Stony Brook University

Medium Access and Security Protocols for Wireless Multihop Networks

A Dissertation Presented

by

Ritesh Maheshwari

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

May 2009

Stony Brook University

The Graduate School

Ritesh Maheshwari

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

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

Dr. Himanshu Gupta, Chairperson of Defense
Associate Professor, Department of Computer Science

Dr. Jie Gao, Committee Member
Assistant Professor, Department of Computer Science

Dr. Xin Wang, External Committee Member
Assistant Professor, Department of Electrical and Computer Engineering
Stony Brook University

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Medium Access and Security Protocols for Wireless Multihop Networks

by

Ritesh Maheshwari

Doctor of Philosophy

in

Computer Science

Stony Brook University

2009

Wireless networks are fraught with several significant performance and security issues for which no equivalent exists in the 'wired' world. Inefficiencies in medium access protocols and interference between concurrent transmissions cause serious capacity limitations. The broadcast nature of wireless medium makes it easier for malicious nodes to eavesdrop and launch various attacks that could be immune to cryptographic solutions. In our work, we study these issues and develop specific solutions. Our contributions are as follows.

First, we tackle the interference issue by using multiple channels in the medium access control (MAC) layer. This is hard because of specific synchronization/negotiation requirements and the so-called deafness problem. We address these issues via two new multichannel MAC protocols called the Extended Receiver Directed Transmission (xRDT) protocol and the Local Coordination-based Multichannel (LCM) protocol.

Second, we show that even when wireless spectrum is not channelized a priori, splitting a given bandwidth into multiple channels is helpful in high data-rate wireless networks. This helps counter the bandwidth independent overheads in the MAC

protocols. Using a carrier sense-based protocol as a case study, we develop and evaluate an Adaptive Multichannel (AMC) protocol that performs channelization adaptively with varying traffic. We also develop software radio-based prototypes to demonstrate the realism and performance potential in such protocols.

Third, we focus our attention on the basic nature of wireless interference itself, and study how interference models that are typically considered in scheduling literature actually behave in practice. We do this work using multiple physical layer technologies, viz., IEEE 802.15.4 and 802.11 PHY layers. We show that the SINR-based physical interference model provides significantly better accuracy than the rest. We also show that while literature has considered a 'thresholded' version of this model for scheduling studies, using the more realistic 'graded' version can extract more capacity from the network.

Finally, we develop and evaluate a novel algorithm for detecting and removing a particularly dangerous eavesdropping attack, called the wormhole attack. Our detection and removal algorithms use only connectivity information, are completely localized, and do not use any special hardware artifact or even location information.

Dedicated to my family and friends

Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank each and every person who has directly or indirectly helped me during the course of my PhD.

First and foremost, my utmost regards go to my dissertation advisor Prof. Samir R. Das. He has been a great advisor and mentor. His knowledge, wisdom and guidance as well as his constructive and honest criticisms have been instrumental in making this dissertation a reality. His open door policy for his students and the casual working atmosphere at the WINGS lab made working towards my PhD a fun learning experience. I would also like to thank Prof. Jie Gao and Prof. Himanshu Gupta for their help and support during the PhD. Finally, I would also like to thank Prof. Xin Wang for serving as a member of my PhD defense committee.

I would also like to thank all my former and current labmates at the WINGS lab – Anand Kashyap, Anand Prabhu, Pralhad, Vishnu, Bin, Zhongheng, Xianjin and Utpal – for making work fun. Many thanks to co-authors and labmates Shweta Jain and Jing Cao, for giving me company during those countless night-outs when we were measuring received signal strengths and PRRs. Very special thanks to Sandra for being a great friend during the past two years. She helped me remain sane and focused. Stony brook wouldn't be the same otherwise.

I would also like to acknowledge my global support group. My school friends – Samrat, Ankit, Rajul and Pooja – for their support in tough times and for always

keeping me grounded. My wingmates from IIT Kharagpur, especially the east coast group – Arul and Gary – for keeping the kgp spirit alive. And my former colleagues from Stony Brook – Yogesh and Chitra – for their ever-enthusiastic calls for parties in the city.

And last but not the least, I thank my family for their full support and my parents for their blessings. This dissertation is dedicated to them.

Chapter 1

Introduction

In recent years, with the popularity of IEEE 802.11-based Wireless LANs, interest in studying other uses of this technology has increased. Real deployments of wireless mesh networks as well as prototypical deployments of wireless sensor networks have contributed in extending this interest to multi-hop wireless networks. But while the WiFi technology has seen tremendous increase in adoption, the network capacity has not scaled proportionately. WiFi radios are now standard in all notebooks, netbooks and smart phones. WiFi connectivity is becoming available not only at homes and offices, but also in public places like coffee shops, airport lounges, book stores etc. The ubiquity of wireless networks, combined with the high-bandwidth demands of today's internet applications which serve multimedia content, has put a huge strain on the capacity of wireless networks. Also, wireless networks are much less secure than wired networks, as an adversary can record the transmissions from a distance. Wireless networks thus, even with their ubiquity, suffer with many problems.

While the problems are manyfold, few of these problems are more important from research perspective. First, the broadcast nature of the wireless physical layer

makes concurrent transmissions hard to achieve. This deteriorates the capacity of the multi hop networks. And second, the broadcast nature of wireless physical layer also makes it easy for an adversary to launch eavesdropping attacks. This can result in an adversary compromising the security of the network. For better multi-hop support, these two issues need to be resolved. In one-hop networks, on the other hand, wireless networks have been relatively more successful. But for futuristic high-speed wireless technologies, the same medium access control (MAC) protocols cannot be directly used. The popular carrier-sense based medium access (CSMA/CA) leads to low throughput even in less-loaded scenarios when used with very high-speed wireless technologies. In this dissertation, we look at these three problems and propose techniques to solve them.

A big hurdle in success of wireless multi hop networks has been the drop in capacity due to wireless interference. In 802.11-like CSMA protocol, interference from one transmission can affect many neighboring transmissions. One transmission can cause all links within a distance from the sender-receiver nodes to shut up. This affects the capacity of the network immensely since multi hop networks strive to achieve higher capacity by trying to schedule multiple parallel transmissions simultaneously. This is also called spatial reuse. With the growing popularity of bandwidth intensive applications like multimedia streaming and VoIP, most real multi hop networks will require a lot of capacity. Thus increasing the capacity of such wireless multi hop networks is an important problem to study.

Another downside of using a broadcast medium like the wireless physical layer is that a malicious user can easily launch eavesdropping attacks and cause immense damage to the network. Thwarting such attacks are very important to keep the network in operation and secure. In a wired network, an adversary can only listen to transmissions if she is part of the network. This usually involves physical access to the network in some manner. The network access can be limited by placing physical

barriers. But in wireless networks, the broadcast nature of the physical layer means that physical barriers are not enough to restrict access. Any node in the proximity of a wireless transmission can hear it. While cryptographic techniques have been employed at upper layers to encrypt important information, such techniques are hard to be used at the physical layer. Thus, at the physical layer, wireless transmissions are open to be heard by anyone within a range. Interestingly, with enough samples, the upper layer encryptions can also be broken by an adversary by just listening to the physical layer transmissions.

While just being physically close to the transmissions are enough for breaking security in one-hop wireless networks, this may not hold true for multi-hop networks. Multi-hop networks can span a much larger area which may not be logistically possible for adversaries to cover. Thus, many transmissions can go unheard. To get over this issue, adversaries again utilize the broadcast nature of the wireless physical layer to disseminate false information in the multi-hop network. These false information usually tune the network routes such that most transmissions go through the part of the network close to the adversaries. Thus, they can hear most transmissions. While dangerous, these *routing* attacks rely on the adversary's knowledge of the upper layer (MAC and network layer) protocols being used in the victim network. For such attacks, thus, proprietary protocols or MAC/network layer encryptions can be used. Recently, new attacks have been proposed which do not rely on such knowledge and can still cause a lot of damage to the operation of victim networks. Thus, this is an important research issue to explore.

Finally, since wireless radios cannot detect collisions, unlike in ethernet, wireless MAC protocols mostly avoid collisions by using collision avoidance mechanisms as used by CSMA/CA like protocols. These protocols (like IEEE 802.11) go through backoffs to randomize the order in which multiple nodes access the channel, thereby reducing chances of collisions. The backoffs comprise major

part of CSMA/CA MAC-layer overheads. They are also bandwidth-independent overheads. Thus, in high-speed networks, they can become comparable to packet transmission times. This leads to inefficiencies in performance, which can negate the performance boost obtained from high-speed physical layer technologies. It is, therefore, imperative to study and solve this issue of MAC-layer inefficiency in very high-speed wireless networks.

1.1 Research Issues

In this dissertation, we look at the above mentioned problems in the following contexts as follows.

1.1.1 Capacity Improvement in Single-hop Networks

High data rate wireless networks (1 Gbps and up) are in the horizon and several standards are in the works. However, the task of designing multiple access protocols for such networks is fraught with new challenges as the bandwidth independent overheads dominate. We show that even when such overheads are kept at a minimum, the performance of multiple access protocols can be very poor. However, performance can be improved significantly by splitting the given bandwidth into multiple channels and running the multiple access protocol independently on these channels. Taking an 802.11-like CSMA/CA protocol as an example we show via a modeling exercise how such channelization can improve performance and why it needs to be adaptive to traffic demand. We develop an Adaptive Multichannel (AMC) protocol and study its performance via simulations. Finally, we develop a 'scaled down' prototype implementation using the USRP/GNURadio platform to demonstrate that adaptive channelization can be practical using appropriate programmable radio hardware and has tremendous performance potential. Taking

our modeling, simulation and experimental results together, our work shows that a throughput gain of a factor of 2 is not unrealistic.

1.1.2 Capacity Improvement in Multi-hop Networks

One way to improve the capacity of multi hop wireless networks is through using different channels for conflicting links. 802.11 defines many non-interfering channels in its operating frequency band. Transmissions carried out in non-interfering bands can go on in parallel even if the sender-receiver pairs are close physically. For multihop networks, protocols using multiple channels for simultaneous transmissions have an immense potential to improve the capacity.

As mentioned earlier, IEEE 802.11 works well in one hop scenarios, but its CSMA approach is detrimental to multi hop network performance. This is because the inherent assumption about wireless interference in such CSMA protocols are usually too conservative. A well studied problem called exposed node problem is a good example to illustrate this point. Here, two senders cannot simultaneously transmit because of the way CSMA works, but it is indeed possible for them to transmit and the respective receivers to receive the packets successfully. To avoid such issues, time slotted multiple access protocols have been suggested. In such TDMA systems, a more careful scheduling of transmissions can give higher capacity. But here too, non-realistic and conservative models have been in vogue. Interference is usually assumed pairwise, such that each pair of link is denoted interfering or not. This is not realistic as a transmission's success depends on all the active links and not just one. Also, interference is usually assumed binary – that is, interference exists or not. In reality, interference is probablistic in nature and depends on the ratio of signal power received at receiver and the amount of interference and noise experienced at reciever at that time. A family of interference models called SINR-based models capture this realistic behavior. Using such models are a

recent research interest area which also has potential to improve the wireless multi hop network capacity.

1.1.3 Securing Wireless Multi-hop Networks

While the interference causes capacity drop in wireless networks, its broadcast nature also facilitates malicious nodes to easily launch eavesdropping attacks. An amazingly simple but very effective eavesdropping attack called the *wormhole attack* has been recently proposed and studied in literature. While routing attacks rely on knowledge of victim network's MAC or routing layer to launch the attack, wormhole attacks do not need such knowledge and can be employed even in presence of proprietary protocols or encryptions techniques. Wormhole attack utilizes the fact that wireless transmissions can always be heard at the physical layer. They capture these transmissions at one point in the network and replay them at another point and vice versa using, e.g., a long ethernet cable. This creates a seeming shortcut in the network which causes route updates such that most traffic passes through the malicious nodes. Thus the adversary can capture most packets. Being easy to deploy but very harmful, wormhole attack needs to be studied closely.

1.2 Contributions

In this work, we make five main contributions to solve the above mentioned three issues. They are as follows.

Novel Multi channel MAC protocols We propose two new MAC protocols – xRDT and LCM MAC – to enable multiple access in infrastructure-less wireless networks. Our first protocol, Extended Receiver Directed Transmission

protocol (xRDT) is based on a well-known single interface multichannel solution called RDT. xRDT solves the problems faced by RDT by using an additional busy tone interface and some other protocol operations. We also develop a novel single interface solution called Local Coordination-based Multichannel MAC (LCM MAC). LCM MAC performs coordinated channel negotiations and channel switching to provide good multichannel support, without the help of any time synchronization. By providing simulation results, we demonstrate the effectiveness of our protocols over two other well-known multichannel protocols – MMAC and DCA – and single channel 802.11

Multichannel MAC for High Speed Networks We make three contributions here. First, we show via analytical modeling that single channel MAC protocol is very inefficient in high data rate networks and channelization can provide the necessary improvement. Second, we develop an adaptive channelization protocol and show via simulations that just channelization is not enough for better performance; channelization also needs to be adapted with varying traffic conditions. Finally, via a 'scaled down' prototype implementation on GNU Radio/USRP platform, we emulate the operation of a high-speed network and show such adaptive channelization can indeed be realized in practice.

Experimental Comparison of Wireless Interference Models We perform extensive modeling and experimentation on a TelosB motes testbed using low power wireless links to compare a suite of interference models for their modeling accuracy. The suite consists of the physical interference model, as well as several common models typically considered in literature for scheduling studies, such as hop-based, range based, distance ratio-based, etc. We first

empirically build and validate the physical interference model via a packet reception rate vs. SINR relationship using a measurement driven method. We then similarly instantiate the other models, and compare their modeling accuracies on the testbed using transmission scheduling experiments. The experiments are very comprehensive, covering 13,000 sets of links for evaluation on our 20 node testbed. We observe that the physical interference model is significantly more accurate than the other models considered for evaluation. We then look closely into the physical interference model itself, and consider its two incarnations – 'thresholded' (overly conservative, but typically considered in literature) and 'graded' (more realistic). We show via solving the one shot scheduling problem, that the graded version can improve 'expected throughput' over the thresholded version by scheduling imperfect links.

Physical Interference Modeling on Commodity WiFi Radios In this work, we use commodity WiFi hardware (specifically, 802.11a) for a comprehensive study on interference modeling for transmission scheduling on a mesh setup. We focus on the well-known physical interference model for its realism. We empirically build the physical interference model via a packet reception rate vs. SINR relationship using a measurement driven method. We propose use of the "graded" version of the model where feasibility of a link is probabilistic, as opposed to using the more traditional "thresholded" version, where feasibility is binary. We show experimentally that the graded model is significantly more accurate (80 percentile error 0.2 vs. 0.55 for thresholded model). However, the graded model has never been considered in algorithmic studies on transmission scheduling. Carrying on further, we develop transmission scheduling experiments using greedy scheduling algorithms for the evacuation model for both interference models. We also develop similar experiments for optimal scheduling performance for the simplified one-shot

scheduling. The scheduling experiments demonstrate clearly superior performance for the graded model, often by a factor of 2.

Detecting Wormhole Attacks We propose a novel algorithm for detecting and removing *wormhole attacks* in wireless multi-hop networks. The algorithm uses only connectivity information to look for forbidden substructures in the connectivity graph. The proposed approach is completely localized and, unlike many techniques proposed in literature, does not use any special hardware artifact or location information, making the technique universally applicable. The algorithm is independent of wireless communication models. However, knowledge of the model and node distribution helps estimate a parameter used in the algorithm. We also extend the detection algorithm for removing wormhole link from the network. We present simulation results for three different communication models and two different node distributions, and show that the detection algorithm is able to detect wormhole attacks with a 100% detection and 0% false alarm probabilities whenever the network is connected with high probability. Even for very low density networks where chances of disconnection is very high, the detection probability remains very high. Simulation results for removal show a 100% removal probability with very less penalty even for highly random scenarios.

In the end, we note that the research issues explored in this dissertation emanate from problems due to the broadcast nature of the wireless physical layer. Some of our solutions, interestingly, also exploit the same broadcast nature for improving the state of the art.

1.3 Outline

The rest of this report is organized as follows. Use of multiple channels as well as other form of diversities in wireless multi hop MAC protocols is discussed in chapter 2. Chapter 3 presents our work on comparing various interference models used in the literature while our study of SINR-based interference models for commodity wifi radios is presented in the chapter 4. In chapter 5, problems with using CSMA/CA MAC in high-speed wireless networks and our proposed solutions are discussed. In chapter 6, our novel algorithm for detecting and removing wormhole attacks is presented and we conclude this dissertation in chapter 7.

Chapter 2

Multi-Channel Protocols for Wireless Networks

2.1 Introduction

Use of multiple frequency channels offers tremendous potential to improve the capacity of a wireless network. This potential has been recognized in existing standards, such as the IEEE 802.11 [36], that can operate on multiple orthogonal channels. Using multiple frequency channels enables conflict-free transmissions in a physical neighborhood so long as pairs of transmitters and receivers can tune to different non-conflicting channels. However, the problem of efficient use of multiple channels to utilize the raw additional capacity is non-trivial for wireless ad hoc or mesh networks.

The research community has been addressing the multichannel question using two very different approaches. The first is a *static* approach based on *topology control*. Here, multiple radio interfaces are used on a node and the emphasis is on assigning frequency channels to these radio interfaces such that two nodes that

communicate directly in the resulting topology have at least one channel in common. As this approach is necessarily static, the approach is often graph-theoretic and is based on models of interference or protocol behavior, and assumptions on average traffic. The papers in literature using this approach pose the problem as essentially an optimization problem [11, 88, 89], [69], [14], [59].

The other approach is more *dynamic*. It relies on the capability of the radio interface to switch channels on the fly with negligible delay. Here, multiple channels can be utilized even with a single radio interface. Generally speaking, this approach can provide a significant performance benefit over a purely static approach (on a per-interface basis) as it can potentially utilize instantaneous traffic or interference information.

Our goal in this work is to develop new MAC protocols for ad hoc networks that use such dynamic approaches. We develop two new MAC protocols. The first protocol, called *extended receiver directed transmission* (xRDT), uses one packet interface and one busy tone interface. Note that we differentiate between a *packet interface* and a *tone interface* to contrast our approach with similar approaches that use a separate control channel and thus two packet interfaces (see, for example, the DCA protocol [97]). Tone interfaces are much simpler to implement than packet interfaces. The second protocol, called *local coordination-based multichannel* (LCM) MAC, only uses a single packet interface. We show, via extensive ns-2 simulations, that these two protocols significantly outperform similar protocols that appeared in literature recently.

The rest of the chapter is organized as follows. In the following section similar multichannel approaches in literature are reviewed to provide a context for our work. In Section III the simple receiver directed scheme is described and its problems analyzed. In Section IV, protocol operations are developed to address these

problems. This constitutes the xRDT protocol. In Section V, the LCM MAC protocol is developed. In Section VI, ns2 simulation results are presented with realistic traffic scenarios and network models. We finally conclude the work in Section VII.

2.2 Background and Related Works

There have been several works on developing new MAC protocols that use multiple channels. We review them in this section to provide a context for our work.

2.2.1 Dynamic Approaches

Approaches based on frequent channel switching are reviewed in this section.

In [17] the authors proposed Slotted Seeded Channel Hopping (SSCH), a link-layer protocol that uses unmodified 802.11 MAC layer. Each node in SSCH switches channels at slot-boundaries in a pseudo-random sequence such that channels for neighboring nodes overlap in time periodically. SSCH, requires time synchronization to implement slotting. Also, to be effective, SSCH must adapt its schedule continuously so that frequently communicating nodes overlap in channels often.

In [105] the authors proposed the Multichannel MAC (MMAC) protocol which is loosely based on the 802.11 power-saving mechanism [36]. MMAC considers time slotted into *beacon periods* of $100ms$ which are again sub-divided into *ATIM window* of $20ms$ and data window of $80ms$.[1] Nodes tune to a common default channel during the ATIM window and perform negotiations for data transmission and channel selection for the data window. Senders pick receivers to negotiate with based on the number of packets for each receiver in their interface queue. Nodes switch to their respective selected channels when data window starts. In some sense,

[1] These possibly could be adapted; but no such protocol exists.

MMAC partitions the network into N sets of nodes during the data phase, where N is the number of channels. The ATIM window serves as a phase for a node to make the best possible decision on which set it should join.

Both MMAC and SSCH require network-wide time synchronization to work. They also constraint the nodes to switch channels only at slot boundaries. Thus, they both cannot utilize channel diversity to the maximum extent as they need to stay in a specific channel for fixed periods of time.

One of the earliest works to utilize dynamic channel switching was the Receiver Directed Transmission (RDT) protocol [102]. In RDT, each node selects a quiescent channel which it always listens to when idle. Any transmitter must switch to the receiver's quiescent channel to transmit. We describe RDT in detail in the next section as one of our approaches (xRDT) is based on the RDT paradigm.

The Dynamic Channel Assignment (DCA) protocol [97], unlike the solutions described above, utilizes two packet interfaces. It employs one of the interfaces as a control interface, which is always tuned to a control channel (common to all nodes). This interface allows senders to do a three-way negotiation with the receivers to decide on a channel to be used for data transmission. The selected channel is then used by the other interface to transmit/receive data packet. As one interface is dedicated to the control channel – every node in DCA is informed of channel usage in its neighborhood and thus can make better decisions while negotiating.

However, DCA uses an extra resource – the control interface. Additionally, the right bandwidth for the control channel is traffic dependent. Wide control channel may result in wastage of precious bandwidth, while narrow control channel may become a bottleneck, resulting in wastage of data channel bandwidth.

We also note here that similar protocols were developed in the past, that split one single channel into multiple subchannels and use only a subchannel for communication [51]. However, the issue, there, was to reduce the overhead of contention.

2.2.2 Static, Multi-radio Approaches

There has been a body of work recently that look at the multi-channel protocols from a different angle. Here, the interest is in using legacy 802.11 protocol with COTS radios – that cannot perform fast channel switching – in multichannel environments. The basic idea is to use multiple radio interfaces assigned (statically, or dynamically – but at a slow time scale) to different channels on each node so that many channels can be used concurrently. However, the channel assignment must be done in a way that interference is minimized. There are several papers in literature that take this broad approach [89], [88], [11], [69], [14], [59]. While such solutions are amenable to implementation with legacy hardware, the static nature of the solutions limit their effectiveness.

2.2.3 Other Related Work

Several other works are also worthy of mention here like the Hop Reservation Multiple Access (HRMA) [110] and Receiver-initiated Channel-hopping with Dual Polling [114] which have been proposed for use with frequency hopping spread spectrum (FHSS) wireless cards.

Also, [62] where multiple radios are used such that some of them have static channel assignment and the rest do dynamic channel switching. Asymptotic capacity models for multichannel networks with multiple interfaces per node were developed in [61].

2.3 Receiver Directed Transmission and Performance Issues

In this section, we will develop an understanding of the issues involved in multichannel operations by revisiting the receiver directed transmission (RDT) approach [102]. RDT uses a clever approach which requires neither a separate control channel nor any form of time synchronized channel access. One of our approaches is based on RDT. However, a straightforward use of RDT with 802.11 MAC results poses some serious problems. Our goal in this section is to describe these issues. We start with describing the RDT approach in detail first.

In RDT, every node is assumed to be have a single interface. Every node also selects (or is assigned) a "well-known" *quiescent* channel for itself. This is the channel the node always listens to when idle. To transmit a packet, a transmitter switches its interface to the quiescent channel of the intended receiver and then transmits using a regular single channel MAC protocol such as 802.11 (with RTS/CTS etc). Following a successful transfer, the sender switches its interface back to its quiescent channel. The protocol assumes that the quiescent channel selection and distribution of this information to the neighboring nodes are done via a separate mechanism. This simplifies the approach greatly in the sense that it is no longer needed that a communicating pair of nodes negotiate a channel beforehand. The receiving channel is always known to the transmitter.

2.3.1 Multichannel Hidden Terminal Problem

The above scheme presents a new form of the well-known *hidden terminal problem* [112]. Similar problems were also observed in [105] in a slightly different context. When a transmitter A, for data transmission, switches its interface from its

quiescent channel p to the receiver B's quiescent channel q, it has no prior information about q's state (i.e., currently ongoing transmissions). For example, there could be another node C in the neighborhood in the same channel q that might be receiving data from a node D that is hidden from A. In case A transmits an RTS for B, it will result in a collision at C.

Note that 802.11 [36] solves the single channel equivalent of this hidden terminal problem by using a virtual carrier sensing mechanism. Senders send RTS and receivers send CTS before a DATA/ACK exchange. Any neighboring node hearing RTS or CTS, will set its NAV (*network allocation vector*) until the end time of ACK transmission. Any node with NAV set remains silent. This ensures that collisions does not happen. For example, A would not have sent RTS to B in the above example as it would have set its NAV for D to C transmission.

