation using both an experimental testbed and detailed simulation modeling. The anycast protocol provides a significantly better packet delivery relative to 802.11 in a variety of ad hoc network models, both regular and random, stationary and mobile. The performance differential was observed to increase when path lengths increase.

Note that when multipath routing is combined with anycast, the forwarding decisions taken at each hop is a local decision. This can easily increase the overall path length unless the forwarding is orchestrated carefully (see the discussion on the value of l at the end of section 2.3.2). Some mechanisms to do this on a per-packet basis has been discussed in [27].

Another point of concern is the operation of the routing protocol. The routing protocol itself suffers from the transient weak channel conditions, and may fail

to discover links that (transiently) fail to deliver routing messages. This does not seem to be a significant problem in the our simulations. However, we anticipate a different method of delivery for routing messages can improve performance (such as using higher transmit power to counteract fading).

Chapter 3

Applications of Anycast in Multichannel and Directional Antenna Networks

3.1 Introduction

In the previous chapter we discussed *anycast*, a new MAC protocol for wireless network that delivers good results in the face of multipath fading and interference in comparison to the 802.11 protocol. In this chapter we will discuss an application of *anycast* in directional antenna and multichannel networks.

It is well known that wireless networks have a limited bandwidth available for communication. This provides a motivation to study network designs which improve the bandwidth utilization. A popular approach is to use multiple channels for communication, known as multichannel networks. Another network model called directional antenna network, uses directional antennas so that the transmission is confined to selected directions with respect to the transmitter, instead of all directions as in regular (omni-directional) networks. Both these network types can potentially improve the bandwidth utilization by increasing the spatial reuse of the available bandwidth.

In multichannel and directional antenna networks just as in regular wireless networks, nodes suffer from deafness and hidden terminal problems. Deafness is said to have occurred when a node makes several futile attempts to communicate with a neighbor who is busy in another transmission and thus is unable to respond to the sender. The hidden terminal problem occurs when a node starts a transmission by incorrectly assuming that the medium is free when in reality there is an ongoing transmission in the neighborhood. The control packet exchange mechanism in 802.11 medium access control protocol (MAC), alleviates the hidden terminal problems in regular networks. This mechanism assumes a single channel network with omni-directional transmissions. Due to the inability of nodes to listen for transmissions in all directions or in all channels in directional antenna and multichannel

networks, deafness and hidden terminal problems may be more rampant in these networks if the 802.11 protocol was used in the MAC layer. In the previous chapter, anycast was proposed for single channel networks to combat multipath fading, where it was able to alleviate losses due to fading by exploiting path diversity. We will see now that by exploiting the same path diversity, anycast is able to alleviate the deafness and hidden terminal problems in both multichannel and directional antenna networks. We will first discuss the anycast application in multichannel networks in section 3.2 followed by directional antenna networks in section 3.3.

3.2 Multichannel Networks

We will first describe the network model that we consider for multichannel networks followed by the description of the base 802.11 like protocol that we extend using the principles of anycast which is followed by an explanation of the anycast extension.

3.2.1 Network Model

While there can be many designs for a multichannel network, we have adapted a "quiescent channel" model that appeared in [87]. In this model, each node in the network is assigned a channel called a quiescent channel. This is the channel to which the node listens to when it is not in transmit mode. This channel assignment is well known to all nodes in the network or can be derived from the node addresses. All channels are used for data transmissions which in a resource constrained network that has a small number of channels, is a more desirable design. Given this network model, we will now describe the receiver directed transmission (RDT) scheme [87], which is a simple adaptation of 802.11 in multichannel networks with the quiescent channel model. We will then use anycast mechanism with RDT to alleviate the deafness and the hidden terminal problems.

3.2.2 Receiver Directed Transmission

In RDT, in order to transmit a packet to the next hop receiver, the transmitting node must switch to the receiver's channel and perform the CSMA/CA mechanism as in 802.11. If this backoff procedure is completed successfully and the medium is still free, the transmitter performs the RTS/CTS exchange with the receiver in that channel. All overhearing nodes invoke their virtual carrier sensing mechanisms. The virtual carrier sensing mechanism in RDT is achieved by maintaining different network allocation vectors for separate channels. Thus, the overhearing nodes set

the NAV corresponding to the channel in which transmission is heard. We distinguish this NAV from the one in regular networks by renaming it as channel NAV or CNAV. Nodes cannot participate in any transmission on a channel as long as the CNAV for that channel is set, but at the same time, nodes are free to switch to another channel for which the CNAV is not set and contend for transmission in that channel. This capability of parallel transmissions can potentially increase the network throughput by a large amount.

We note that due to the node's inability to listen to all channels at the same time, it may not have the current state of the channel it intends to transmit in. Thus, when a node switches to a new channel for transmission, it may inadvertently act as a hidden terminal causing collision for an ongoing transmission. Similarly, it can suffer from the deafness problem if the intended receiver happens to be busy in another transmission.

3.2.3 Anycast Extension of Receiver Directed Transmission

The anycast mechanism is capable of alleviating the deafness and hidden terminal problems in RDT by exploiting path diversity in the transmission channel. The multipath routing layer may be instrumented to maintain multiple paths on each channel in the network, and provide these node addresses to the MAC layer. Thus, in anycast, the transmitting node switches to the receivers' channel and multicasts a RTS packet to multiple potential next hop receivers in that channel and waits for a CTS. Reception of CTS from any one of the next hop nodes indicates that the channel has been reserved, thus, the transmitter sends data to the receiver from which it received CTS. In case the transmitter did not receive CTS from any nexthop receiver, it retries upto 6 times.

We can see from the protocol description that, anycast would be more successful in alleviating the deafness and hidden terminal problems, because it tries to negotiate medium access simultaneously with more than one nexthop nodes. This parallel negotiation process greatly increases the probability of success. Note that, the multichannel anycast protocol is similar in principle to its single channel counterpart and thus we can use the same protocol stack without changes in the hardware in both networks.

3.3 Directional Antenna Networks

We will now proceed to discuss the application of anycast in directional antenna networks. We will first describe the network model and the directional antenna design that we consider in our work. Description of the base 802.11 like

directional antenna model and the anycast extension follow next.

3.3.1 Network Model

We have studied an "electrically steerable antenna" which can change the antenna direction through beamforming. The same antenna model was used in [94]. The only difference is that we have used eight antenna directions with a beamwidth of 45^o each. This antenna is also able to transmit omni-directionally. We further assume that the antenna gain is same in both omni and directional modes. This may be easily achieved by reducing the transmit power when transmitting in the directional mode. Nodes are able to determine the direction of an incoming transmission by measuring the angle of arrival of the strongest signal. This information provides the relative direction of next hop neighbors and this direction information is cached at the routing layer along with the routes to various destinations. Having described the network model we will now proceed to discuss the directional virtual carrier sensing (DVCS) [94] protocol followed by the anycast extension.

3.3.2 Directional Virtual Carrier Sensing

In DVCS, if a node is idle, it switches its antenna to omni-directional mode in which it can hear transmissions from all directions. When a node needs to transmit a unicast packet to a receiver, and it is aware of the direction of the receiver, it invokes the CSMA and the backoff mechanism during which if the node does not hear any transmission from the intended receiver's direction, it beamforms the antenna to that direction and sends a RTS toward the receiver in that direction. The receiver upon receiving this RTS, orients its antenna in the direction from where the maximum signal strength is received, and sends a CTS in that direction provided that it senses a free medium in that direction. A successful RTS/CTS exchange is followed by data/ACK exchange in the same manner as in the 802.11 protocol. Nodes that overhear RTS/CTS exchange must invoke their virtual carrier sensing mechanism. Nodes maintain separate network allocation vectors for different antenna sectors instead of a single vector. We distinguish this NAV from the NAV in 802.11 by naming it as directional NAV or DNAV. Thus, when making a decision to contend for the medium, nodes check if the DNAV for the direction of transmission is set. If this is not the case, the node is free to contend for the medium in that direction. Otherwise, it must wait until the DNAV expires. Meanwhile, the node is still allowed to transmit in those directions for which the DNAV is not set.

When a node switches from directional transmission or reception mode to the omni-directional mode, it is possible that it has missed some control packet exchange that took place while it was in the directional mode. Thus, the node no

longer has the current state of the medium. This may lead to the hidden terminal problem. Also while a node is busy in transmission or reception from a direction, a neighbor being unaware of this state might try to communicate with this node from a different direction. This is the well known deafness problem occurring in directional antenna networks. We will see in the next section how *anycast* is able to alleviate these problems.

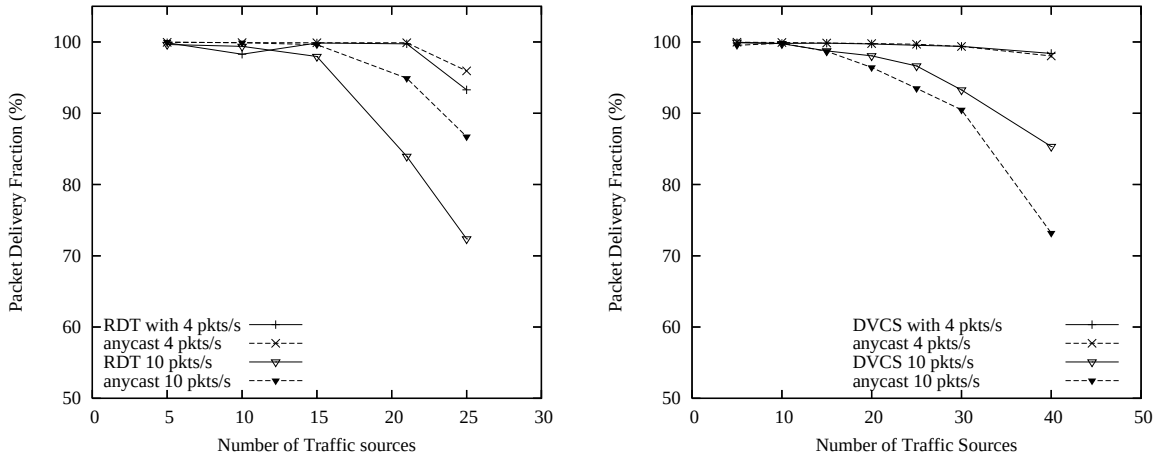
3.3.3 Anycast Extension of Directional Virtual Carrier Sensing

Once again we note that in anycast, the multipath routing protocol may be able to provide more than one next hop neighbor for forwarding data to the destination. The routing layer may be instrumented to maintain different paths for different directions (antenna orientations) and provide multiple next hop options in a particular direction to the MAC layer. Thus, in anycast, the transmitter multicasts MRTS to multiple nexthop neighbors in the same direction and wait for CTS in response. Upon receiving a CTS from any one of the receivers, the sender transmits data to that receiver. All overhearing nodes invoke their directional virtual carrier sensing mechanism just as in DVCS. If the sender does not get any CTS in response to its MRTS, it may retry upto 6 times after appropriate backoff mechanism.

We observe that, since there may be multiple nexthop choices for forwarding the packet, the probability of atleast one of them responding with a CTS is higher in comparison to the case when there is only a single next hop choice as in DVCS. Thus in anycast, if deafness prevents one node from responding to a sender who is trying to communicate with it, due to the path diversity provided by anycast, another node may respond and forward the data packet. Once again we note that, the directional antenna version of anycast is quite similar to the omni directional version as well as the multichannel version described earlier.

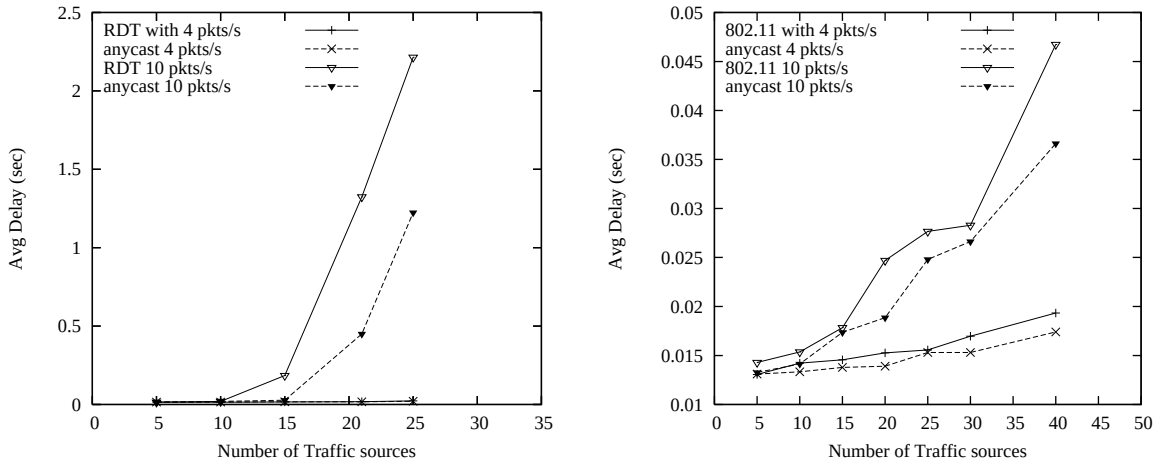
3.4 Performance Evaluation

We implemented the multichannel and directional antenna protocols in the popular ns-2 simulator. We used multipath AODV in the routing layer with appropriate modification so that the routing layer can maintain separate paths for separate channels or directions. We performed experiments in a static scenario with 100 nodes placed randomly in a 1000x1000m area. We ran experiments for different scenarios with 5, 10, 20, 25, 30 and 40 traffic connections and with data rates of 4pkts/s and 10pkts/s where the packet size was 512 bytes. In the multichannel network experiments, there are three channels available for communication. Figure 3.1(a) shows the graph of packet delivery fraction achieved by RDT and multi-



(a) Multichannel network (b) Directional Antenna network.

Figure 3.1: Packet delivery fraction vs number of traffic sources for anycast and 802.11 like protocols



(a) Multichannel network. (b) Directional Antenna network.

Figure 3.2: Average per hop delay vs number of traffic sources for anycast and 802.11 like protocol.

channel anycast protocols when the number of traffic connections is varied at two different rates (4 pkts/s and 10pkts/s). Similarly figure 3.2(a) shows the average per hop delay for the same scenario. The results clearly show how anycast outperforms RDT both in terms of delay and packet delivery fraction. As the number of traffic sources increases the difference between the two protocols constantly increases and at high load scenarios with 25 sources and 10 packets per second, anycast delivers 88% packets while 802.11 delivers only 73%. This result clearly shows the advantage of anycast in high load network when the problem of deafness is more prominent in multichannel networks.

In the directional antenna experiments, we set the beam-width each of the 8 antenna sectors to 45^o. Figures 3.1(b) and 3.2(b) show packet delivery fraction and average per hop delay graphs for both anycast and DVCS in directional antenna networks. We see that, anycast has a better performance compared to DVCS as it shows a higher packet delivery fraction when the network load is increased. Anycast delivers 12% more packets to the destination and incurs 16% lower delay in the scenario with 40 sources and 10 packets per second. Our results confirm that anycast is more robust in high load scenario where deafness is more common in directional antenna networks.

3.5 Conclusion

By anycasting the deafness problem in a multichannel or directional antenna network may be alleviated if not solved without the use of additional hardware or a separate control channel and even without synchronization requirement. Anycast can alleviate these problems by exploiting the availability of different routes to the destination. Thus, if one of the next hop nodes is “deaf”, another node may be able to route the data packet. Similarly, if a transmission is interrupted by a hidden terminal, the transmitter may be able to re-negotiate the channel with a different neighbor thereby, reducing the possibility of another collision. We have presented anycast in single channel, multiple channels and directional antenna networks. It is also possible to use the same protocol in hybrid networks containing all three features. Thus, unlike other protocols that were designed either for multichannel or for directional antenna networks, anycast is suitable for both types as well as single channel and omni-directional networks.

Chapter 4

MAC Layer Multicast in Wireless Multihop Networks

4.1 Introduction

Wireless ad-hoc networks have various applications in military, conferences, sensor networks and emergency operations. Many of these applications need one-to-many (multicast) communication. In multicast communication a single sender may send data to multiple receivers in the network. Such multicast communication can be very useful in the military where a commander might need to coordinate the activities of his troops and send critical instructions. Video and audio multicast are popular multicast applications among civilians where, a single sender sends video/audio data to multiple receivers.

Multicast communication can be achieved by sending multicast data to all receivers in the network via flooding. This approach may reduce the overall network efficiency due to unnecessary transmissions. These transmissions may be reduced or limited if the network is aware of routes to the multicast receivers so that the data could be sent only to the multicast receivers via predetermined routes. Several routing protocols have been developed to determine such routes from senders to multicast receivers ([24],[25],[31],[33],[44],[46]). Routes in ad-hoc networks might traverse various nodes to reach the receivers. Thus, multicast data may need to be transmitted across various hops before it reaches all multicast receivers. Since wireless links are prone to errors, data may not always be received correctly at the next node along the route. Such errors may not be tolerable by the multicast application, in which case an error recovery mechanism may be required. Certain error recovery mechanism might be implemented at the upper layer by requesting positive acknowledgments or feedback from the multicast receivers. However, this mechanism will require the sender to buffer data locally until the feedback has been received. This technique may increase delay in data delivery if the sender and receivers are separated by large number of hops. Sometimes such delay is not tolerable for example in voice applications large delays might make the data

unintelligible.

It is well known that an efficient and reliable medium access control (MAC) protocol is capable of removing inefficiency caused due to transmission errors. For several years MAC layer techniques have been used to improve the reliability and efficiency of one-to-one (unicast) communication where a sender communicates with a single receiver in the network. Various techniques to improve data delivery are implemented in the IEEE 802.11 MAC protocol which is the most widely accepted MAC layer protocol for both wireless LAN as well as ad-hoc networks. This protocol implements positive acknowledgment to provide reliable transmission of unicast data to the next hop node in the route and implements a retransmission policy in case of transmission failure. However, no such policy is implemented for multicast data in the 802.11 MAC protocol. However, upper layers may choose to use the same facility for multicast communication as well by explicitly sending multiple copies of multicast data, one for each next hop in the route, thus forcing the MAC layer to treat each copy as individual unicast data. This method however, may substantially increase the network load. In a wireless medium, a single transmission may be received by multiple receivers hence sending multiple copies of the same data is an unnecessary overhead. Thus, we can see that there is a reasonable ground to research MAC layer protocols that can potentially improve the performance of multicast communication and several attempts have been made in this direction to achieve greater efficiency in multicast communication.

In this chapter we propose a MAC protocol which can improve the efficiency of multicast communication. Our protocol is based upon the concepts of the popular IEEE 802.11 MAC protocol. In fact, we have developed a multicast extension of IEEE 802.11 protocol and evaluated its performance against IEEE 802.11 protocol and some other related approaches. We implement the protocol in the popular *ns-2* simulator and experiment with multicast routing protocol. Our approach demonstrates superior performance in terms of *packet delivery fraction* as well as *delay* compared to the IEEE 802.11 protocol.

The rest of this chapter is organized as follows. In section 2 we describe the medium access mechanism in IEEE 802.11 for unicast and multicast communication. Then in section 3 we describe our protocol. We show performance analysis and results in section 4. In section 5 we describe some recent work that propose reliable MAC layer protocols for multicast and/or broadcast traffic. We present conclusion and future works in section 6.