However, the virtual carrier sensing mechanism is not sufficient to prevent collisions in multichannel environment where only a single interface is used. This is because the control packets now could be sent in different channels and one interface can only work on one channel at a time. Also note that a simple solution to this problem would be for A to wait for the longest packet transmission time before attempting transmission after switching channel. But this is clearly inefficient.

2.3.2 Deafness Problem

A second problem, called *deafness*, arises because an intended receiver may currently be in transmit mode, transmitting in the quiescent channel of a third node. This will cause the transmission attempt to fail as the receiver will not respond; it is deaf as it is not tuned to its quiescent channel at this time. In 802.11, this means that the transmission will be retried – after a backoff, that increases exponentially after multiple such failures, suspecting congestion. This wastes network resources and causes unfairness. Also, it is indeed possible that the receiver comes back to its

quiescent channel when its current packet transmission is over; however, the transmitter remains in the backoff unaware of this event, waiting for the backoff timer to expire. By the time the latter indeed attempts the next retry, the receiver could have switched to another channel for transmission.

Our simulations (not described here due to lack of space) indicate that such situations occur frequently enough at high load causing throughput to decrease. Note that the packet could be dropped after multiple retries when the retry limit has been exceeded. This usually will have terrible consequences with upper layer routing protocols, which might suspect a link breakage and start new route computations, usually a high overhead operation.

Similar deafness problems have been noted before in the context of directional antennas [31]. Similar situation happens in the basic 802.11 protocol as well, but one *additional* hop away. This situation has been referred to as *information asymmetry* in literature [115] and is one cause of fairness problems in 802.11.

We develop two protocols to eradicate the effects of the problem mentioned in this section. The first protocol, xRDT, tries to *solve* the problems using extra resources but using the same framework as the RDT approach. The second protocol, LCM-MAC, instead *prevents* the above mentioned problems from occurring, by using a novel technique of channel negotiation based on local coordinations. We describe these approaches and their relative merits in the following sections.

2.4 xRDT: Extended Receiver Directed Transmission Scheme

xRDT adds two mechanisms to RDT to address the multichannel hidden terminal and deafness problems. They are described below. For the purpose of explaining this protocol assume that the quiescent channel advertisement is done by an separate mechanism. Thus, every transmitter knows the receiver's quiescent channel. The quiescent channel assignment process is explained separately.

2.4.1 Addressing Multichannel Hidden Terminal

One solution of the hidden terminal problem is to implement a "channel memory" that helps propagate the channel state to a potential transmitter *at all times*. An easy and well-known way to implement "channel memory" is by using *busy tones* [112] [35] [119]. Busy tones are single frequency tones used for signaling. The advantage of using busy tones relative to using a separate control channel is that the issue of determining the right bandwidth to allocate for the control channel does not arise. Also, the channel gain for both the data channel and the busy tone channel (or, the control channel for that matter) must be the same for the techniques to operate correctly. This means frequencies must be close – closer than the *coherence bandwidth* of the data channel. This is relatively easier to do for a single frequency tone. In addition, hardware requirement is simpler as only a tone interface is needed instead of a packet interface.

We assume that there is a different tone channel b_c for each data channel c. However, one single tone interface is sufficient. A receiver, when receiving a data packet on channel c turns on the tone in corresponding busy tone channel b_c. This enables a potential transmitter that has just come to channel c to learn about any

receiver in the neighborhood by sensing on the busy tone channel b_c. If the busy tone channel is indeed found busy, a transmitter would defer its transmission on the data channel. This deferment is designed exactly similar to the collision avoidance mechanism in 802.11 [36] that uses a variation of the p-persistent protocol [57]. For brevity, this mechanism is not discussed here.

Note that use of the busy tone prevents any collision of data packets.

2.4.2 Addressing Deafness

There are simple solutions to deafness; but they all require additional resources. For example, note that deafness arises because a radio interface is half-duplex. So, in transmit mode it is deaf to any reception. So, if two interfaces are used, one for transmission and the other for reception [62], the problem is solved trivially. In this work, we take an approach that simply softens the impact of deafness instead of completely eliminating it.

Recall from Section III that a receiver might return back to its quiescent channel while the transmitter is still in backoff. A notification that a potential receiver is available to receive data can preempt this backoff and ready the transmitter to transmit immediately following. One way to achieve this would be for the "deaf" nodes to broadcast a *Data Transmission Complete* or DTC notification message in its own quiescent channel. This will ensure that all potential transmitters (who may be in backoff) come to know of the receiver's availability. They now can break out from backoff and start the transmission process. A contention resolution is necessary to resolve between multiple such transmitters. This can be done simply by following the contention resolution scheme in 802.11 [36].

Notice that DTC does not prevent deafness from occurring – transmitters will still send RTS to deaf receivers – but it will alleviate it by capitalizing on the fact that the deaf node will return to its quiescent channel before switching to another

channel for transmission. This small window of opportunity is utilized by making the deaf receiver send the DTC to "wake up" the backed off transmitters.

2.4.3 Complete Protocol Description

Briefly, the Extended RDT (xRDT) protocol is the same as RDT, except that now (i) there is a receiver busy tone on the appropriate busy tone channel, and (ii) a DTC message after the data transfer is over in the quiescent channel. The details follow.

2.4.3.1 Start of Transfer

In xRDT, every node listens to its quiescent channel when idle. A transmitter A switches channel to the receiver B's quiescent channel q. Then it senses carrier on both q and the busy tone channel b_q using the two interfaces. If any channel is found busy, it uses a contention mechanism similar to 802.11, which we do not describe for brevity. When the channels are found idle (after the appropriate contention resolution, if any), A sends RTS to B on channel q. B after receiving RTS turns on busy tone for b_q. The busy tone works as an implicit acknowledgment for the RTS. On hearing the busy tone, A transmits DATA to B. When DATA transmission is complete, B turns off the busy tone. Then again, after an appropriate interframe spacing, busy tone is turned on briefly as an acknowledgment. This stays for a normal ACK packet duration. The setting of the interframe spacing guarantees that no other transmission in the vicinity can start in the interim. Absence of this busy tone-based acknowledgment signals the transmitter that retransmission is necessary. The retransmission is tried after a backoff. This backoff again is similar to 802.11.

2.4.3.2 End of Transfer

After the transfer is complete, A switches back to its own quiescent channel (say, p). After proper interframe spacing, it broadcasts a DTC message in this channel if the channel is idle (both p and b_p). Channel sensing eliminates the possibility of DTC collision. If channel is sensed busy DTC transmission is deferred till channel becomes idle. If DTC is indeed sent, any other node C waiting for A in channel p cancels its current backoff timer, erases all backoff and contention window related states, and acts as if it is attempting to transmit a fresh packet.

2.4.3.3 Next Transfer

If A also has another packet to transmit, it prepares to transmit that packet right after transmitting the DTC for the previous transmission. This means again – as in 802.11 – setting the contention window to the minimum value and picking a random backoff time. A transmits – after switching to the receiver's quiescent channel – if its backoff expires earlier than C's. Else, C transmits to receiver A. If this is the case, when C to A transfer is complete, A again attempts to transmit its packet after completing its *remaining* backoff time.

2.4.4 Selection of Quiescent Channel

A good quiescent channel selection for all nodes is required for maximizing parallelism in the network. When the traffic profile generated by each node is similar, selection of a good quiescent channel become equivalent to finding a solution to the max k-cut problem in the G^2 connectivity graph of the network (each 2-hop neighbors have an edge in G^2). Any of the approximation algorithms [32] existing in the literature can be used to do the channel assignment in that case.

In case, the traffic profile changes dynamically, we propose a periodic channel

selection mechanism using channel load as a criterion. A node measures the load on all channels by snooping on other channels during its idle time. Traffic directed towards itself is discounted when calculating load on its current quiescent channel, q. If load on the least loaded channel, l, is lower than the load on q, l is chosen as the new quiescent channel. To avoid oscillations, the difference is mandated to be above a threshold for any change to take place.

2.5 Local Coordination-based Multichannel (LCM) MAC

The receiver directed approach described before suffers from a limitation. It requires an additional busy tone interface which is hard to engineer because of the requirement that channel gains for the busy tone channels and corresponding data channels need to be the same. Eliminating the busy tones is not an option as they serve as a mechanism to solve the multichannel hidden terminal problem.

In this section, we develop an alternative approach called LCM-MAC where busy tones are not used and each node is required to have only one interface. The neighboring nodes go through local coordinations to generate *transmission schedules*. A transmission schedule consists of a period when only control packets are transmitted (also called *control window*) followed by a period when only data packets are transmitted (a *data window*). LCM, essentially, is a two stage protocol, such that:

- All control packets are transmitted in the same channel during the control window. All nodes in a neighborhood are tuned to this same channel at this time.
- All data packets are transmitted concurrently in different channels in the data

window.

The first stage helps ensure that nodes become aware of transmissions in the neighborhood (this avoids the Multichannel Hidden Terminal problem as well as the Deafness problem). Data packets are transmitted concurrently at different channels to exploit parallelism.

The common channel used in the control window is called the *default* channel. Unlike the quiescent channel in xRDT, the default channel in this case is common to all nodes. The default channel is used as a control channel during the control window and as a data channel during the data window.

The key idea in LCM protocol is to setup transmission schedules without the use of any time synchronization. Senders use a contention resolution mechanism similar to 802.11 to gain access to the default channel during the control window. A sender then negotiates a channel to be used during the data window with the intended receiver. Once the negotiation is over, it releases the channel to let other senders contend for its access.

When control window gets over, the communicating nodes switch to their respective selected channels and exchange DATA and ACK. This constitutes the data window. After data window is complete, all these nodes switch back to the default channel for another round of negotiations. The time line showed in Figure 2.1 illustrates the scenario, when all the nodes are in the same radio neighborhood.

The protocol is similar in detail to the MACA-P [10] protocol and the POW-MAC [76] protocol for transmit power control. LCM also has some similarities with the MMAC [105] protocol in channel negotiations. However, MMAC follows a rigid schedule and the negotiations are for long term. Thus, its benefit is limited by traffic conditions. MMAC also requires tight time synchronization for the protocol to work whereas LCM has no such requirements.

We now describe the protocol operation in detail.

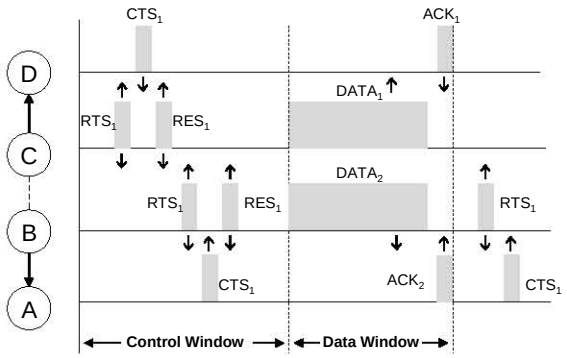


Figure 2.1. A example trace showing the working of LCM MAC in a simple three hop network with 2 channels. The subscript for each packet indicates the channel in which the packet is sent.

2.5.1 Detailed Protocol Operation

In this section, we try to detail the methodology used for setting up a schedule without global synchronization. We also talk about how negotiations are performed during the control window and how channels are selected to avoid conflicts during the data window. Later, we talk about the limitations of the protocol and how to overcome them.

2.5.1.1 Control Window Operation

Any node unaware of a control/data window schedule can propose a schedule. Otherwise, it follows a schedule it already knows. When a node has a packet to send and is unaware of any schedule, it transmits an RTS (as in 802.11) in the default channel with a proposed schedule. This node is called a *master* node. The schedule can be defined by two additional fields in the RTS packet: (i) time left for data window to start (control window duration) and (ii) the data window duration. A RTS packet also contains a list of free channels at the sender for transmission during the data window. We use a concept of *Multichannel NAV* for this purpose. This is the

same as NAV used in 802.11, except that now NAV is a vector with one element for each channel. If the NAV for a particular channel is set, then that channel is deemed busy otherwise, it is free.

If n is the number of channels, then only n negotiations are possible in a neighborhood in the control window - as it will exhaust the list of channels for data communication. Thus, if T_{neg} is the time needed for a successful negotiation (explained next), then control window duration is set to n times T_{neg}. But, in case the master node has heard only k ($<n$) negotiations in the last control window, it sets the control window size to (k+1) times T_{neg}. The data window size is set to the time needed for DATA-ACK exchange with the proper interframe spacings.

On receiving RTS, the receiver can accept the schedule by replying with a CTS. The CTS also contains a channel id selected from the channel list in RTS. This selected channel is one of the free channels at both sender and receiver's positions. The CTS also contains the schedule information.

When sender receives a CTS, it transmits another packet called RES (for *reserve*) containing the schedule and the selected channel id. RES is needed to allow all neighbors of the transmitter to be aware of the channel to be used for communication. Any node hearing a CTS or RES packet will set its Multichannel NAV for the channel whose id is included in the packet for an appropriate duration of time (end of data window). All such nodes also note the schedule mentioned in the packet and follow that schedule unless they have already been following another schedule. Thus the schedule is propagated to all the nodes in the one-hop neighborhood of the sender and the receiver.

In case, the receiver could not find a common free channel from the channel list in the RTS, it replies with a channel id value of -1 in the CTS to signal the sender to retry in the next schedule. The sender will not respond with RES to such a CTS message. Any neighboring node hearing such a CTS will ignore it.

2.5.1.2 Data Window Operation

After a successful RTS and CTS exchange, the transmitter-receiver pair has now agreed on the channel to be used and all potential interferers have set their NAVs for this channel to allow this transmission to proceed without conflict. Other nodes who overheard the CTS/RES packets are now free to start their own negotiations (using RTS/CTS/RES exchange) as long as they can finish the negotiation within the control window and their data transmission takes time less than or equal to the data window length mentioned in the schedule. At the end of control window the transmitter-receiver pairs switch to their selected channels. This starts the data window. DATA and ACK are transmitted in the selected channel. At the end of the data window, the transmitter-receiver pairs return to the default channel implicitly signaling the start of another control window.

The nodes who heard a schedule in the control window but are not participating in data transmissions, remain in the default channel during the data window. They are not allowed to communicate during this time.

2.5.1.3 Similarities With 802.11

The other details of the protocol are similar to 802.11. For example, it follows the identical interframe spacings and collision avoidance strategies by using physical carrier sense and backoff before transmitting an RTS. If there is no CTS in response to an RTS, RTS is retransmitted with an increased backoff exactly similar to 802.11. This covers for the case where there is an RTS collision because of a significant load. If the backoff gets over in the same control window the RTS is retransmitted, while if it gets over during the data window the node waits for control window to start again.

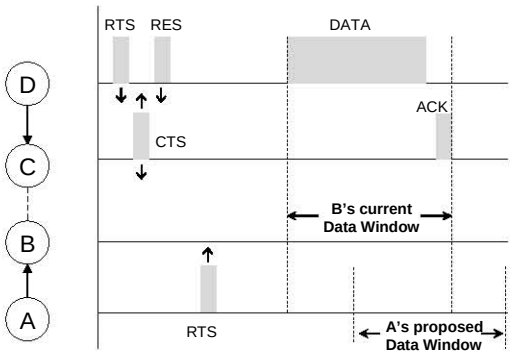


Figure 2.2. Example demonstrating how schedules can be adapted in the LCM MAC protocol and the use of the inflexible bit.

2.5.1.4 Adapting Schedules

Consider the scenario in Figure 2.2, node D sends RTS to node C setting up a schedule which node C accepts by sending a CTS. This CTS is heard by node B and it also starts following that schedule. Now, if node A, hidden from C and D, sends an RTS to B, it may propose a different schedule. The protocol dictates that B has to follow C's schedule. Thus, it cannot reply to A. However, in some cases, with some additional protocol operations the schedule can be adapted.

To explain this we come back to the notion of *master node*. A master node is one that proposes a *new* schedule in its RTS. Thus, in Figure 2.2, D could be the master node, or it could have itself heard a schedule from some other node in its neighborhood and might be following that schedule. We can say the same for node A. If node A is not a master node, then it is impossible for the $A-B$ communication to go on at this time as they both are following different schedules and they cannot violate them. But, in case A is a master node, B can actually reply proposing a change in A's schedule to match its own. Because A is a master node, it can very well accommodate its schedule to facilitate its data transfer to B.

To achieve this, we introduce an *inflexible bit* field in RTS. The inflexible bit is set to zero if the sender of RTS can change its schedule if needed (i.e., it is a master node). On receiving the RTS with inflexible bit set to zero, B can send a CTS with a changed schedule to match its own. On receiving it, A would send a RES with the changed schedule. Note that because neighboring nodes are not meant to follow schedules proposed in RTS, but only in CTS and RES, changing schedules in this manner does not affect any neighborhood node.

2.5.1.5 Improving Efficiency

The constraint in LCM MAC that all nodes in a neighborhood tune to the default channel during control window solves almost all the problems associated with multichannel operations. But, this also leads to an inefficiency – all the non-default channels remain idle during the control window, resulting in a considerable loss of bandwidth. The control window constraint cannot be relaxed as it can lead to more inefficiency in the form of collisions due to hidden terminals and drops due to deafness.

We counter this inefficiency by letting senders transmit a maximum of k (>1) DATA packets to the receivers per negotiation; or in effect, increasing the data window duration by a factor of k. This amortizes the loss of bandwidth during the control window over a number of data transmissions. But it is not clear what should be the value of k. For example, if the size of data window is too large, many senders might run out of packets to send to the respective receivers – resulting in more wastage of bandwidth. If the data window duration is too short, the protocol may still remain inefficient.

We propose a way to adapt the data window duration by doing the following. For every node, if d is the intended receiver for the next packet in the outgoing interface queue, then the node counts the number of packets in the queue that have

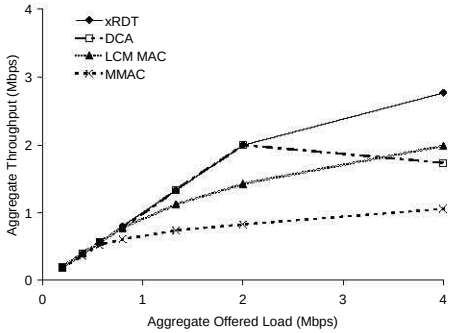
d as the next hop. This information is included by all nodes in their CTS and RES packets for the neighbors to learn about their future transmission requirements. Thus, at the start of the next control window, all nodes have information about the requirements of other neighboring nodes. A *master* node can then take an average of these values (or use some other heuristic) to come up with a reasonable k to be used in current schedule. To eliminate outliers, we also constraint k to be within n where n is the number of channels.

When data window starts, a sender, after sending the current DATA packet can proceed to transmit the next one only if: a) it receives an ACK for the current DATA packet and b) the time left for data window to end is greater than the time required for another DATA/ACK exchange.

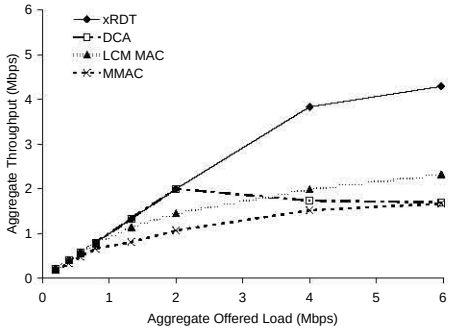
2.5.1.6 Question of Starvation

It is possible that some nodes suffer periods of starvation. To see this, consider Figure 2.2 again. It is possible that A is inflexible, and thus A to B transfer cannot proceed with the current control and data window for B. In that case, A simply has to continue trying in its next control window. It is possible that B again follows a schedule set by another node hidden from A. In such similar cases B can starve – as it is not able to receive packets intended for it. As the average route length increases in the network, the probability of more than one schedule operating in the network increases. This, as a result, could cause more starvation. However, because each transmission starts with a contention period that uses randomization, it is unlikely that a single node will suffer for a long time.

For our implementation, we used an ad-hoc approach to alleviate the problem. Whenever a node suffers long periods of starvation, it disrupts a negotiation which is imposing a second schedule on it by causing a control packet collision. For example, after hearing a CTS from C (intended for D), B can cause a collision at



(a) 6 Channels.



(b) 13 Channels.

Figure 2.3. Throughout vs. load in ns2 simulations with 1Mbps channels, 100 nodes in a 500m $\times$ 500m area.

C when C was supposed to receive the RES from D. In case, A attempts to send a RTS to B again, it can safely reply with a CTS now.

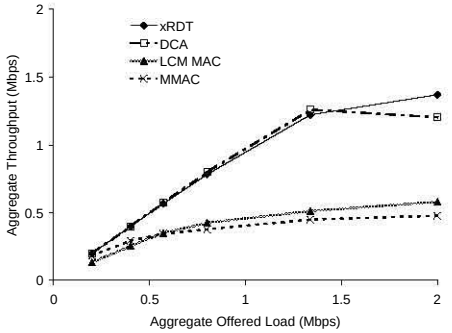
2.6 Simulation-Based Performance Evaluation

We evaluate xRDT and LCM MAC and compare them against two known multi-channel protocols, DCA and MMAC, using the $ns2$ simulator with CMU wireless extensions [71]. The simulations were performed for two scenarios with varying density of nodes. Following common parameters were used in each experiment,

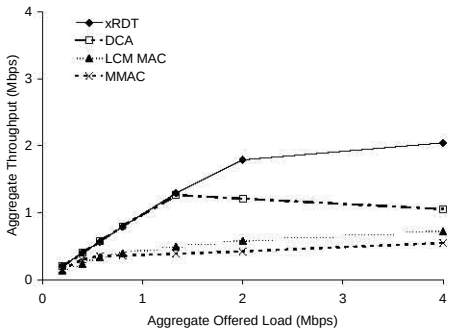
most of them being $ns2$ defaults. The transmission range of each node was set to 250m, the carrier sense range to 500m and the bit rate of each channel to 1 Mbps. The wireless propagation model used was Two Ray Ground. Number of nodes in each scenario was 100 while the number of flows was 50. Each flow consisted of constant bit rate (CBR) traffic generated at the source node with data packet size of 1000 bytes. The data packet generation rate for each flow was varied to vary the load in the network and simulations were done for different number of channels. Static routing was used with routes computed at the start of the simulation. Each simulation is performed long enough for the output statistics to stabilize. Each data point in the plots is an average of five runs where each run used a different randomly generated topology.

We simulated five protocols xRDT, LCM MAC, DCA, MMAC and IEEE 802.11. For DCA, if there were n channels available, then 1 channel was designated as the control channel and the rest $n - 1$ were used as data channels. For MMAC, the specified values in [105] of 80ms for data window and 20ms for the ATIM window were used. Two 100 node network scenarios – with varying density – were simulated. The first (second) scenario was created by randomly placing 100 nodes in a 500m $\times$ 500m (1000m $\times$ 1000m) area. 6 and 13 channel results are presented in Figures 2.3, 2.4 and for the two scenarios. We can make the following observations from these performance plots.

The two-interface protocols usually perform better than the single-interface protocols. xRDT provides much superior performance among all protocols. DCA is a close second, except at high loads. Also, DCA's performance suffers for 13 channel experiments, likely because of the control channel becoming a bottleneck resulting in wastage of data channel bandwidth. Note that both xRDT and DCA use two interfaces, although the second interface in xRDT is a much simpler busy tone interface. So, it is fair to compare them together.



(a) 6 Channels.



(b) 13 Channels.

Figure 2.4. Throughout vs. load in ns2 simulations with 1Mbps channels, 100 nodes in a 1000m $\times$ 1000m area.

LCM MAC performs better than (or equal to) MMAC at all times, although LCM is much better in the 500$\times$500 scenario. The degradation in 1000$\times$1000 scenario is probably due to starvation problem as mentioned in section 2.5.1.6. Also note that MMAC's performance is not good at low loads. This is due to the large data window size. At low loads senders run out of packets to send to the receivers present in their current channel. As they cannot change channel until the end of data window, this results in wastage of bandwidth. Once again, note that both these protocols use one interface; so it is fair to compare them together. LCM MAC also

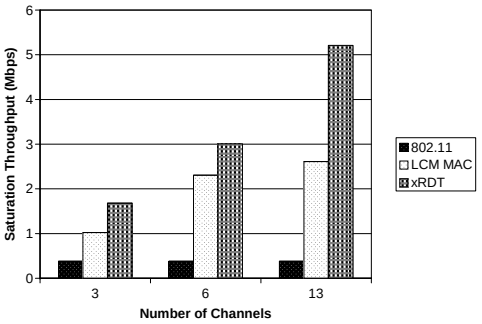


Figure 2.5. A chart comparing saturation throughput of xRDT and LCM MAC with varying number of channels in a 500m $\times$ 500m area. The chart also includes single channel 802.11 for baseline comparison.

does not give proportional improvement with the increase in channels. We discuss more about it later.

Another goal of our work was to show the kind of performance benefits that can be derived from use of multiple channels in wireless networks. For this, we plotted the average saturation throughput of xRDT and LCM MAC in figure 2.5 with varying number of channels (n) and compared them against 802.11. Even though change in number of channels would not affect 802.11 as it is a single channel protocol, we plot 802.11 for different number of channels for ease of comparison. The earlier mentioned scenario with 100 nodes in 500$\times$500 area was used for this plot.

From Figure 2.5 it is clear that xRDT gives more than n times better capacity than 802.11. We attribute the benefit to two factors. Firstly, due to the receiver directed transmission paradigm, xRDT does not face problems related to control channels like those faced by other protocols. It does not waste any bandwidth during control window periods, unlike MMAC and LCM MAC, neither does it face control channel bottleneck issues, unlike DCA. Secondly, due to lesser contention per channel and lesser control packet overheads, xRDT's per-channel performance comes out to better than 802.11. It should also be noted that xRDT's saturation

throughput increases proportional to the increase in channels.

LCM MAC also gives substantial improvement over 802.11, nearing n times for the 3 and 6 channel experiments. But, the saturation throughput does not increase proportionately for 13 channels. This can be attributed to the earlier mentioned problem about loss of bandwidth during the control window (Section 2.5.1.5). Infact, the loss of bandwidth is $O(n^2)$ as it is proportional to the product of number of non-default channels $(n-1)$ and control window size, and control window size is itself proportional to n.

2.7 Conclusions and Future Work

We have presented two multichannel MAC protocols – xRDT with a packet and a busy tone interface, and LCM MAC, with just one interface. Simulation results reaffirm our belief that the receiver directed paradigm has more potential than both control channel based and time-synchronization based protocols. In addition, our asynchronous single interface solution LCM MAC gives performance benefits at par with synchronization based protocols.

Our future work will involve implementing and testing the studied protocols in real wireless testbeds. We will also develop efficient broadcast mechanisms, as broadcast is a useful operation in multihop networks. At this time, the broadcast must be done individually in each channel. Quiescent channel advertisement can be handled in a similar manner. Many more intelligent approaches are possible to solve the starvation problem in LCM MAC and our ad-hoc solution only serves as an incentive to do so. Channel switching time has been ignored in our work as in similar works in literature. However, this presents a practical issue, and we intend to study it in a real implementation.

Chapter 3

Experimental Comparison of Wireless Interference Models

3.1 Introduction

Wireless sensor network researchers have traditionally assumed sporadic, low rate communications between sensor network nodes. But interest has grown recently in studying high rate sensor network scenarios []. E.g, when aggregating sensor data, while the data rate may be low at source, the network may be quite busy near the sink where nodes multiplex many flows. Also, some sensor nodes may be idle for most of times, but can be very active momentarily when an event happens or a target needs to be tracked, for example. Such scenarios need high data rate communication between nodes. Interference between simultaneously transmitting nodes is the single biggest factor that hampers wireless capacity. Thus, practical approaches for modeling interference on wireless links are critical for understanding wireless network behavior. Fundamentally, the MAC layer protocol must be able to schedule transmissions on links in an interference-free fashion. There are several

interference models that have been considered in the literature and used in transmission scheduling studies. They vary from oversimplified range-based models to fairly realistic SINR-based physical models [46]. The general research approach in most cases has been to carefully balance the modeling realism with a specific research goal, e.g., achieving a performance bound (in algorithmic studies) or making a practically viable implementation (in testbed studies). However, there is a general lack of understanding of the accuracy of various interference models, or how much a less accurate model hurts in transmission scheduling, or whether the SINR-based model can be made 100% accurate in a practical setting. Our work addresses this gap by developing a practical, measurement-driven methodology. To the best of our knowledge our work is the *first systematic experimental comparison study of wireless interference models from the point of view of TDMA transmission scheduling.*

Our general approach is as follows. For the purpose of concreteness in evaluation, we choose TDMA transmission scheduling [77, 94, 116] as the MAC layer model. We specifically target motes and 802.15.4-based low-power sensor networks. We do expect that the general experimental methodology should be applicable for a variety of wireless networks, though actual results could vary depending on specific radio characteristics.

We consider several interference models popularly considered in literature. For example, in the *hop-based* model, interference is specified relative to the communication graph [93]. In the *range-based* model, any node within certain geographical distance from a receiver is assumed to interfere. In the *protocol model* [46], a distance-based relationship exists between the intended sender-receiver pair and any potential interferer. More recently, researchers have started using SINR-based models. These models are also called *physical models* [46]. While physical models have been used in the design of cellular (one-hop) networks for a long time [91],

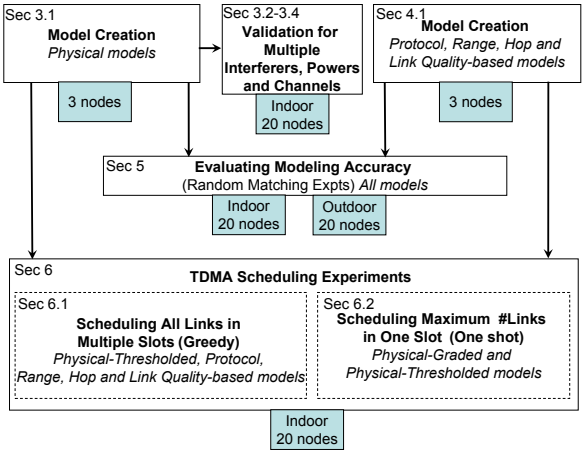


Figure 3.1. Block diagram summarizing the experimental steps in this chapter.

their use in multihop networks for protocol design is fairly recent [22, 45, 73].

Physical models require special attention. Unlike the other models – physical model is not 'pair-wise.' In physical model, a set of nodes transmitting simultaneously may potentially cause enough interference to disrupt an ongoing transmission, while each node transmitting individually may not be able to do so. Also, the physical model introduces a notion of 'graded' interference, while many other models use a notion of 'binary' interference, i.e., interference either exists or it does not. This will play an important role in our evaluations.

3.1.1 Overview of Approach

Our work in purely measurement-based. We use the TelosB motes-platform [75] that uses the Chipcon CC2420 radio with the 802.15.4 PHY layer [111]. Our broad evaluation approach is as follows. See Figure 3.1 for a block diagram.

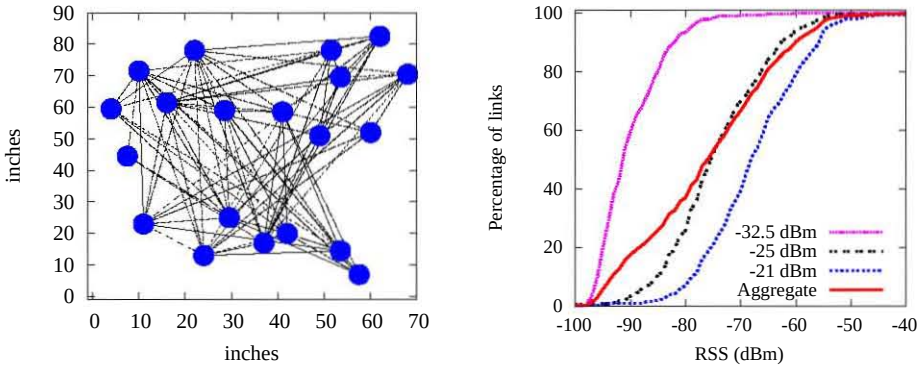
1. We instantiate each model separately using a three node setup (sender, receiver and interferer). This includes the physical model and all pairwise models.

2. We put the physical model through an extra validation step – validating for use with multiple interferers, diverse transmit powers and multiple overlapping channels. The physical model requires this step as it is supposed to be independent of these three concerns. (The other models are pairwise and do not consider multiple interferers. Also, they have to be instantiated separately with different transmit powers and channels using step 1 above.)

3. We evaluate modeling accuracy for all models for transmission scheduling use. This step essentially uses a random sampling study using random matchings. This step brings out a new insight about the physical model – the 'graded' version of the model is more accurate than the commonly used 'thresholded' version.

4. We use actual TDMA scheduling experiments for further comparison across models. Here, we go through two sets of experiments. First, we use traditional greedy scheduling techniques for all models for scheduling all network links following a given demand vector. This step, however, cannot use the 'graded' physical model as algorithms are yet unknown for this. Thus, we show the benefits of this model with an exhaustive search using a simpler, one-shot scheduling experiment.

Two 20-node testbeds are used for most validation and evaluation, except that a 3-node testbed is used for initial model creation. The testbeds are referenced along with specific experiments in Figure 3.1. We will start with a description of the experimental platform in Section 3.2. The rest of the chapter is laid out in the above



(a) Topology

(b) CDF of RSS values

(c) Setup environment

Figure 3.2. (a) Topology of the indoor 20 mote setup for -32.5 dBm transmit power. Links shown have at least 99% PRR. This results in average degree of about 9. (b) CDF of RSS values observed in this testbed for different transmit powers used. (c) A picture of the indoor deployment environment.

sequence. The appropriate section numbers are noted in Figure 3.1 for the benefit of the reader.

3.2 Experimental Platform and Setup

We use TelosB motes [75] that use CC2420 radio [111]. The radio is is compliant with the IEEE 802.15.4 [49] PHY layer standard in the 2.4 GHz ISM band and operates at the nominal bit rate of 250 Kbits/s. The radio provides some flexibility

in terms of choice of frequency and transmit power that has been quite useful in our work. A custom MAC layer is implemented to enable TDMA transmission scheduling. The necessary details about our setup is described below.

3.2.1 Channels

The CC2420 radio can operate in various frequency channels of 5 MHz bandwidth in the 2.4 GHz ISM band. Channel switching in CC2420 can be done dynamically in steps of 1 MHz [111]. This gives us the capability to create partially overlapped (i.e., interfering) channels useful to study inter-channel interference in wireless networks [72]. We use three such channels in this work and we refer to them as channels f_A, f_B, and f_C, with center frequencies 2480 MHz, 2479 MHz and 2478 MHz respectively. These frequencies are chosen specifically because they do not overlap with the 802.11 channels in the region of the world where the experiments were done. These channels overlap by various degrees. Note that given the radio restrictions (5 MHz channel bandwidth and center frequency set at 1 MHz intervals) we can use only 3 channels to experiment with partially overlapped channels. A further shift of the center frequency creates orthogonal channels (i.e., center frequencies 3 MHz or more apart). We have experimentally verified them as non-interfering[1] and thus they are not useful here. We conducted most of our experiments on a single channel (channel f_A). For multichannel experiments, we tuned the receivers to channel f_A and sender/interferers to one of the 3 channels depending on the experiment.

[1]This observation is also supported by the transmit spectral mask values mentioned in the radio datasheet [111].

3.2.2 Received and Transmit Powers

The CC2420 radio provides a measure of the received signal strength (RSS) in dBm, which is an estimate of signal strength averaged over 32 bit 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. Since packet reception is not always possible for weak signals, we read the RSS from the register periodically to obtain signal strength even when the packet is not received.

The CC2420 datasheet [111] specifies that the transmit power can be programmed between -25 to 0 dBm in 8 steps. But we verified experimentally that the power levels can be varied at a finer scale from -32.5 to 0 dBm.[2] Thus, we have the choice of picking from a wide range of power levels.

3.2.3 MAC Layer and Measurement Process

We have implemented a simple TDMA protocol in TinyOS-2.0 [5] in which motes transmit at designated time instants without performing carrier sensing or backoff as in the default MAC implementation in TinyOS. We achieve time synchronization between nodes in the testbed as follows. One mote outside the testbed is directly connected to a laptop via USB. This mote and laptop combination is loosely referred to as the 'base station' (BS). The base station is positioned in such a way that all network motes can directly talk to the base station using the maximum transmit power (0 dBm). This is the power the base station also uses. The base station periodically (500 ms intervals) transmits 'beacons' that the motes use to synchronize their clocks. For multichannel experiments, we have used multiple motes in different channels on the base station so that beacons can be transmitted on all channels.

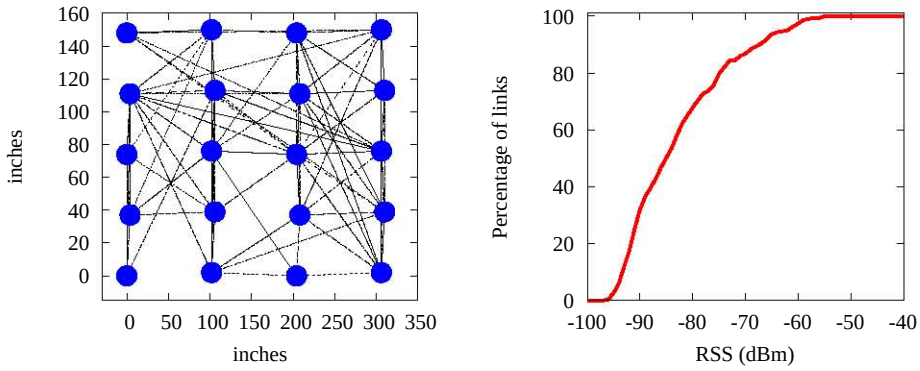
Since much of our work is related to concurrent transmissions and TDMA

[2]This undocumented feature was confirmed by the mote manufacturer [1].

scheduling, we have also implemented a 32 KHz precision timer to achieve low jitter between the actual and the scheduled transmission start times across motes. This is necessary to observe capture effect since capture depends upon the arrival times of overlapping transmissions [118]. If a receiver synchronizes its radio to a stronger signal, a late arrival of weaker signal does not affect the stronger signal reception. But in the converse case of stronger signal arriving later, both transmissions can be lost. To avoid this case, the stronger signal must arrive no later than the synchronization time, i.e., the duration of the start frame delimiter (SFD). This time is 128 μs for CC2420. We have experimentally observed that the maximum jitter in transmission start times in our setup is less than this value.

The base station also acts as a command and control center for the network for the measurement process. Any measurement activity in the testbed is initiated by broadcast 'command' message(s) from the BS. The command message contains specific instructions for each node and the nodes then start the necessary 'activity' (RSS measurements, packet transmissions etc., possibly at the scheduled time instants as indicated in the command). Similarly, when the 'activity' is over (the period of activity is pre-determined), the BS mote sends 'poll' messages to motes to collect measurement data one at a time. These protocols are fairly straightforward owing to one-hop connectivity between the base station and motes and we do not describe these details here.

It is important to note that care is taken so that all measurements are done within the timing beacon interval so that the beacon do not interfere with the measurements. But they are repeated in different beacon intervals for obtaining desired confidence levels.



(a) Topology

(b) CDF of RSS values

(c) Deployement area

Figure 3.3. (a) Topology of the outdoor 20 mote setup for 0 dBm transmit power. Links shown have at least 99% PRR. This results in average degree of about 8. (b) CDF of RSS observed in this testbed. (c) Google Maps image of the parking lot environment where the testbed was deployed.

3.2.4 Experimental Setups

We use three different experimental setups in this work. They vary in size, transmit power, area of deployment and deployment environment.

Testbed with 3 motes: This consists of three nodes – one receiver, one transmitter and one other transmitter acting as an interferer. This setup is used for instantiating the various interference models we compare in this work. The receiver is kept stationary and the positions and transmit powers of the transmitter and interferer nodes are varied to cause various interference patterns at the receiver. This testbed is used in a large indoor area for the physical interference modeling in Section3.3. For modeling the pairwise models in Section 3.4 it is moved to the same environment that the model is used (i.e., one of the two environments below).

Indoor testbed with 20 motes: This setup consists of a static 20 motes testbed deployed indoors in a quiet office environment. The 20 motes are placed in a random fashion on a 7.5 ft long, 6 ft wide tabletop (Figure 3.2(c)). Since this testbed is exercised the most, the motes are powered through their USB interface from power outlets for convenience. Transmit powers from the lower range are chosen for this setup according to the area of deployment. This enables multiple simultaneous transmissions without making the resulting network graph too sparse. The resulting network topology for the testbed when a transmit power of -32.5 dBm is used is shown in Figure 3.2(a). The average degree of the nodes in the network graph comes out to be about 9. The cumulative distribution function of received signal strength (RSS) observed at receivers of all 380 links for three different transmit powers is shown in Figure 3.2(b). These are the three power values that would be used later in our experiments in this testbed.Also the CDF of aggregate data is shown. This shows that the RSS is well distributed over a range.

Outdoor testbed with 20 nodes: The final setup consists of 20 motes placed outdoors in an open parking lot. (See Figure 3.3(c)). This testbed was temporarily

setup on a weekend when there were sufficient empty spaces. The nodes are placed in a grid like topology as shown in Figure 3.3(a) for convenience. These motes are powered through batteries since there is no easy way to power them through USB in an outdoor environment. While the previous tabletop testbed uses transmit powers from the lower end, this setup uses the highest possible transmit power, 0 dBm[3]. The cumulative distribution function of received signal strength (RSS) values observed at receivers of all 380 links for this setup is shown in Figure 3.3(b).

We end this section with a note on power. We have noticed that use of battery power reduces transmit range. This might mean that the transmit power set by the program is not the transmit power actually used, depending on power sources. Thus, it will not be appropriate for the reader to compare range and related data across experimental testbeds, as we have used different power sources in different cases (USB/mains and battery). However, range and related parameters are profiled separately for the indoor and outdoor scenarios. So, these differences do not play any role in our results.

3.3 Building Physical Interference Model

The physical interference 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 ΣI is the sum of the interference powers experienced

[3]The setup and choice of powers are somewhat related. For an interference study, we need a setup where there are enough concurrent transmissions possible on some links, as well as there are enough opportunities of interference on others individual and cumulative. Otherwise, there is a danger of arriving at trivial conclusions.

at the receiver caused by the group of interferers (transmitting concurrently), the model predicts the relationship between the bit error rate (BER) and SINR, where SINR = $\frac{S}{\Sigma I+N}$. This relationship depends on radio properties such as modulation. The packet error rate (PER) is directly related to BER and depends on coding. The packet reception rate (PRR), a quantity we will evaluate directly, is simply $1 -$ PER and thus again is directly related to SINR.

Typically, the PRR vs. SINR curve makes a sharp transition from low to high PRR values with increasing SINR. The rising part of the function has been described as the *transition region* in [131]. Since scheduling applications need a 'binary' model, the curve is typically 'thresholded' and is described as a step function changing from 0 and 1 at a specific value of SINR, called the *SINR threshold* or *capture threshold*. This variant of the physical model is henceforth referred to as *thresholded* physical interference model. The original model will be called the *graded* physical interference model.

3.3.1 Modeling with Single Interferer

To build the physical model, one needs to find the PRR vs. SINR relation. We do this empirically by simply taking many measurement samples of S, I and N for the three node setup (Section 3.2.4) – sender, receiver and interferer, thus directly computing SINR as $\frac{S}{I+N}$. The samples vary in the values of S and I. This variation is obtained by changing the distances between transmitter-receiver and interferer-receiver pairs. The transmitter receiver distance is varied from 1 foot to 64 feet[4] in discrete steps. For each such transmitter-receiver distance, the interferer-receiver distance is also varied from 1 foot to 64 feet.

The following measurements are performed in three successive steps for each

[4]The TelosB datasheet [1] documents that its indoor RF range upto 64 feet.

transmitter-receiver and interferer-receiver distance pair. Each experiment in each step is preceded by the base station sending command message(s) and followed by the base station sending poll messages to collect the data. All packet transmissions in the testbed are done with 128 byte packets.

1. *Noise estimation:* Noise is measured by sampling the RSS register in the CC2420 radio when there is no other transmission. The receiver samples its RSS register every 20 ms for a period of 6 seconds. Using the valid values thus obtained[5] the average noise at the receiver in the network is computed.

2. *Pairwise RSS measurement:* Transmitter and interferer take turn to send 1000 packets in succession to the receiver. Each packet transmission time is approximately 4 ms. The receiver samples the RSS register every 3 ms to obtain RSS on its link with the corresponding sender. (More frequent sampling did not change the measured RSS.) It is possible that some of these samples may have been taken when the sender is not transmitting. Such samples are filtered out from the dataset by comparing it with the noise estimate obtained in step 1. The average of RSS value from transmitter is taken as S, while the RSS from interferer is taken as I for calculating the SINR. This entire step is repeated for 8 different transmit powers covering the entire transmit power range of CC2420 radio from -32.5 dBm to 0 dBm. In all, this results in 64 experiments.

3. *Concurrent transmission:* In each experiment, the transmitter and the interferer 'concurrently' transmit 1000 packets each. The receiver records the number of packets it received correctly from the transmitter. This defines the packet reception rate (PRR) for the transmitter-receiver link in the presence of the interferer. This step is also repeated for 8 different transmit powers

[5]Not all read attempts for the register produce valid values [111].

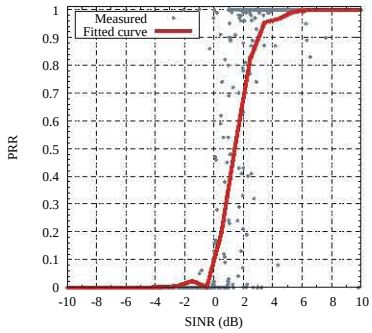


Figure 3.4. PRR vs. SINR relation for single interferer measurements on a 3 node setup. The fitted curve on the aggregated data (bold,red) is shown for reference.

covering the entire transmit power range of CC2420 radio from -32.5 dBm to 0 dBm to exactly correspond to the step 2 above.

The above three steps are done in succession so that noise and RSS measurements (in steps 1 and 2) are as fresh as possible when PRR is measured in step 3. This is to avoid any form of noise/RSS fluctuations over time. In the measurement time period we did not observe any statistically significant fluctuations. For example, when steps 1 and 2 are repeated we obtained samples statistically similar as before.

For each PRR obtained in step 3 above, the SINR is calculated using measurements from step 1 and 2. A scatterplot showing the results of this experiments is shown in Figure 3.4. The results show that for SINR greater than about 5 dB, PRR is almost 100%. As mentioned before, there is a *transition region* [131] 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 [131] albeit for a different radio). The PRR trails down to 0 below (-3) dB. We also show a fitted curve using a linear interpolation of average values in buckets of 1 dB each. This fitted curve provides the PRR vs. SINR model that will used in our later analysis. This model can also be used directly by a scheduling algorithm.

3.3.2 Validation with Multiple Interferers

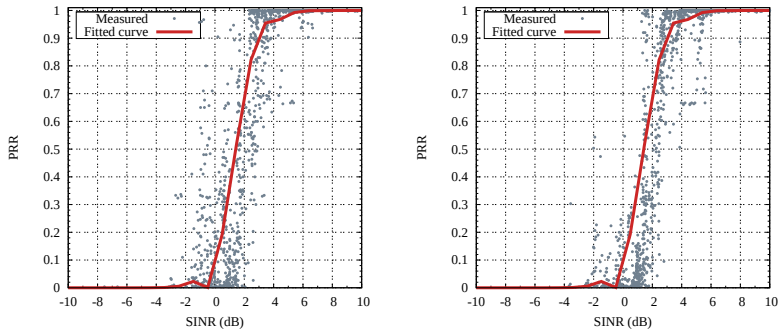
While in theory the physical model is dependent only on the received powers and not on the number of interferers, previous work has made an observation in the contrary, albeit for an older generation radio (CC1000) [107]. Much was attributed to hardware imperfections and measurement noise. Since we undertake a measurement-based paradigm, it is of interest to validate the above empirically derived model in presence of multiple interferers. In the next sub-section, we will also extend this validation for multiple channels and multiple transmit powers. *These validations are key to assumption that only received powers drive the model and not any other parameter.*

We develop a systematic methodology for validation with multiple interferers. Let us denote by $RSS_r^p(s)$ the received signal strength at node r when a node s transmits with transmit power p; and by N_r the ambient noise at r. Assume that a set of nodes, Φ, is active simultaneously transmiting at power p. Then, we also denote the PRR at r from a node $i \in \Phi$ as $PRR_r^p(i, \Phi)$. In this case, the SINR at r for node i is given by

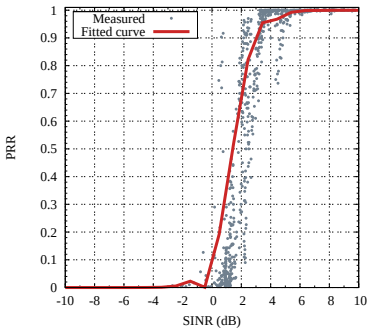
$$SINR_r^p(i, \Phi) = \frac{RSS_r^p(i)}{N_r + \sum_{\forall j \in \Phi, j \neq i} RSS_r^p(j)} \tag{3.1}$$

$SINR_r^p(i, \Phi)$ above can be computed from individual pairwise RSS measurements done separately. $PRR_r^p(i, \Phi)$ can be directly measured by making the nodes in Φ transmit together at power p and by measuring PRR at node r for packets transmitted by i. This provides a data point for the PRR vs. SINR relation. In fact, just one single experiment with the nodes in Φ transmitting together can provide PRR at any node $r \notin \Phi$ for each sender $i \in \Phi$. Such experiments can be repeated for different Φ and p in different settings, providing many data points for the PRR vs. SINR relation.

Similar measurements as in Section 3.3.1 are performed in three successive steps on the 20-node indoor testbed to this end. Only one transmit power is used



(a) 4 transmitters (3 interferer). (b) 5 transmitters (4 interferers).