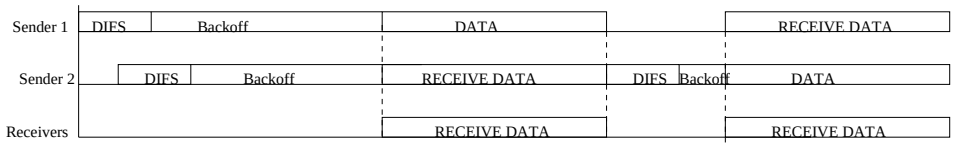


Figure 4.1: Access mechanism for multicast and broadcast transmission in IEEE 802.11

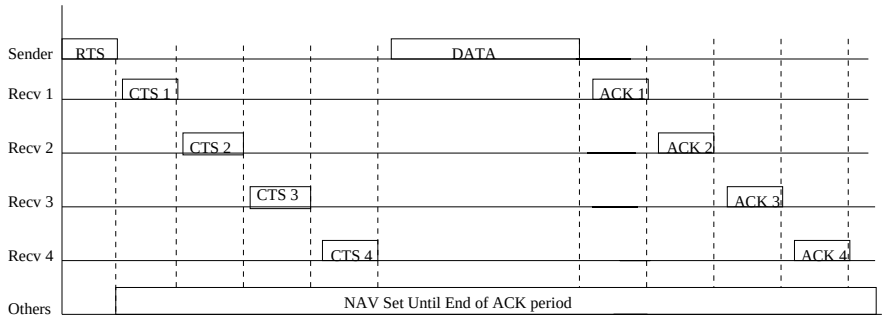


Figure 4.2: Multicast extension to 802.11 protocol.

4.2 Multicast Transmission in IEEE 802.11

In this section, we will briefly review the mechanism for multicast data transmission in IEEE 802.11 protocol. When a node has broadcast or multicast data to transmit, it performs channel access in accordance to the Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) protocol as in the IEEE 802.11 DCF described in Chapter 1. But unlike unicast transmission, multicast data is transmitted without any control packet exchange or acknowledgment. Figure 4.1 illustrates this multiple access mechanism for multicast packets. After completing the carrier sensing and collision avoidance procedure, the transmitter sends the DATA packet. All receivers that detect the transmitted packet correctly would receive the DATA packet and send it to the routing layer. The routing layer may decide that the packet needs to be forwarded if the node is an intermediate node in the multicast route. This node would then use the same access mechanism to forward the packet. This mechanism does not provide protection from hidden terminals neither does it guarantee that DATA was received correctly by all intended next hop nodes as their is no acknowledgment from the receivers.

4.3 Multicast MAC Protocol

We have developed MAC layer multicast as an extension to the IEEE 802.11 DCF protocol which can be used with any multicast routing protocol. We have tested our protocol with multicast AODV [25] but the scope of our protocol is not limited to any particular routing protocol. In this section we will describe our MAC layer approach to provide reliable multicast.

4.3.1 Multicast Extension of IEEE 802.11

We have implemented reliable multicast MAC within the IEEE 802.11 framework. We have used a similar approach in a previous work [84] but with a different goal of MAC layer "anycast" to achieve path diversity and thereby, combat fading and adverse channel conditions. Before we describe the protocol we will describe the changes introduced to the MAC layer frames.

We modify the RTS frame to include multiple next hop node addresses as in [84]. The RTS frame in 802.11 originally carries only one next hop node address since it is used only for unicast transmission. But the MAC layer packet header contains space for including three more addresses typically used to insert addresses of access points, senders, receivers etc. We can use this space to fit in four addresses for next hop nodes. This design choice helps us keep the RTS frame no larger than that in 802.11. The CTS frame is modified to include the receiver's (node that sends the CTS in this case) priority order which, we will explain later, is determined from the RTS frame. This helps the original sender to differentiate between the CTS sent by different nodes (CTS and ACK frames do not carry the sender's address). DATA packet header is modified to include the addresses of all those nodes from which CTS was successfully received. Finally ACK frames are modified to include the receiver's priority order, determined from the received DATA packet. Henceforth, we will refer to the modified control and DATA packets as RTSExt, CTSExt, DataExt and ACKExt. We will now describe our protocol in the next paragraph.

When the MAC layer receives a multicast DATA packet from the upper layer it first invokes the CSMA/CA mechanism as used in IEEE 802.11 protocol. After performing the collision avoidance procedure and when the medium is idle, the transmitter transmits an RTSExt frame to request access to the medium from atmost 4 next hop nodes in the multicast route because RTSExt may carry only upto 4 next hop addresses. Only those nodes which are part of the multicast route and whose addresses are included in the RTSExt must prepare to respond with CTSExt frames. All other nodes must invoke their virtual carrier sensing mechanism and defer medium access until the end of the current transmission. Since multiple routing nodes may exist in the next hop, the sender may expect multiple CTSExts. If all the

CTSExt frames are sent simultaneously, they may not be correctly received. Thus we need to devise a method to prevent simultaneous transmissions. In our approach we allow the CTSExt to be sent one after another by deliberately introducing a fixed amount of delay between successive transmissions. Thus, each receiver calculates the time it must wait before sending its CTSExt frame. This time is based upon the priority order conveyed via the RTSExt frame. This priority order is nothing but the position index of each receiver's own address in the RTSExt frame. The wait times are calculates as follows. The Nth receiver waits for a time equal to $N \times SIFSDuration + (N-1) \times CTSDuration$, where N is the position index of its address in the RTSExt frame. Thus the first node waits for SIFSDuration and the 4th one waits for $4 \times SIFSDuration + 3 \times CTSDuration$ before transmitting the CTSExt. A node transmits the CTSExt only if it does not hear any other transmission that could potentially interfere with the DATAExt that it will receive next. Thus, if during the wait period, if any node senses a busy medium, it must cancel the transmission of CTSExt. But if the overheard transmission is actually a CTSExt frame that was sent in response to the same RTSExt, it is not considered as a competing transmission and it is safe to send the local CTSExt. Since each CTSExt is sent at its own slot, the transmitter is able to receive the CTSExt frames and determine from the order in the CTSExt the addresses of nodes from which CTSExt was received. Successful reception of any CTSExt implies that the medium has been successfully reserved for that next hop node, but it is not the case for those nodes which had failed to send CTSExt. Thus, at the end of the waiting period (the time required by all next hop nodes to send CTSExt), the transmitter sends DATAExt to those next hop nodes from which it successfully received the CTSExt. Each next hop node that had sent CTSExt receives the DATAExt and waits for its turn for sending ACKExt in the same way as it waited for the CTSExt, only this time, the priority order is determined by the position index of addresses in the DATAExt, instead of the RTSExt. The wait times in this case are $N \times SIFSDuration + (N-1) \times ACKDuration$, where N is the position index of the node address in the DATAExt. If the sender does not receive ACKExt or CTSExt from some next hop nodes, it resends the DATAExt after appropriate backoff mechanism and RTSExt/CTSExt exchange with those nodes. This method of control packet exchange and retransmission policy provides efficiency to multicast communication by reducing the time required to recover from packet losses due to errors in the wireless medium. Fig 4.2 illustrates the multicast extension proposed here.

Until now we explained how the protocol would works when there are 4 or less next hop nodes. But as the number of multicast receivers increases in size, the number of next hop nodes in the multicast route also increases. In case there are more than 4 next hop nodes, the transmitter needs to cluster the next hop nodes into

different groups of size atmost 4 each and transmit data to one group at a time. We will explain the clustering method in the next paragraph. We will first describe the problem that motivates the formation of clusters.

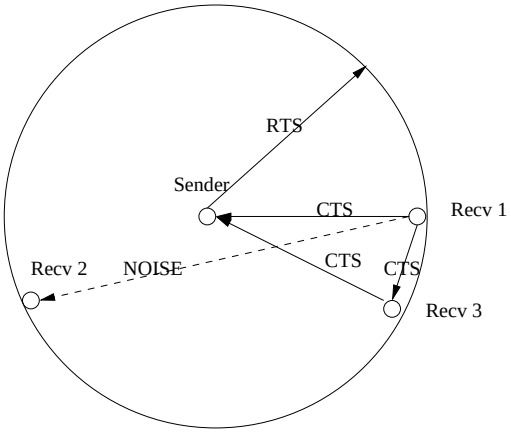


Figure 4.3: Neighbor unable to respond due to interference with CTS sent by another neighbor

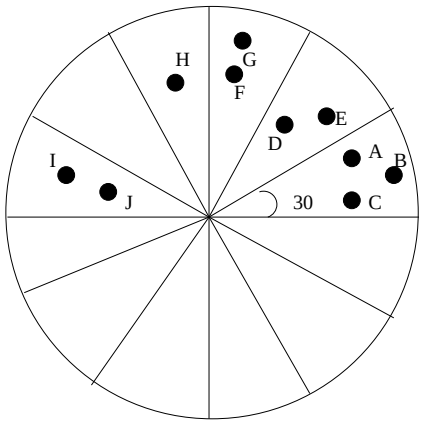


Figure 4.4: Clustering to group together non conflicting multicast next hop nodes.

We have discussed earlier that some nodes in the network might not receive certain transmissions correctly simply because the transmitter is not within their receive range. Such nodes may still hear noise in the medium due to which they may not participate in any transmission as long as the medium is not free from noise. These nodes must set their NAV to EIFS duration and refrain from participating in any communication. We observe that this problem may cause some next hop nodes that have determined that they must send CTSExt, to cancel the transmission

of their CTSExt. Since in our protocol as well as in [42] and [54], multiple CTSExt may be sent, it is possible that some nodes do not send CTSExt because of interference caused by CTSExt sent by other nodes although they are sent in response to the same RTSExt. This may happen due to the distance between these nodes is such that the received power of the CTSExt sent by the one node is below the receive threshold at the other thus making it difficult for the node to decode the packet. Therefore, the node treats this packet as noise which causes the virtual carrier sensing mechanism to be invoked, which inhibiting the transmission of CTSExt. This scenario is illustrated in fig 4.3. Here, nodes 1 and 2 are beyond each others transmission range but within the carrier sensing range. When node 1 transmits CTSExt, node 2 would sense a busy medium and defer transmission by an EIFS period. Node 3 would hear the CTSExt correctly and determine that the CTSExt was meant for the same multicast sender and will go ahead and send the CTSExt itself. The sender will send DATAExt to receivers 1 and 3 alone and retry transmission for receiver 2. While this scheme is still reliable, the network incurs an extra wait period due to the unnecessary inclusion of receiver 2 in the RTSExt frame. If the transmitter is made aware of such node pairs that conflict with each others transmissions, and it requests CTSExt only from those next hop nodes that do not conflict, this problem can be eliminated. Thus, the transmitter may 'cluster' nodes into different groups such that nodes in the same group are always within each others transmission range and thus do receive each CTSExt correctly. There are many ways to achieve this clustering. One way is by determining the local network topology, i.e. topology including only the one hop nodes via location information. Location information may be available with the use of Global position system (GPS). Since the transmitter only needs relative locations of its neighbors, it may calculate relative location via angle of arrival and distance measurements from the received signal instead of using additional hardware for GPS. Another way to achieve clustering is by exchanging neighbor lists with all neighbors and group together nodes that are each others neighbors. This method would require additional message exchanges. Any of these methods can be effectively used to form these clusters. However, since the clustering mechanism does not require the knowledge of exact location, we use a different and simpler approach to achieve clustering. We calculate the approximate direction of the neighbors with respect to the transmitter via angle of arrival of the received signals. With this knowledge and simple geometry we can claim that neighbors that lie within the same quadrant of a circle which is drawn with the transmitter at the center and which approximately defines the transmission range of the transmitter are each others neighbors. We re-order the list of next hop nodes obtained from the routing table to group together nodes that lie within the same quadrant. We illustrate this with a simple example in fig 4.3.1. Here, nodes A through G are within the same quadrant and thus are within

each others transmission range but since we cannot have groups larger than 4 we choose A through D to form the first group. Nodes E through H should form the second group while nodes I and J should form the third group. We then make three copies of the same data each containing different groups of node addresses. By clustering the next hops in this manner we ensure that if the channel is idle at all the nodes in the same group, the nodes will all transmit CTSExt in response to the RTSExt and will not defer transmission due to interference due to CTSExts sent by other nodes in response to the same RTSExt. This reduces the time wasted in waiting for CTSExt from those receivers that will not be able to send CTSExt as they perceived a prior CTSExt as noise.

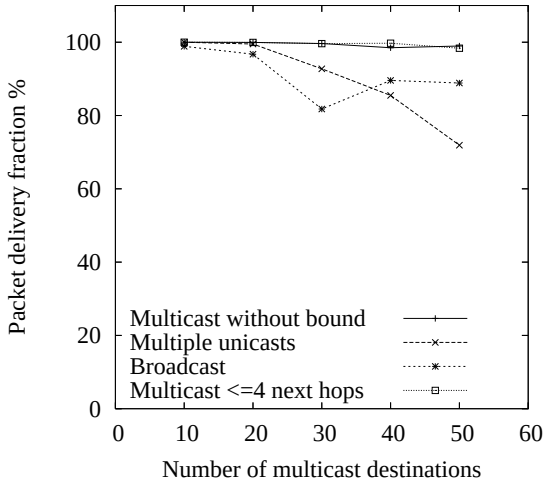


Figure 4.5: Packet delivery fraction with a two ray ground propagation model with 100 nodes.

4.4 Performance Evaluation

We have used network simulator *ns-2.26* to implement the multicast MAC protocol. In this section we will describe the experimental setup and results obtained. We will also briefly explain the protocols used to compare performance.

4.4.1 Experimental Setup

We have implemented four different approaches to provide multicast at the MAC layer. We will see later that there are several works that implemented some

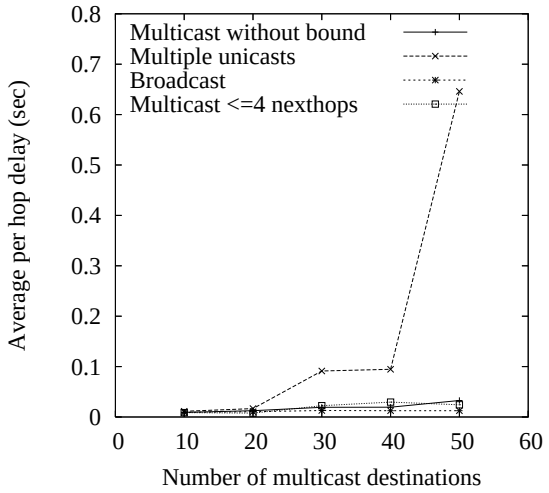


Figure 4.6: Average per hop delay with a two ray ground propagation model with 100 nodes.

sort of multicast MAC. In IEEE 802.11 protocol multicast data is sent to all neighbors via a single transmission after performing CSMA/CA. The next hop nodes who receive this transmission, filter the packets depending upon the multicast address associated with the packet. If the next hop node is one of the multicast receivers, it accepts the packet and if it is an intermediate node in the multicast route, it resends the data to other receivers. We will refer to this method as the *broadcast MAC*. This method does not provide reliability as it does not have any error recovery or retransmission policy. One method to achieve reliable MAC layer multicast is to treat a single multicast data packet as N unicast packets where N is the number of next hop nodes that are either multicast receivers or intermediate nodes in the route. Each unicast packet is then transmitted using CSMA/CA with virtual carrier sensing and RTS/CTS exchange as implemented in the IEEE 802.11 protocol. This method provides reliability via acknowledgment and retransmission but it also brings about a larger delay in packet delivery apart from increasing network load. We will refer to this protocol as *multiple unicast* MAC protocol. Another method to achieve reliable multicast is to send multicast packets using RTS/CTS exchange with all multiple next hop addresses in RTS and DATA packets as used in MACAM [54] and MMAC [42]. This method also provides reliable transmission but due to the presence of large number of address in the RTS frame, the packet size may be increased by a large. This is in violation of the idea that control frames must be small so that they are less prone to errors and collisions, thus defeating the purpose

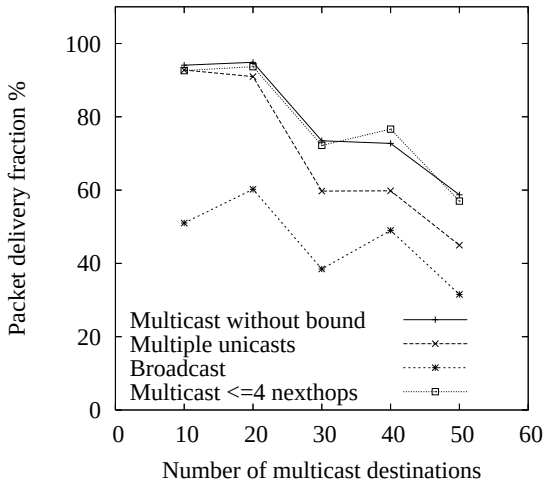


Figure 4.7: Packet delivery fraction with a Ricean fading propagation model with 100 nodes.

of control frames altogether. [1] In our implementation of MMAC / MACAM we have artificially reduced the size of the RTS frame so that it is the same size as the original RTS frame in IEEE 802.11 protocol. We made this change to make a fair comparison with our protocol. Although we have observed that the performance of these protocols is much worse if the control packet sizes are not controlled in this manner. We have implemented all four methods and evaluated their performance against one another.

We have set up the experiment using a grid of size 1500x300 with 100 nodes. There is one multicast sender with different number of receivers (10,20,30,40 and 50). The sender sends 4 multicast UDP packets per second. The simulation runs for 900 simulation seconds. We use 2Mbps data rate and a nominal transmission range of 250m with the carrier sensing range of 500m. We use the two ray ground propagation model in the physical layer in one set of experiments and a *Ricean fading model* from [83] for another set of experiments. We experimented with the latter model to further motivate the importance of using reliable MAC layer multicast. The same physical layer model was used in [17] and [84].

[1] We will describe this protocol in more details in the related works section.

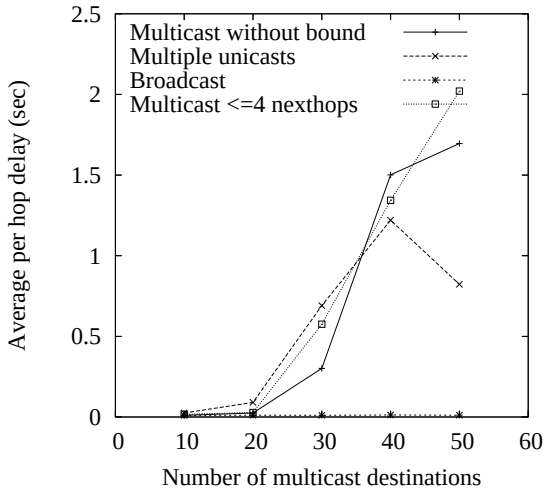


Figure 4.8: Average per hop delay with a Ricean fading propagation model with 100 nodes.

4.4.2 Results

We instrument the experiments to calculate the *packet delivery fraction* and *average per hop delay* in the network. The *packet delivery fraction* is calculated as

$$\frac{\text{no. of packets delivered}}{\text{no. of multicast receivers}}.$$

Similarly the *average per hop delay* is calculated as

$$\frac{\text{per packet delay}}{\text{number of hops between sender and receiver of the packet}}.$$

Experimental results clearly show that in the absence of a reliable MAC protocol, the network suffers from a large amount of packet loss. These losses are mainly due to collisions with other transmissions in the network. Fig 4.5 shows the *packet delivery fraction* achieved with various MAC protocols. Due to the absence of a retransmission policy the broadcast MAC protocol is able to deliver only 88 % of DATA when the number if multicast receivers increases to 50. Unicast MAC also shows poor performance although it ensures reliable delivery through ACK and retransmission policy. The poor performance of multiple unicast MAC is attributed to the delay incurred in sending multiple unicast packets to along every hop in the route to multicast receivers. This delay causes packet loss in queues at various nodes. Multiple unicasts also contributes to the increase of the overall network load since a single multicast packet is treated as N unicast packets, N being the number

of next hop nodes. The load is further increased due to contention with the next hop nodes that need to send the data further downstream along the routes. Thus, the *packet delivery fraction* for multiple unicast MAC is only 71 % for 50 multicast receivers which is lower than that achieved by the broadcast MAC protocol. On the other hand, both reliable multicast schemes achieve higher *packet delivery fraction* compared to the other schemes due to their ability to utilize the network bandwidth more optimally by delivering data to multiple next hop nodes via a single transmission i.e. proper exploitation of the broadcast nature of the wireless medium and yet ensuring reliable delivery by implementing retransmission policy in case of errors. Both our protocol and MMAC provide high *packet delivery fraction* of 98 % for even large number of multicast receivers, however, the good performance of MMAC may be a little exaggerated because as we have mentioned earlier, we have artificially reduced the size of RTS frames in MMAC as larger frames might lead to performance degradation due to collision of control packets. Later in the related work section we shall point out other problems that may arise in MMAC.

Fig 4.7 shows the *packet delivery fraction* in the presence of Ricean fading model in the physical layer. The comparative performance of all but broadcast MAC protocol is similar in this case except for the fall in the absolute performance. Here we observe that broadcast MAC which performed better than unicast MAC in the two ray ground propagation model, actually performs much worse with the fading model. The main reason for this degradation is the absence of retransmission policies which becomes of more importance in adverse channel condition.

Fig 4.6 plots the average per hop delay incurred by each of the four protocols in the two ray ground propagation model. We observe that broadcast MAC achieves the least delay which is again due to the absence of any loss recovery mechanism. Unicast MAC incurs the maximum delay mainly due to the increased network load and queuing delays. The queuing delay is a direct result of increased network load owing to the multiple copies of the multicast packets in the network. The multicast MAC protocols incur very low delay compared to the unicast MAC protocol since they efficiently utilize the broadcast nature of the medium while providing reliability.

Fig 4.8 plots average per hop delay incurred in the Ricean fading scenarios. We observe that all reliable protocols incur much higher delay than in the two ray ground model. This is due to the increase in the number of retransmissions required to recover from losses due to adverse channel conditions. Multicast MAC protocols incur higher delay in these scenarios than the other protocols but this is due to statistical reasons. Note that it is possible that some packets that arrive at a node may be dropped from the interface queue. The probability of such drops increases when the packet has incurred higher delay. These dropped packets do not contribute to the delay calculations in our experiments. The exclusion of such high delay

packets attributes for lower calculated delay in unicast MAC protocol as compared to the multicast protocols. Low delay in broadcast is again due to the absence of retransmission policies.

4.5 Related Work

Some recent works have explored MAC protocols for reliable multicast and broadcast. [28], [85],[89] present solution requiring the use of busy tones and control packet exchange to achieve reliability and solution to *hidden terminal problems*. These protocols require additional hardware to send busy tones which might not be economical in real life. The broadcast support medium access (BSMA) protocol [48] is one of the first works that employ exchange of control packets to provide reliable MAC layer broadcast. Before sending data, the sender transmits an RTS frame and waits for CTS from all receivers, which are sent simultaneously causing collision at the sender. This protocol requires the use of a direct sequence spread spectrum (DSSS) receiver with capture capability and assumes that the simultaneous signals can be captured by the DSSS radio. This protocol tries to avoid hidden terminal problem through this approach. Even with the availability of such radios, the receiver can capture colliding packets with a very low probability as analyzed in [93].

The protocol in [47] uses a similar approach without assuming a DSSS radio. In this work, the senders and receivers assume that a collision after RTS transmission is due to multiple CTS frames and the sender continues to transmit DATA. There is no ACK transmission, thus this approach does not provide retransmission policy, it only tries to alleviate hidden terminal problem. The assumption of collision in this protocol is unrealistic in a dense medium where the collision may be due to another transmission and not due to CTS frames sent simultaneously.

Batch mode multicast MAC [93] is another protocol that employs control packet exchange to alleviate hidden terminal problems and achieve reliable transmission. In this protocol, the transmitter does an RTS/CTS exchange with all the next hop nodes in the route before data transmission, which is followed by a round of *request for ACK (RAK)* and ACK transmissions. This requires the senders and next hop nodes to reserve the medium for a relatively long interval of time $N \times (T_{RTS} + SIFSDuration + T_{CTS} + SIFSDuration) + T_{DATA} + SIFSDuration + N \times (T_{RAK} + SIFSDuration + T_{ACK} + SIFSDuration)$, where N = number of next hop multicast receivers. This approach does not fully utilize the broadcast nature of the broadcast medium, leading to wasted bandwidth. Similarly, broadcast medium window (BMW) [49] achieves reliable broadcast by sending the broadcast packet as unicast packets to each neighbor in a round robin

fashion while allowing other neighbors to receive the data without requiring acknowledgment. The sender transmits an RTS to a chosen neighbor and the neighbor responds with a CTS. The CTS contains the sequence numbers of packets that could not be received. The sender retransmits the missing packets as well as the current packet. All other nodes may receive the packets and update their list of received data. The sender then transmits an RTS to the next neighbor and repeats this process. This approach achieves reliability but increases the data delivery latency because each neighbor needs to wait for its turn to request missing data from the sender and thus the sender still needs to buffer all unacknowledged data.

MMAC [42] is very similar to our work. Here, the authors present an extension of IEEE 802.11 protocol called multicast MAC (MMAC). In this work, the sender transmits multicast data packet to the next hop nodes and waits to receive acknowledgments. The acknowledgments are sent according to a schedule calculated from the position index of the next hop address in the data packet. There is no upper bound to the number of next hop addresses that may be included into the data packet. Thus the data packet size increases by the number of addresses included in the header. The amount of time the sender has to wait before all the ACK frames have been received is $N \times (T_{ACK} + SIFSDuration)$, where N is the number of next hop nodes. At 2Mbps data rate $T_{ACK} = 56\mu sec$, $SIFSDuration = 10\mu sec$. Thus, for $N = 8$, the wait time is 528 μsec. If in the meantime a mobile node happens to enter the sender's collision domain, it would sense an idle medium and might initiate a new data transmission. Apart from a mobile node straying into the transmission range, those nodes which are beyond the receiving range but in the carrier sensing range of the sender will also be free to contend for the channel after an EIFS duration which is equal to $SIFSDuration + 8 \times ACK + DIFSDuration = 508\mu sec$ (DIFSDuration = $50\mu sec$). From these calculations it is clear that for $N \geq 8$, there is a possibility of ACK collisions at the sender leading to retransmission attempts by the sender. On the other hand, it is possible that while the receiver is waiting for its turn to send ACK, another node is trying to transmit DATA to the receiver. The receiver will not respond to any DATA transmissions before the ACK timeout period. This may cause the sender to retry several times leading to an increased contention window size and in extreme cases dropping the packet and initiating route error and discovery processes even though the route actually exists. This is the well known *exposed node* problem in wireless ad-hoc networks and it is somewhat increased in MMAC.

The loss recovery method used in MMAC is similar to multicast scheme use in MACAM [54] and our protocol. In both approaches the sender sends a single multicast RTS frame to all the neighbors and waits for CTS frames. The RTS frame is overloaded to contain the addresses of all the multicast next hop nodes. Thus the RTS frame size is larger than the size of the frame in IEEE 802.11. CTS frames are

transmitted in a time based priority schedule. In both protocols there is no upper bound on the number of next hops that can be included in the RTS frame. Thus the RTS frame in MMAC is larger than that in 802.11 making the RTS frame itself prone to collisions due to hidden terminals. The effect of increased RTS size is not evaluated in these papers. These approaches also do not implement the clustering method we have described earlier in our protocol.

4.6 Conclusion and Future Directions

We have presented a simple extension to IEEE 802.11 protocol to provide reliable multicast MAC protocol. This approach can be easily incorporated in the IEEE 802.11 protocol to provide performance enhancement for multicast communication. Further work in this direction is required to implement this concept in a testbed scenario. In future, we will implement this protocol in a testbed using Berkeley motes similar to the one used in [84] to provide a proof of concept implementation.

Chapter 5

Experimental Study of Physical Interference Model for Wireless Networks.

5.1 Introduction

Practical approaches for modeling interference on wireless links is critical for understanding wireless network behavior. This is because the MAC layer protocol must fundamentally be able to schedule transmissions on links in an interference-free fashion. MAC layer protocols are always based on an interference model, often implicitly. For example, 802.11 protocol using RTS/CTS [13] essentially "assumes" that any other transmitter that can send/receive packets to/from the intended transmitter/receiver can interfere with the transmission. Such assumptions are made more directly for TDMA transmission scheduling protocols [69, 60, 82, 101], where interference-free transmission schedules are computed for the links in the network. For the scheduling algorithm to work, it must assume an interference model that states how links interfere.

Other than guiding the MAC protocol design, understanding interference also leads to better understanding of the network capacity. In fact, interference model and transmission scheduling together specify the network capacity [39]. In addition, understanding of interference can guide protocol design for QoS or other utility metrics [58]. It can also guide selection of different transmission modes – such as channel selection [92, 78], transmit power control [56] or selection of beams with switched-beam directional antennas [79].

In the past literature, researchers have assumed a unit disk model for the wireless communication range. While this model makes algorithm design simple, it is far from being realistic. In recent literature, there has been an interest in using more realistic models in simulations. In [83], the authors have designed models to include multipath fading and shadowing along with the path loss model for the popular ns2 simulator [63]. More recently, researchers have stressed realism

in interference modeling and realistic SINR-based models have been used. These models are also called *physical models* [39]. While physical models have been used in the design of cellular (one-hop) networks [80] for a long time, their use in multihop networks for protocol design is fairly recent [22, 38, 65]. Several recent measurement-based works have argued in favor of using physical model because of its realism [30, 72, 52]. In this chapter we evaluate the physical interference model for its accuracy. We make two contributions. First, we develop a systematic, measurement-based modeling approach for the physical model. While a specific radio has been used, we hope that our methodology will be useful for other radios as well. Second we provide results that quantify the accuracy of the physical model in the context of TDMA scheduling. The goal is to validate the accuracy of the physical interference model in a testbed.

Since this is an experimental work, the choice of testbed is important. We have chosen the Berkeley motes platform (specifically TelosB architecture [66], that use the Chipcon 2420 radio [97]) for this work. This choice gives us a radio which is very well documented in a complete manual [97] and a MAC protocol that can be implemented purely in software. They are very affordable and popularly used. Thus, our results will be directly useful to the community. Another option would be using 802.11 radios for physical layer and a software-based approach to implement MAC-layer protocol [70]. However, here we will suffer from lack of documentation about the radio and will have to rely on certain amount of reverse engineering. Other approaches using custom, programmable, high-speed radio platforms (e.g., gnuradio [3] or Rice's WARP kit [6]) are also possible and probably ideal since we will have physical layer access to the radio. However, this will require an expensive testbed. We will study these other platforms in our future work for a more complete understanding.

The rest of the chapter is organized as follows. In Section 5.2 we develop the physical interference model specifically for the motes. Performance results are presented in Section 5.3. Related work and Conclusions are presented in Sections 5.4 and 5.5, respectively.

5.2 Building Physical Interference Model

5.2.1 Experimental Platform

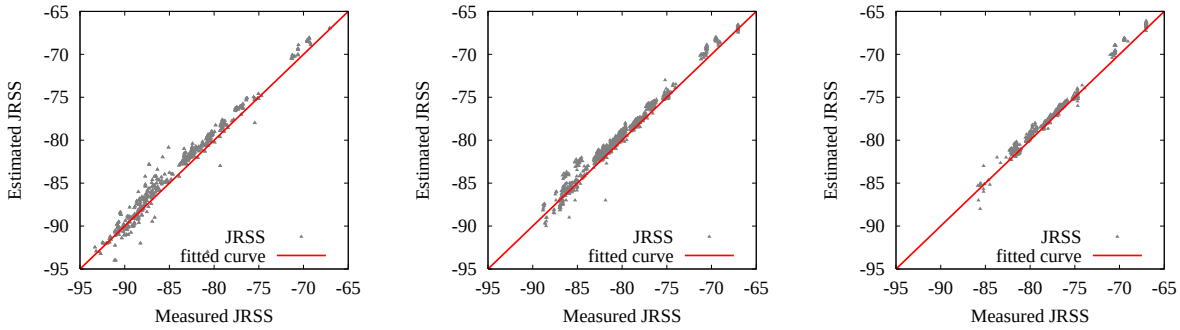
Our experimental testbed consists of 20 TelosB motes [66] based on Berkeley mote architecture [102] which we program using TinyOS 2.0 [40]. Each TelosB mote has a CC2420 radio [97] which is compliant with the IEEE 802.15.4 physical layer standard. The CC2420 radio operates in 2.4 GHz ISM band with an effective

data rate of 250 Kbps. To avoid interference with 802.11 networks, we have tuned the mote radios to IEEE 802.15.4 channel 26 since it stands clear of the IEEE 802.11 channels used in North America.

CC2420 provides a measure of the received signal strength (RSS), which is an estimate of signal strength averaged over the last 8 symbol periods ($128\mu s$) and is continuously updated. This value can be either read directly from the RSS register or obtained from the metadata in the received packet. In our work, we obtain the RSS value by reading the register right after the start frame delimiter (SFD) of a packet is received. An interrupt from the radio enables the mote to recognize the reception of SFD. This allows us to obtain the RSS even if the rest of packet is not correctly received. RSS is expressed in dBm.

To ensure that we can have a multihop testbed in a small space, we have used the minimum possible transmit power in the motes. All motes use the same transmit power. All motes are powered directly via USB so that variabilities due to different battery levels can be eliminated. The motes are placed in a random fashion on a tabletop. Experimental data are collected by another mote directly connected via USB to a laptop. All experimental data are transmitted directly to this central mote and the data is analyzed on the laptop (this mote and laptop combination is loosely referred to as 'base station'). All communication to and from the base station happens at the maximum transmit power so that all motes in the network can communicate with the base station directly.

The default MAC layer in TinyOS is a simple Carrier Sensing Multiple Access (CSMA) protocol. Since the interference models we evaluate are independent of the MAC layer, all MAC functionalities including carrier sensing are disabled. Nodes transmit synchronously so that success probabilities can be evaluated experimentally. The synchronous transmission is achieved by a separate mechanism described as follows. The base station (BS) mote sends its system clock in beacons sent periodically (every 500ms) at the maximum power. All motes in the network listen to these beacons and synchronize their clocks to the beacon time. The BS mote also acts as a command center for the network. Whenever a synchronous transmission by more than one node is needed for an experimental evaluation, e.g., evaluating a TDMA schedule, the BS mote sends commands to each of those nodes instructing the start time of the transmissions. Since the nodes are time synchronized, this enables synchronous transmission. Note that in [90] the authors also used a similar technique for synchronized transmission. We independently evaluated jitter in the transmission start times. The maximum jitter was less than $128\mu s$, which is the time to receive the SFD. This level of synchronization was sufficient to eliminate the possibility of not being able to capture a stronger signal that arrives later while a weaker signal is present, consequently losing both packets [103].



(a) Two transmitters (b) Four transmitters (c) Six transmitters

Figure 5.1: Validation that interference is additive. The scatterplots show $JRSS(m)$ against $JRSS(e)$ for different number of interferers. The plots also show that $JRSS(m) = JRSS(e)$ explains the observed statistics very well and that there is hardly any dependency on number of interferers.

5.2.2 SINR-based Model

The SINR-based model describes the success probability of a transmission (modeled in terms of *packet reception rate* or *PRR*) when one or more interferers are contributing to the interference at the receiver of the intended transmission. If S is the signal power received at the intended receiver from the sender, N is the noise power at the receiver and I_{joint} is the combined interference power experienced at the receiver caused by the group of interferers (transmitting at the same time as the sender), the model predicts the relationship $\beta(\cdot)$ between the bit error rate (BER) and SINR:

$$BER = \beta(SINR), \text{ where } SINR = \left(\frac{S}{N + I_{joint}}\right).$$

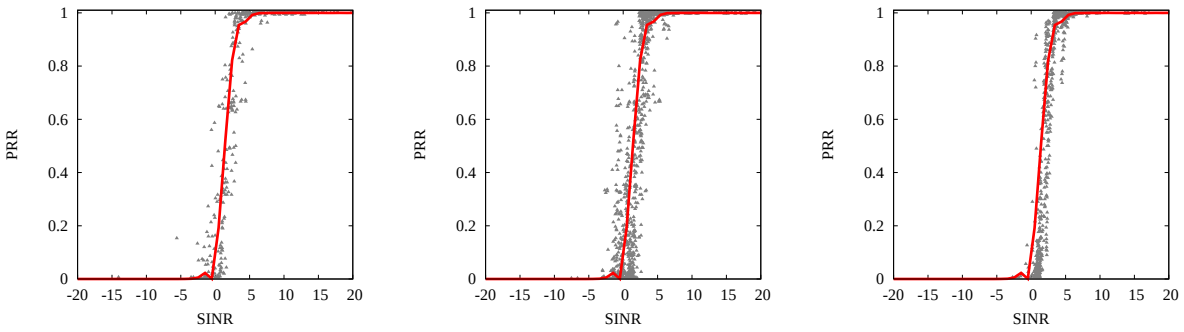
The function β depends on radio properties such as modulation. Packer error rate or PER is directly related BER and depends only on encoding. Thus, the above equation can simply be rewritten by replacing BER by PER:

$$PER = \beta'(SINR).$$

See [80] for further exposition of the nature of β and the relationship between BER and PER for various common modulation and encoding schemes. PRR is given by simply

$$PRR = 1 - PER = 1 - \beta'(SINR).$$

It is important to note that the nature of the functions β or β' is shaped like the mirror image of the letter 'Z', with zero or negligible error rate for high SINR, very



(a) 2 transmitters (1 interferer). (b) 4 transmitters (3 interferers). (c) 6 transmitters (5 interferers).

Figure 5.2: PRR vs. SINR for different number of interferers. Also, the fitted curve on the aggregated data is shown.

high error rate for low SINR and a sharp fall between the two. The falling part of the function has been described as the *transition region* in [109]. For simplicity of modeling, often in literature β or β' is 'thresholded' and described as a step function going from 1 to 0 at a specific value of SINR, typically called the *SINR threshold* or *capture threshold* (β_T).

I_{joint} is usually the sum of individual signals received at the receiver from the interferers [80]. However, a recent result in [90] has questioned this using measurements in a different mote radio hardware. They have also observed that unlike the above equations the SINR threshold depends on the number of interferers and the signal power. Our modeling experience (described in Section IV), however, shows that modeling I_{joint} by the sum of individual interference powers is sufficient and the functions β and β' or the SINR threshold β_T do not depend on any signal or interference power or number of interferers.