(c) 6 transmitters (5 interferers).

Figure 3.5. PRR vs. SINR for different number of interferers. The fitted curve on (bold, red) is shown for reference.

(-32.5 dBm). One other important difference here is that there are more than one interferer and thus more than two concurrent transmitters, i.e., $\Phi > 2$. This makes step 3 only slightly more elaborate. Here, a set of nodes Φ 'concurrently' transmit 1000 packets each. All nodes $r \notin \Phi$ act as receivers. Each receiver records the number of packets it received correctly from each transmitter. This defines the packet reception rate (PRR) for different links in presence of a set of interfering transmissions. The set Φ was chosen randomly out of the 20 nodes in the network. The size of set Φ was varied from 3 to 6. 100 such random sets are used for each chosen value of $|\Phi|$.

At the end of the measurement process, we have $100 \times |\Phi| \times (20 - |\Phi|)$ data

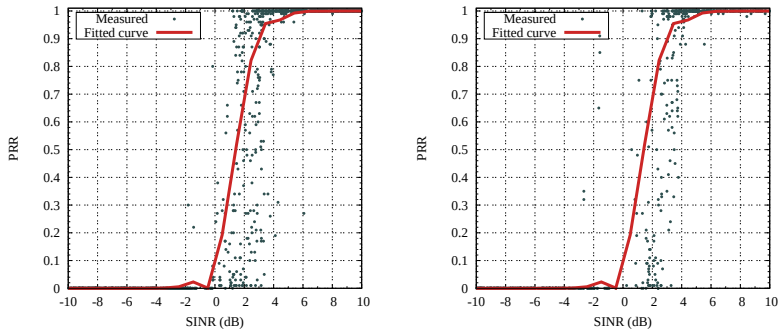
points for the PRR vs. SINR relation for each $|\Phi|$. We show these as scatterplots in Figure 3.5 categorizing into different values for $|\Phi|$. For brevity, only 4, 5 and 6 transmitter cases are presented, which means 3, 4 and 5 interferers, respectively. This categorization is specifically intended to demonstrate that the relationship is independent of the number of interferers and interference does work in an additive fashion as the theory predicts (at least upto the extent of 5 interferers that we could study). The fitted curve developed in the previous subsection is shown as well for comparison. Note the excellent fit. The coefficient of determination (R^2) values for these experiments with respect to the fitted curve is always over 0.90.

A separate set of analysis (not reported here for brevity) also revealed excellent agreement between measured aggregated interference and sum of the individual RSS's from the interferers with $R^2 = 0.99$. It shows that the observations in [107] is quite likely due to imperfections and high degree of measurement noises in older generation hardware.

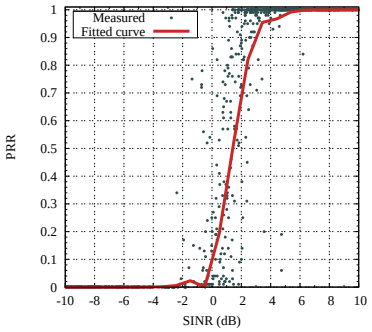
3.3.3 Validation with Multiple Channels and Transmit Powers

So far, we have used only one channel (channel f_A) and the same transmit power (-32.5 dBm) in our multiple interferer modeling experiments. Since often scheduling protocols use multiple channels [] and different transmit powers [] to exploit diversity, we want to also validate whether the PRR vs. SINR relationship depends upon transmit power or transmission channel. While an exhaustive evaluation is combinatorially explosive, we have carried out a large number of experiments to validate that the empirical PRR vs. SINR relationship obtained in Figure 3.5 does hold for various power levels and channels. For lack of space we report a subset of the results in Figure 3.6.

The same methodology is followed as before. Step 2 is repeated with senders transmitting at three different transmit power levels and on three channels (See



(a) single channel, power -25 dBm (b) single channel, power -21 dBm



(c) multi channel, power -32.5 dBm

Figure 3.6. PRR vs. SINR results for 3 transmitters with different transmit powers and channels. Single channel experiment results with transmit powers of -25 dBm and -21 dBm are shown in (a) and (b) while multichannel experiment result with transmit power of -31.5 dBm is shown in (c). The fitted curve (bold, red) is shown for reference.

Section 3.2). Homogenous transmit powers have been chosen to reduce the number of experiments. Each experiment is repeated so that the receivers can measure noise and RSS in each channel. In step 3, the channels are selected randomly for randomly chosen set of transmitting nodes, Φ. A subset of results is shown in Figure 3.6 shown against the same fitted curve. Note that, generally speaking, the SINR vs PRR relation remains fairly independent of different transmit powers and use of multiple overlapping channels. We again have an R^2 value of at least 0.90 for these results.

3.3.4 Discussions

While one could like a better overall confidence than 0.90, we attribute the remaining variations to hardware differences between individual motes and measurement errors. Given our experience vis-a-vis prior work [107], we feel that low-power radios have matured enough that a purely measurement-based SINR profiling independent of any other parameter is possible and is usable in scheduling studies.

We have also investigated whether the analytical BER vs. SNR curves can be directly used instead of profiling the PRR vs. SINR relation via measurements. Such analytical curves can be derived from the knowledge of modulation/coding and the noise processes. See [49] for the analytical BER vs SNR curves for 802.15.4. We found that this curve is about 2 dB shifted towards the left from the fitted curve we have derived here. Much can be attributed to this difference – from measurement errors in the low-cost radio to the fact that spectral characteristics of interference is different than AWGN noise assumed in the analytical curve. Calibrating the analytical model with measurements would be interesting, but is of little value in the work we are pursuing here.

3.4 Pairwise Interference Models

One goal of our work is to experimentally compare interference models. Our comparison points will be various pairwise interference models that are commonly used in literature. They consider interference within *only pairs of links* as opposed to sets of links as in the physical interference model. The advantage of pairwise models is that the interference can be represented in terms of a *conflict graph* [50], which makes modeling and analysis straightforward. For example, for scheduling one simply needs to find an independent set of nodes in the conflict graph. In this section, we present the pairwise models and the empirical techniques used to instantiate the models.

Before we describe the models, let us define some notations. Assume that the network graph is denoted by $G(V, E)$. A communication link between two nodes $u, v \in V$ (u is the sender and v is the receiver) is denoted by $l(u, v) \in E$. Assume that the physical distance between the two nodes is $d(u, v)$. In the following, we enumerate the conditions under which each model predicts that a link $l(x, y)$ interferes with another link $l(u, v)$. These models consider this interference in a *binary* sense — PRR on $l(u, v)$ will be 0 if $l(x, y)$ interferes, else the PRR will be 1. This is make it amenable to conflict graph representation.

3.4.1 Description of Models

Hop-based model: Hop-based interference model [103] states that link $l(x, y)$ interferes with link $l(u, v)$, if node x is within k hops of v in the graph G. k is usually 1 or 2. Many scheduling works [94] have used this model to simplify the interference assumptions.

Range-based model: The range based model uses two range or distance parameters, namely, transmission range (d_T) and interference range (d_I). It assumes that

for any link $l(u,v) \in E$, $d(u,v) \leq d_T$. It also states that $l(x,y)$ interferes with $l(u,v)$, if $d(x,v) \leq d_I$. Many modeling and protocol studies [13] in wireless networks use such a model (often referred to as disk model). Range-based model is believed to be more accurate than the hop-based model, but it does not take into account the capture effect – if the sender and receiver are close enough, the packet can be successfully received even when there is an interferer present close by.

Protocol model: The protocol model was first introduced in [46]. This model also assumes a concept of transmission range as before, i.e., for any link $l(u,v) \in E$, $d(u,v) \leq d_T$. The model also states that link $l(x,y)$ will interfere with $l(u,v)$ if $d(x,v) \leq (1+\Delta)d(u,v)$, where $\Delta \geq 0$. This model improves on the range-based model by making interference dependent on the ratio of the distances between sender-receiver and interferer-receiver and thus tries to address the capture effect. Δ is assumed to be independent of the distance $d(x,v)$ and $d(u,v)$.

Link quality-based model: Models using any concept of distance or SINR require pairwise distance or signal strength measurements. This may not be feasible always. To address this we introduce a new model that defines interference based on link quality as measured by PRR in absence of interference from another link. In this model, link $l(x,y)$ will interfere with $l(u,v)$ if link $l(x,v)$ has a PRR more than a given threshold (*interference threshold*).[6] It is also assumed that the link $l(u,v)$ already has a strong quality, characterized by a high PRR (PRR larger than an given threshold called *transmission threshold*). Note that transmission threshold must be larger than interference threshold.

[6]Note that link $l(x,v)$ may not exist in the network graph G. So consider it hypothetical.

Model	Parameters for indoor scenario (Transmit power = -32.5 dBm)	Parameters for outdoor scenario (Transmit power = 0 dBm)
Hop-based	k =1 (1 hop)	k =1 (1 hop)
Range-based	$d_I = 45$ inch	$d_I = 255$ inch
Protocol	$\Delta = 0.36$	$\Delta = 0.67$
Link quality-based	interference threshold = 0.0	interference threshold = 0.0
Physical (thresholded) interference	PRR vs. SINR model (Sec. 3.3) SINR threshold = 5 dB	PRR vs. SINR model (Sec. 3.3) SINR threshold = 5 dB
Physical (graded) interference	PRR vs. SINR model (Sec. 3.3)	PRR vs. SINR model (Sec. 3.3)

Table 3.1. Summary of model parameters used in experiments.

3.4.2 Instantiating Models

Just like the physical model in Section 3.3 the above pairwise models must be instantiated. This means that various model parameters need to be determined. But unlike the physical model, for which we separately verified the additive nature of interference, the *classical* definitions of pairwise models are taken as the ground truth (as listed in Section 3.4.1) and we just instantiate them through measurements. In our work, transmission threshold is set at 99%. All links with PRR equal or more than 99% are considered links in the network graph G. Using this definition of link, we say that an interferer interferes with a link if the concurrent transmission from the interferer and the transmitter causes the receiver to receive less than 99% of packets from the transmitter. Using this definition of interference, we perform experiments to instantiate various models. We follow a similar methodology with the three node setup as in Section 3.3.1 (single interferer modeling).

To instantiate the range-based model, the following technique is used. The distance between the transmitter and the receiver is slowly increased from a very small value. The transmit power is kept constant. The PRR of link from the transmitter

to the receiver drops below 99% at some distance. This distance is the transmission range, or d_T. To measure the interference range, d_I, we first keep both the transmitter and the interferer at distance d_T from the receiver. Then the interferer is slowly moved further away from the receiver. For each such distance, PRR is measured at the receiver, when both transmitter and interferer are active concurrently. The PRR usually starts close to 0% and increases with increasing distance of the interferer. The distance at which PRR on transmitter-receiver link crosses 99% is taken as the interference range, d_I.

For the protocol model, we use Δ such that $d_I = (1 + \Delta)d_T$. Values of d_I and d_T are obtained from the above experiments. Since, Δ should not depend on the distances between transmitter-receiver or interferer-receiver, any pair of distances which cause interference should suffice. Thus, using d_I and d_T is sufficient.

The link quality-based model is instantiated in a similar manner by using an *interference threshold* such that the PRR on the transmitter-receiver link drops below 99%. This directly corresponds to the PRR on the interferer-receiver link when interferer is at distance d_I from receiver and the transmitter is at distance d_T from the receiver. For hop-based model, k=1 (one hop) as well as k=2 (two hop) models are evaluated. It was found that one-hop model gives better accuracy and is thus considered henceforth.

The above instantiation experiments are performed both for the indoor and outdoor 20 nodes testbeds separately using transmit power of -32.5 dBm and 0 dBm respectively. The resulting parameters are listed in Table 3.1. For completeness the physical interference models are also included here.

3.5 Comparing Interference Models

Our goal here is to compare various pairwise interference models with the SINR-based physical model for TDMA transmission scheduling. Since scheduling essentially determines 'feasible' transmission sets (links) in each slot subject to an interference model, a model's responsibility is to describe which sets of links are feasible together and which are not.

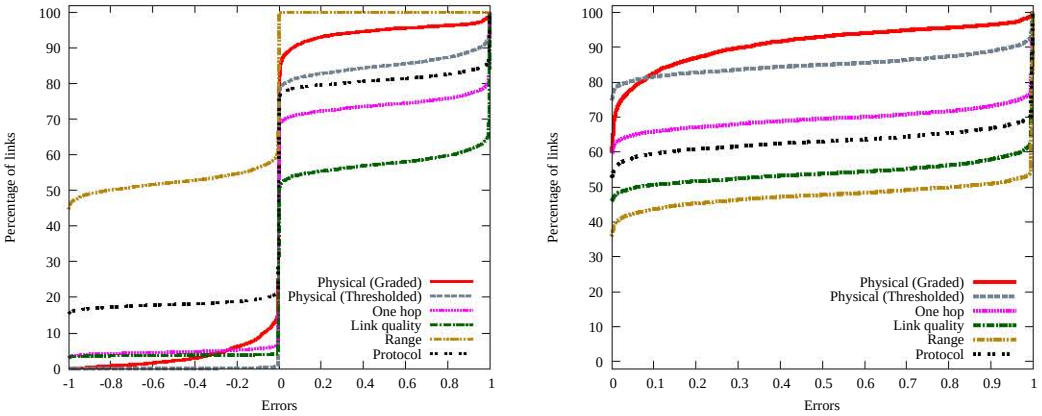
As evaluating all sets of links is intractable (there are exponentially many such sets), the best way to compare the interference models is to do a sampling study by comparing modeling *accuracy* in predicting the feasibility of a *randomly chosen set of links*. The measure of modeling accuracy is simply the difference between the measured throughput for the chosen set of links and the predicted throughput per the given model.

We conducted the experiments in two different setups – our 20 node indoor testbed and the 20 node outdoor testbed, as described in Section 3.2.4. We used -32.5 dBm transmit power in the indoor testbed and 0 dBm in the outdoor testbed. All links with PRR (in absence of interference) equal or more than 99% are considered links in the network graph G[7].

3.5.1 Use of Random Matchings

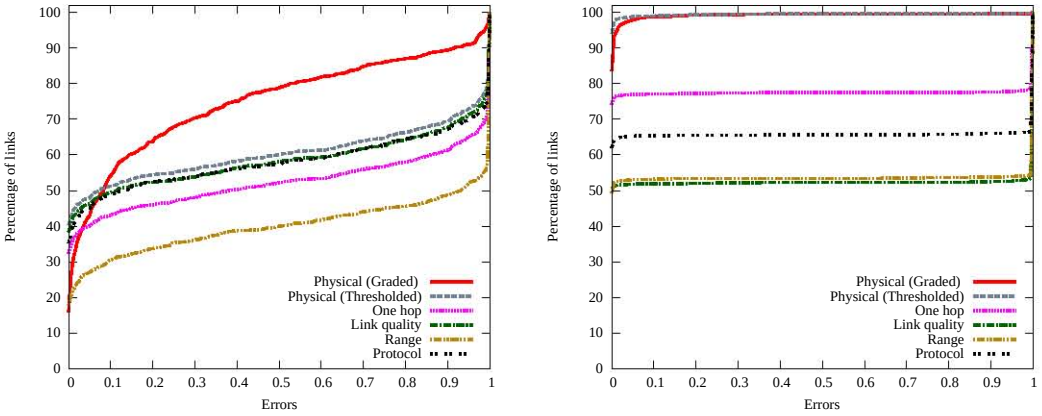
For the sampling study as mentioned before, it is possible to do some optimizations. Any scheduling algorithm must avoid the so-called *primary interference*, i.e., interference between links with a common endpoint in the network graph. Thus, the

[7]It is possible that if this transmission threshold is different, the conclusions we draw from our results will also be different as the topology in use is different.



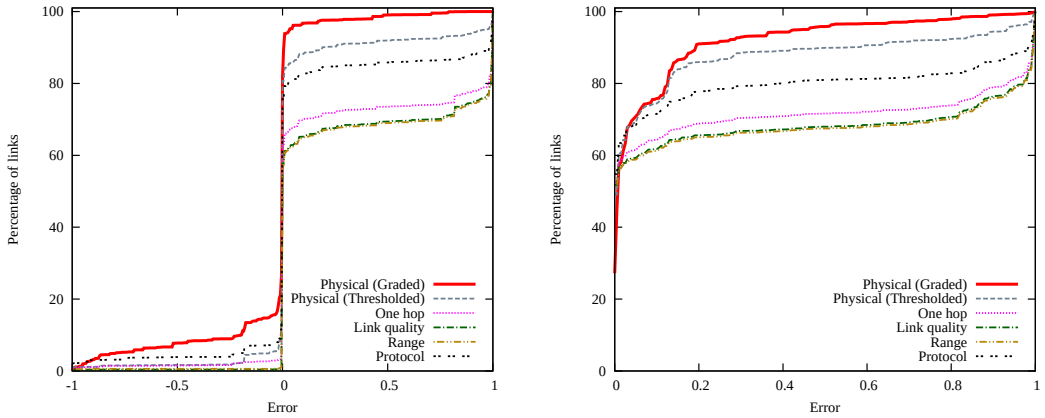
(a) Actual error. (b) Absolute error.

Figure 3.7. Indoor testbed (-32.5 dBm transmit power): CDF of modeling errors (per Equation 3.2) for different interference models. (Absolute error is simply the absolute value of the actual error.)



(a) Absolute error in transition region. (b) Absolute error in non-transition region.

Figure 3.8. Indoor testbed (-32.5 dBm transmit power): CDF of absolute modeling errors (per Equation 3.2) for different interference models, with data split into transition and non-transition regions.



(a) Actual error. (b) Absolute error.

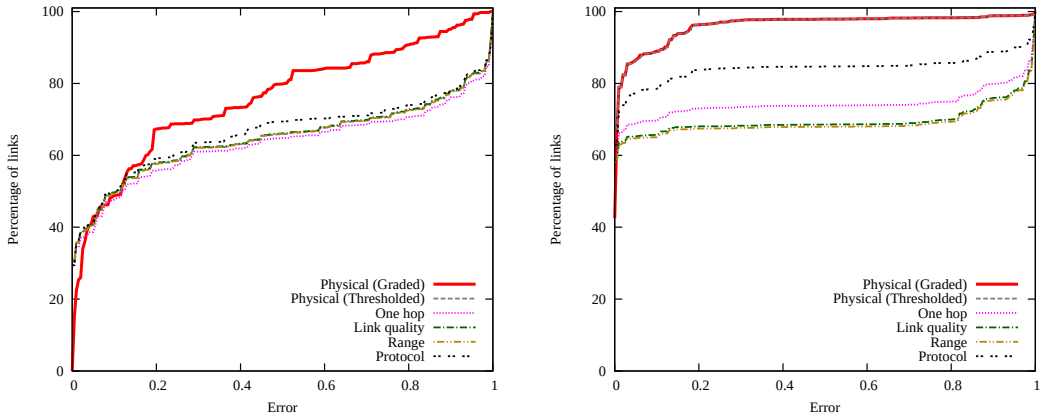
Figure 3.9. Outdoor testbed (0 dBm transmit power): CDF of modeling errors (per Equation 3.2) for different interference models. (Absolute error is simply the absolute value of the actual error.)

algorithm must choose a *matching*[8] on the network graph as we are only considering unicast transmissions. Thus, instead of using random subset of links, we can use random matchings. There is no point in evaluating non-matchings as they will never be scheduled by any algorithm. Interestingly, use of matchings does not necessarily reduce the complexity of the problem as there can be exponentially many matchings. Thus, we still need to do random sampling. Choosing random matchings is intractable as well. Thus, we resort to a heuristic mentioned in the appendix to pick random matchings.

About 13,000 random matchings are used for the indoor experiments and about 3,000 for the outdoor, providing significant data sets. Such large data set also makes the study relatively independent of the topology used. This is because we could independently verify that the data set included many instances of links well distributed over the relevant range of SINR values.

For each randomly selected matching the actual throughput (normalized) of each link is evaluated when all links in the matching are transmitting concurrently

[8] A matching is a set of links such that no two links have a common end point.



(a) Absolute error in transition region.

(b) Absolute error in non-transition region.

Figure 3.10. Outdoor testbed (0 dBm transmit power): CDF of absolute modeling errors (per Equation 3.2) for different interference models, with data split into transition and non-transition regions.

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 matching, 1000 concurrent transmissions are done over all links in the matching to calculate throughput.

3.5.2 Modeling Error

Each random set of matching is used as input to a predictor that evaluates the link throughput predicted by each interference model discussed in Sections 3.3 and 3.4. Note that all links in a given matching may not be deemed feasible by a given interference model. Thus, for the binary models, the throughputs of all conflicting links in a matching are assumed to be 0 and those of the non-conflicting links are assumed to be 1. For the physical model, the throughputs are simply the PRRs as determined by the PRR vs. SINR relation.

The modeling error is evaluated in the following fashion. Given a matching M_i consisting of $|M_i|$ links we denote the measured throughput for j-th link

in this matching as $\Gamma_i^j(measured)$ and the predicted throughput by model k as $\Gamma_i^j(model(k))$. Then the modeling error for the interference model k with respect to the j-th link in the matching M_i is given by

$$error_i^j(model(k)) = \Gamma_i^j(measured) - \Gamma_i^j(model(k)). \tag{3.2}$$

3.5.3 Experimental Results

Experiments are performed in the two 20-node testbeds. For the indoor testbed the lowest transmit power (-32.5 dBm) and for the outdoor testbed the highest transmit power (0 dBm) are used. The cumulative distribution function (CDF) of the modeling errors (Equation 3.2) is plotted to compare various interference models. Figure 3.7 and Figure 3.9 show the CDF plots for indoor and outdoor testbeds respectively. Note that the very smooth nature of the indoor results is due to a very large dataset (13,000 matchings).

From the CDF results, we can see that overall the graded physical interference model is the most accurate. The 90-percentile error is about 0.25 for indoor and about 0.2 for outdoor experiments. The 80-percentile error is down to 0.07 and 0.12, respectively. Also, note that while the accuracy is good, it is not excellent. We will come back to this question momentarily. The thresholded physical model is a close second to the graded model, with 80-percentile error close to zero for indoor and 0.15 for outdoor experiments. But the 90-percentile error for the thresholded model is very high, close to 0.9 for indoor and 0.6 for outdoor. This high error is due to the fact the thresholded physical model is quite accurate for links outside the transition region, but links in the transition region are predicted to have zero throughput. The percentage of links which lie in transition region for our experiments is approximately 20%. Thus thresholded model gives large error for 20% of cases.

The pairwise models have higher error. The best one among them roughly trails the physical model by 12-18 percentile points for the same error target. They also exhibit some interesting characteristics. Note the nature of the absolute error plots (Figure 3.7(b) and 3.9(b)) – first a sharp rise near 0, then relatively flat and then again a sharp rise near 1. This denotes a bimodal error distribution – most errors are either very low or very high. The reason for this is the binary nature of these models. Note also some of these models have significant bias, they tend to either under-estimation or over-estimation (see Figures 7(a) and 9(a)). Sometimes this bias is not even consistent. This happens for the range-based model that over-estimates in the indoor scenario and under-estimates in the outdoor scenario. Much of these problems is related to the fact that these models depend on estimation of a single model parameter. Among these models, the hop based and the protocol model perform relatively better, but this is again scenario-specific.

Now, let us get back to modeling accuracy question for our best model – the graded physical model. It was observed before, *albeit* with an older mote/radio platform [131], that the links in the transition region are hard to estimate accurately. This is because a slight measurement error makes a significant difference in the estimate. To investigate this issue further in our platform, we split out the results presented in Figures 3.7(b) and 3.9(b) into two parts, for the transition and non-transition regions. Recall that from our model instantiation experience in Section 3.3, we found that the transition region for CC24020 radio is -3 to 5 dB. The new plots are shown in Figures 3.8 and 3.10. Note the poorer accuracy of all models in the transition region relative to the non-transition region. But the graded physical interference model still performs better than all other models. The thresholded physical model does much worse than the graded physical model in the transition region. It performs as bad as the other pairwise models. It is interesting to note that both graded and thresholded physical models are *very accurate* for the

non-transition region case, 90-percentile error is about 1% for the indoor setup and 80-percentile error is about 1% for the outdoor scenario.

We summarize our general findings below:

1. Generally speaking, the graded physical interference model outperforms any other. However, the overall accuracy is not perfect, particularly in outdoors.

2. If a binary interference model is to be used (all existing scheduling algorithms rely on such models), thresholded physical model is still the best overall, but this is worse that the graded model.

3. The best performing pairwise model is about 12-18 percentile points poorer than the physical model depending on the environment and error target.

4. If a pairwise model must be used, either protocol or hop-based approach should be preferred. Hop-based model worked well in our indoor experiments and the protocol model for outdoor experiments.

5. The range-based model, while widely used in literature, performs quite poorly in both testbeds. This is even with relationship to a much simpler hop-based model.