5.2.3 Measurements

In order to compute SINR, we measure signal, noise and interference powers separately. Received signal power is measured by the radio in absence of interference i.e., when there is only one transmitter. This is simply the transmitted signal power that reaches the receiver after path loss, shadowing and multipath fading with the added noise component.[1] We measure RSS at each node in the network for packets received from every other node in an otherwise "quiet environment," i.e., no other node except the said transmitter is active. These measured values serve

[1] We did not observe signal power changing appreciably over time other environmental conditions remain same. Similar observations were also made in [91].

as either signal or interference power for computing SINR depending on whether the transmitting node is the intended transmitter or an interferer in case multiple transmitters could be active simultaneously.

The experiments are performed as follows.

1. *Noise estimation:* Noise is measured by sampling the RSS register in the CC2420 radio when there is no transmission. We sample the RSS register every 20 ms for a period of 6 seconds and using the valid values thus obtained[2] compute the average noise at every node in the network.

2. *Pairwise RSS measurement:* Each node takes turn to broadcast 1000 packets of 128 bytes each, while all other nodes act as receivers. Each receiver reads the RSS value when it detects the start frame delimiter of a packet as described before. Note again the entire packet does not need to be received correctly for this. These RSS values are used to compute average signal strengths for each link in the network.

3. *Multiple concurrent transmitters:* Here, in each experiment k nodes transmit 1000 packets in synchronized fashion where k is varied from 2 to 6. These k nodes are chosen out of 10 nodes randomly selected from the network. This constituted 837 experiments. All senders transmit at the same time. Every other node acts as receiver. The number of concurrent transmitters is limited to 6 simply to limit the number of experiments to be performed. Each receiver records the number of packets it received correctly from each transmitter and all RSS values sampled. This defines the packet reception rate (PRR) for different links in presence of a set of interfering transmissions.

5.2.4 Model Creation

The first step in the model creation is verifying if interference is indeed additive. One important reason for doing this verification is a recent work [90], where it was observed, *albeit* using a different mote and radio platform, that total interference power, I_{joint} – when multiple interferers are active – may not be the sum of the individual interference powers. The authors also found that the total interference power I_{joint} was influenced by number of interferers. We perform a careful evaluation of this aspect and reach a different conclusion. *In our observation, when multiple interferers are active, the total interference power experienced is indeed the sum of individual interference powers.* To see this, take a look at Figure 5.1. Each

[2]Not all read attempts for the register produce valid values [97].

subfigure shows a scatterplot for the joint RSS measured ($JRSS(m)$), borrowing the terminology from [90]) and joint RSS estimated ($JRSS(e)$). $JRSS(m)$ is the measured joint RSS from the experiments in step 3. $JRSS(e)$ is simply the sum of interference powers that are obtained from the average RSS measurements in step 2. Note that the observation samples are very close to the $JRSS(m) = JRSS(e)$ line. The coefficient of determination, R^2, for the $JRSS(m) = JRSS(e)$ model is found to be very good (0.9962). This is evidence that interference acts additively in our test platform.

In the second step we develop the PRR vs. SINR model. To do this, we consider each experiment done in step 3 above in isolation. Note that for each experiment, each receiver records the PRR for each transmitter active in that experiment. We take turn to consider one of these transmitters as the *sender* and the rest as *interferers*. We compute $JRSS$ for the set of interferers as the sum of the average RSS values for interferers recorded at that receiver in step 2. Similarly, the average RSS value recorded in step 2 for the sender provides the signal power for computing SINR. The signal, sum of interference powers and noise are used to compute SINR at the receiver for the concerned sender. The PRR for this sender and the computed SINR is plotted in the scatterplot in Figure 5.2. For a particular experiment, this is repeated for every sender by fixing the receiver, and then repeated for every receiver. Combining all experiments we get the scatterplots in Figure 5.2, categorized into different number of interferers. This categorization is done specifically to demonstrate that the PRR vs. SINR relationship is independent of the number of interferers.

These results show that at SINR greater than about 5 dB, PRR is almost 100%. As mentioned before, there is a *transition region* [109] between (-3) to 5 dB where packets are received with a probability less than 1. This region is somewhat noisy and predictability is poor (also observed in [109]). The PRR trails down to 0 below (-3) dB. Overall the nature of PRR vs. SINR relationship is similar to that observed in [90] [109], except that in our case the relationship is fairly independent of number of interferers.

For use in later modeling, we develop a fitted curve on the aggregated data in Figure 5.2. We do this following the method used in [52] for similar modeling. We obtain the fitted curve using a linear interpolation of average values in buckets of 1 dB each. This fitted curve is shown in each subfigure of Figure 5.2 for reference and comparison with the experimental data. It provides the PRR vs. SINR model that can be used by a scheduling algorithm, for example.

5.3 Performance Results

The same 20 motes testbed described before is used to perform a relative performance evaluation of the interference models described so far. The evaluation consists of two separate scheduling experiments: (i) experiments with schedules generated by a greedy algorithm; and (ii) experiments with randomly chosen set of links scheduled together. The experiments are very comprehensive, covering 13,000 sets of links for evaluation.

To get started, we assume that the physical interference modeling (Section 5.2) has already been done and we have the PRR vs. SINR relationship (β'). Now, in a the given network we simply need to instantiate the model. To do this, we estimate the noise at each node, determine the RSS (average) and PRR between each node pair (each direction). This is not unlike the steps done in Section 5.2.

Transmission threshold is set at 99%. All links with PRR equal or more than 99% are considered links in the network graph G. The scheduling algorithm can only handle 'binary' transmission probabilities, i.e., a link can either be scheduled with absolute certainly or it cannot be. Thus the PRR vs. SINR relation (β', β'') was thresholded at 5 dB (SINR threshold) to handle the physical interference and the maximum interference models. (Note from Figure 5.2 that PRR is almost 100% when SINR $\geq$ 5 dB). Knowledge of RSS's between node pairs can now determine whether a transmission on a link is 'feasible'[3] given a set of other links active at the same time, according to the physical interference model.

5.3.1 Performance of Scheduling Algorithms

In this section, we study the performance of the physical interference model when used by scheduling algorithms to make scheduling decisions. We choose a simple greedy scheduling algorithm similar to the one used previously in [22]. The algorithm takes as input a traffic load (an ordered set of links to be scheduled and the number of packets to be scheduled on each link). The algorithm generates a schedule in a greedy fashion. The schedule is simply sets of links such that the links in each set can be scheduled simultaneously. The algorithm provides as many sets as needed to schedule all packets on all links.

The interference model essentially specifies which set of links are 'feasible' together. The greedy algorithm takes each link in the specified order and schedules it with the first available set where it is feasible given the conflict relation specified

[3]Feasibility here means whether or not, the transmission on a given link will be successful, given that a set of links is scheduled together.

by the model. If the link is not schedulable according to the feasibility criterion, it is placed in a new set. .

The model accuracy is checked in the following fashion. For a given load the model provides a specific schedule. This schedule provides a *predicted packets/set*. This is simply the total number of packets (in the specified load) divided by the number of sets needed to schedule all packets. We then evaluate the model accuracy by evaluating the schedule it generates using a direct experiment on the testbed. The average PRR for each scheduled link on each set is evaluated over 1000 runs of the same schedule. Then all PRRs for all links on all sets in the entire schedule are summed up and divided by the number of sets to determine the *measured packets/set*. The difference, *measured minus predicted packets/set*, determines the modeling error. Note that modeling error here can only be negative – a perfect PRR (100%) in all cases will make the measured equal to the predicted. Indeed this is what we observed for the physical interference models for three different loads. The schedules predicted by the model matched exactly with the results of the direct measurements and therefore the modeling error was non existent.

5.3.2 Evaluating Models Based on Random Subset of Links

While the greedy scheduling results gives us some idea of the accuracy of the interference model, it does not provide a complete picture. *First*, the algorithm is not optimal. In fact, the optimal algorithm is intractable. The above evaluation checks only the sets of links that are deemed feasible by the model, not the sets that are not. The greedy scheduling algorithm as well as the algorithms known in literature work with a 'binary' model of interference. They cannot schedule a set of links where the probabilities of transmission success is non-zero but less than 100%. Thus, SINR-based physical model has to be 'thresholded' to make it usable by the scheduling algorithm.

We will now try to address these issues with a different evaluation approach. The accuracy of a model can only be determined by looking at how well it predicts the 'feasibility' of any given set of links. One way to do this would be to enumerate all possible subsets of links in the network and then test for feasibility of each of these subsets, both according to the model and also in reality. The outcomes can then be compared to determine modeling errors. However, in this approach, the number of subsets is exponential in the network size. We can reduce the number of experiments to perform by random sampling, i.e., simply evaluating a large number of randomly generated subsets instead of exhaustively evaluating all possible subsets.

Thus we select a random subset of links from the network graph eliminating those that violate the primary interference condition. We evaluate the actual

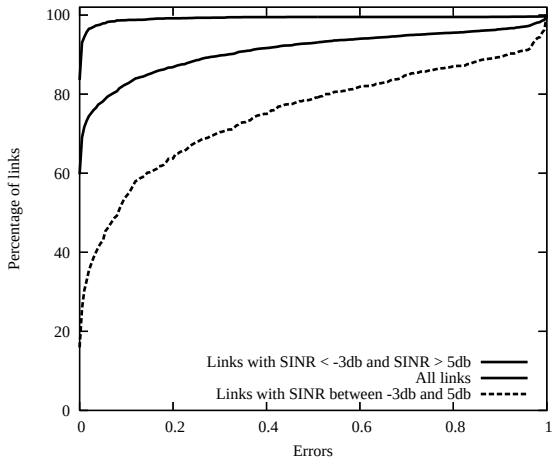


Figure 5.3: CDF of absolute modeling errors for the physical interference model, with all data and data split into transition and non-transition regions.

throughput (normalized) of each link when all links in the subset are active simultaneously in the testbed. The normalized throughput is simply the number of packets received on each link divided by the number of packets transmitted on this link. For each subset, 1000 simultaneous transmissions are done over all links to calculate throughput.

Modeling Errors

Each random subset is used as input to a predictor that provides the link throughput predicted by the physical interference model. Note that all links in a given subset may not be deemed 'feasible' by the model and therefore the PRRs are determined by the corresponding SINR vs PRR relationship derived earlier in this chapter.

5.3.3 Results of Experiments with Random Subset of Links

We plot the modeling accuracy of the physical interference model in terms of cumulative distribution of absolute errors. This absolute error is computed as the difference between the observed PRR through direct measurements and the PRR predicted by the model. We observe in Figure 5.3 that about 40% of links have higher than 90 percentile error. Such low accuracy is quite in contrast with the perfect predictions observed in the results presented before. We conjecture that this poor accuracy was due to the probabilistic nature of the SINR vs PRR model

in the transition region. It was observed before, albeit with a different mote platform [109], that the links in the transition region are highly unreliable. Thus, the SINR-based modeling cannot model the transition region with good accuracy. To validate this hypothesis in our platform, we split out the results into two parts, for the transition and non-transition regions in Figure 5.3. Recall that in our model the transition region is -3 to 5 dB. Note the poor accuracy of the physical interference model in the transition region relative to the non-transition region. It is interesting to note that the model is extremely accurate for the non-transition region case, 90-percentile error is about 1%. However, the accuracy is obviously much poorer for transition region case. We can conclude several things form these results. First, this reaffirms the observations in [109]. Second, excellent modeling accuracy in the non-transition region means that scheduling algorithms that treat links as 'binary', will have excellent results with these two models. Note that all scheduling algorithms known to us are of this type.

5.4 Related Work

A recent paper by Brar et al. [22] can be considered complimentary to our work. Here, the authors investigate algorithms for physical interference model and show via simulations that physical interference modeling leads to more efficient schedules relative to the protocol interference model. However, the simulations use very straightforward propagation and radio models. We also arrive at similar conclusions, albeit via a more elaborate experimentally based method.

Researchers have only begun to study effect of interference in wireless networks using experimental methods. The authors in [109] have studied the *transition region* and quantified its effects. The analysis in the paper is also supported by experimental validation using a motes testbed, though with a different (CC1000) radio. Many of our observations are also similar. Another work [90] by the same group has considered the effect of multiple interferers. They however concluded that the SINR threshold is dependent on number of interferers and the joint interference is not necessarily the sum of individual interference powers. As described in Section 5.2, our conclusions are different, and we have derived a more classical model [80]. In a different work [91], the authors have concluded from measurements on MicaZ motes with CC2420 radios, that RSSI is a good estimate of link quality. This observation is also confirmed by the success of our SINR-based models.

Experimental work has also considered 802.11-based systems to study interference behavior. The difference here is that the sender-side (carrier-sense) behavior in the MAC protocol must also be modeled. Notable articles are as follows. Single

and multiple interferer scenarios have been modeled in [81] and [52], respectively. The need for modeling multiple interferers has been motivated in [30].

5.5 Conclusions

There are two 'take home' points in this paper. First, we develop and validate a method to instantiate physical interference models for use in TDMA scheduling. Second, we demonstrate the accuracy of the physical interference model via extensive experimentation on a motes testbed. The general conclusion is that the SINR-based physical interference model has excellent accuracy if the transition region can be discounted. If the TDMA scheduling uses a binary model of interference (all known algorithms do), ignoring the transition behavior is perfectly acceptable.

A question can arise as to whether the conclusions here are radio-specific, as everything was done on a single radio. Use of different radios is beyond the scope of this paper. We like to think that the general conclusions are radio independent. Even if they are not, we believe that the general methodology would be useful for wireless network researchers for studying interference models with other radios. A study of similar nature using 802.11 PHY layer is a topic of our future study.

Chapter 6

Distributed Protocol for Max-min Fairness in Wireless Mesh Networks

6.1 Introduction

A common problem observed in wireless multihop networks is a situation where externally offered load entering the network exceeds the network capacity. If the network capacity is exceeded, packets are queued en-route to the receiver resulting in higher end-to-end packet delays, and wastage of bandwidth when packets are dropped at intermediate nodes. Unfair distribution of bandwidth among users is another challenge that a network designer needs to address specially in distributed ad-hoc and mesh networks. In this context, an appropriate and viable solution is a maxmin fair rate allocation[53] in which resources are allocated in order of increasing demand such that no user gets a resource share larger than its demand and users with unsatisfied demands get an equal share of the resource. Also a user with unsatisfied demands cannot increase its resource share without reducing the share of others who are already using equal or lesser amount of the resource.

Our goal in this chapter is to develop a distributed max-min fair queuing mechanism that enforces this notion of fairness for multihop flows in wireless mesh networks. We compute the maxmin fair rate of a multihop flow by computing the maxmin fair rate at each hop along its path and finally enforcing the rate offered to the flow at the most constrained hop in the path. This approach provides the framework for a multihop maxmin fair rate allocation as well as bounds the rate at which packets are injected in the network to the maximum rate at which it can be delivered to the destination. Although our queuing mechanism can work with any reasonable MAC protocol, we find that the IEEE 802.11 MAC seriously deviates from fairness principles in certain scenarios [51],[99],[43]. In order to reduce MAC layer unfairness, we replace the exponential backoff mechanism in 802.11, with virtual time based CSMA (VTCSMA) which is a backoff scheme based upon packet

arrival time.

VTCSMA [64] provides a distributed first come first serve medium access to contending nodes. This approach ensures that the scheduling order computed at the upper layer is also enforced in the MAC layer. The VTCSMA protocol was designed for single hop networks, and our work extends it for multihop networks. This is nontrivial as problems such as hidden terminals and starvation must be addressed. Our queuing method and the MAC layer protocol together form a complete protocol suite that computes and enforces max-min fair scheduling in wireless mesh networks in a distributed manner.

The rest of this chapter is organized as follows. In section 6.2, we will explain the background, theory and definition of max-min flow control in the context of wireless multihop networks. We will then describe our upper layer protocol in section 6.3 followed by the MAC layer solution in section 6.4. We present performance evaluation in section 6.5 and related work and conclusions in sections 6.6 and 6.7.

6.2 Background

In wireless networks, transmission between a pair of neighboring nodes (also called single hop flow) interferes with a transmission between another pair if either the two single hop flows have a common transmitter or receiver or if the transmitter or receiver of one is within two hop distance from the transmitter or receiver of the other. The two hop consideration is due to the assumption of an 802.11-like protocol where any transmission can interfere up to two hops. We model these interfering flows using a contention graph, henceforth called *flow contention graph*, where nodes are single hop flows on the network graph and edges are drawn between two nodes if the flows interfere. An example of the flow contention graph is shown in Figure 6.1.

Given this notion of flow contention graph, earlier work [43] has considered max-min fair rate of single hop flows. In our work, we consider end-to-end multihop flows as multiple single hop flows that can go over a sequence of links. We first treat these single hop segments as individual flows and then extend the idea of fairness to multihop flows. To demonstrate the technique let us first describe the notion of feasibility and max-min fair allocation.

A feasible rate allocation essentially constrains the rate allocation for each flow such that the sum total of the rates allocated to all flows belonging to a *clique* in the flow contention graph do not exceed the network capacity. A rate allocation is max-min fair if it is feasible and the only way a flow can get higher rate is by reducing the rate of some other flow that has been allocated equal or lower rate.

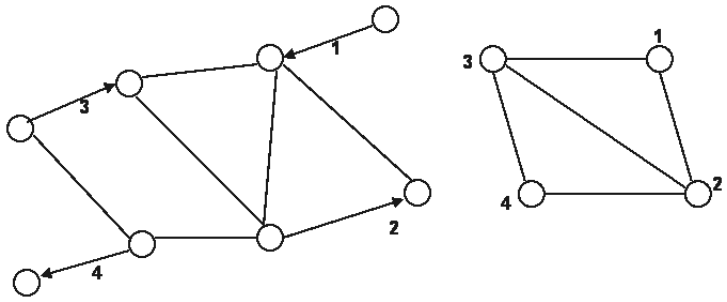


Figure 6.1: Network graph and the corresponding flow contention graph.

Formal definitions are below.

Definition 1 (Feasible Rate Vector) *Assume that C is the link capacity in the wireless network. Let R represent a vector that represents transmission rates r_i allocated to each flow f_i in a "clique" in the flow contention graph. If F is the set of flows in the clique, then the vector R of rates r_i is feasible if*

$$r_i \geq 0, \sum_{\forall f_i} r_i \leq C.$$

Definition 2 (Max-min Fair Rate Allocation) *A feasible rate vector is max-min fair if for any flow f_i, the allocated rate r_i cannot be increased while maintaining feasibility without decreasing r_j for some flow f_j for which $r_j \leq r_i$ [18]. Flows f_i and f_j do not need to belong to the same clique.*

Prior work [43] has shown that a feasible rate vector R is max-min fair if and only if each flow has a bottleneck clique with respect to R. Bottleneck clique is defined as follows.