3.6 Evaluating Scheduling Performance

The previous section evaluated the accuracy of various interference models in predicting the feasibility of a randomly chosen set of links. While these evaluations are very comprehensive, they only evaluate modeling accuracy, but do not directly model real performances when used in a scheduling algorithm. This is because a scheduling algorithm considers only specific subsets of links for feasibility. This is

entirely algorithm dependent. To gain some insight here, we now study the performance of various interference models for making actual scheduling decisions. We limit our work only to the indoor testbed using transmit power -32.5dBm. Our work here is split into two parts. First, we study all models using a greedy scheduling algorithm – similar to the one used in literature [22, 103, 121] – for scheduling all links in the network. The graded physical model, however, cannot be considered here, as no scheduling algorithm exists in current literature to account for the probabilistic (non-binary) behavior in this model. So, in the second part, we separately consider the graded model and evaluate its performance for a simplified scheduling problem (one-shot scheduling [43]).

3.6.1 Scheduling All Links Using Greedy Algorithm

For a fair comparison, we use the same greedy scheduling algorithm for all models. It is simple to implement and performance bounds are known for specific models [18, 22, 103]. The link demand vector is an input to the algorithm. The demand for a link is simply the number of packets to be scheduled on the link. The schedule is a sequence of slots with a feasible set of links to be scheduled in each slot. In our implementation, each slot is equivalent to one packet (128 bytes) transmission and processing time in the mote (12.5ms). The greedy algorithm works as follows.

Input: Network graph $G = (V, E)$, demand vector on the links $f = (f_1, \ldots, f_{|E|})$ and interference model. The interference model specifies which set of links are 'feasible' together.

Output: Schedule $S = \{S_1, S_2, \ldots, S_\tau\}$, where S_k is a feasible set of links scheduled in the same slot. τ is the schedule length.

Algorithm:

1. Order and rename links such that $f_1 \geq f_2 \geq f_3 \ldots \geq f_{|E|}$.

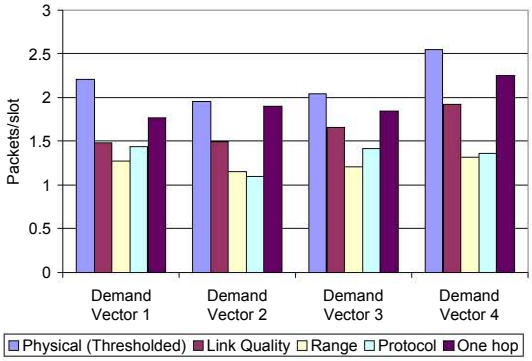


Figure 3.11. Measured aggregate throughput for various interference models for four different link demand vectors (indoor testbed, -32.5 dBm transmit power).

2. Set $i = 1$, $S = \phi$, $\tau = 0$. (Initial schedule is empty.)

3. Schedule link i in the very first available slot where it can be scheduled interference-free according to the given interference model. If no such slot of feasible, increment τ and schedule the link in the last slot. (Incrementing τ is equivalent to creating a new empty slot at the end of the current schedule.)

4. Repeat step 3 above f_i times.

5. Increment i. Go back to step 3 until $i > |E|$.

The physical (thresholded), link quality-based, Range-based, protocol and 1-hop models are considered. As mentioned before, the graded physical model is not considered as the greedy algorithm handles feasibility in a binary sense (either feasible or not). This model will be considered separately in the next subsection.

The models are compared in the following fashion. Different models generate different schedules for a given demand vector. Four different demand vectors are considered for experiments. The links are split into two equal sets randomly. One set has one packet each. The other set has i packets each for vector i. The schedules

generated by each model are evaluated using direct TDMA scheduling experiments on the testbed. Due to modeling inaccuracy some slots may have links scheduled that are infeasible in the experiments. This leads to packet losses. The lost packets are retransmitted. To do this, fresh schedules are computed with only packets lost in the previous attempt constituting link demands. All schedule computation is done by the 'base station' which also has access to all packet loss information (see section 3.2). This procedure is repeated until all packets are successfully transmitted.[9] The above constitutes one trial. Trials are repeated 1000 times for each demand vector and the performance is averaged to determine the *measured aggregate throughput* in packets/slot. The results are presented in Figure 3.11. As expected, physical model has the best throughput, 1-hop model a close second, losing about 5%–20% of throughput. The range-based model generally performs the worst, losing more than 40% of throughput in all cases.

The results here are in general agreement with the observations in the previous section (see Figure 3.7) except for the link quality model. Relative performance of this model is better in scheduling than what we saw before. This is likely due to significant conservative estimates (note large positive errors in this model in Figure 3.7) in this model that works favorably here. This is because the given demand vectors have packets on all links.

[9]Note that this strategy may get 'stuck,' where links scheduled in one slot are all infeasible in reality, and the same links are scheduled in one slot again during retransmissions. This will lead to repeated losses as the same scheduling pattern will continue. This could be addressed (but not perfectly resolved) via randomizing the order of the links considered in the algorithm. We did not, however, see this behavior in our experiments. Note that this is a fundamental problem for using an inaccurate interference model, and does not have a perfect solution.

3.6.2 Graded vs. Thresholded Physical Model: One Shot Scheduling

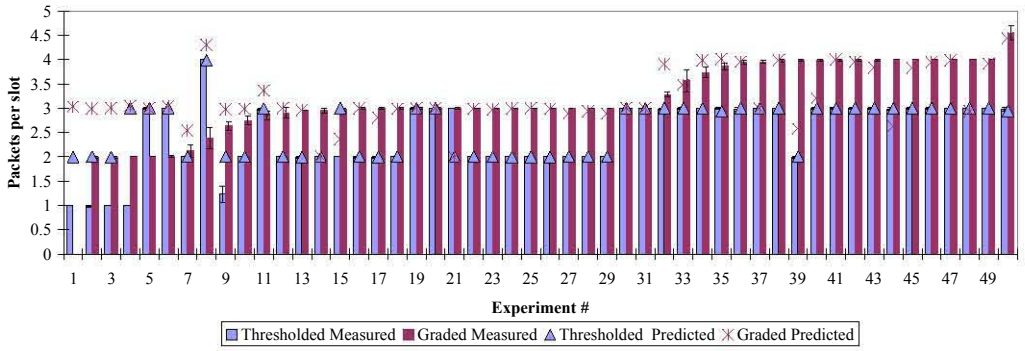


Figure 3.12. Results of the One Shot Scheduling experiment comparing the thresholded vs. graded physical interference models (indoor testbed, -32.5 dBm transmit power).

We did not consider the graded physical model above as the greedy scheduling can only use a binary model (links are either feasible or not). However, the graded model has proved to be the most accurate in our evaluations in Section 3.5. So, it is also instructive to investigate its potential for scheduling.

The thresholded model does perform excellently outside the transition region, but its performance is quite poor in the transition region. The graded model, while not excellent in the transition region either, is still much better than its thresholded counterpart. For example, in the indoor case, the 70-percentile error in the graded model is about 20%, while in the thresholded model it is close to 90% (see Figure 3.8a). A natural question arises: Can this improved accuracy for the transition region be gainfully used in scheduling? Another way to argue this would be to say that the thresholded model is unduly conservative. It only allows transmissions

with very high (close to 1) probability of success. Can we gain extra capacity by allowing transmissions with less than perfect success probability? Note that extra capacity could be substantial if there are many links in the transition region.[10] To address this question we need to develop new scheduling algorithms that can treat links as non-binary.

A comprehensive treatment of this topic is beyond the scope of this work, where we want to focus on measurements only. However, to make our observations stronger, we study here a simplified scheduling problem called "One Shot Scheduling" [43] and experiment with scheduling algorithms both for graded and thresholded physical interference models. The one shot scheduling problem picks a subset S of links to be scheduled from a given set L such that the aggregate throughput is maximized. We redefine throughput as 'expected throughput' as we are dealing with probabilistic transmission success. The one shot scheduling problem for the thresholded physical model is intractable [43]. But, for small size of L, it is computationally feasible to exhaustively look for the optimal subset S_{opt} to be scheduled. Any set of schedulable links has to be a matching. Thus, we can pre-select L as a matching. With a 20 node network $|L|$ is upper-bounded by 10. Thus, exhaustive search is feasible to obtain optimal schedules for both models. One needs to evaluate only 1024 possibilities.

The experiments are done on the indoor testbed as follows. First, we obtain the connectivity graph of the network. Since we are comparing only physical models, here we define network links as those with SNR greater than the the SINR threshold (5 dB). For each experiment, we pick a random matching L from the connectivity graph such that the $|L|$ is equal or close to 10. For each model, we estimate the throughput of each subset S of L and then choose the optimal subset

[10] We are sometimes using the expression "links in transition region" to mean links with SINR in the transition region.

S_{opt} which provides the maximum aggregate throughput. Note that the S_{opt} for the graded model can be different than S_{opt} for the thresholded model. Thus, for each experiment, we schedule both these subsets one by one in the testbed to find their respective throughputs (using a process similar to used in Section 3.5). The receiver for each link records the respective PRR and the base station collects this information at the end of the step and determines the aggregate throughput. We perform 50 such experiments with different random choice of L each time and each experiment is repeated 5 times to obtain an average throughput for each model as well as confidence intervals.

The results are shown in Figure 3.12. Throughput is expressed in terms of *average number of packets successfully transmitted per slot*. This is the Y-axis. The individual experiments (i.e., different choices of L) are shown on the X-axis, sorted in the order of increasing throughput for the graded model for visual clarity. For each experiment, the throughput for thresholded model is drawn as a bar graph on the left while the throughput for graded model is drawn on the right. 95% confidence intervals are shown using error bars. They are usually very small, particularly for the thresholded model. The model predictions are also shown.

It is easy to see that in 90% of the experiments, graded model gives higher throughput. Overall, the graded model got 3.14 packets/slot successfully transmitted per experiment while thresholded model got 2.45 packets/slot. This is an improvement of about 28%. However, the modeling error (difference between predicted and measured throughput) is significant for the graded model in about 30% of the cases, while this is true only for 10% of the cases for the thresholded model. This is expected as the graded model schedules links in the transition region that has relatively poor predictability. However, we do see that the thresholded model is not perfect either.

This simple one shot scheduling experiment shows the power of using graded

physical interference model instead of using the more conventional thresholded model for use in scheduling. We expect that the general observations here applies for other types of wireless networks also, and not just with low-power radio links.

3.7 Related Work

A recent paper by Brar et al. [22] can be considered complimentary to our work. Here, the authors investigate algorithms for physical model and show via simulations that physical interference modeling leads to more efficient schedules relative to the protocol 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, but in the context of low-power wireless links.

Researchers have only begun to study effect of interference in wireless networks using experimental methods. The authors in [131] 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 [107] 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. They also observed slightly different behaviors dependent on received power ranges. As described in Section 3.3, our conclusions are somewhat different with the newer radio platforms, and we have derived a more classical model [91].

In a different work [108], 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. In

a closely related work [128], the authors investigated the accuracy of range-based interference model by conducting experiments with Mica2 motes. They concluded that it is inaccurate and proposed a new protocol to detect run-time radio interference relation among nodes.

Experimental work has also considered 802.11-based systems to study interference behavior [54, 63, 80, 92]. The difference here is that the sender-side (carrier-sense) behavior in the MAC protocol must also be modeled. This phenomenon is absent in TDMA scheduling. Notable articles are as follows. Single and multiple interferer scenarios have been modeled in [92] and [54], respectively. The need for modeling 'graded' interference has been demonstrated in [82]. The need for modeling multiple interferers has been motivated in [34].

3.8 Conclusions

There are several contributions in this work. *First*, we develop and validate a purely measurement-based method to instantiate SINR-based physical interference model. *Second*, we compare the accuracy of different interference models via extensive experimentation on two different motes testbeds – low power, indoors and high power, outdoors. The general conclusion is that the physical interference model provides the best accuracy. But it is still far from being perfect (90-percentile error about 20-25%). Many commonly used models such as hop-based, range-based and protocol model have poorer accuracy. In case a pairwise model must be used, our experience indicates that the range-based model should be avoided. This provided the worst performance across the board, while the experience with other models varied. *Third*, we observe that while the thresholded physical interference model is used in existing scheduling algorithms, it is overly conservative and does not utilize links in the so-called transition region. We have shown the potential of scheduling

such links by directly using the graded physical model in a scheduling problem. We expect that this observation will generate new research in exploiting the graded nature of the physical model for better scheduling.

Appendix: Choosing Random Matching

Assume, M is the matching to be picked randomly. We first choose $|M|$, the size of M. Ideally this should be chosen randomly based on the probability of choosing matchings of different sizes. Since this is a hard problem we use the probability of choosing a matching of size $|M|$ 'approximated' to be $\frac{C(|E|,|M|)}{2^{|E|}}$, where $C(n,k)$ is "n choose k." Once the size is picked randomly based on this probability, a random matching is computed by simply selecting random links in sequence and by putting them in a set so long as the set remains a matching. For some sizes, the sequence in which links are chosen may not provide a matching of the desired size. In such cases, this trial is discarded and a new random trial is used.

Chapter 4

SINR-based Interference Modeling on Commodity WiFi Hardware

4.1 Introduction

TDMA-based transmission scheduling can potentially extract the optimal capacity from a wireless network. On the other hand, CSMA-based MAC protocols, such as used in 802.11, are known to have poor performance in heavy traffic situations. Several measurement studies have documented this in real world scenarios [53,95]. CSMA protocols are also not easily amenable to rigorous mathematical modeling for throughput and capacity [40]. The problem with 802.11 is expected to be worse in mesh networks, where high capacity backbone links are needed and multihop interference plays a significant role.

This makes a case for developing robust TDMA-based transmission scheduling on 802.11 hardware using the existing 802.11 PHY layer. The reason this is attractive is that 802.11 platforms are now commodity, and firmware modifications to alter the standard MAC layer are possible to obtain new MAC functionalities [78,90].

On the theoretical side, transmission scheduling algorithms and their performance bounds are well-investigated in literature [22, 45, 73]. Thus, a significant foundation is available to drive the above pursuit. However, practical questions remain. The algorithms are driven by an interference model. Consideration of unrealistic or inaccurate models hurts performance in two obvious ways. Overly aggressive schedules may be infeasible in reality. Overly conservative schedules may not have enough concurrent transmissions even when they are actually possible. Neither is good for high throughput. Our goal in this work to understand this interference modeling aspect for transmission scheduling. We take a measurement-driven approach using commodity 802.11 hardware.

It is important to note that developing a complete TDMA system on a stock implementation of 802.11 PHY layer involves many aspects other than interference modeling and scheduling. Tightly synchronized time slots must be designed that are small enough. Traffic loads on the links must be estimated. Schedule must be computed and distributed with an appropriate centralized or distributed scheme. Overheads for all these must be minimized. These are independently strong areas of research and are beyond our scope. Our work here focuses on two basic aspects in this entire problem space: (i) interference modeling using measurements, (ii) studying actual scheduling performance on a 802.11 mesh testbed using known algorithmic approaches.

There are several interference models that have been considered for transmission scheduling studies. They vary from oversimplified range-based models to fairly realistic SINR-based physical interference models [46]. Because of their realism, several recent studies have used physical models [21,22,45,73]. The general goal here has been to develop algorithms to maximize throughputs, given specific traffic loads on the network links. However, all such works take a theoretical view

of the problem, where they evaluate achievable capacity bounds or develop near-optimal algorithms. There is little knowledge in the community about practicality of such approaches. Our work is one of the first attempts to bridge this gap using a measurement-driven approach. We also address the modeling accuracy question and how it might impact scheduling performance.

Physical models are based on SINR (signal to interference plus noise ratio). The relationship between bit error rate (BER) and SINR is exploited and it is assumed that if the SINR is sufficiently high, more than a given SINR threshold β, the BER is negligible and thus packet transmission is successful with very high probability. Lower SINR is not used for scheduling transmissions. We call such models "thresholded." However, if we move away from the thresholded paradigm, lower SINRs can still be used for scheduling packet transmissions, albeit with non-negligible packet error rates. Depending on the application, this may require retransmissions from the link or upper layer. Thus, scheduling may directly use the actual BER vs SINR relation instead of thresholding it. We call this "graded" physical model. We show via real scheduling studies that graded models provide potential for higher throughputs, while only thresholded models have been considered in existing scheduling literature (e.g., [22]).

The rest of the paper is organized as follows. In Section 4.2, we describe the experimental setup of the 802.11a mesh network. In Section 4.3, we describe how concurrent transmissions are acheieved for the purpose of this study. Section 4.4 evaluates the accuracy of the two physical interference models. Section 4.5 investigates scheduling performance. Related work and conclusions are presented in Sections 4.6 and 4.7, respectively.

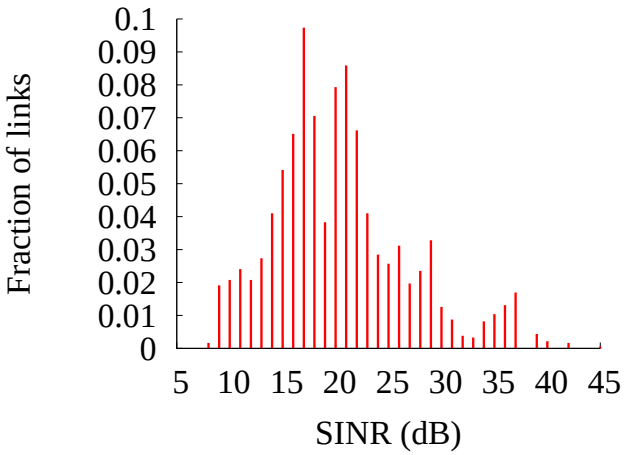


Figure 4.1. Distribution of SINR for the links in the chosen activation sets in the testbed setup.

4.2 Experimental Platform and Setup

The mesh network testbed consists of 11 'network' nodes. Each network node is monitored by a 'sniffer' node co-located with the network node, thus needing a total of 22 nodes in the testbed. The purpose of the sniffer will be explained momentarily. Each node is essentially a single-board computer (SBC), meant for embedded use, with an 802.11a/b/g interface. We use Soekris 4511 and 4826 boards [3] and Atheros 5213 chipset-based 802.11a/b/g mini-PCI cards manufactured by Winstron connected to a 5 dBi rubber-duck antenna. While the testbed uses two different types of boards, all 802.11 interfaces are identical. All experiments reported here are conducted in channel 44 of the 802.11a band using a single PHY layer rate 6 Mbps with transmit power of 11 dBm, and UDP packets with payload size 1470 bytes. These choices have been done with careful considerations to be discussed momentarily. The boards run pebble Linux with a mix of Linux 2.6.17 and 2.4.26 kernels with `madwifi` device driver version 0.9.4 for the 802.11 interface.

The 802.11 interfaces in the network nodes are set up in 'ad hoc' mode and the

sniffer nodes in 'monitor' mode. As mentioned before, each sniffer is co-located with the corresponding network node. The sniffer's purpose is simply to record all packet transmissions from its corresponding network node. The sniffer does this by simply running a packet capture tool like `tcpdump`. The radiotap header supportis used so that RF level information (e.g., received signal strength (RSS) and noise powers) as well as accurate timestamps can be collected. More on the timestamps in the next section.

The co-location eliminates possibilities of collision at the sniffer. This is because the sender's signal is much stronger than any possible interferer. Note that the sniffer could be avoided if all actual packet transmissions in the air could be recorded on a transmitting 802.11 interface with accurate timestamps. However, commodity 802.11 interfaces and device drivers do not facilitate this.[1]

4.2.1 Deployment Choices

We use 802.11a in preference to more widely used 802.11b/g to reduce external interference. We have verified that no other 802.11a transmissions exist in our testbed location. All experiments are done at quiet times with nobody around the testbed area. This is to avoid signal strength variations due to movement of people. The lowest possible PHY layer rate (6 Mbps) and a large packet size is chosen for the experiments. This is because, at higher rates or smaller packets, the sniffers cannot capture all packets in our low-cost embedded hardware, likely due to inefficiencies in interrupt processing. In our future work we plan to augment the experiments for all rates and other packet sizes.

[1]Tools such as `athstats` can be used to provide certain aggregated information about `madwifi` devices like packets sent/received, transmit/receive errors etc. But our work needs detailed per packet information.

Arbitrary pre-deployed topologies are not suitable for the goal of our measurement study. The reason for this is that for a good understanding of the modeling accuracy and scheduling success, the topology should be such that a range of SINRs (from low to high with many intermediate values) are possible in the network links. Otherwise, trivial conclusions are possible. For example, think of a network where the SINRs on most links for most choices of *activation sets* (i.e., set of links that are transmitting concurrently) are either too high or too low. Such networks can provide a very high degree of predictability in transmission success, and do not form interesting test cases.

We deployed the testbed in one part of our department building in an approximately 4000 sq ft area. The choice of transmit power (11 dBm) was influenced by the requirement of obtaining a range of SINRs for the links the activation sets we experiment with. The choice of activation sets will be discussed in Section 4.4.2. The distribution of SINRs is shown in Figure 4.1. We will see later that about 10-30 dB is the interesting range of SINRs. Beyond this, the link is either perfect or non-existent. In particular, the range 15-25 dB is the intermediate range, where the link works with intermediate, 20%–90%, packet reception rate. Such intermediate links are important realities in wireless networks. We note that our topology provides sufficient links within the range of interest.

All network and sniffer nodes are connected via Ethernet LAN to a central computer that acts as the control center. The central computer instructs the nodes to perform specific experiments, which are either transmitting or receiving/capturing packets as per the role of specific nodes. The received/captured packet traces are collected centrally over the Ethernet for later analysis.

Base transmit power (dBm)	Increase in transmit power (dB)	Avg. increase in RSS (dB)
0	17	16.545
5	12	12.152
7	10	9.827
9	8	8.053
11	6	5.744
13	4	3.953
15	2	1.980

Table 4.1. Average increase in RSS with increase in transmit power.

4.2.2 Measuring Signal Strengths

Signal strengths between nodes are needed for calculating the SINR of a link for an activation set. We use radiotap headers in the captured packets to measure RSS and the noise at the receiver. While this measurement is straightforward, it has one limitation. Radiotap headers can be obtained only when packets are actually received.[2] Thus, RSS for very weak links cannot be measured. However, these links may indeed generate enough interference.

We overcome this limitation by a power translation mechanism. The idea here is to measure the RSS on such links with a higher (say, by X dB) transmit power. Then the original unknown RSS would be X dB lower than the measured. This strategy would work so long as the radio is well-calibrated and the measurement noises are limited. Thus, there is a need for validation directly on the testbed. For validation, we (i) use a base transmit power, (ii) measure RSS on all links for this

[2]Some 802.11 chipsets such as Intersil's Prism [9] let sample specific registers on the interface card to measure RSS at any time, regardless of whether a packet is being received. However, in our knowledge the Atheros chipset we used does not have this facility.

power (of course, only on those where packets are received), then (iii) increase the transmit power by X dB to bring it to the maximum possible transmit power (17 dBm), and (iv) measure RSS on all the above links for this power again, and finally (v) correlate the increase in RSS on the links with X dB. The validation results are shown in Table 4.1. Note that average increase in RSS matches closely with X, typically within 5% in dBm values.

4.3 Achieving Concurrent Transmissions

We begin this section by first noting a subtle and important point. We do not 'explicitly' schedule concurrent transmissions in this work. Doing this with a very tight time synchronization on commodity 802.11 radios requires intricate engineering that is beyond the scope of this current paper.[3] Since our work is a measurement study only, we strictly do not need to do this explicitly. Instead, we achieve concurrent transmissions implicitly in the following fashion. See Figure 4.2 for an illustration.

i. Start a long sequence of back-to-back transmissions (with carrier-sense and backoffs disabled) on a chosen set of links. This is the activation set.

ii. All transmissions are captured at the co-located sniffer and timestamped using a synchronous time base. All successful packet receptions are also captured at the receiver of each link and timestamped similarly. As we will show in Section 4.3.2, the transmissions are somewhat jittery. Thus, some post-processing is needed.

iii The captured traces at the sniffers are post-processed to determine the actual sets of transmissions that overlap with a pre-defined degree of overlap (50%

[3]Research groups have achieved this only for course time synchronization [78, 85].