Definition 3 (Bottleneck Clique) *Given a max-min fair rate vector R, a bottleneck clique cl_i is that clique for which flow $f_i \in cl_i$, $\sum_{\forall f_k \in cl_i} r_k = C$, and allocated rate r_i of f_i is equal or greater than the allocated rate r_k of any other $f_k \in cl_i$. The largest clique in the network is the bottleneck clique for the flows it contains.*

6.2.1 Max-Min Rate Calculation

Based on the above, prior work [43] has provided a mechanism to compute max-min fair allocation of rates on single hop flows in the network. The technique simply determines all cliques in the flow contention graph. Since this can be computationally intractable, heuristics are used for the clique computation. Starting with

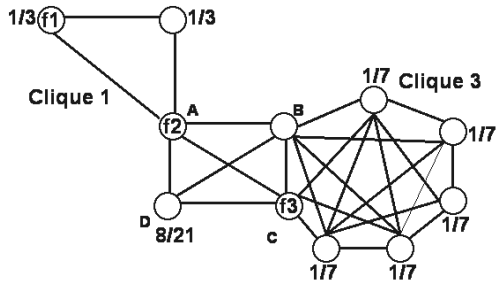


Figure 6.2: Illustrating computation of fair rates.

the largest clique, each flow in the clique is allocated equal share of the remaining capacity except the ones that have already received an allocation. The remaining capacity is simply the capacity C minus the already allocated rates. The allocation is started with the largest clique, as this clique is always the bottleneck for the flows belonging to this clique and thus determines the fair rate allocation of these flows.

For the benefit of the reader, we illustrate the procedure using the example of Figure 6.2. Assume capacity $C = 1$. There are three cliques with 3, 4 and 7 nodes respectively with some common vertices's (A, B, C) corresponding to network flows. The procedure starts with clique 3, assigning a rate of $\frac{1}{7}$ to each vertex of clique 3. Then it turns to clique 2. Since B and C have already been allocated their rates, A and D are allocated the remaining capacity equally. Each of them gets $\frac{1}{2}(1 - \frac{2}{7}) = \frac{5}{14}$. But since the rate allocated to A by clique 1 is only $\frac{1}{3}$ which is less than the rate being offered by clique 2, it receives only $\frac{1}{3}$ rate, while node D finally gets $1 - 2 \times \frac{1}{7} - \frac{1}{3} = \frac{8}{21}$ part of the bandwidth.

6.3 Upper layer Protocol to achieve Max-min fair scheduling

In the prior section, we have described how to compute max-min fair rates for single hop flows in the network. In this section, we develop a queuing mechanism that computes and allocates max-min fair rates to multihop flows. The protocol has three components: "clique formation protocol" that computes the allocations locally on single hop segments of multihop flows; "back pressure protocol" that assigns fair rates to multihop flows; "rate enforcement protocol" which essentially controls the scheduling and enforces that no flow exceeds its allocated rate.

6.3.1 Clique Formation Protocol

In order to compute fair rates for all flows in the network in a distributed fashion, each network node needs to obtain the flow contention graph that represents its *local neighborhood*. The local neighborhood of a node consists of its neighbors that can be reached in up to two hops. A two-hop message exchange protocol gathers enough information to build the local flow contention graph. This can be done by sending "hello" messages and rebroadcasting the contents so that the two-hop neighbors of the original sender can receive the messages as well. These "hello" messages are similar to "hello" messages that many routing protocols (e.g., AODV [75]) employ to maintain neighborhood information; so we do not consider them to be additional overheads except the additional content. Frequency of such exchange for our protocol objective should be the granularity of any topology change or traffic changes (in terms of origination of a new flow or expiry of an existing flow).

Each node i maintains and includes in the "hello" messages, information about the single hop flows that a node originates, receives or routes. These single hop flows may be segments of multihop flows. This information includes the flow id (f_m), the nexthop receiver of the flow (node j) and the rate allocated to the flow ($r_{m,i}$) at node i. Thus, the "hello" messages contain a set of tuples $f_{m,i,j} =< f_m, j, r_{m,i} >$. We will refer to the set of $f_{m,i,j}$ tuples as the *local flow set* (L_i) for node i. Apart from L_i, node i also includes in the "hello" messages, the same information about the flows that interfere with its transmissions. We will refer to this set as the *interfering flow set* or (I_i). The I_i is the union of *local flow sets* L_j of all nodes within the two hop neighborhood of node i. Thus, if N_i is the set of one and two hop neighbors of node i then,

$$I_i = \bigcup_{\forall j \in N_i} L_j.^{1} \tag{6.1}$$

After receiving messages from all neighbors, node i is able to construct a *neighbors interfering set* or P_i such that,

$$P_i = \bigcup_{\forall j \in N_i} I_j. \tag{6.2}$$

This information is sufficient [43] for node i to compute the flow contention graph representing its neighborhood and calculate all cliques in this graph. The fair share of bandwidth of all members of the bottleneck clique in the network is simply the ratio of the bandwidth and the size of the clique [43].

[1] Here we would like to mention that when computing the union or intersect of sets, a node only considers the $< f_m, j >$ pair from the tuple while $r_{m,i}$ is used in rate computations at upstream and downstream nodes.

We cannot obtain the size or content of the bottleneck clique in the entire network due to the hardness of the problem. But we can find all cliques and compute the bottleneck clique in the local neighborhood consisting of few nodes, in reasonable time. Thus, for every flow the node keeps track of the *local bottleneck clique* corresponding to that flow and computes rate, say S. If after subsequent "hello" message exchanges, the node sees that other flows in this clique insist on getting less than rate S, it redistributes the residual rate among other flows in the clique and recomputes the *local bottleneck clique*. Thus, we may claim that, at the steady state, the rate of each flow in the network is equal to that offered by the flow's *local bottleneck clique* which is the max-min fair rate of the flow.

Let us explain this with an example in *Figure 6.2*. This figure represents a flow contention graph of the network. Clique 3 is the largest clique in the network and thus is a *local bottleneck clique* for all member flows. Flow A in the graph is a member of both clique 1 and clique 2. The rates offered by the cliques to flow A are $\frac{1}{3}$ and $\frac{5}{14}$ respectively. Thus although clique 2 is the largest clique for flow A in terms of size, clique 1 is the bottleneck clique as it allows a rate lower than clique 2.

6.3.2 Back Pressure Protocol

In the previous section, we treated multi-hop flows as multiple single hop segments of the flow thereby assigning rates to each segment of the flow at the local bottleneck cliques. We now introduce the notion of a *global bottleneck clique* for multihop flows as the clique at which the flow receives the least rate along its path. A more formal definition is as follows.

Definition 4 (Global Bottleneck Clique) *A global bottleneck clique for a multihop flow is the clique containing the single hop flow segment $f_{m,i,j}$ (flow id m, from node i to node j) of the multihop flow $F_{m,a,b}$ (flow id m, from source a to destination b), where the offered rate $S_{m,i,j}$ at node i is less than the rate offered at any other node k along the flow's path.*

Consider *Figure 6.2* again. A multihop flow F in the figure is represented by three single hop flow segments – $f1$, $f2$ and $f3$. The rate offered at each of these segments are $\frac{1}{3}$,$\frac{1}{3}$ and $\frac{1}{7}$ respectively. Thus clique 3 is the global bottleneck clique for flow F since it offers the least rate compared to other cliques along the path from source to destination.

If the rate provided at upstream nodes of a multihop flow is larger than the rate offered at the *global bottleneck clique*, packets may be queued and dropped at the forwarding nodes. Similarly, if the rate offered at downstream nodes is higher than the rate allocated at the global bottleneck clique, the allocated rate will remain

unused instead of being utilized by other flows with unfulfilled demands. In order to prevent such wastage of bandwidth, we introduce a back pressure protocol in which each node limits a multihop flow's rate to the minimum of the rates provided at the next hop, at the previous hop and at the current hop. The source and destination of the multihop flow, limit the flow's rate to the minimum of the computed rate and that offered at the next or previous hop respectively. This scheme achieves what the authors in the paper [95] have tried to achieve by a more complex token generation process. Due to this *back pressure* mechanism, the rate offered by the global bottleneck clique for the flow is propagated to all nodes along the path from the source to the destination of the flow. The extra bandwidth available after applying the back pressure technique is distributed among other flows after the next hello message exchange and the *local and global bottleneck cliques* are recomputed. A detailed mathematical analysis of the token based back pressure technique is presented in [95] which also applies to our technique.

6.3.3 Rate Enforcement Protocol

In order to enforce the assigned rates, the protocol needs to ensure that the rate at which the packets are transmitted follows the rate computed by the *clique formation protocol* and the *back pressure protocol*. We employ a timer based mechanism to "release" packets at the computed rate. A flow may be served only if there is a packet that has been "released" for transmission. Every node that has packets to send, runs a timer, which we will refer to as the *release timer*. The interval of release timer is calculated dynamically and depends upon the number of contending flows in the local neighborhood. When the release timer fires, the node checks if there is a flow from which a packet can be "released". A packet can be "released" if the flow to which the packet belongs has used less than its allocated rate otherwise the next flow is considered. This scheme ensures that each flow receives no more that the rate computed by the clique formation and back pressure protocols, thereby enforcing the computed rates.

6.4 Virtual Time Based MAC Protocol

The three step upper layer protocol that we proposed in the previous section can be used in conjunction with any reasonable MAC layer protocol in wireless network. However, we know from [99],[51],[43] that the commonly used IEEE 802.11 MAC protocol suffers from several unfairness issues. This is due to several reasons including exposed terminals, hidden terminals and the backoff policy used in 802.11. We have developed a medium access protocol to complement our

scheduling scheme. Our MAC protocol performs a packet arrival based backoff mechanism known as virtual time CSMA (VTCSMA) [64] rather than random exponential backoff mechanism used in 802.11.

The VTCSMA MAC protocol implements a first come, first serve access to the shared medium by emulating a single server multiple queue system. Only here the queues are maintained at different nodes in the network and the scheduling decision must be made in a distributed manner. In order to achieve this distributed scheduling process, each node in the network maintains two clocks, *real clock* and *virtual clock*, to measure the passage of *real time* and *virtual time* respectively. Both clocks may be initialized to zero and the real clock runs at a constant rate. The virtual clock runs η times faster than the real time clock while the medium is idle (unless the two clocks are in sync, in which case they run in lock steps). The virtual clock is stopped whenever the medium becomes busy and it resumes when the medium is idle again. When the virtual clock of a node passes the arrival time of the packet in the head of its queue, the packet is transmitted. If all nodes in the network share the same wireless medium and follow this transmission rule, the first-come first-serve scheduling is trivially achieved in a distributed manner. The analysis in [64] shows that this protocol can potentially provide a higher goodput as compared to random access CSMA.

VTCSMA as described above provides fair medium access when all nodes are within a single collision domain i.e., all nodes are within receive range of one another. Since in a single collision domain, nodes can "hear" transmissions from each other, the virtual clocks run almost in sync or atleast at the same average rate. The average rate is calculated as the rate at which the virtual time progresses with respect to progress of real time. The average rate of virtual clock at any node depends upon the contention level it experiences. Also since a packet is transmitted only when the virtual time reaches the packet arrival time, the throughput achieved by a node is also a function of the average rate of the virtual clock. In a multihop network, the contention experienced by nodes differ from one region to another. It is easy to construct scenarios where some nodes experience larger contention than their neighbors thereby getting fewer chances to transmit than other nodes. This phenomenon may lead to unfair share of bandwidth and even starvation. Figure 6.3(c) shows a typical scenario where this may happen. Here node 5 being in the carrier sensing range of both nodes 0 and 3, faces higher contention than either node 0 or node 3 which do not contend with one another. Therefore, the average rate of node 5's virtual clock is lower than that of 0 and 3. We suggest a two step approach to address this problem in the multihop extension of the VTCSMA protocol described in the next section.

6.4.1 VTCSMA in Wireless Multihop Networks

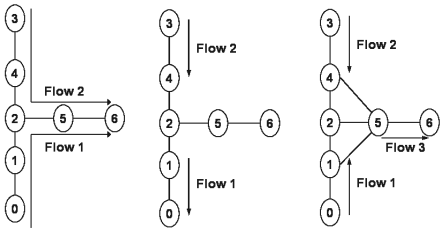
We have proposed a multihop VTCSMA MAC protocol that alleviates the starvation problem of VTCSMA. We borrow the virtual carrier sensing and solution to hidden terminal problem from IEEE 802.11 where nodes maintain "network allocation vectors (NAV)" and exchange RTS/CTS control packets to maintain channel state and to notify potential interferers of the impending transmission.

To solve the starvation problem in VTCSMA, we propose that every packet must carry the virtual time stamp of the transmitting node and every node in the network must follow a two step approach to prevent starvation. In the first step which we name "good neighbor approach", nodes reduce the possibility of starvation of their neighbors by adjusting their virtual clock to minimum of the virtual time stamp from overheard packets and the time measured by the local virtual clock. The second step which we name "bad neighbor approach" is invoked when a node that has packets to transmit, overhears another packet with a virtual time stamp that is ahead of its own virtual time by more than a fixed threshold (an indication of starvation). The starving node then sends a jamming message that conveys this situation to all receivers in its vicinity, forcing all nodes to invoke their collision recovery mechanism i.e setting the NAV and withholding all transmissions. Here we propose an additional network allocation vector called "soft NAV". When a node detects a jamming signal or a collision, it waits for the medium to become idle again and then sets a "soft NAV" in addition to the regular NAV. During this "soft NAV" state or "soft state", nodes do not run their virtual clock and do not initiate any transmission, but they may receive unicast transmissions and send acknowledgements. While neighboring nodes are in the "soft state", the starving node gets the opportunity to transmit its backlogged packets. At this time, nodes with faster virtual clocks adjust their clocks in the manner of the "good neighbor approach". This two step approach is instrumental in reducing the difference between average rate of virtual clocks in the network which prevents starvation in the network.

6.5 Results

We evaluated the performance of our queuing protocol and compared with a first-come-first-serve scheduling mechanism that schedules packets in the order they arrive in the queue at each node without consideration for the flow to which they belong. We have also compared the performance of the two MAC protocols in conjunction with each scheduling protocol. We used fairness index and goodput as the metrics to evaluate performance.

Definition 5 (Fairness Index) *If a system allocates resources to n contending users,*



(a) TCP flows from node 0 to node 6 and node 3 to node 6.

(b) Information Asymmetry: UDP flows from node 1 to node 0 and node 3 to node 4.

(c) Perceived Collision: UDP flows from node 0 to node 1, node 3 to node 4 and node 5 to node 6.

Figure 6.3: Network graphs of representative scenarios

such that the i^{th} user receives an allocation x_i, then fairness index is defined as

$$f(x) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2}, x_i \geq 0.$$

Definition 6 (Goodput) *Goodput is defined as the number of application layer data bits successfully received at the receiver over the total span of time for which the application layer sent data.*

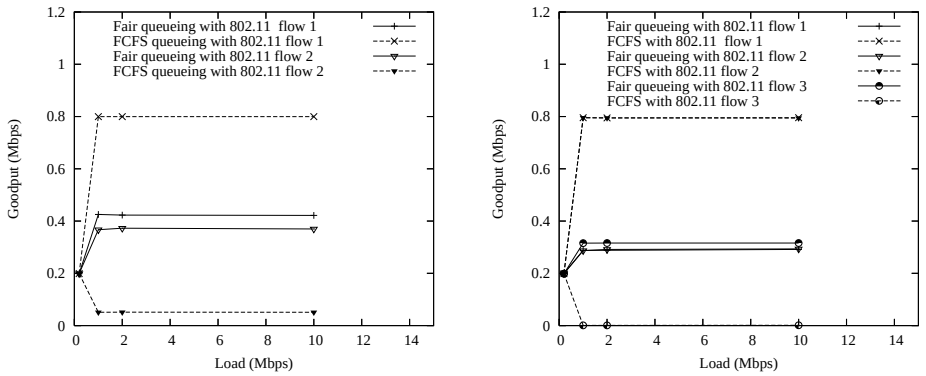
We have used network simulator ns2 version 2.27 [34] for all simulations. We have experimented with both small scenarios that represent specific problems that arise in multihop networks as well as random scenarios with varying packet rates and number of traffic sources.

6.5.1 Max-min Fair vs FCFS Scheduling with IEEE 802.11

We placed 7 nodes in a network as shown in *Figure 6.3(a)*. We set up two TCP flows in the network, flow 1 from node 0 to node 6 and flow 2 from node 3 to node 6. We present the result of this experiment in table 6.1. We observe that the max-min fair scheduling protocol distributes the bandwidth more evenly between the two flows with flow 1 achieving a rate of 53kbps and flow 2 achieving

Table 6.1: Goodput vs load for symmetric scenario of *Figure 6.3(a)* with two TCP flows from node 0 to node 6 and node 3 to node 6

Flow	FCFS Queue(Kbps)	Fair Queue(Kbps)
1	169.46579	52.94678
2	0.70691	51.1774



(a) Information Asymmetry (b) Perceived Collision

Figure 6.4: Goodput vs load for networks in Figure 6.3(a), 6.3(b) and 6.3(c)

51kbps, but in FCFS scheduling, flow 1 receives a goodput of 169kbps while flow 2 is starved.

In the network shown in Figure 6.3(b) two UDP flows represent the information asymmetry (IA) scenario [51]. Here, node 1 that originates flow 1 is within the carrier sensing range of node 4 which receives flow 2. On the other hand, node 3 that originates flow 2 does not have any information about flow 1 because it is beyond the transmission range of both node 1 and node 0. Since node 3 is unaware of transmissions by node 1, it is possible that node 3 attempts to transmit data while a transmission between nodes 1 and 0 is going on. These transmissions from node 3 may not be received correctly at node 4 due to interference with transmissions from node 1 causing multiple retransmission attempts by node 3. These retransmissions, in 802.11 based MAC protocols, lead to a larger contention window at the sender thus reducing its probability of acquiring the medium. This is reflected in the results shown in *Figure 6.4(a)*, where the goodput achieved by flow 1 is more than 75% larger than that achieved by flow 2.

In *Figure 6.3(c)*,we constructed a perceived collision [51] scenario with UDP flows from node 0 to node 1, node 3 to node 4 and node 5 to node 6. In a perceived collision scenario, three flows ‘1’, ‘2’ and ‘3’ are such that flows ‘1’ and ‘2’ do not contend with one another but flow ‘3’, contends with both flows ‘1’ and ‘2’.

Since the flow in the middle has to defer for the flows on each side, and therefore faces more contention compared to the neighboring flows, it gets fewer chances to transmit packets. Results in *Figure 6.4(b)* show that the middle flow receives very little share of the bandwidth while flows '1' and '2' each are able to receive 80% higher bandwidth share.

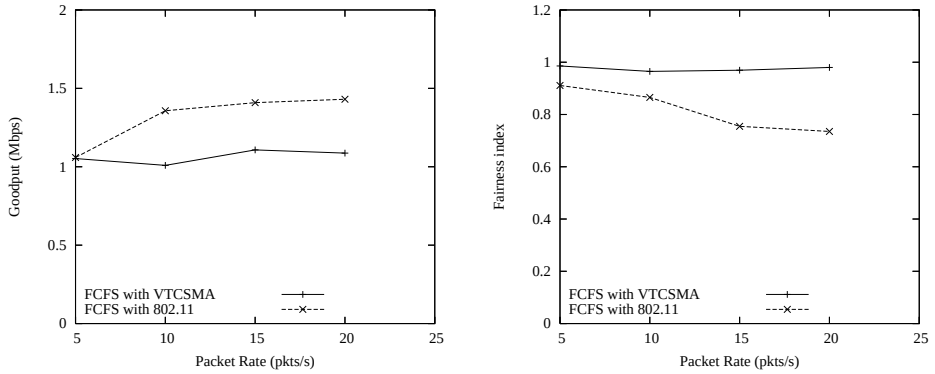
When maxmin fair scheduling is used in both information asymmetry and perceived collision scenarios, we observe that the contending flows form a clique in the network and thus equally divide the bandwidth among each other thereby achieving nearly equal goodputs as shown in *Figure 6.4(a)* and *Figure 6.4(b)*.

6.5.2 Multihop VTCSMA vs IEEE 802.11

We performed some experiments to demonstrate the advantage of using multihop VTCSMA over IEEE 802.11. We randomly placed 50 nodes in a network of size 1500x300m. Each node in the network transmits packets to a randomly selected neighbor. The virtual clock rate in VTCSMA is 200 times the real clock rate. The packet size is 512 bytes and we vary packet rates and compare fairness index and goodput for multihop VTCSMA and IEEE 802.11 in *Figure 6.5(b)* and *Figure 6.5(a)* respectively. We observe that VTCSMA achieves nearly perfect fairness index but lower goodput compared to 802.11. Here 802.11 achieves a higher goodput compared to VTCSMA but the fairness index graph shows that this is at the cost of unfair distribution of bandwidth among flows. The lower bandwidth utilization in fair scheduling protocols is due to the conflicting nature of the two goals. In [60] the author explains the difficulty of simultaneously achieving both fairness and maximizing bandwidth usage.

6.5.3 Maxmin and FCFS Scheduling with Multihop VTCSMA and IEEE 802.11

We randomly placed 50 nodes in a network of size 1500x300m and selected multihop flows between random pairs of nodes in the network. We experimented with 5,10,15 and 20 traffic connections that transmit UDP packets of size 1000 bytes at a rate of 10pkts/s. We compared the goodputs and fairness index5 of the two scheduling protocols under varying load conditions and the plots are shown in *Figure 6.6(a)* and *Figure 6.6(b)*. We observe that with 20 traffic sources, maxmin scheduling with VTCSMA MAC provides a fairness index above 0.9 while fairness index in maxmin scheduling with 802.11 MAC protocol drops to 0.8. FCFS with VTCSMA is more fair compared to FCFS with 802.11. Also note that max-min fair scheduling with VTCSMA in the MAC layer outperforms all combinations in



(a) Goodput vs packet rate. (b) Fairness Index vs packet rate.

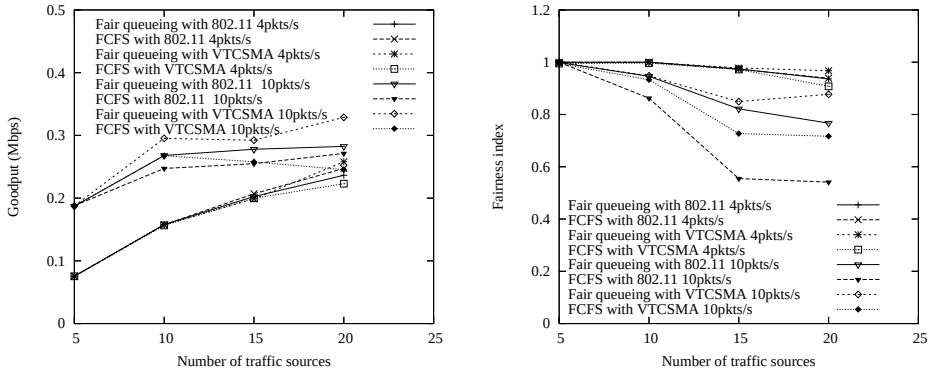
Figure 6.5: Multihop VTCSMA and IEEE 802.11 MAC and FCFS in 50 node random multihop networks.

terms of both fairness index and goodput. These results clearly demonstrate the advantages of the protocol suite that we have proposed in this work.

6.6 Related Work

Fair scheduling of flows in a wireless multihop network has been a popular topic of research for several years. In some of the earlier works, researchers have focused on providing a MAC layer solution for fair bandwidth allocation. In [99] the authors have proposed a scheduling discipline to schedule packets on an arrival time and packet size basis with concepts similar to virtual time CSMA. We discussed earlier in this chapter the drawbacks of using virtual time for scheduling in multihop networks. Since this scheme was suggested for wireless LAN, the authors did not discuss the problems that may arise in wireless multihop networks. Similarly the scheme suggested in [50] and [51] schedules packets on a priority order, where the priorities are learned from information piggy backed on control and data packets. These papers also provide MAC layer solutions and fairness is achieved by appropriate backoff policy.

In [60], the authors have provided a two tier solution to provide maxmin fair allocation for local flows and to maximize the network throughput. In the first step, the protocol achieves the fairness model by selecting a set of flows and then in the second step, the protocol tries to maximize the bandwidth utilization by scheduling the maximum independent set subject to the selection of the flows in the first phase. Since the problem of finding the maximum independent set is NP-complete, the authors implement a minimum degree greedy algorithm. The distributed imple-



(a) Goodput vs packet rate. (b) Fairness Index vs packet rate.

Figure 6.6: Fair queuing with multihop VTCSMA and IEEE 802.11 in 50 node random multihop networks.

mentation of the global model proposed in the paper requires that each time there is a change, the new information must be disseminated throughout the network in order to maximize network throughput. A backoff based protocol is used to achieve the local fairness model and to implement the minimum degree greedy algorithm for maximizing bandwidth utilization.

In [43] the authors allocate maxmin fair rate to single hop flows in a multihop network and the fair rate of each flow is limited by the share provided by the bottleneck clique. The fair rate of a flow is calculated by computing the rate provided by the largest clique in the flow's flow contention graph and the fair rates are achieved by a backoff based MAC protocol. The authors in [95] present an algorithmic perspective of max-min fair allocation in wireless multihop networks. The network model used in this work is different from what we used in our work. Here each node in the network has a locally unique frequency, thus there is no location dependent contention. Unlike [43], flows are multihop flows and the fair rate of a flow in the network is limited by the share provided by the bottleneck link along the path of the flow.

6.7 Conclusion

We have defined max-min fairness in terms applicable to multihop flows in wireless mesh networks. We have then developed a protocol suite to achieve max-min fairness in a distributed manner in the network. Our solution consists of an upper layer protocol for achieving max-min fairness that can be used with any MAC protocol. This protocol suite also consists of a fair MAC protocol that schedules

flows on a first in first out basis. This MAC protocol truly complements our upper layer protocol to provide a complete implementation of max-min fair scheduling in mesh networks. We have presented a comprehensive performance evaluation of the protocols and compared performances with IEEE 802.11 and FCFS scheduling protocols.

Chapter 7

Collision Avoidance in a Dense RFID Network

7.1 Introduction

RFID (radio frequency identification) [35] is an automatic identification system that consists of two components – readers and tags. A tag has an identification (ID) stored in its memory that is represented by a bit string. A reader is able to read the IDs of tags in the neighborhood by running a simple link-layer protocol over the wireless channel. In a typical RFID application, tags are attached or embedded into objects in need of identification or tracking. In the most common application of RFID (e.g., supply-chain management), RFID tags simply serve the purpose of UPC bar codes. By reading all the tag IDs in the neighborhood and then consulting a backend database that provides a mapping between IDs and objects, the reader learns about the existence of corresponding objects in the neighborhood. This way RFID readers also act as identification and/or proximity sensors.

RFID tags can be either *active* or *passive* depending on whether they are powered by battery. We are interested in passive tags in this chapter. Passive tags are prevalent in supply chain management as they do not need a battery to operate. This makes their lifetime unlimited and cost negligible (only few US cents per tag). The power needed for passive tags to transmit their IDs to the interrogating reader is supplied by inductive coupling between the reader and tag antennas. The reader "energizes" the tags in the vicinity with RF power continuously for the entire read operation. In the most prevalent form of the technology, part of this power is used to transmit a response back to the reader (using a process called *backscattering*) after appropriate modulation and coding via the tag's electronics.

While RFIDs have mostly been used in supply chain management so far, our interest in this chapter is studying their performance in a very dense deployment scenario as will be common in "smart environment" applications. In such applications, we envision that there will be a lot of tiny readers deployed in a dense fashion – much like a sensor network – observing the tagged environment around them by

reading tags continuously or periodically. There will also be a lot of tags around in such environments. This will certainly be the case in smart home or office scenarios as RFID tags will soon replace the UPC bar codes for any item we buy in stores.

However, several collision problems might occur when multiple readers are used within close proximity of each other. Thus, the concurrent read operations must be coordinated appropriately. We will elaborate on these problems in the following section. Current generation RFID systems do not address the multi-reader coordination problems effectively because of their emphasis on supply chain where multiple readers are rarely used in the same physical space.

In this chapter, we design and evaluate a simple carrier sense-based MAC protocol to avoid collisions in multi-reader scenarios. We build it specifically for a tiny Berkeley mote-based platform [102] for deployments in smart environment applications. The goal in this chapter is to describe the design choices we made, the protocol operation and preliminary performance results. The key feature of this design is the use of an RFID tag antenna as an apparatus to measure receive signal strength and the mote platform to sample it. While many other sophisticated solutions (e.g., use of TDMA-based approaches or multiple frequencies) are possible, the approach we present is simple, requires a bare minimum of electronics to build and performs effectively.

The rest of this chapter is organized as follows. We present our system design in Section 7.2, followed by the description of the MAC protocols in Section 7.3, their performance evaluation in Section 7.4 and concluding remarks in Section 7.5.

7.2 System Design

In this section, we present the hardware design for a RFID reader that uses carrier sensing to avoid collisions. This system consists of an OEM RFID reader module, a host micro-controller and a received signal strength indicator.

7.2.1 RFID Reader Module

We use the SkyModule^{TM}M1-mini [2] multi-protocol 13.56 MHz OEM RFID reader module for our work. The read range of this reader is up to 7cm with the internal antenna. The actual range is somewhat dependent on the size of the tag antenna and also the tag orientation. It can read upto 20 tags in a second. It is capable of communicating with a host micro-controller over the TTL, SPI and I2C interfaces. The reader module is capable of responding to ASCII and binary commands sent by the host micro-controller. It can select, read and write RFID tags.

The host controller can also read and write the reader's memory and system registers to put the reader in low power sleep mode and to wake it up from sleep. The small footprint and low power requirement makes it suitable for being integrated with the processor radio modules used in RFID-based sensor networks.

7.2.2 Host Micro-controller

We have interfaced the Skyetek RFID reader to a mica2dot processor radio module. Mica2dot is based on the well-known Berkeley mote architecture [102] and is manufactured by Crossbow technologies [1]. Equipped with Atmel's Atmega128L 4MHz, 8 bit micro-controller and Chipcon's CC1000 radio, mica2dot can communicate with the RFID reader module via the TTL interface and with the central computer over a 433 or 900MHz wireless link. This setup enables untethered communication between a central controller and the RFID readers. Mica2dot can be programmed with the TinyOS operating system [4, 40].

7.2.3 Received Signal Strength Indicator

Much of our work has centered around building and experimenting with this module. SkyeModule^{TM}M1-Mini uses a Texas instruments TI-S6700 multi-protocol transceiver. This transceiver does not provide received signal strength of the signal received from tags or neighboring readers. Since we could not obtain the received signal strength directly from the reader, we have built a signal strength indicator circuit that can provide an accurate estimate of the signal strength received from other readers in the neighborhood. This signal strength indicator is later used by the MAC protocol designed to avoid reader-reader and reader-tag collisions.

The Tag-it RFID tag manufactured by Texas Instruments is used to measure the signal level at any point in the reader antenna system. It is often used as charge level indicator to design reader antenna [96] by simply removing the IC from the tag. When the tag is brought in the RF field of a reader's antenna system, a voltage is induced in the parasitic capacitor on the tag. This is a high frequency sine wave whose amplitude varies with the amount of voltage induced in the tag's antenna due to the reader's RF field. In order to measure this signal amplitude accurately, we use an IF limiting amplifier that takes this signal as input and provides a steady voltage as a logarithmic (in db) measure of the input signal amplitude. This voltage can serve as the received signal strength indication (RSSI). We have used the AD8306 chip [10] as the high precision limiting-logarithmic amplifier. The chip provides a perfect linear relationship between the output voltage and the input signal level in

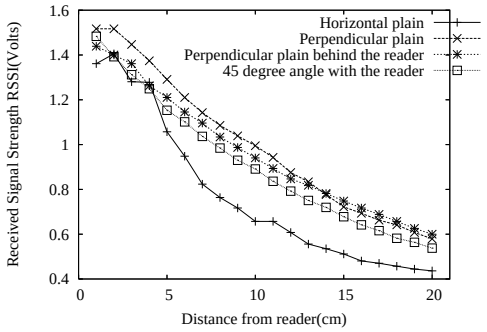


Figure 7.1: Received signal strength vs. distance between a reader transmitting RFID commands and our RSSI circuit.

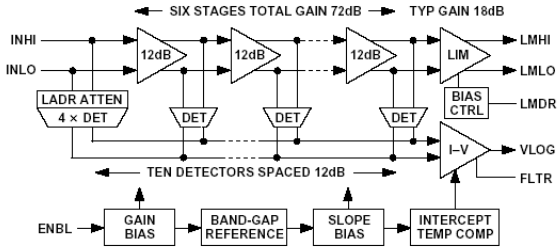
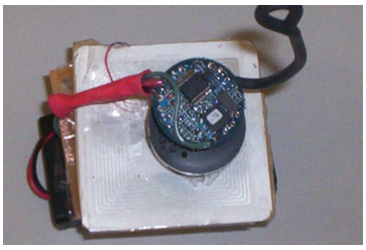


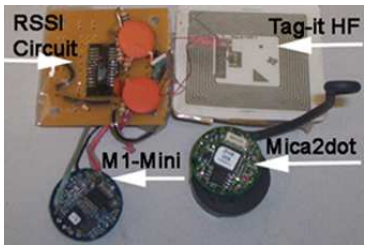
Figure 7.2: Circuit diagram for the received signal strength indicator (RSSI) circuit[11].

db. We connected the output from the charge level indicator (Tag-it HF RFID tag) as a differential input on SIG_{INHI} and SIG_{INHLO} of the circuit shown in Figure 7.2. The RSSI voltage as measured by this circuit is available at V_{RSSI} and can be sampled by an ADC (analog to digital converter) to "sense" the presence of an active reader in the neighborhood. We use one of the mica2dot's ADCs for this purpose.

To understand the characteristics of our prototype, we measured the variation of the RSSI values obtained from this circuit with distance from an active reader. The results (Figure 7.1) show that the RSSI progressively diminishes with distance from the reader as expected. We performed this experiment with the RSSI circuit moving away from the reader in the perpendicular plane with respect to the reader antenna. We did this for both sides of the reader. We also moved the RSSI indicator sideways from the reader antenna, i.e., in the same plane as the reader antenna. We measured the RSSI at an angle of 45° with respect to the reader's antenna as well. This set of experiments indicate that the radiation pattern from the reader's antenna is not perfectly omni-directional.



(a) RFIDMote.



(b) Components of the RFIDMote shown separately.

Figure 7.3: RFIDMote and its components.

7.2.4 RFIDmote

The RFID reader module is connected to mica2dot mote that serves as the host micro-controller and communicates with it via the TTL interface. The output of the RSSI circuit described above is connected to ADC2 on the mica2dot and the PW0 port on mica2dot provides the external enable switch to the RSSI circuit. Thus, when the received signal strength is needed, the PW0 port provides the voltage to enable the RSSI circuit and the signal strength is obtained by sampling on ADC2. The RFID reader module, mica2dot mote and RSSI circuit together form the complete system that we have used to evaluate the proposed MAC protocol. We will henceforth refer to this complete system as the **RFIDMote** (Figure 7.3).

7.2.5 Power Consumption

Since the target application is an RFID sensor network with battery driven RFIDmotes, power consumption is an important design consideration. The RFIDMote is powered using a 3V power supply consisting of two AA size batteries. We have measured that the RSSI circuit consumes 14 mA current when it is turned on by applying a voltage on the external enable switch. The RSSI circuit is turned on only when the RFIDMote needs to sense the carrier before instructing the reader to start a new transmission. The RFID Reader module consumes 10 mA current when it is in the idle mode, 60 uA in sleep mode and 60 mA when scanning for tags. Since the RFID reader takes about 100ms to wake up from the sleep mode, we keep the reader in IDLE mode at all times, except if the RFIDMote is itself in sleep mode.

The mica2dot can operate at a low power mode with the radio turned off (8 mA current consumption) or in a sleep mode($\leq$ 1uA current consumption). The radio is turned on only when the RFIDMote needs to communicate tag data. The radio consumes 27 mA in the transmit mode and 10 mA in the receive or idle mode.

Table 7.1: Power Consumption of RFIDMote at 3V input.

RFIDMote	Mica2dot	RFID Reader	RSSI Circuit	CC1000 Radio	Current Used(mA)
Sleep	SLEEP	SLEEP	OFF	OFF	0.007
Idle	IDLE	SLEEP	OFF	OFF	8
Ready	IDLE	IDLE	OFF	OFF	18
Carrier Sensing	IDLE	IDLE	ON	OFF	32
Scanning for tags	IDLE	SCAN	OFF	OFF	68
Transmit data	IDLE	SLEEP	OFF	Transmit	35
Receive data	IDLE	SLEEP	OFF	Receive	18

Based upon these known or measured values we estimate the current consumption of RFIDMote in various states and tabulate the results in Table 7.1. A designer can use these values as a guidance for protocol design. Note that channel sensing (i.e., sampling RSSI values) is much less expensive than scanning for tags. Given that the channel sensing is only momentary relative to scanning for tags, channel sensing can provide valuable energy savings as it eliminates wasteful scanning.

7.3 Protocols

We implemented three protocols to evaluate tag reading performance in a multi-reader environment. These three protocols – *naive protocol*, *random protocol* and *CSMA protocol* – are discussed in this section. Since we do not have control over the reader firmware, we have implemented these protocols in RFIDMote in software using TinyOS.

7.3.1 Naive Protocol

In the naive protocol, the RFIDMote transmits a reader-tag inventory request at constant intervals. If two readers are placed in such a way that their interrogation zones overlap, it is possible that some tags would escape detection due to collision

(reader-reader collision). Also if two readers are active at the same time and they are close to each other, the signal from one reader would interfere with the tag responses received from the other (reader-tag collision). Since the readers send commands at the same fixed intervals, these collisions may be repeated and it is possible that some tags are never read by any reader. This is a naive reading procedure and is quite prone to reader-tag and reader-reader collisions.

We implement this protocol on the mica2dot using TinyOS. The mica2dot starts a timer using the `call Timer.start (TIMER_ONE_ SHOT, interval)` command and when `event Timer.fired()` is signaled, the mica2dot sends a "read" command to the reader via the TTL interface. The reader now attempts to read the IDs of all tags in its interrogation zone. In this mode, the reader executes the STAC anti-collision protocol, to prevent tag-tag collision discussed earlier. When the reader gets a tag response, it sends the response to the mica2dot via the TTL interface. When all tags have been read, the reader sends a special "read complete" command to indicate that it has completed the execution of the anti-collision protocol and there are no more tags to be read. When the mica2dot receives the "read complete" command, it stores the tag IDs read by the reader. The central computer polls each RFIDMote one at a time to receive the tags read by the readers.