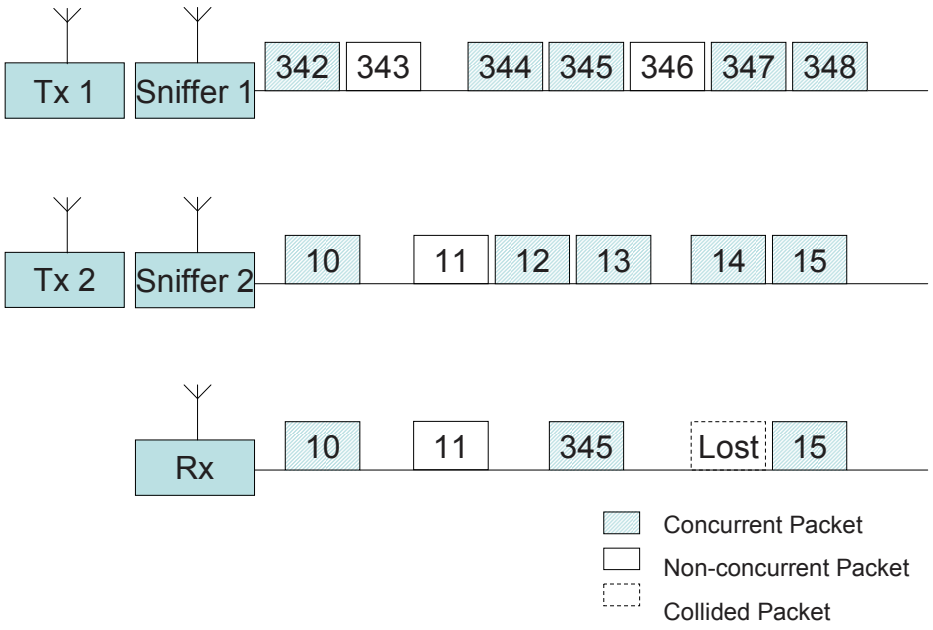


Figure 4.2. An example of how concurrent transmissions are implicitly achieved in our work shows two nodes transmitting simultaneously to a receiver. Sniffer co-located with the transmitters record their corresponding transmitter's packets. These packets are later analyzed to find which pair of packets is really concurrent by comparing their timestamps. An example timeline is shown, where back-to-back packets are transmitted with a slight jitter. Some packets undergo collisions at the receiver. Post-processing on the traces captured by sniffers 1 and 2 give us concurrent packets. Analysis of the receiver trace reveals which of these packets are received correctly and which are lost.

in our experiments). These sets of transmissions are deemed 'concurrent'[4]. All other transmissions are ignored.

iv. Success or failure of each transmission from each set of concurrent transmissions is noted from the receiver trace. 802.11 frame sequence number is used in identifying frames.

v. Finally, from the above data, the packet reception rate (PRR) statistics is computed over a large number of such sample concurrent transmissions for each link in the chosen activation set.

Several technical details are important to understand how the above is achieved. We discuss these in the following subsection. Then, we will describe some validation experiments to support our techniques.

4.3.1 Technical Details

a) *Obtaining synchronized clocks.* The 802.11 cards in the network nodes are set up in ad hoc mode. The time synchronization function (TSF) in 802.11 automatically synchronizes the clocks in the interface cards. To ensure that sniffers also utilize TSF, they are also made to run the ad hoc mode on a virtual interface, in addition to running the monitor mode on the main interface. Note that we do not disable beacon transmissions that are needed for clock synchronization using TSF. Microseconds resolution TSF timestamps are recorded in all captured frames in the sniffers or the receivers from the radiotap header. *Thus, all captured frames in the network have a common time base (TSF time).* This is useful for step iii above.

[4]For details regarding the capture phenomenon, look at the related work section for a discussion.

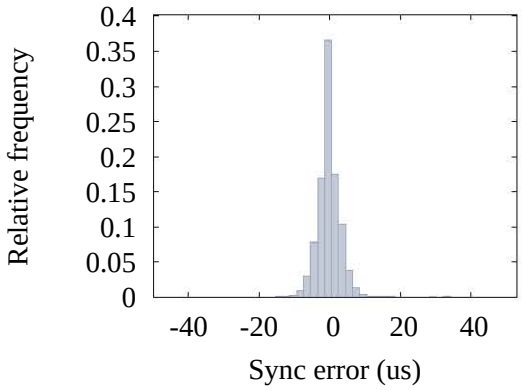


Figure 4.3. Distribution of the synchronization error (difference between the recorded timestamps at the sniffer and at the receivers for the same packet).

b) *Disabling carrier-sense.* To achieve this we used the antenna switching technique [30]. The 802.11 interface uses two antenna connectors for diversity. We have only one antenna connected to one connector, keeping the other connector unconnected. Using driver-level commands, any one of the connector can be selected as the receiving/transmit antenna. Selecting the unconnected antenna as the receiving antenna effectively disables carrier sense. This is useful for step i above.

c) *Disabling backoffs.* This is again achieved via driver-level commands by explicitly setting the initial backoff window size to zero and disabling retransmissions. Thus, there is only one transmission attempt with no backoff. The transmitted packets are MAC layer broadcasts; there are no ACKs.

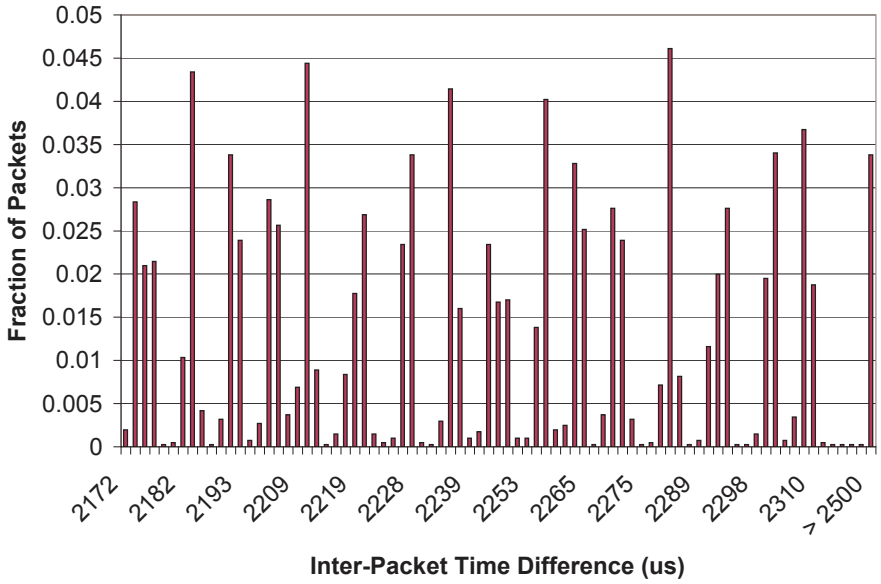
4.3.2 Experimental Validation

We have validated a), b) and c) above experimentally to ensure that they indeed work satisfactorily. To validate a), we essentially evaluate how good the TSF time

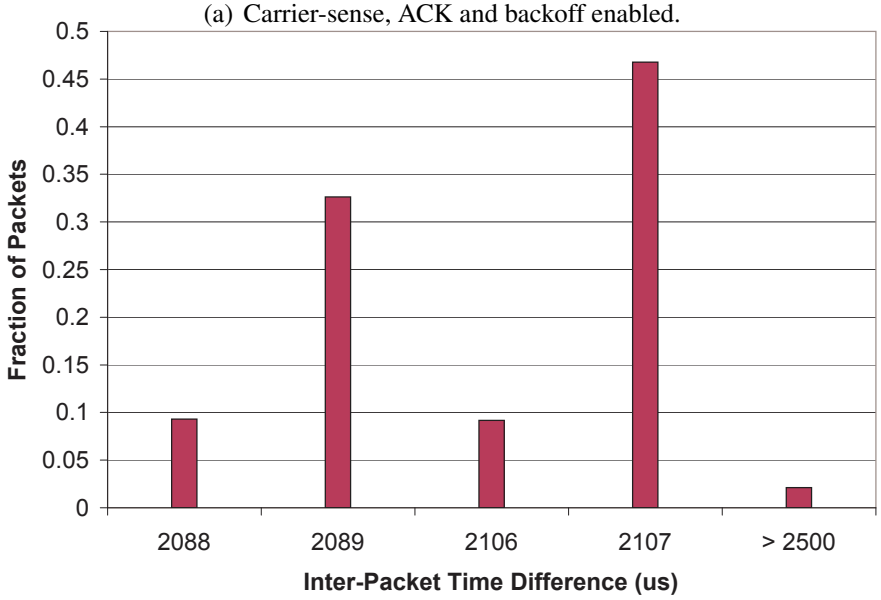
synchronization is on our testbed. To do this, we conduct the following experiment. Each node in the testbed is made to transmit alone and all the other nodes receive these transmissions. As usual, the transmitted frame is also captured at the sender's sniffer. Each packet is uniquely identified by the MAC address and frame sequence number in all captured traces. Each packet's receive timestamp on each receiver is compared to the same packet's timestamp on the sender's sniffer. Figure 4.3 shows the distribution of the synchronization error (difference between these timestamps). Note the sharp peak at zero. Median error is 2 μs for the absolute difference. Note that this time is very small compared to the average packet transmission time in our experiments (2089 μs).

We also validate b) and c), i.e., that the disabling of carrier-sense and backoff is working well. We perform an experiment where two network nodes concurrently transmit back-to-back UDP packets (size 1470 bytes) thus generating saturated loads. The nodes are kept within perfect carrier sense distance. The corresponding sniffers captures all transmissions from the senders. We observe that when carrier-sense and backoff are not disabled, the two senders share the medium and the aggregated transmit rate is 5.7Mbps, where one is 2.5 Mbps and the other is 3.2 Mbps (as per the captured packets at the sniffers). When carrier-sense and backoff are disabled, each sender transmit at 5.55 Mbps, almost doubling the aggregate sending rate.

We also analyze the inter-packet times in the above experiment from the 'merged' packet trace captured at the sniffers as another form of validation. Their distributions are shown in Figure 4.4. Note 16 distinct peaks of inter-packet times without disabling carrier sense, ACK and backoff (denoting 16 possible backoff values in 802.11a). When they are disabled, we get two peaks with one corresponding to the packet time. There are also two other smaller peaks. We attribute



(a) Carrier-sense, ACK and backoff enabled.



(b) Carrier-sense, ACK and backoff disabled.

Figure 4.4. Distribution of inter-packet times (difference between start-time of successive packets) at the sniffers for two scenarios. Note packet time is 2089μs.

the occurrence of these peaks to synchronization error, occasional beacon transmissions, hardware imperfections and unexplained software delays, or even occasional carrier-sensing directly from the antenna connector itself. Note also that in both cases a small fraction of packets have very high delay. These are clubbed together in the plot as their values take a very wide range. All these jitters are the reason why we do not rely on the scheduling system to be able to make perfect concurrent transmissions. This is also the reason post-processing is used to identify what sets of packets are really concurrent.

4.4 Building and Evaluating Physical Interference Model

The physical interference 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 ΣI is the sum of the interference powers experienced at the receiver caused by the group of interferers (transmitting concurrently), the model predicts the relationship between the bit error rate (BER) and SINR, where SINR = $\frac{S}{\Sigma I+N}$. This relationship depends on radio properties such as modulation. The packet error rate (PER) is directly related to BER and depends on coding. The packet reception rate (PRR), a quantity we will evaluate directly, is simply $1 - \text{PER}$ and thus again is directly related to SINR.

Typically, the PRR vs. SINR curve makes a sharp transition from low to high PRR values with increasing SINR. The rising part of the function has been described as the *transition region* in [131], albeit for a different radio technology.

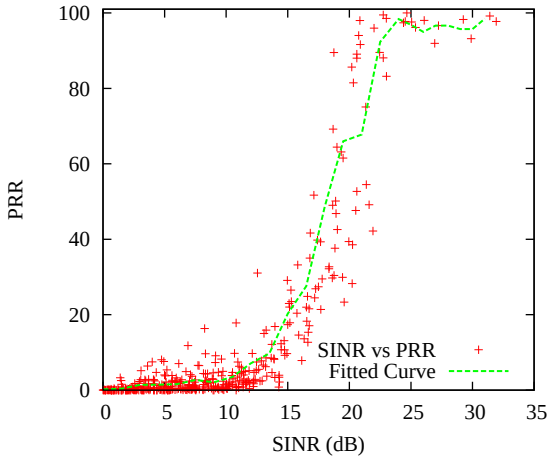


Figure 4.5. PRR vs. SINR relationship from measurement data.

Since scheduling applications need a 'binary' model (see, e.g., [22]), the curve is typically 'thresholded' and is described as a step function changing from 0 and 1 at a specific value of SINR, called the *SINR threshold* or *capture threshold*. This variant of the physical model is henceforth referred to as *thresholded* physical interference model. The original model will be called the *graded* physical interference model.

4.4.1 Model Building

To build the physical model, one needs to find the PRR vs. SINR relation. We do this empirically by simply taking many measurement samples of S, I and N for a separate three node setup – sender, receiver and interferer, thus directly computing SINR as $\frac{S}{I+N}$. PRR is measured by noting the fraction of packets received at the receiver when the sender and interferer transmit concurrently. The above three node measurements use identical hardware and measurement mechanism as described in Sections 4.2 and 4.3. This means that each node is accompanied by a sniffer, and concurrent transmissions are identified by postprocessing captured packet traces.

To derive a statistically meaningful relationship between PRR and SINR, many different samples of <PRR,SINR> are needed. We change the distances between transmitter-receiver and interferer-receiver pairs as well as change the transmit powers so that S and I can vary over a wide range. For each distances and transmit power setting (i.e., constant S and I), several thousands concurrent transmissions are used to compute PRR. This provides one <PRR,SINR> sample. The samples are shown in the scatterplot in Figure 4.5. We also show a fitted curve using a linear interpolation of average values in buckets of 1 dB each. This fitted curve provides the PRR vs. SINR model that will beused in our later analysis. This model can also be used directly by a scheduling algorithm. PRR is close to zero for SINRs less than about 10 dB and close to 100% for SINR more than about 25 dB. The transition region is thus quite wide. Note many intermediate PRRs are possible in the range of 15-22 dB. These considerations are important in choosing a topology for our study as discussed in Section 4.2. Note that measurements on large mesh testbeds have shown significant number of intermediate level links. See, e.g., evaluations on MIT roofnet in [12] or Rice University testbed in [26].

4.4.2 Model Evaluation

Our goal now is to evaluate the accuracy of the physical model for transmission scheduling. In particular, we will compare the two versions of physical model here – graded and thresholded, and also the thresholded model for different SINR thresholds. Since scheduling essentially determines 'feasible' activation sets (set of links) in each slot subject to an interference model, a model's responsibility is to describe which sets of links are feasible together and which are not.

Any scheduling algorithm must avoid the so-called *primary interference*, i.e., interference between links with a common endpoint in the network graph. Thus,

the algorithm must choose a *matching*[5] on the network graph as we are only considering unicast transmissions. Thus, instead of using random subset of links, we use random matchings. There is no point in evaluating non-matchings as they will never be scheduled by any algorithm. Choosing random matchings is intractable as well. We resort to a simple heuristics (not described here for brevity) to pick random matchings.

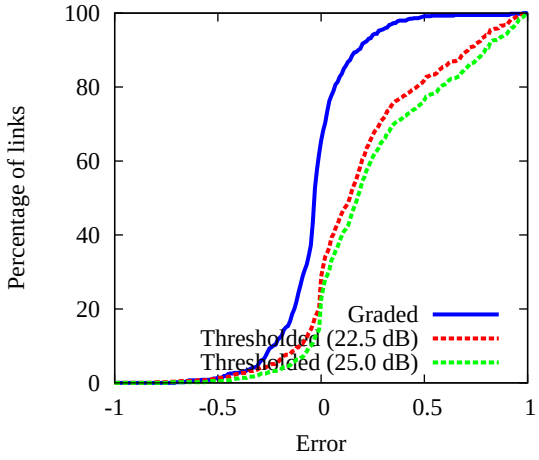
It is important to define which are actually links in our mesh testbed. We assume that for each ordered pair of nodes A, B, there is a network link from node A to node B if transmissions from A to B have 90% or more PRR in the absence of any interference. While we would have liked to use a higher PRR threshold, this also reduces the number of concurrently usable links. Note from Figure 4.5 that very high PRR (95% or beyond) is rare in the testbed.

566 random matchings are used providing a significant data set. Such a large data set also makes the study relatively independent of the topology used. This is because the data set includes many instances of links well distributed over the relevant range of SINR values. See Figure 3.2(a) again.

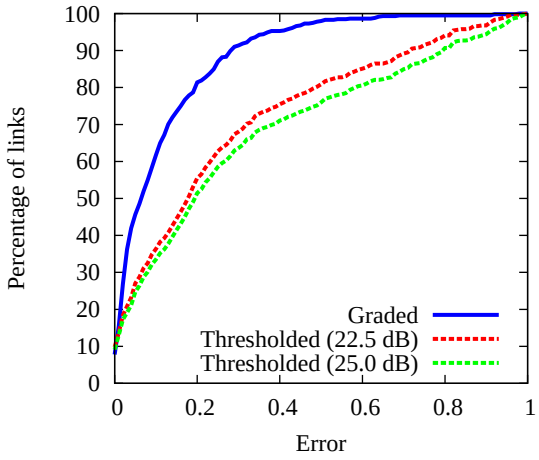
For each randomly selected matching, the actual throughput (normalized) of each link is evaluated when all links in the matching are transmitting concurrently in the testbed. The normalized throughput is simply the number of packets received on each link divided by the number of packets transmitted on this link. The method described earlier in Section 4.2 is used to find concurrent transmissions and throughput is calculated over only those packets which are actually concurrent. For each matching, about 3000 concurrent transmissions in the matching are used to calculate throughput.

Each random matching is used as input to a predictor that evaluates the link throughput predicted by each interference model. For the thresholded model, the

[5] A matching is a set of links such that no two links have a common end point.



(a) Actual error.



(b) Absolute error.

Figure 4.6. CDF of modeling errors for thresholded and graded interference models.

throughputs of all links in a matching that have above-threshold SINR are assumed to be 1. The rest are assumed to be 0. For the graded model, the throughputs are simply the PRRs as determined by the PRR vs. SINR relation (Section 4.4).

The modeling error is evaluated in the following fashion. Given a matching M_i consisting of $|M_i|$ links we denote the measured throughput for j-th link in this matching as $\Gamma_i^j(measured)$ and the predicted throughput by model k as $\Gamma_i^j(model)$. Then the modeling error with respect to the j-th link in the matching M_i is given by

$$error_i^j(model) = \Gamma_i^j(measured) - \Gamma_i^j(model). \tag{4.1}$$

The cumulative distribution function (CDF) of the modeling errors (Equation 4.1) is plotted in Figure 4.6. Errors for the graded and thresholded versions of the physical model are shown separately. Two different thresholds are used for the thresholded model 25 dB and 22.5 dB to demonstrate the impact of choice of threshold. Figure 4.6(a) shows the CDF of actual modeling error while Figure 4.6(b) shows the CDF of absolute error. Out of the evaluated links, 50 percent links show an absolute error below 0.05 for graded model and 0.2 for the thresholded model. The 80 percentile absolute error is in fact much lower for graded model (0.2) and very high for the thresholded model (0.5–0.6). This shows that the graded model is much more accurate than the thresholded model in predicting the accuracy of scheduling. There is a slight difference in choice of threshold too. Note, however, this issue is actually a problem for the thresholded model, as the right threshold to use is not easy to determine in a practical set up.

The CDF of actual error shows whether a model is biased towards under or over-estimation of expected throughput. In Figure 4.6(a), the thresholded model is clearly underestimating throughput – a sign of overly conservative modeling. This is expected as the thresholded model allows links to be scheduled only with high probability of success. The graded model, on the other hand, is relatively unbiased.

4.5 Evaluating Scheduling Performance

The previous section evaluated the accuracy of the two incarnations of the interference models in predicting the feasibility of a randomly chosen set of links. These evaluations only focus on modeling accuracy, but do not directly model real performances when used in a scheduling algorithm. This is because a scheduling algorithm considers only specific subsets of feasible links. This is entirely algorithm dependent. To gain some insight here, we now study the performances of the two physical interference models for making actual scheduling decisions.

To do this, however, we need to use scheduling algorithms. One concern here is that the two interference models behave quite differently. The thresholded model is binary, while the graded model is not. In our knowledge, all scheduling algorithms in literature deal with binary models and not with probabilistic models. However, using probabilistic models directly in scheduling has potential for improved throughput. This has been partially addressed by the model accuracy evaluation in the previous section, where it has been shown that the thresholded model can be overly conservative. It only allows transmissions with very high (close to 100%) probability of success. Can we gain extra capacity by allowing transmissions with less than perfect success probability? Note that extra capacity could be substantial if there are many links in the transition region. To address this question we need to develop new scheduling algorithms that can treat links as non-binary.

A comprehensive treatment of this topic is beyond the scope of this paper. Here, we want to focus on measurements only. To demonstrate the potential of the graded model in scheduling we use a two-part approach:

- In the first part, we study the two models using the *greedy scheduling algorithm*. Greedy scheduling has often been considered in literature – both for physical (thresholded) [22] and other simpler interference models [103, 121].

The goal there has been primarily to investigate its optimality properties. We extend greedy scheduling to the graded model; though its performance bound remains unknown.

- In the second part, we use optimal scheduling using both models. Since optimal algorithms for either model for the general scheduling problem are open questions, we use a simplified scheduling problem (*one-shot scheduling* [43]) here. The advantage here is that exhaustive searches are possible to determine the optimal for small networks such as ours.

The following two subsections describe these two parts respectively.

4.5.1 Scheduling Using Greedy Algorithm

We use the same greedy scheduling algorithm for both models. It is straightforward to implement and performance bounds are known for specific models, including the thresholded version of physical model [18, 22, 103]. The link demand vector is an input to the algorithm. The demand for a link is simply the number of packets (each packet takes one slot to transmit) to be scheduled on the link. The schedule is a sequence of slots with a feasible set of links to be scheduled in each slot such that demands of all links are satisfied. This is sometimes called the *evacuation model*. We describe the greedy algorithm below for a binary interference model (thresholded physical model in our case). We will describe later how it is modified to run under the graded model.

Input: Network graph $G = (V, E)$, demand vector on the links $f = (f_1, \ldots, f_{|E|})$ and the interference model. The interference model specifies which set of links (activation sets) are 'feasible' together.

Output: Schedule $S = \{S_1, S_2, \ldots, S_\tau\}$, where S_k is a feasible set of links scheduled in the same slot. τ is the schedule length.

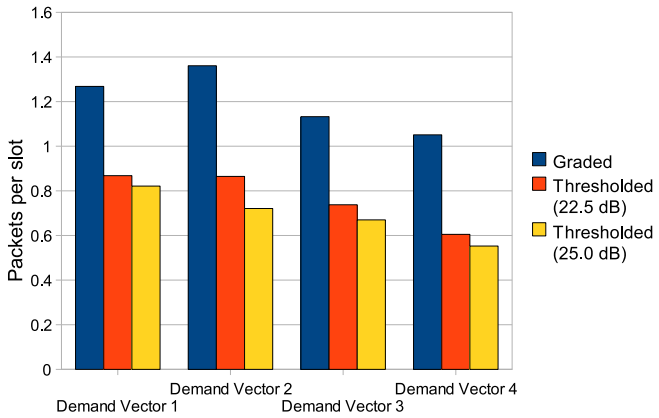


Figure 4.7. Results of greedy scheduling showing measured aggregate throughput for thresholded and graded physical models for different link demand vectors.

Algorithm:

1. Order and rename links such that $f_1 \geq f_2 \geq f_3 \ldots \geq f_{|E|}$.
2. Set $i = 1$, $S = \phi$, $\tau = 0$. (Initial schedule is empty.)
3. Schedule link i in the very first available slot where it can be scheduled interference-free according to the given interference model. If no such slot of feasible, increment τ and schedule the link in the last slot. (Incrementing τ is equivalent to creating a new empty slot at the end of the current schedule.)
4. Repeat step 3 above f_i times.
5. Increment i. Go back to step 3 until $i > |E|$.

For non-binary models such as the graded model, the algorithm is modified as follows. There is no real notion of feasibility now. Any set of links can be scheduled together, providing probabilistic packet deliveries on the constituent links. Thus, we use the notion of *expected throughput* in a slot to design the greedy algorithm. The expected throughput is the sum of the PRRs in all scheduled links per the PRR vs. SINR relation defining the interference model. In the greedy choice step above

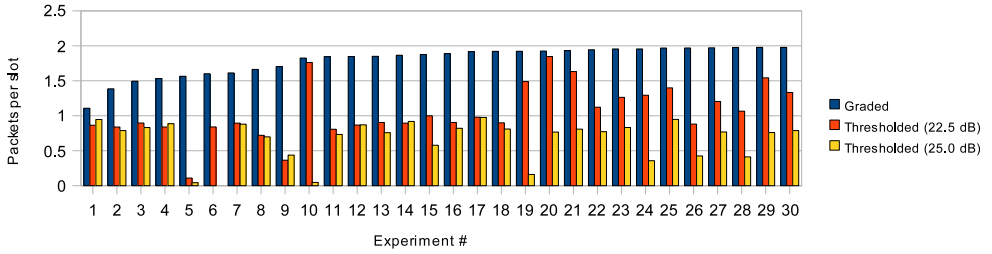


Figure 4.8. Results of the One Shot Scheduling experiment comparing the thresholded and graded physical models.

(step 3), the link i is scheduled in the first available slot so that its addition to that slot does not decrease the expected throughput in that slot.