7.3.2 Random Protocol

The naive protocol is prone to reader-reader and reader-tag collisions. A simple method to reduce the chances of collision is the introduction of randomization in the reading schedules. Thus, if the readers choose to backoff for a random interval before sending a read command, the probability of collision may be lower. We introduce a random access protocol in which the mica2dot in RFIDMote, sends a read command to the reader after waiting for a random interval. In TinyOS this random interval is generated by using the `RandomLFSR` component. The size of the window may be varied by masking the 16 bit random number generated via the `RandomLFSR` component. Thus, if the desired window size is 2^7ms, we mask the random number by a bitwise AND with 0x3F. When the mica2dot on the RFIDMote is ready to send a read command to the reader, it goes into a random backoff state by starting a timer for a random duration by executing `call Timer.start (TIMER_ONE_SHOT, (call Random.rand()) & cw)`, where, `cw` is the masking integer to limit the value of the generated random number within the desired window size. When `event Timer.fired()` is signaled, mica2dot sends a read command to the reader, which then immediately starts the RFID transmission. Since the RFIDMotes choose to send commands after random intervals, the commands from two readers would not be concurrent with high probability, given that the window size is sufficiently large. In case there is a collision, it is less likely

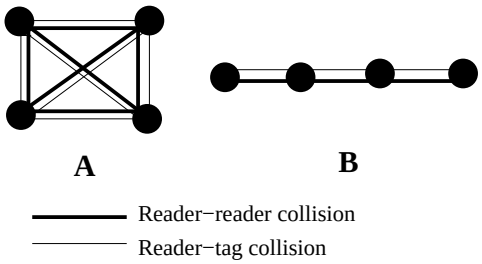


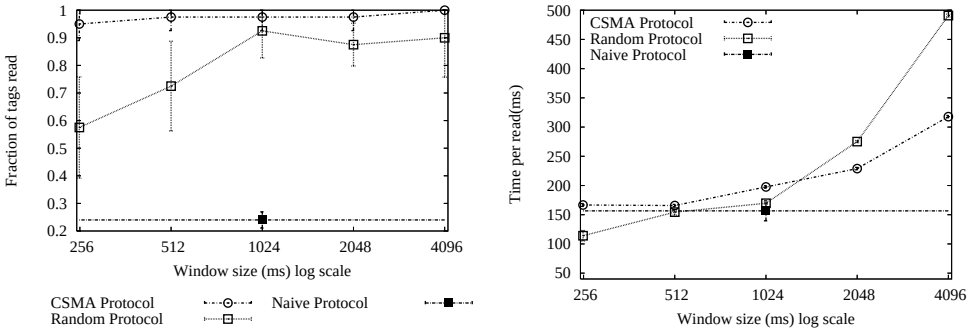
Figure 7.4: Conflict graphs for (A) square grid and (B) straight line configurations.

that the collision will recur for subsequent read commands because the RFIDMote re-selects the interval each time it sends a command.

7.3.3 CSMA Protocol

Here, when the mica2dot on the RFIDMote is ready to send a read command to the reader, it starts a backoff timer for a random interval, by executing `call Timer.start (TIMER_ONE_SHOT, (call Random.rand()) & cw)` command. Meanwhile, the mica2dot continuously samples the voltage on ADC2 to which the RSSI circuit is connected. If the voltage read from the ADC is less than a threshold voltage throughout the backoff interval, i.e., until `event Timer.fired()` is signaled, mica2dot sends the "read" command to the reader. In case the mica2dot senses that the medium is busy, i.e., it reads a voltage higher than the threshold voltage on the ADC2 port, it stops the timer by issuing the `call Timer.stop()` command which prevents `event Timer.fired()` from being generated. The mica2dot then continues to sense the medium and when the medium becomes free and stays free for a random duration between 1 and 16 ms, it restarts the timer. This carrier sensing and backoff procedure, enables the RFIDMote to make a more informed decision about scheduling the RFID transmission, that in turn further reduces the chances of reader-reader and reader-tag collisions.

A note is due on the choice of threshold voltage. We have observed that reader-reader collisions occur when the voltage read from the ADC is greater than 1 V which corresponds to a maximum distance of about 10 cm between the readers. The reader-tag collisions occur at a slightly higher voltage, when the two readers are about 5 cm apart. At this distance, the tag may be able to receive signals from both readers. Thus, to solve both reader-reader and reader-tag collisions, we chose the lower of the two, i.e., 1 V as the threshold voltage.



(a) Accuracy. (b) Time per read.

Figure 7.5: Accuracy and time taken per read vs. window size for four readers in a square grid.

7.4 Performance Evaluation

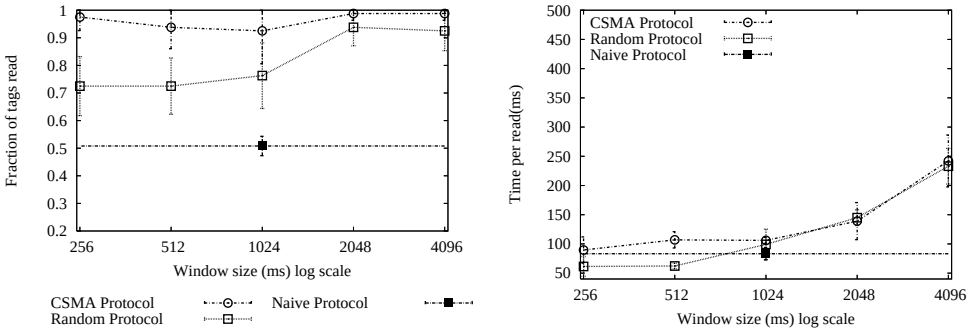
We will now discuss the experimental setup and analyze the performance of the RFIDMotes with the protocols discussed in the previous section. We have used accuracy and time taken per read as two performance metrics in our experiments. Let us first define these metrics.

Definition 7 (Accuracy) *Accuracy is the ratio of the number of unique tags read by all readers to the total number of tags in the interrogation zone of all the readers.*

In order to compute the accuracy of the system we need to determine the number of tags in the interrogation zone of the readers. We activated readers one at a time and allowed them to read all the tags in their respective interrogation zones without any interference from other readers. We recorded the number of unique tags that were read by all readers in each experimental setup. This is the maximum number of tags in the entire interrogation zone. We then use this number for calculating accuracy.

Definition 8 (Time per read) *Time per read is the ratio of the maximum time taken to complete all reads to the number of tags read.*

The maximum time is the time taken by the reader that finishes last and the number of tags read is the total number of unique tags read by all readers. Time is calculated from the point the RFIDMote starts the timer before sending the read command to the RFID reader and until the reader sends the “read complete” response indicating that there are no more tags to read.



(a) Accuracy. (b) Time taken per read.

Figure 7.6: Accuracy and time taken per read vs window size for four readers in a straight line.

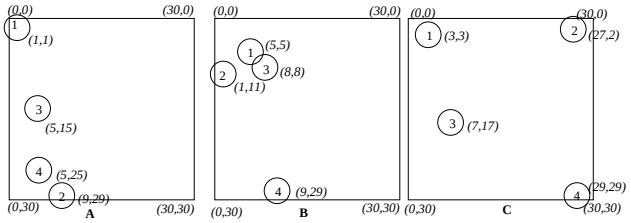
7.4.1 Experimental Setup

We built and programmed four RFIDMotes and arranged them in different configurations (or topologies) to experiment with the protocols described before. There are 25 tags distributed uniformly in the area . The experiments are controlled by a central computer that broadcasts commands to the RFIDMotes to run specific protocols with specific parameters (e.g., window size) and collects results at the end of the experiments. Each individual experiment is repeated 20 times and average performance metrics are presented. For protocol comparison identical configurations (RFIDMotes and tags) are used.

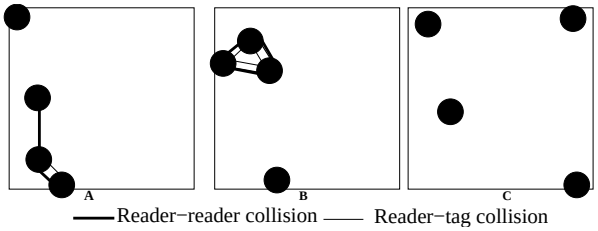
It is expected that the performance of the protocols will be influenced by the density of the RFIDMotes as this influences how probable the collisions are. Thus, for each configuration we experiment with we show a conflict graph to demonstrate what types of collisions are likely. The conflict graph shows an edge between two nodes (RFIDMotes) that can potentially collide. A thick edge is drawn to denote reader-reader collision and a thin edge is drawn to denote reader-tag collision. The conflict graph is determined via a separate experimental evaluation. More edges in the conflict graph means more gain from the use of carrier sensing.

7.4.2 Results

We first show the results of some hand created topologies. We placed four RFIDMotes very close to each other in a square. The conflict graph of this topology is shown in Figure 7.4A. This is a dense topology in which all readers collide with one another. We measured the accuracy and time per read. The results along with the 95% confidence interval are shown in Figures 7.5(a) and 7.5(b) respectively.



(a) Random configurations with locations of RFIDMotes shown. The scale is in cm.

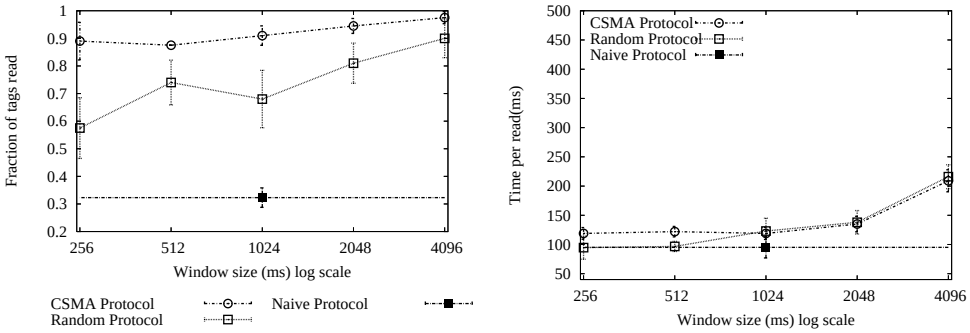


(b) Conflict graphs for the random configurations.

Figure 7.7: Random configurations and their conflict graphs.

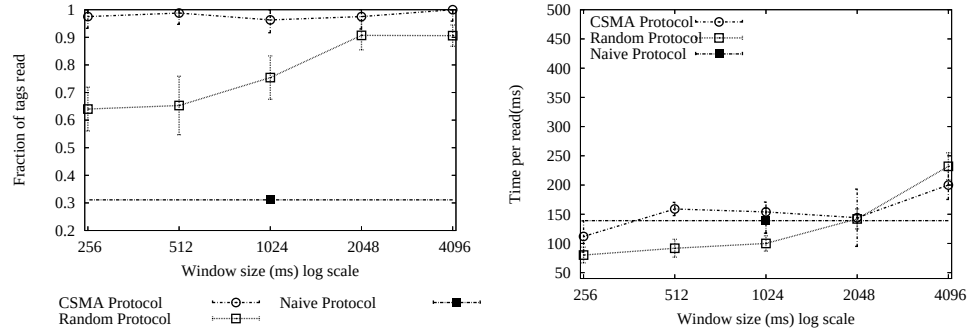
The horizontal axis shows the varying window size for random and CSMA protocols and the vertical axis shows the accuracy and time per read for each protocol. The naive protocol is shown as a straight line since window size is not a parameter here. The accuracy graph shows that the CSMA protocol achieves much better accuracy than the naive protocol. It is much better than the random protocol when the window size is small. The random protocol improves when the window size is increased, which is obviously due to the increase in the diversity of intervals chosen by each RFIDMote due to larger window size. This improvement comes at the cost of longer time taken to read each tag as seen in Figure 7.5(b).

We then placed four readers in a straight line. The conflict graph for this setup is shown in Figure 7.4B. We plot the accuracy and time consumed in reading each tag in Figures 7.6(a) and 7.6(b) respectively. This is a less dense topology compared to the grid before and only the adjacent readers can collide. This is the reason why the naive protocol is now able to read more tags than before, but the accuracy still remains poor compared to the random protocol. The CSMA protocol, still performs much better than the rest. Here, we notice that at smaller window sizes, the time taken per tag by CSMA is larger than the random protocol. The reason for this lies in the functioning of the STAC anti-collision protocol. In STAC, when a reader does not receive any tag response during a slot, it sends the “end slot” command earlier than the slot in which it receives a response. This means that the size of an “empty” slot, i.e, a slot in which the reader cannot successfully decode



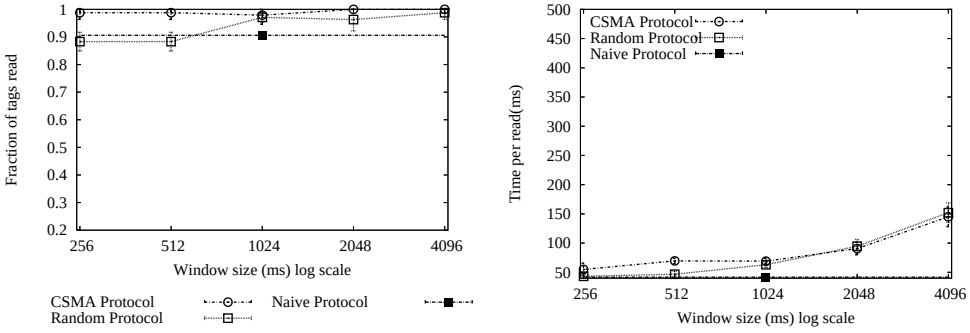
(a) Accuracy. (b) Time per read.

Figure 7.8: Accuracy and time per read vs. window size for the scenario in Figure 7.7(a)A.



(a) Accuracy.

Figure 7.9: Accuracy and time per read vs. window size for the scenario in Figure 7.7(a)B.



(a) Accuracy. (b) Time taken per read.

Figure 7.10: Accuracy and time per read vs. window size for the scenario in Figure 7.7(a)C.

a tag response, is smaller than a slot in which tag response is heard successfully. Thus, since the random and naive protocols are able to read fewer tags successfully, due to reader-tag or reader-reader collisions, they complete the reads faster than the CSMA protocol.

We will now show results for three random configurations. These configurations with the location of the RFIDMotes in the 2D plane and their conflict graphs are shown in Figures 7.7(a) and 7.7(b), respectively.

The performance results are shown in Figures 7–9 for these three configurations. Note that the configurations A and B have several conflicts and C has none. Thus, as expected CSMA provides much superior performance in configurations A and B and the naive protocol performs the worst. The protocols perform almost similarly in configuration C due to the absence of conflicts. But still CSMA has a slight advantage because it appears that occasional stray signals still cause a few collisions in the other two protocols.

Finally, note that 95% confidence interval for the CSMA has been usually much smaller than the random protocol. Thus, the performance of the CSMA protocol is more predictable.

7.5 Conclusion

In this chapter, we have developed a CSMA-based MAC protocol to address reader-reader and reader-tag collision problems in RFID networks. In order to realize this protocol in a working system, we have built the carrier sensing capability in a commercially available HF RFID reader OEM module and implemented the MAC protocol on the reader. We have created topologies that may represent actual deployment scenarios and ran some experiments to analyze the performance of the protocol. We have shown that the protocol is indeed able to achieve superior performance relative to other alternatives that do not rely on carrier sensing. While carrier sensing is an established technique for multiple access and is indeed expected to perform very well, our work demonstrates the feasibility of using carrier-sensing as an add-on at a low cost for tiny HF readers that otherwise have not been developed for multi-reader environments.

We are currently in the process of augmenting our testbed to a larger number of RFIDMotes and evaluating performance in more varied deployment scenarios.

Chapter 8

Future work and Conclusion

8.1 Future Work

The large socio-economic impact of wireless technologies and the increasing demand for mobile Internet access has motivated research and development in the wireless networking discipline. Most of the research until recently was based upon simulation of wireless network protocols. Due to the unreliable nature of the wireless links, it is hard to accurately model wireless communication in simulation. Thus, there is an increasing need to validate the research performed thus far on real platforms. Although some of the works presented in this thesis have been tested in Berkeley mote platforms [102], a more rigorous validation through implementation on the commercial wi-fi radios or gnuradio [3] will provide a better insight on the performance on the protocols. An implementation of CSMA based RFID readers on UHF RFID readers is also a topic future research. In terms of new protocol design, the SINR model presented in Chapter 5 may be used in a CSMA based MAC protocol as well to improve the network utilization. An initial design of this protocol is shown in Table 1 and Table 2. This design may be incorporated in the RTS/CTS exchange in 802.11 in place of the network allocation vector to reduce the harmful affect of the exposed terminal problems.

8.2 Conclusion

The main contribution of this thesis is efficient medium access protocols for wireless networks. We have designed medium access schemes for ad-hoc and mesh networks that improve the performance and robustness of existing MAC solutions. Our solutions provide efficient techniques to improve packet delivery ratio, decrease end to end delay in data transmission, provide fair medium access to multihop flows in the network and improve the throughput in the network. This thesis also con-

Input: SINR threshold β, Signal S, Interference I and Noise N levels at neighboring active receivers.
Output: Initiate transmission
for *Network node* s_j **do**
  foreach *Active neighboring receiver* r_i *receiving data with signal* S_i, *noise* N_i *and interference power* I_i **do**
    if ΔI *is the interference power between link* $s_j \rightarrow r_i$ *if* s_j *starts transmission* **then**
      if $\frac{S_i}{I_i+\Delta I+N_i} \geq \beta$ *at* r_i **then**
        choose random number n between 1 and cw; **if** $n \geq cw \times constant$ **then**
          Initiate transmission.
        end
      end
    end
  end
end

tributes a simple and distributed solution to collision problems in dense RFID networks.

We have designed anycast which is resilient to transient link losses in wireless ad-hoc networks. Anycast provides a significantly better performance in terms of packet delivery ratio as well as end to end delay. Anycast is able to provide this performance benefit by interacting with routing and physical layer so that it can use path diversity in the channel on various next hop links to improve probability of successful packet delivery.

We find an application of anycast in multichannel and directional antenna networks where, by exploiting path diversity, anycast is able to alleviate deafness problems without the use of additional hardware and network resources. The Anycast idea is not limited to any particular MAC scheme, so it may be implemented in conjunction with any other available MAC solution that uses busy tones or additional control packet exchange to provide further performance benefit. Further, we have applied anycast-like multiple control packet exchange mechanism, to improve MAC layer reliability for multicast data transmission. This approach can be easily incorporated in the IEEE 802.11 protocol to provide performance enhancement for multicast communication.

We demonstrate that the physical layer can provide useful information about channel conditions in terms of signal, noise and interference levels. This information and the SINR vs PRR model that we present in this thesis can be used to design

Input: Interference I and noise N levels at the node and signal level S from potential transmitter while there are other transmissions in the neighborhood.
Output: Agree to receive new transmission
for *Network node* s_i **do**
  if $\frac{S}{I+N} \geq \beta$ *at* r_i **then**
    choose random number n between 1 and cw; **if** $n \geq cw \times constant$ **then**
      Agree to receive new transmission.
    end
  end
end

accurate transmission schedules. Such a design will be quite useful in improving the reliability of the schedule as well as in fully utilizing the available network resources.

We have designed a max-min fair scheduling protocol for multihop flows in wireless mesh networks. We have also developed a first-in first-out medium access protocol that complements the scheduling protocol to provide a complete protocol suite to achieve max-min fair bandwidth distribution among contending flows in a multihop mesh network. This protocol suite consists of a rate computation protocol that computes fair rates for each single hop segment of a multihop flow, a back pressure protocol that extends this rate computation to multihop flows, a rate enforcement protocol that ensures that the computed rates are followed at each node in a distributed manner and a virtual time based MAC protocol that ensures that the same computed rate is followed at the MAC layer.

As our contribution to RFID networks, we have designed an efficient, distributed medium access protocol for dense RFID networks. We have developed a CSMA-based MAC protocol to address reader-reader and reader-tag collision problems that reduce the accuracy of reading tags in a dense reader environment. We have designed and built a carrier sensing circuit using a RFID tag as an antenna and a log amplifier chip to convert the signal detected by the antenna into received signal strength indicator. We have used this circuit in an RFID reader module and implemented a carrier sensing multiple access mechanism to alleviate the reader-reader and reader-tag collisions. While carrier sensing is an established technique for multiple access and is indeed expected to perform very well, our work demonstrates the feasibility of using carrier-sensing as an add-on at a low cost for tiny HF readers that otherwise have not been developed for multi-reader environments.

Bibliography

[1] Crossbow Technologies,Inc. http://www.xbow.com.

[2] Skyetek,Inc. http://www.skyetek.com.

[3] The GNU Radio Project. http://www.gnu.org/software/gnuradio/.

[4] TinyOS Community Forum http://www.tinyos.net.

[5] IBM Extends Moore's Law to the Third Dimension. http://www.physorg.com/news95575580.html, April 2007.

[6] Rice University Wireless Open-Access Research Platform (WARP). http://warp.rice.edu/.

[7] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz band, IEEE Standard 802.11a–1999, 1999.

[8] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 2: Higher-speed Physical Layer (PHY) extension in the 2.4 GHz band-Corrigendum1, IEEE Standard 802.11b–2001, 1999.

[9] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band, IEEE Standard 802.11b–2003, 2003.

[10] Analog Devices. *5 MHz-400 MHx 100dB High Precision Limiting-Logarithmic Amplifier AD8306*.

[11] Analog Devices. *AD8306 Evaluation Board EVAL-AD8306EB*.

[12] 3.56 MHz ISM band Class 1 Radio Frequency Identification Tag Interference Specification: Candidate Recommendation, Version 1.0.0. Technical Report MIT-AUTOID-WH-002, MIT Auto ID Center, 2003.

[13] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11–1997, 1997.

[14] EPC Generation 1 Tag Data Standards Versioni 1.1 Rev.1.27, May 2005.

[15] EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz Version 1.0.9 , January 2005.

[16] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specificationsAmendment : Enhancements for Higher Throughput, IEEE Sraft Standard 802.11n–2007, 2007.

[17] B. Sadeghi, V. Kanodia, A. Sabharwal and E. Knightly. Opportunistic Media Access for Multirate Ad Hoc Networks. In *Proceedings of the 8th International Conference on Mobile Computing and Networking (ACM MOBICOM'02)*, pages 24–35, September 2002.

[18] D. Bertsekas and R. Gallager. *Data Networks*, chapter 6. Prentice-Hall, 1992.

[19] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi. Using Channel State Dependent Packet Scheduling to Improve TCP Throughput over wireless LANs. *ACM/Baltzer Wireless Networks Journal*, pages 91–102, 1997.

[20] S. Biswas and R. Morris. Opportunistic Routing in Multi-hop Wireless Networks. *SIGCOMM Computer Communication Review*, 34(1):69–74, 2004.

[21] S. Biswas and R. Morris. ExOR: Opportunistic Multi-hop Routing for Wireless Networks. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 133–144, New York, NY, USA, 2005. ACM Press.

[22] G. Brar, D. M. Blough, and P. Santi. Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, pages 2–13, New York, NY, USA, 2006. ACM Press.

[23] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the 4th International Conference on Mobile Computing and Networking*, pages 85–97, October 1998.

[24] C. Chiang and M. Gerla. On-Demand Multicast in Mobile Wireless Networks. In *Proceedings of the Sixth International Conference on Network Protocols*, page 262, Washington, DC, USA, 1998. IEEE Computer Society.

[25] Charles Perkins and Elizabeth Royer. Multicast Using Ad Hoc On-Demand Distance Vector Routing. In *Proceedings of the 5th International Conference on Mobile Computing and Networking*, pages 207–218. ACM Press, August 1999.

[26] Charles Perkins and Elizabeth Royer and Samir R. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. RFC 3561, July 2003.

[27] R. R. Choudhury and N. H. Vaidya. MAC-layer Anycasting in Ad hoc Networks. *SIGCOMM Computer Communication Review*, 34(1):75–80, 2004.

[28] Chun-Yuah Chiu, E.H. Wu and Gen-Huey Chen. A Reliable and Efficient MAC layer Broadcast (Multicast) Protocol for Mobile Ad hoc Networks. In *Global Internet and Next Generation Networks, GlobeCom 2004*, pages 2802– 2807, December 2004.

[29] M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross-Layering in Mobile Ad Hoc Network Design. *Computer*, 37(2):48–51, 2004.

[30] S. M. Das, D. Koutsonikolas, Y. C. Hu, and D. Peroulis. Characterizing Multi-way Interference in Wireless Mesh Networks. In *Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pages 57–64, New York, NY, USA, 2006. ACM Press.

[31] E. Madruga and J. Garcia-Luna-Aceves. Multicasting Along Meshes in Ad-Hoc Networks. In *IEEE International Conference on Communications*, volume 1, pages 314–318, 1999.

[32] A. Ephermides. Ad hoc Networks: Not an Ad hoc Field Anymore. *Wireless Communications and Mobile Computing, Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):441–448, 2002.

[33] Ewerton L. Madruga and J. J. Garcia-Luna-Aceves. Scalable Multicasting: The Core-Assisted Mesh Protocol. *ACM/Baltzer Mobile Networks and Applications, Special Issue on Management of Mobility*, 6(2):151–165, apr 2001.

[34] K. Fall and K. Varadhan. *ns* Notes and Documentation. http://www-mash.cs.berkeley.edu/ns/, 1999.

[35] K. Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[36] K. Fische. RFID Tag Market to Approach $3 billion in 2009. http://www.instat.com.

[37] V. Fodale. 4Q06 WLAN Market Share Report. Technical Report IN0703417WL, IN-Stat, March 2007.

[38] J. Grönkvist and A. Hansson. Comparison between Graph-based and Interference-based STDMA scheduling. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad hoc Networking & Computing*, pages 255–258. ACM Press, 2001.

[39] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[40] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System Architecture Directions for Networked Sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[41] G. Holland, N. Vaidya, and P. Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 236–251. ACM Press, 2001.

[42] Hrishikesh Gossain and Nagesh Nandiraju and Kumar Anand and Dharma P. Agrawal. Supporting MAC Layer Multicast in IEEE 802.11 based MANETs: Issues and Solutions. In *29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, pages 172–179, 2004.

[43] X. L. Huang and B. Bensaou. On Max-min Fairness and Scheduling in Wireless Ad-hoc Networks: Analytical Framework and Implementation. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 221–231, New York, NY, USA, 2001. ACM Press.

[44] J. Xie, R. Talpade, T. McAuley and M. Liu. AMRoute: Ad Hoc Multicast Routing Protocol. *ACM Mobile Networks and Applications (MONET) Journal, 7(6):*, pages 429–439, Dec 2002.

[45] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia. Exploiting Medium Access Diversity in Rate Adaptive Wireless LANs. In *Proceedings of the 10Th Annual International Conference on Mobile Computing and Networking*, pages 345–359, New York, NY, USA, 2004. ACM Press.

[46] Jorjeta G. Jetcheva and David B. Johnson. Adaptive Demand-Driven Multicast Routing protocol (ADMR). Internet Draft, draft- jetcheva-manet-admr-00.txt, work in progress, June 2001.

[47] K. Tang and M. Gerla. MAC Layer Broadcast Support in 802.11 Wireless Networks. In *21st Century Military Communications Conference Proceedings Volume 1*, pages 544–548, Oct. 2000.

[48] K. Tang and M. Gerla. Random Access MAC for Efficient Broadcast Support in Ad hoc Networks. In *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE Volume 1*, pages 454 – 459, 2000.

[49] K. Tang and M. Gerla. MAC Reliable Broadcast in Ad Hoc Networks. In *Military Communications Conference, IEEE Communications for Network-Centric Operations: Creating the Information Force. IEEE Volume 2*, pages 1008 – 1013, Oct. 2001.

[50] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 200–209, New York, NY, USA, 2001. ACM Press.

[51] V. Kanodia, A. Sabharwal, B. Sadeghi, and E. Knightly. Ordered Packet Scheduling in Wireless Ad Hoc Networks: Mechanisms and Performance Analysis. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 58–70, New York, NY, USA, 2002. ACM Press.

[52] A. Kashyap, S. Ganguly, and S. R. Das. A Measurement-Based Approach to Modeling Link Capacity in 802.11-based Wireless Networks. In *To appear in ACM MOBICOM '07*, Montreal, CA, 2007. ACM Press.

[53] S. Keshav. *An Engineering Approach to Computer Networking : ATM Networks, the Internet, and the Telephone Network*, chapter 9. Addison-Wesley, 1998.

[54] Ki-Ho Lee and Dong-Ho Cho. A Multiple Access Collision Avoidance Protocol for Multicast Service in Mobile Ad Hoc Networks. In *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring Volume 3*, pages 1793 – 1797, April 2003.

[55] C. E. Koksal, H. Kassab, and H. Balakrishnan. An Analysis of Short-Term Fairness in Wireless Media Access Protocols (poster session). In *Proceedings of the 2000 ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems*, pages 118–119, New York, NY, USA, 2000. ACM Press.

[56] M. Krunz, A. Muqattash, and S.-J. Lee. Transmission Power Control in Wireless Ad Hoc Networks: Challenges, Solutions, and Open Issues. *IEEE Network*, 18(5):8–14, Sept/Oct 2004.

[57] P. Larsson. Selection Diversity Forwarding in a Multihop Packet Radio Network with Fading Channel and Capture. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5:79–282, October 2001.

[58] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. T. Capbell. INSIGNIA: an IP-based Quality of Service Framework for Mobile Ad Hoc Networks. *Parallel Distributed Computing*, 60(4):374–406, 2000.

[59] Z. Li, S. Nandi, and A. K. Gupta. Modeling the Short-term Unfairness of IEEE 802.11 in Presence of Hidden Terminals. *Performance Evaluation*, 63(4):441–462, 2006.

[60] H. Luo, S. Lu, and V. Bharghavan. A New Model for Packet Scheduling in Multihop Wireless Networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 76–86. ACM Press, 2000.

[61] M. Marina and S. R. Das. On Demand Multipath Distance Vector Routing in Ad Hoc Networks. In *Proceedings of the International Conference on Network Protocols*, pages 14–23, Dec. 2001.

[62] N. F. Maxemchuk. Dispersity Routing. In *Proceedings of the IEEE International Conference on Communications*, pages 41:10–41:13, 1975.

[63] S. McCanne and S. Floyd. Network simulator ns-2. In *http://www.isi.edu/nsnam/ns/*, 1997.

[64] M. L. Molle and L. Kleinrock. Virtual Time CSMA: Why Two Clocks Are Better than One. *IEEE Transactions on Communications*, COM-33 - 9:919–933, September 1985.

[65] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *In Proceedings of 25th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1 – 13, April 2006.

[66] Moteiv Corporation, San Framsisco, CA. *TMote-Sky: Ultra Low Power IEEE 802.15.4 Compliant Wireless Sensor Module*, November 2006.

[67] A. Nasipuri, R. Castaneda, and S. R. Das. Performance of Multipath Routing for On-Demand Protocols in Ad Hoc Networks. *ACM/Kluwer Mobile Networks (MONET) Journal*, 6(4):339–349, 2001.

[68] A. Nasipuri and S. R. Das. On-demand Multipath Routing for Mobile Ad Hoc Networks. In *Proceedings of the 8th. IEEE International Conference on Computer Communications and Networks* , pages 64–70, Boston, October 1999.

[69] R. Nelson and L. Kleinrock. Spatial-TDMA: A Collison-free Multihop Channel Access Protocol. *IEEE Transactions on Communication*, 33:934–944, Sept. 1985.

[70] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald. SoftMAC-flexible Wireless Research Platform. In *4th Workshop on Hot Topics in Networks (HotNets-IV)*, Nov 2005.

[71] A. Nogee. RFID Tags And Chips: Changing The World For Less Than The Price Of A Cup Of Coffee. Technical Report IN0402440WT, In-Stat, December 2004.

[72] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of Link Interference in Static Multi-hop Wireless Networks . In *Proceedings of Internet Measurement Conference (IMC)*, pages 305–310, 2005.

[73] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi. On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad hoc Networks. In *Proceedings of the 1st ACM International Symposium on Mobile Ad hoc Networking & Computing*, pages 3–10. IEEE Press, August 2000.

[74] C. Perkins, E. Royer, and S. R. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. http://www.ietf.org/internet-drafts/ draft-ietf-manet-aodv-07.txt, Nov 2000. IETF Internet Draft (work in progress).

[75] C. Perkins, E. Royer, and S. R. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. http://www.ietf.org/internet-drafts/ draft-ietf-manet-aodv-13.txt, February 2003. IETF Internet Draft (work in progress).

[76] R. Punnoose, P. Nikitin, and D. Stancil. Efficient Simulation of Ricean Fading within a Packet Simulator. In *Proceedings of IEEE Vehicular Technology Conference (VTC 2000)*, pages 764–767, 2000.

[77] M. Pursley, H. Russell, and J. Wysocarski. An Improved Forwarding Protocol for Updating Channel State Information in Mobile Fh Wireless Networks. In *IEEE Communications for Network-Centric Operations: Creating the Information Force*, volume 2, pages 967 – 971, October 2001.

[78] K. N. Ramachandran, E. M. Belding-Royer, K. C. Almeroth, and M. M. Buddhikot. Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. In *25th IEEE International Conference on Computer Communications*, pages 1–12, 2006.

[79] R. Ramanathan. On the Performance of Ad Hoc Networks with Beamforming Antennas. In *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 95–105, New York, NY, USA, 2001. ACM Press.

[80] T. Rappaport. *Wireless Communication: Principles and Practice*. Prentice-Hall, 2002.

[81] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based Models of Delivery and Interference in Static Wireless Networks. *SIGCOMM Computer Communication Review*, 36(4):51–62, 2006.

[82] I. Rhee, A. Warrier, M. Aia, and J. Min. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 90–101, New York, NY, USA, 2005. ACM Press.

[83] R.J. Punnoose, P.V. Nikitin and D.D. Stancil. Efficient Simulation of Ricean Fading within a Packet Simulator. In *52nd IEEE Vehicular Technology Conference*, volume 2, pages 764–767, September 2000.

[84] S Jain and S Das. Exploiting Path Diversity in the Link Layer in Wireless Ad hoc Networks. In *Proceedings of World of Wireless, Multimedia and Mobile networks, WoWMoM 2005*, pages 22–30, June 2005.

[85] S. K. S. Gupta, V. Shankar and S. Lalwani. Reliable Multicast MAC Protocol for Wireless LANs. In *IEEE International Conference on Communications Volume 1*, pages 93–97, 2003.

[86] S. E. Sarma, S. A. Weis, and D. W. Engels. RFID Systems and Security and Privacy Implications. In *Workshop on Cryptographic Hardware and Embedded Systems*, volume 2523 of *Lecture Notes in Computer Science*, pages 454–470, 2002.

[87] N. Shacham and P. King. Architectures and Performance of Multichannel Multihop Packet Radio Networks. *IEEE Journal on Selected Areas of Communication*, SAC-5(6):1013–1025, 1987.

[88] S. Shakkottai, T. Rappaport, and P. Karlsson. Cross-layer Design for Wireless Networks. *IEEE Communication Magazine*, 41(10):74 – 80, Oct 2003.

[89] W. Si and C. Li. RMAC: A Reliable Multicast MAC Protocol for Wireless Ad Hoc Networks. In *International Conference on Parallel Processing*, pages 494–501, 2004.

[90] D. Son, B. Krishnamachari, and J. Heidemann. Experimental Study of Concurrent Transmission in Wireless Sensor Networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pages 237–250, New York, NY, USA, 2006. ACM Press.

[91] K. Srinivasan and P. Levis. RSSI is Under Appreciated. In *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets 2006)*, 2006.

[92] A. P. Subramanian, H. Gupta, and S. R. Das. Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks. In *To Appear in 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks* , San Diego, California, USA, June 2007.

[93] M.-T. Sun, L. Huang, A. Arora, and T.-H. Lai. Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks. In *Proceedings of the 2002 International Conference on Parallel Processing*, pages 527–536, Washington, DC, USA, 2002. IEEE Computer Society.

[94] M. Takai, J. Martin, R. Bagrodia, and A. Ren. Directional Virtual Carrier Sensing for Directional Antennas in Mobile Ad-hoc Networks. In *ACM Mobihoc*, pages 39–46, June 2002.

[95] L. Tassiulas and S. Sarkar. Maxmin Fair Scheduling in Wireless Networks. In *21st Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2-21, pages 763–772, June 2002.

[96] Texas Instruments. *HF Antenna Design Notes Technical Application Report Literature Number 11-08-26-003*, 3 edition, September 2002.

[97] Texas Instruments. *CC2420 Radio Datasheet*, 1.3 edition, October 2005.

[98] F. A. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part-ii - The Hidden Terminal Problem in Carrier Sense Multiple-Access Models and the Busy-Tone Solution. *IEEE Transactions in Communications*, COM-23(12):1417–1433, 1975.

[99] N. H. Vaidya, P. Bahl, and S. Gupta. Distributed Fair Scheduling in a Wireless LAN. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 167–178, New York, NY, USA, 2000. ACM Press.

[100] J. Wang, H. Zhai, and Y. Fang. Opportunistic Packet Scheduling and Media Access Control for Wireless LANs and Multi-Hop Ad Hoc Networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, volume 2, pages 1234 – 1239, March 2004.

[101] W. Wang, X.-Y. Li, O. Frieder, Y. Wang, and W.-Z. Song. Efficient Interference-Aware TDMA Link Scheduling For Static Wireless Networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing And Networking*, pages 262–273, New York, NY, USA, 2006. ACM Press.

[102] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. *Computer*, 34(1):44–51, 2001.

[103] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting the Capture Effect for Collision Detection and Recovery. In *IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, pages 45– 52, May 2005.

[104] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *21st Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.*, volume 3, pages 1567– 1576, 2002.

[105] W. Ye, J. Heidemann, and D. Estrin. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks. *ACM/IEEE Transactions on Networking*, 12(3):493–506, June 2004.

[106] W. Zaumen and J. J. Garcia-Luna-Aceves. Shortest Multipath Routing Using Generalized Diffusing Computations. In *Proceesings of IEEE Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1408–1417, March 1998.

[107] H. Zhai, J. Wang, X. Chen, and Y. Fang. Medium Access Control in Mobile Ad Hoc Networks: Challenges and Solutions: Research Articles. *Wireless Communication and Mobile Computing Special Issue on Ad Hoc Wireless Networks*, 6(2):151–170, 2006.

[108] M. Zorzi and R. R. Rao. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance. *IEEE Transactions on Mobile Computing* , 2:337–348, Oct-Dec 2003.

[109] M. Zuniga and B. Krishnamachari. Analyzing the Transitional Region in Low Power Wireless Links. In *Proceedings of the 1st IEEE International Conference on Sensor and Ad Hoc Communication and Networks*, pages 517–526, 2004.