The two models are compared in the following fashion. Four different demand vectors are considered for experiments, 1 through 4. The links are split into two equal sets randomly. One set has one packet each. The other set has i packets each for vector i. The models generate different schedules for a given demand vector. The schedules generated by each model are evaluated using scheduling experiments on the testbed. Each slot in the schedule is an activation set and concurrent transmissions are scheduled and identified in the same way as described in Section 4.3 by sending 3000 back-to-back packets transmitted on all links in the activation set concurrently. As outlined in that section, only actual concurrent packets are used for calculating the reception rate at the receiver on a link. This gives us the throughput on each link in a slot. All slots are evaluated for throughput in the same manner. Throughput is measured for each slot in number of packets per slot and is averaged for the entire schedule. The results are presented in Figure 4.7.

The results show that graded model consistently gives higher throughput than the thresholded models (for either threshold). On an average, the percentage improvement of graded model against thresholded model with SINR threshold of 22.5 dB is 56% and with SINR threshold of 25.0 dB is 74%. This demonstrates the potential of using the graded model directly in scheduling.

4.5.2 One Shot Scheduling

One limitation of the above study is that while the same greedy scheduling is used for both models, it remains inconclusive regarding why the graded model is performing better. Is it due to a better modeling accuracy or due to a better algorithm? While the performance bound for the greedy algorithm for the thresholded model is known, it is quite loose [22]. Also, the bound for the graded is unknown. Thus, we do not have any useful tool to answer this question.

So, in order to make our observations stronger, we study a simplified scheduling problem called "One Shot Scheduling" [43], where we can reasonably implement optimal algorithms for both models. Thus, only modeling accuracy will play any role.

The one shot scheduling problem picks a subset S of links to be scheduled from a given set L such that the aggregate throughput is maximized. We redefine throughput as 'expected throughput' as we are dealing with probabilistic transmission success. The one shot scheduling problem for the thresholded physical model has been shown to be intractable [43]. But, for small size of L, it is computationally feasible to exhaustively look for the optimal subset S_{opt} to be scheduled. Any set of schedulable links has to be a matching. Thus, we can pre-select L as a matching. With a 11 node network $|L|$ is upper-bounded by 5. Thus, exhaustive search is feasible to obtain optimal schedules for both models. One needs to evaluate only 31 possibilities.

The experiments are done as follows. First, we obtain the connectivity graph of the network. We again define network links as those with PRR greater than 90% in absence of any interference. For each experiment, we pick a random matching L from the connectivity graph such that the $|L|$ is equal or close to 5. For each model, we estimate the throughput of each subset S of L and then choose the optimal subset S_{opt} which provides the maximum aggregate throughput. Note that the S_{opt}

for the graded model can be different than S_{opt} for the thresholded model. Thus, for each experiment, we schedule both these subsets one by one in the testbed to find their respective throughputs. The same method as used earlier in greedy scheduling (described in Section 4.3) is used to schedule concurrent transmissions. We also determine the aggregate throughput for each subset using the same method as used before. We perform 84 such experiments with different random choice of L each time.

Due to lack of space, results for randomly chosen 30 experiments (out of the 84) are shown in Figure 4.8. Throughput is expressed in terms of *average number of packets successfully transmitted per slot*. This is the Y-axis. The individual experiments (i.e., different choices of L) are shown on the X-axis, sorted in the order of increasing throughput for the graded model for visual clarity. For each experiment, the throughput for graded model is drawn as a bar graph on the left and the throughputs for thresholded models are drawn as a bar graphs on its left.

In all experiments, graded model gives higher throughput. Overall (for all 84 experiments), the graded model improves throughput per slot by 77% over the thresholded model with threshold of 22.5 dB and by about 146% over the thresholded model with threshold of 25.0 dB. Note again choosing the right threshold is tricky with extensive experiments.

This simple one shot scheduling experiment again establishes the power of using graded physical interference model instead of using the more conventional thresholded model for use in scheduling.

4.6 Related Work

Interference modeling using the physical model or promoting the use of such models have been the topic of several recent 802.11-based empirical modeling and evaluation work. However, these are based on the default CSMA/CA MAC protocol of 802.11 and part of their effort goes into modeling the carrier-sense aspects and how transmission capacity is shared. In [52], authors investigated the impact of carrier sensing. In [83], Padhye *et. al.* developed a measurement-based methodology to characterize link interference in 802.11 networks. They pointed out that interference between links is not "binary" in practice. In [34], the authors showed that pairwise interference modeling is often not accurate and multiple interferers must be accounted for. Techniques for generating interference maps have been developed in [81]. Measurement-based modeling to evaluate 802.11 link capacities have been used in [54,87,92]. We also use measurement-based modeling, albeit for a different goal.

Several papers also studied capture effects on 802.11 with different modulations (802.11b and 802.11a) and chipsets [29, 58, 64, 96]. Experience with the capture model varied somewhat depending on the actual technology used. This varied from scenarios, where the second packet out of two overlapping packets cannot be captured if it arrives after the preamble period of the first packet, to cases where it can be captured when the MIM (message-in-message) mode [64, 96] is implemented. Note that we consider 50% overlapped of packets as concurrent and do not give any special consideration for first or second packet. The reason for this is that independent analysis revealed no statistical difference of the capture behavior for the first or second packet, likely because MIM is implemented in the Atheros chipset we use.

TDMA-based MAC on commodity 802.11 hardware has been a topic of several

investigations [30, 78, 85, 90]. None of these, however, directly address the interference modeling question. However, they do address many implementation issues that are perfectly complimentary to our goal. We note that synchronization achieved by software-level implementations in these papers are not tight enough to have one packet per slot at the data rate we use. Roughly, slot sizes in the order of 10ms have been achieved, while our packet time is about 2ms in this paper. However, future firmware level developments are expected to address these limitations.

Moving over to non-802.11 platforms, evaluations similar to this paper are available. Concurrent transmissions on low-power sensor motes have been studied in [107, 118, 131]. Our recent works [67, 68] on motes platforms also studied interference modeling aspects. Some of the approaches used in the current paper are similar to those we used in [67]. Note that TDMA-based scheduling in mote-class radios such as 802.15.4 is easier to achieve as the radios are better documented than 802.11 radios and data rates are much slower.

4.7 Conclusions and Future Directions

Our work makes the following contributions. First, we evaluate the accuracy of physical interference modeling on an 802.11a testbed. We show that the thresholded model commonly used for scheduling has much poorer accuracy relative to the graded model that we promote for scheduling use. The graded model is not perfect, however. The 80 percentile error is 0.2 (normalized to maximum link throughput 1 packet/slot). Second, we show using two types of scheduling experiments that the accuracy question hurts the performance of the thresholded model badly. The graded model achieves a better throughout by a factor of roughly 2. This stems primarily from the overly conservative behavior of the thresholded model that schedules perfect links only. Our recommendation for future work is thus to embrace

the reality and investigate scheduling algorithms that exploit the graded model for creating high-performance mesh backbones using 802.11.

A careful reader will note that a physical layer rate reduction would improve PRR for a given SINR. One can argue that the thresholded model can always be used if the rate can be adjusted to obtain a high enough PRR for the available SINR. While this is true, standardized protocols and hardwares allow rate adjustments with only a very course granularity (e.g., only a few available rates in 802.11). Also, many mesh networks may use long distance links [85] and may thus operate at the lowest possible rate. Thus, our approach of scheduling imperfect links is still very useful. A comprehensive study of rate control along with scheduling imperfect links is also an important direction for future work.

Chapter 5

Adaptive Multichannel Protocols for High-speed Wireless Networks

5.1 Introduction

At the beginning of this decade, many regulatory authorities worldwide (including FCC in US) set aside a large swath of spectrum (7 GHz wide) in the 60 GHz band for unlicensed use. This has promoted a large number of innovations in developing very high data rate (1 Gbps or above) wireless link technologies for the local area. The general goal is to provide 'wired' equivalent performance – roughly equivalent to gigabit Ethernet that is now widely available. There have been several developments around this goal. For instance, the upcoming WirelessHD standard [8] can use up to 4 Gbps links over short distances (10m) for high-definition audio/video applications. The link technology actually allows up to 25 Gbps. The upcoming 802.11VHT standard [2] (VHT stands for very high throughput) is expected to provide at least 1 Gbps data rate and is expected to go over much longer distances.

As one goes for high data rates as above, the efficiency of the MAC protocol

reduces. This happens due to the following reason. The per-packet MAC protocol overhead can be broadly classified in two parts – *bandwidth independent* and *bandwidth dependent* [113, 123]. The bandwidth independent part slowly becomes substantial as one improves the physical layer data rate. For the same packet length in bits, the transmission time of the packet and the bandwidth dependent overhead reduce proportionately with physical layer data rate; however, the bandwidth independent part stays the same. We will show later that the efficiency of a 802.11 like MAC protocol can easily be limited to only 50% with just 5 nodes if the data rate exceeds 1 Gbps. While this issue has apparently been noticed in simulation exercises in the 802.11VHT group [2], the research community is yet to undertake the challenge of developing efficient random access protocols for the very high speed regime. This issue is not limited to CSMA or 802.11-like protocols alone and can happen for TDMA protocols as well. However, we will limit discussions for CSMA only for its clear suitability for data networks.

The goal of our work is using *adaptive channelization* as a mechanism to improve MAC protocol efficiency in the very high speed regime. The basic idea is to split the available single-channel bandwidth into multiple smaller channels. Each individual channel now has a smaller bandwidth supporting a proportionately slower data rate. This helps mask the bandwidth independent overhead. We will demonstrate this with analysis in Section 5.3. However, such channel splitting carries its own overhead as guard bands must be used. Thus, the number of channels must be chosen appropriately to strike a balance of different overheads. Also, such channelization must be adaptive to the traffic demand. For example, a smaller (larger) number of channels may be appropriate when a small (large) number of nodes are active or when traffic demand is low (high). The challenge is to adapt the system appropriately to ensure an optimum operating point at all times.

In some ways, adaptive channelization that we propose is reminiscent of the

subcarrier allocation problem in OFDMA [125]. However, in OFDMA a centralized entity (base station) maps the available set of OFDM subcarriers into a set of sub-channels to be allocated to the active links. The mapping is done based on channel state information and traffic on the links to improve the overall spectral efficiency and is renewed periodically in a TDM fashion. In contrast, our goal is to develop an entirely distributed random access model agnostic to the physical layer. The goal is to optimize channel access efficiency in presence of significant bandwidth independent overhead.

Our main contributions are as follows. First, we show via analytical modeling that single channel MAC protocol is very inefficient in high data rate networks and channelization can provide the necessary improvement (Sections 5.3 and 5.4). Second, we develop an adaptive channelization protocol and show via simulations that just channelization is not enough for better performance; channelization also needs to be adapted with varying traffic conditions (Sections 5.5, 5.6, 5.7). Finally, via a 'scaled down' prototype implementation on GNU Radio/USRP platform, we emulate the operation of a high-speed network and show such adaptive channelization can indeed be realized in practice (Sections 5.8 and 9). Related works and conclusions appear in Sections 2 and 10 respectively.

5.2 Related Works

Several approaches dynamically allocate variable amount of spectrum in cognitive radio based networks. For example, the KNOWS system [126] develops distributed allocation techniques for contiguous time-spectrum blocks to maximize the use of fragmented spectrum [127]. Spectrum is allocated based on pre-determined traffic demands, interference criterion, and bandwidth allocated to interfering transmissions. Similar spectrum allocation problems are also considered in [109], but in the

context of cellular dynamic spectrum access networks, and are solved in centralized fashion. Several market-driven auction mechanisms [129, 130] have also been proposed that dynamically auction variable amount spectrum to wireless nodes based on demands and bids.

Independent of cognitive radio or dynamic spectrum access, several works in current literature study adapting channel width dynamically to improve different performance measures. In [28], the authors make a case for adapting the channel width in wireless networks using 802.11 networks as a case study. However, they primarily study the impact of adapting channel width on data rate, power consumption and communication range on a single wireless link. In [74], the authors study a spectrum distribution problem in the context of 802.11 WLANs by providing wider channels to the more congested APs and smaller channels to less congested ones. The approach is aimed more towards load balancing than addressing MAC protocol overheads and can serve as complimentary to our approach. Similar problem is also solved in the FLEX system [122].

There is a rich literature on multichannel CSMA-based wireless MAC protocols. Here, the channelization is already fixed, and the broad goal is to develop CSMA MAC protocols to efficiently utilize multiple channels. A sampling of major works in this space is as follows. In MMAC protocol [106], the authors augment the 802.11 MAC protocol such that all nodes meet at a common channel periodically to negotiate the channels to use for transmission in the next phase. In SSCH [17], the authors propose dynamic switching of channels using pseudo-random sequences. The idea is to randomly switch channels such that the neighboring nodes meet periodically at a common channel to communicate. In DCA [120], the authors use two radios – one for the control packets (RTS/CTS packets) and another for data packets. The channel to send the data packet is negotiated using the control packets,

and the data packets are sent in the negotiated channels. Similar control channel-based MAC protocols have been used in [51]. In AMCP [104], the authors uses a similar notion of a control channel, but a single radio and focus on starvation mitigation. An asynchronous control channel based MAC protocol is proposed in [15] that solves multi-channel coordination problems and assigns channels to links using a channel assignment algorithm. In [66], two different multi-channel protocols, xRDT and LCM-MAC, are proposed that do not require separate control channels.

The fact that channelization can improve MAC protocol efficiency was observed originally in [70] in the context of Ethernet. The authors in [113, 123] noticed the impact of bandwidth independent overheads on MAC protocol efficiency.

There is also a significant amount of literature on channel assignments for multi-radio, multichannel networks (e.g., [89]), as well as channel selection for TDMA scheduling (e.g., [14, 59]). They are orthogonal to our work and we do not discuss them here.

5.3 Case for Channelization

We illustrate the advantage of channelization by a simple example. Suppose, two transmitters are sharing a channel of bandwidth B. See Figure 5.1. Assume that each packet transmission carries a bandwidth independent overhead of Δ time units and the packet transmission (including any bandwidth dependent overhead) takes Ttime units. Splitting the channel into two subchannels of bandwidth $B/2$ each (ignore the guard band question for now) makes the transmission time increase to $2T$ keeping the bandwidth independent overhead part the same. Four packet transmissions from the two nodes now takes $4T + 2\Delta$ time as opposed to $4T + 4\Delta$ time in single channel. The speedup will obviously be substantial if Δ/T is large. An added advantage is that the number of contenders in each channel could reduce

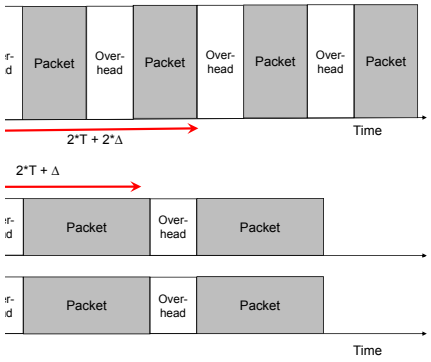


Figure 5.1. Demonstrating the benefit of channelization.

leading to further speedups.

In an 802.11-like CSMA/CA protocol, the bandwidth independent overhead is the backoff time that is counted in terms of slots. The slot size must be at least the sum of maximum propagation time between two nodes and the carrier sense interval. If there is a non-negligible time synchronization error, it must be accounted for in the slot size as well. Thus, the slot size has a lower bound that is independent of data rates. The analysis in the following subsection shows the impact of a constant slot size on MAC efficiency at high data rates.

5.3.1 Why Does Single Channel Work Poorly?

We start with the widely used model of 802.11 developed by Bianchi in [19]. It assumes a single collision domain (single hop network) and ideal channel conditions (perfect carrier sensing and no capture) with the network under saturated load. According to this model, the probability τ that a node transmits in an arbitrarily chosen time slot is given by,

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)}, \tag{5.1}$$

where p is the 'constant and independent' probability that a packet collides when transmitted regardless of the number of retransmissions suffered, also called as the *conditional collision probability*, W is the size of the minimum contention window and m is the maximum backoff stage such that the maximum contention window is $2^m W$. The 'constant and independent' probability assumption is a key approximation in this model.

By definition,

$$p = 1 - (1 - \tau)^{n-1}. \tag{5.2}$$

[19] shows that the above two equations have a unique solution in the two unknowns τ and p and they can be solved by numerical techniques.

These expressions pave the way to derive expressions for normalized throughput S. Assume as in [19], P_{tr} is the probability that there is at least one transmission in a slot. Since n nodes contend on the channel, and each transmits with probability τ in a slot,

$$P_{tr} = 1 - (1 - \tau)^n. \tag{5.3}$$

The probability P_s that a transmission occurring on the channel is successful is given by the probability that exactly one node transmits on the channel, conditioned on the fact that at least one node transmits, i.e.,

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}. \tag{5.4}$$

Now, assume that the slot time is σ and the packet time is T_p. Ignore all interframe spacings and header overheads. Consider the basic access with no RTS/CTS or ACK. (These can be added but they generate distracting details.) By a straightforward application of renewal theory, the normalized throughput or the long run fraction of time spent in successful transmissions is given by,

$$S = \frac{P_{tr} P_s T_p}{(1 - P_{tr})\sigma + P_{tr} T_p}. \tag{5.5}$$

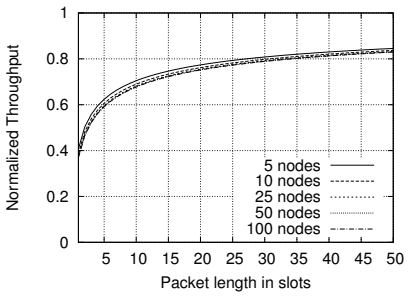


Figure 5.2. Normalized throughput versus packet time (in slots) for a single channel 802.11-like network. Optimal contention window is assumed.

Note that the packet time in slots (T_p/σ) is an influential determinant of throughput. Smaller values mean lower throughput. It is easily seen that for higher physical layer speeds T_p/σ will tend to be smaller. This is because slot size σ has a lower bound that is independent of speed, as we discussed before. For example, the propagation time of RF signals at 150m is 0.5μs. Carrier sense interval can easily add another 0.5μs, assuming that the fastest available A/D chips can digitize 512 Msamples/sec and 256 samples are used to sense carrier. Thus 1μs can serve as a lower bound on the slot size. For a 1000 byte packet, the packet time is 8μs for a 1GBps link ,and 0.8μs for a 10Gbps link. This gives us (T_p/σ) = 8 and 0.8, respectively. Note that these numbers are very conservative. We have assumed a larger than average packet size.

We plot throughput versus packet time for various number of nodes in Figure 5.2 as per Equation 5.5. For this plot, the minimum contention window size W is assumed to be optimal for the number of nodes and the optimal value is used to generate this plot. The optimal is computed via straightforward numerical techniques by computing the throughput for different values of W and choosing the optimal for presenting in the plot. For unoptimized W (such as in 802.11) the performance is

likely to be worse. Still, we note very poor throughput for a very realistic range of packet times in our context. Packet times 1-5 slots mean efficiency between 0.4-0.6 even with an optimized window.[1]

Of course, efficiency improves with larger packet sizes. But packet sizes cannot be increased arbitrarily as packet error rates will increase for a given bit error rate in the underlying wireless channel, for a given SINR, modulation and coding. Also, from a more practical point of view packet coalescing to increase packet size may not be possible depending on packet generation/forwarding rates from the upper layer.

5.3.2 Modeling Multichannel Benefit

To see the benefit of splitting the channel up into multiple subchannels, assume that the given channel is divided into k smaller channels of the equal bandwidth. The above model can now be modified to compute the resulting throughput. Assume for simplicity that transmitters choose channels randomly for transmission and then contend on that chosen channel. Thus, on average there are now n/k nodes competing in each channel. The transmission probability, P_{tr}, becomes

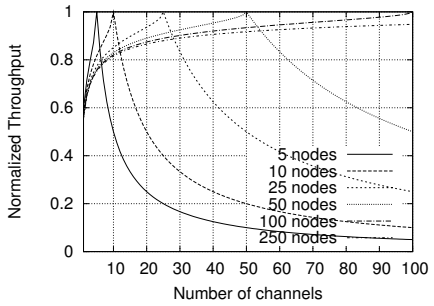
$$P_{tr}(k) = 1 - (1 - \tau)^{n/k}. \tag{5.6}$$

The successful transmission probability, P_s, becomes

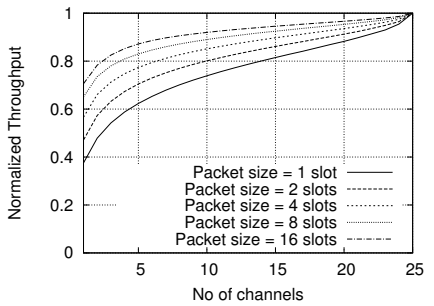
$$P_s(k) = \frac{(n/k)\tau(1-\tau)^{(n/k)-1}}{P_{tr}(k)}. \tag{5.7}$$

The packet time T_p is now longer as each channel now has a factor k smaller bandwidth. Thus, the packet time is kT_p, and the normalized throughput $S(k)$ of each

[1] In Equation 5.5 similar performance with different number of nodes is not surprising. It is an artifact of the use of optimal window. Figure 9 in [19] also has a similar observation.



(a) Packet time = 4 slots for single channel.



(b) Number of nodes = 25.

Figure 5.3. Normalized throughput of a 802.11-like network in a multichannel setting. Single collision domain and implicit ACK are assumed. Optimal contention window (for number of nodes per channel) is assumed for a fair comparison.

channel is given by,

$$S(k) = \frac{P_{tr}(k)\ P_s(k)\ kT_p}{(1 - P_{tr}(k))\sigma + P_{tr}(k)\ kT_p}. \tag{5.8}$$

All channels being identical, the aggregated normalized throughput in all k channels in also $S(k)$.

5.3.3 Results

In Figure 5.3 we present the throughput $S(k)$ as a function of the number of channels k, packet time in slots (T_p/σ) and number of nodes n. As before the range

of packet time in slots has been been chosen carefully to reflect the realistic values possible in high data rate networks. We have also been careful with the choice of minimum contention window size W. *To clearly demonstrate the multichannel advantage we have used the optimal W for each choice of n and k pairs.* While the optimal may not be achievable in a real protocol or it may require complex estimation or adaptation that may be expensive [124], from the perspective of the analytical modeling this makes the fairest demonstration of the performance benefit of multichannel. To see this, assume that the single channel case is already sub-optimal because of a poor choice of W (assume, smaller than optimal). Now when we split the channel but not modify the contention window, we may get closer to the optimal as contention reduces due to channel splitting. Thus, it will be unclear how much benefit is due to channel splitting and how much due to a more suitable contention window with split channel as opposed to single channel. If we choose the optimal in all circumstances, the comparison is clearer.

In Figure 5.3 we have plotted the normalized throughput $S(k)$ as per Equation 5.8 versus number of channels k. In Figure 5.3(a), the packet time is fixed at 4 slots and the number of nodes is varied. The throughput reaches optimal (100%) when the number of channels is equal to the number of nodes, as there is no contention and the optimum window size is 0. On the left of the optimal point, there are more nodes and less channels, thus throughput suffers due to contention. On the right of the optimal point, there are more channels and less nodes, so throughput suffers as many channels remain unused.

Note very poor efficiency for single channel even with an optimum contention window (about 0.56). In Figure 5.3(b), the number of nodes is constant at 50 and the packet time and the number of channels are varied. Note again that for small packet time, the single channel efficiency is very poor. For example, for small packets (1 slot), it is about 0.38. However, efficiency rapidly increases with increase in

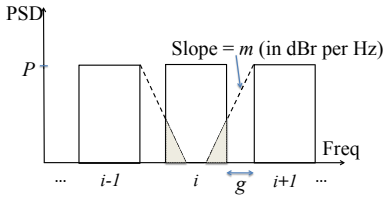


Figure 5.4. Channels with guard band.

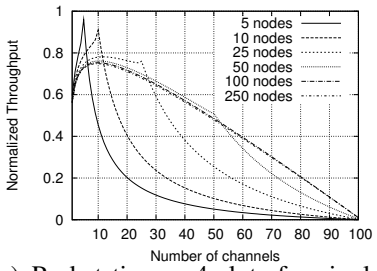
number of channels. For example, the efficiency increases by 50% with just 5 channels and doubles with 20 channels. The rate of increase in efficiency tapers off with larger number of channels. This means that just a handful of channels can make a significant performance impact.

5.4 Guard Bands

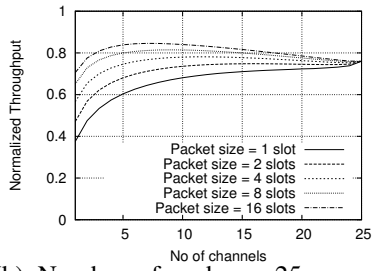
Channelization, however, comes with an overhead. When we divide a spectrum of bandwidth B in to k channels each of width b, there should be enough guard band separation between the channels, so that concurrent transmissions are possible on on each channel without interference. In practice, due to the non-linearity of power amplifiers, radio leakage occurs on each channel. The amount of this leakage determines the guard band separation needed between two adjacent channels. In Figure 5.4, we show three channels separated by guard bands of width g. A simple analysis below determines the guard band so that we can reevaluate the multichannel advantage assuming the existence of guard bands.

Assume that regardless of the number of channels to use, the total power budget remains the same. This means that the power spectral density or PSD, P, remains same for all channels in any channelization.[2] Assume also that leakage follows

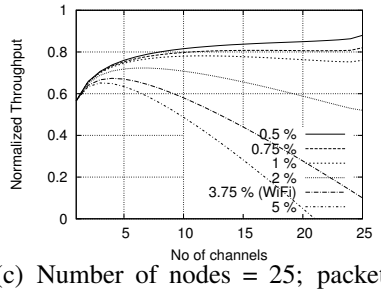
[2]It could be slightly higher than P, in fact as much as $PB/(B-(k-1)g)$, assuming no power



(a) Packet time = 4 slots for single channel; guard band 1% of channel bandwidth.



(b) Number of nodes = 25; guard band 1% of channel bandwidth.



(c) Number of nodes = 25; packet time = 4 slots for single channel.

Figure 5.5. Normalized throughput for various number of nodes, number of channels and packet sizes, showing that an optimal number of channels exists.

a straight-line decay pattern. Thus the amount of out of band leakage will be a function of the PSD, P and the slope m of the decay.[3] Now, the width of each channel, when there are k channels is given by, $b = (B - (k-1)g)/k$.

One way to design guard bands would be to ensure that the SINR in each channel i is greater than the receive threshold β, where the interference comes solely due to leakage from the adjacent channels. Ignoring path loss, the signal power in channel i is given by Pb and the interference due to radio leakage from adjacent

is wasted in the guard bands. But we ignore these intricacies here.

[3]This is not unlike standards based spectral masks(e.g., in 802.11a/g where a dBr loss per MHz is specified beyond the channel bandwidth).

channels is given by (the area of the two shaded triangles shown in the Figure 5.4),

$$m\left(\frac{P}{m}-g\right)^2 \tag{5.9}$$

The SINR in channel i must be atleast equal to the SINR threshold β. Thus,

$$\frac{Pb}{N_0 b + m\left(\frac{P}{m}-g\right)^2} = \beta \tag{5.10}$$

Equation 5.10 has simple solutions when $(P/m - g)$ is zero, b or a fraction of b. The solution at $(P/m - g)$ equal to zero refers to the case when the leakage from adjacent channels do not affect channel i at all, and the interference (given by Equation 5.9) from adjacent channel is zero. This is the upper bound on g. Since this bound is constant and all the other solutions of g are too specific to a scenario, we will use a constant guard band formulation to numerically evaluate the multichannel advantage in presence of guard bands.

Throughput $S(k)$ with guard bands can be computed by using Equation 5.8 except that now the packet time is slightly different. It is no longer kT_p, but $\frac{k \cdot B}{B-(k-1)g}T_p$.

In Figure 5.5, we show plots similar to Figure 5.3 except that a constant guard band (1% of the channel bandwidth B) is used throughout. An additional plot with varying guard band width is also presented. Note that the guard band indeed makes an impact on the performance with larger number of channels. For example, even with 1% guard band, 24% bandwidth is wasted for 25 channels. Unless guard band is too small, both too few and too many channels hurt performance. Thus, depending on the actual packet sizes the optimal number of channels are typically small (more in the order of 10 rather than 100). Note also from Figure 5.5(a) that the channel efficiency never reaches 100% due to the guard band wastage.

Note that, some radio technologies like OFDM [16] does not require guard bands between adjacent channels. In such cases, channelization will lead to much better performance benefit as shown in Section 5.3.

5.5 Adaptive Multichannel MAC Protocols: Background

5.5.1 Need for Adaptation

The preceding analysis demonstrates that channelization is effective in improving efficiency for high data rate wireless networks. The analysis shows that an optimum number of channels exists depending on the operational parameters. If there is no guard band wastage, the optimum number is equal to the number of contending nodes. The optimum number is smaller when guard band is non-zero. The actual number depends on parameters such as guard band size and packet size. Thus a mechanism to adapt the number of channels with the number of contending nodes is useful. We will describe simple protocols for channel adaptation in Section 5.6. In spirit, the idea of channel adaptation is somewhat similar to the adaptation of contention window in CSMA protocols [124]. It is a hard problem as it requires sophisticated estimation mechanisms to estimate the number of contenders. Fortunately, some degree of success has been reported in literature [20, 25]. In the simplest form these techniques track the collision probability p by continuously measuring idle times, successful transmit times, busy times and unsuccessful transmit times. They use a moving average-based method to give importance to more recent measurements. Finally, using the estimated value of p, the equations 5.1 and 5.2 are solved to determine the number of nodes n. We will use a similar methodology in estimating the number of contending nodes when evaluating our protocol in Section 5.7.

In its most general form, the channelization does not need to be uniform. Different channels can be of different widths. This could be useful in certain scenarios. For example, if the packet sizes are not uniform then clearly links transmitting smaller packets should use smaller channels and vice versa. However, such optimizations in a dynamic scenario can be complex. We will use only uniform channelization in this work.

5.5.2 Protocol Design Background

As discussed in Section 5.2, there is a host of multichannel MAC protocols in current literature. But they are all geared for networks with pre-configured or fixed channelization (such as in IEEE 802.11). Such networks have the operating frequency band divided into a predefined set of channels and the multichannel protocols try to optimally use all the channels in the network. Protocols like SSCH [17], and xRDT [66] specifically fall into this category, and thus do not satisfy our requirement. Negotiation-based protocols such DCA [120], MMAC [106], LCM-MAC [66] that utilize an out of band mechanism (e.g., a control channel or a synchronized time period for control) to negotiate the channel to be used for transmission can be used, but need to be suitably modified to support channel adaptation. Also, one fundamental issue these protocols suffer from is a bottleneck in their out of band mechanism (control channel or control period) as they do per-packet or per-cycle negotiations which may lead to a significant amount of control traffic in certain high traffic scenarios. Larger control channel, for example, can alleviate this issue. But this can waste control channel bandwidth for some other traffic conditions. (Note that in our model the control channel must be curved out of the provided channel bandwidth B). The control channel bottleneck problem has been studied in several contexts, and the right size of the control channel has been analyzed [120]. However, this requires adapting the control channel bandwidth with

traffic demonstrating another need for adaptation.

We propose an Adaptive Multi-Channel (AMC) MAC protocol that mimic our modeling of multichannel operation in the previous sections. In particular, the protocols have the following properties:

1. The available frequency band B is adaptively divided into a near-optimal number of channels. To enable this the radio interface on each node is capable of changing center frequencies and channel bandwidths on the fly, within the given frequency band.

2. Each node has only one physical radio interface for data packets. This makes interface costs reasonable. However, we do assume a high degree of capability on the part of interface (see below). The node also has a control interface and a low bandwidth control channel in the simplest implementation of the protocol. However, as will be discussed in Section 5.6.3 these are not strictly required.

3. The data interface has a very general reception ability. Let us refer to the ordered set of tuples <center frequency, bandwidth> of each channel as a channel configuration. Assuming that the maximum possible split is k-way, there are k possible channel configurations, with $1, \ldots, k$ channels. Thus, there can be $k(k+1)/2$ possible channels in the system. We assume that the interface is able to receive on all possible $k(k+1)/2$ channels at the same time.[4] This ability (i) removes the need for informing the receiver about the channel to be used before transmission, and thus eliminates a significant control

[4]In principle, this can be achieved using a wide-band RF front-end that can tune to the entire bandwidth and have parallel software paths that decode signals on each of the defined channels. In order to achieve high data rates, each of the software paths can be assigned to high speed parallel processors [44, 65].

overhead; (ii) allows for correct packet reception always (barring collisions and channel errors) by reducing the deafness and multichannel hidden terminal problems [66];[5] and (iii) allows different nodes to have different channel configurations for transmissions and still packets may be received correctly.

4. The interface is half-duplex. It can either transmit or receive at one time. When transmitting, it can only transmit in one channel.

5. The onus of channel selection is purely on the sender. For the purpose of our study here, we assume a simple scheme where the sender selects a channel randomly and then contends and transmits only on that channel. This happens for each packet transmission.

6. When not transmitting, the interface always listens on all channels and keeps a population estimate on a continuous basis. As indicated before, techniques available in literature to build such population estimates can be used [20, 25].

5.6 Adaptive Multi-Channel Protocol: Operation

In this paper we will only study a distributed protocol that works in a single collision domain (single cell). While single collision domain is a limitation of our current study, our goal in this work is to explore the opportunities in channelization for high data rate networks – rather than promoting specific protocols, scenarios or architectures.

The simplest version of the Adaptive Multichannel (AMC) protocol that we will study here uses a low bandwidth control channel and a separate control interface. The protocol uses minimal control traffic to communicate certain status information

[5]These problems arise in several multi-channel protocools because nodes can listen to only one channel at any instant [66].

(see below). We assume that the control channel traffic does not interfere with the data channels and that its propagation characteristics are at least as good as the data channel. As will be discussed in Section 5.6.3, both the control channel and control interface are not strictly necessary.

5.6.1 Protocol Operation

The protocol is centered on two basic operations: 'split' and 'merge.' Given a k channel configuration, the 'split' operation moves the system to the $k+1$ channel configuration. The 'merge' operation does the opposite. Since the available bandwidth B is known and channels are all equal, knowledge of k is sufficient for the nodes to learn what channels are in use for communication.

The split and merge operations are implemented by broadcast SPLIT and MERGE control packets in the control channel. These broadcasts can be initiated by any node when the optimal channelization according to the current population estimate (aggregate in all channels) does not match with the current channelization. If the population estimate is n, then the analysis in Sections 5.3 and 5.4 can provide the corresponding optimal number of channels, $k^*(n)$. This becomes the threshold for 'merge' and 'split' operations. If the current channelization, k, is less than $k^*(n)$, then the node should 'split'; if k is greater than $k^*(n)$, the node should 'merge'.

The SPLIT and MERGE packets ensure that all nodes can keep track of the current number of channels used. To provide a degree of fault tolerance against lost control packets, each node also periodically broadcasts the current number of channels (according to its own information) through BEACON packets on the control channel. Upon receiving such BEACON packets a node changes its understanding of the current channel configuration to the minimum of its own information and the value contained in the BEACON. This minimizes the 'period of vulnerability'

only to the interval until the next successful BEACON reception. Note, however, while during this period different nodes can use different channel configurations for transmission, packet reception is still possible because of the assumption that nodes can receive in any channel in all possible configurations. Also see the discussion on multicell operation in Section 5.6.3.

Use of randomness can prevent synchronous behavior and thrashing. For example, more than one node can otherwise broadcast SPLIT messages almost back to back based on the same estimate causing the channels to be split more than necessary only to merge back momentarily. Much of these are matters of details and can be fine-tuned for a given architecture.

5.6.2 Discussions

Two important multichannel-related performance issues do not impact the AMC protocol.

Control Channel Bottleneck: As observed in prior work [120], the control channel has the potential to become a bottleneck when used in multichannel operations. While this is true for negotiation-based protocols like DCA [120], bSMART [127] etc, AMC does not send explicit control messages for negotiation or coordination between senders and receivers for transmissions. Control packets are used only for merging, splitting and beaconing actions that are not frequent.

Deafness and Multichannel Hidden Terminal: Deafness occurs in a multichannel protocol when a sender does not know the state of its receiver (transmitting or ready to receive) [66]. In AMC, all nodes are always listening to all channels (except when actually transmitting). This helps address deafness. Multichannel hidden terminal arises when a node switches its channel to transmit on another channel, but it does not know the 'state' of that new channel, i.e., possible ongoing transmissions in

the interference neighborhood [66]. But in AMC, a node transmits only in one channel and always knows the state of all channels (except for periods when it is transmitting).

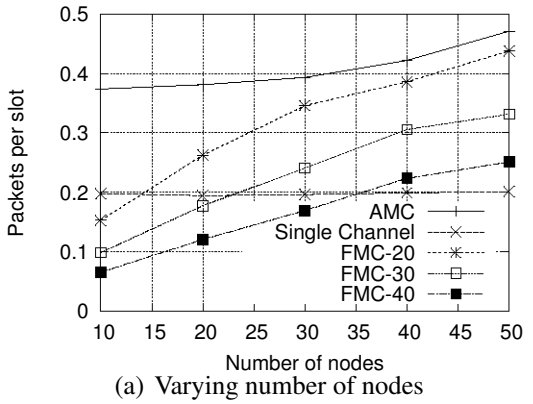
5.6.3 Improvements and Extensions

Several alternative approaches are possible around the same basic idea presented above.

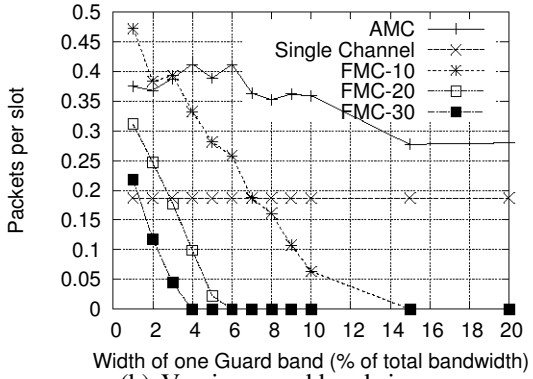
No Control Channel: Instead of sending the SPLIT, MERGE and BEACON packets in a separate control channel, a node can send them in one or more data channels. The upside of this approach is that control channel and the extra control interface are not needed. The downside is that nodes which are busy transmitting will miss these packets (due to the half-duplex assumption). However, since the BEACON is periodic, the 'period of vulnerability' is bounded.

No Control Messages: The control messages (SPLIT, MERGE etc.) are not strictly needed either. All the operations performed by nodes using the control messages can be done by sensing the spectrum alone. The intuition behind this is that the frequency domain representation of the signal in the operating band B should look quite different with different channel configurations. Thus, straightforward 'spectrum sensing' over the entire bandwidth B can reveal the channel configuration in use.

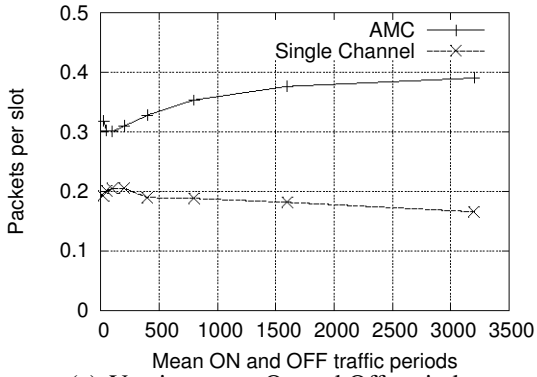
Multicell or Multiple Collision Domain Operation: We have studied the AMC protocol only for a single collision domain operation. For multiple collision domains – either for single- or multi-hop operation – it is possible that different parts of the network use different channel configurations for optimal throughput. Thus, it is possible that some nodes might 'see' different channel configurations being used by different nodes in the neighborhood. This is not necessarily a problem if nodes are



(a) Varying number of nodes



(b) Varying guard band size



(c) Varying mean On and Off periods

Figure 5.6. Simulation comparison of the AMC protocol with fixed multichannel (FMC) and single channel protocols (25 nodes, packet time = 1 slot, guard band width (g) = 1% of total bandwidth B).

assumed capable of receiving packets simultaneously in all possible channel configurations as indicated before in Section 5.5. It is, however, possible that overlapped channels are created due to different channel configurations used by different nodes. This may introduce unintended interference. A control channel based protocol with suitable optimizations can alleviate this effect and lead the system to its optimal performance goal. We leave this design as an open question and a topic of our future work.

5.7 Simulation Results

We have developed a slotted-time discrete event simulator to simulate the AMC and fixed channel multichannel protocols. While the analysis in Sections 5.3 and 5.4 shows the theoretical limits of the possible improvement due to channelization, the simulation results can show the real improvements possible when adaptation overheads are also taken into account. We compare our AMC protocol to a simple fixed multichannel (FMC) and single channel CSMA protocols. The fixed multichannel protocol is similar in most aspects to our AMC protocol except that it does not adaptively change the channelization, and all nodes channelize the spectrum into the same number of channels. A comparison with fixed multichannel protocol helps illustrate the need for adaptive channelization with varying traffic. The FMC protocol has a parameter k, which denotes the number of channels.

We use an 'on-off' traffic model for each node indicating bursty (on) periods alternating with silence (off) periods. The periods are exponentially distributed with chosen means which are set equal in the all results reported here. Note that the analysis in Sections 5.3 and 5.4 has used saturated traffic. Thus, the simulation results always do not directly correspond to the analysis results.

A single collision domain is assumed and no channel error is modeled. Every node generates packets of constant sizes for its neighbors during its on period. Exponential backoff mechanism is used by every node for control as well as data transmissions, with a maximum of 6 backoff stages (i.e., $m = 6$). The minimum contention window W for each channel is optimized for the population estimate of that channel (as discussed in Section 5.5). A simple table (pre-computed) lookup achieves this. We vary the number of nodes, guard band width, and the mean on and off periods. The following parameters are used whenever otherwise not specified: packet time in slots (T_p/σ) is 1, the guard band size is 1% of the single channel bandwidth B, both mean on and off periods are 1000 slots and the number of nodes are 25.

Figure 5.6 shows the aggregate throughput in packets/slot from fairly long simulation runs with varying parameters. FMC-k denotes the fixed multichannel protocol with k channels. Evidently, AMC beats any FMC protocol or the single channel protocol almost always. On average, the improvement over single channel is about 100%. While FMC protocols tend to do better than single channel, they are almost always poorer than AMC.

In Figure 5.6(a) FMC protocols have higher throughput with increasing number of nodes because of the increase in effective offered load. AMC also has higher throughputs with increasing number of nodes, but offers relatively stable behavior. Single channel performance does not change as it is always under saturation. In Figure 5.6(b), we note that with larger guard bands AMC is even more preferable. FMC protocols can offer very poor performance if the guard band wastage is large. AMC is always able to choose the appropriate channelization for the best throughput.

In Figure 5.6(c) exposes the weakness of the AMC protocol in that it relies on estimation of number of nodes and adapts relatively poorly when on-off periods are

small while the aggregate offered load is the same. The performance differential is seen to be almost 25%. However, its performance relative to single channel remains high.

5.8 Software Radio Implementation

In this section, we demonstrate the advantage to be gained from adaptive channelization using a prototype implementation on a 6 node software radio-based network. We use the GNURadio/USRP platform [4, 7]. While this platform is by no means high data rate, our 'scaled-down in speed' implementation still demonstrates the following.

(i) We show that in the USRP/GNURadio platform the slot time must be in the order of tens of milliseconds for an effective implementation of a CSMA protocol. This means that with the highest feasible data rate in this platform (1Mbps), packet time in slots (T_p/σ) is quite small even for reasonably large packets (e.g., in the order of a few KBs or smaller). This opens up the possibility of a performance boost via channelization.

(ii) We show that adaptive channelization is feasible and effective on this platform with some careful engineering.

In the following we describe the relevant details of the platform, our design choices for channelization, and experimental results.

5.8.1 Prototype Platform

GNURadio [4] is an open source software development platform that provides several signal processing blocks necessary to implement software defined radios using

low-cost RF hardware and general purpose computers. The Universal Software Radio Peripheral (USRP) [7] is the most commonly used RF hardware along with GNURadio. The USRP motherboard has 4 high-speed analog to digital converters (ADCs), each at 12 bits per sample and 64 MSps. There are also 4 high-speed digital to analog converters (DACs), each at 14 bits per sample and 128 MSps. In principle, it allows an effective receive bandwidth of up to 32 MHz and and transmit bandwidth of up to 50 MHz. These 4 input and 4 output channels are connected to an FPGA. The FPGA, in turn, connects to a USB2 interface and thereby to a host computer. The baseband samples are transferred between the the USRP motherboard and the host computer using the USB2 interface. Each sample that is sent to and received by the USRP is 16 bits. The USB2 interface can support a data rate of up to 480 Mbps nominal, or 250-300 Mbps effective. Due to this limitation, the maximum rate at which the samples can be sent to or received from USRP is limited to 8 MSps of complex samples (16-bit I and Q samples).

Daughter boards implementing the RF front end can be plugged in on the motherboard. Daughter boards have direct access to the ADC and DAC converters. For our prototype implementation, we have used the RFX2400 daughter board [7] that operates in the 2.3 – 2.9 GHz band, though the methods described below are general for any frequency band.

5.8.2 Fine-grained Channelization

When the RF front-end in USRP is set to operate in receive mode, it continuously senses a 20 MHz chunk (limitation of RFX2400 daughter board) of spectrum centered at any specified center frequency. The received time domain analog signal is digitized into I/Q samples at a fixed rate of 64 MSps by the ADC and sent to the FPGA. In the FPGA, the digitized I/Q samples are down-converted from intermediate frequency (IF) to the baseband and then *decimated* according to the required

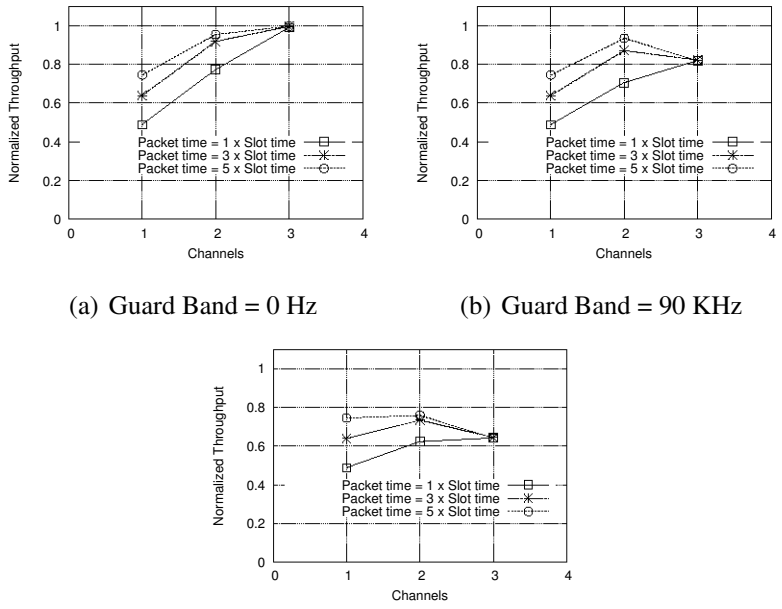
channel width and sent to the host machine through the USB2 interface. In the decimation process, the baseband I/Q samples (at 64 MSps) are sent through a low pass filter and then through a down-sampler to filter out samples that are outside the frequency band of interest. The decimation rate is software controllable from the host machine. In order to filter out samples outside a band of F_{rx} Hz around the center frequency, we use the following formula:

$$F_{rx} = f_{ADC} \cdot \frac{E \cdot b_S}{D \cdot M}, \tag{5.11}$$

where f_{ADC} is the sampling rate of ADC, D is the decimation rate, M is the number of samples that make a symbol, b_S is the number of bits in each symbol and E is the spectral efficiency[6] of the modulation scheme used. For example, assuming GMSK modulation (that we use in our study) with a spectral efficiency of 1 bps/Hz, 1 bit per symbol and 2 samples per symbol, a decimation rate of 16 filters out samples that are outside a frequency band of 2 MHz (± 1 MHz around the center frequency). In the USRP, the decimation rate can be changed from 4 to 256 (in multiple of 2) thus allowing us to tune to channels of width in the range of 62.5 KHz – 8 MHz assuming the other parameters as above.

When the USRP is in transmit mode, the data to be transmitted are modulated into baseband I/Q samples using any specific modulations scheme in the host machine and sent to the FPGA. The I/Q samples are *interpolated* at a specified rate in the FPGA depending on the frequency band they need to occupy and then sent to the DAC to be converted into analog signals. The DAC in USRP operates at a constant rate of 128 MSps. The interpolation rate is software controllable and controls the channel width occupied by any transmission. In order to restrict a transmission within a frequency band of F_{tx} Hz, we use the following formula similar to the one

[6]Spectral efficiency (bps/Hz) refers to the amount of information that can be transmitted over a given bandwidth in a specific communication system.



(a) Guard Band = 0 Hz (b) Guard Band = 90 KHz

(c) Guard Band = 180 KHz

Figure 5.7. Throughput vs. number of channels and packet size for different guard bands.

used in receive mode:

$$F_{tx} = f_{DAC} \cdot \frac{E \cdot b_S}{I \cdot M}, \tag{5.12}$$

where f_{DAC} is the rate at which the DAC operates and I is the interpolation rate. For example, again assuming GMSK modulation, 1 bit per symbol and 2 samples per symbol, an interpolation rate of 32 restricts the transmission to a frequency band of 2 MHz around the center frequency. In the USRP, the interpolation rate can be changed from 4 to 512 (in multiples of 2), thus allowing us to send data in channels of width in the range of 62.5 KHz – 16 MHz, assuming other parameters as above.

5.8.3 CSMA Protocol Implementation

In our implementation, carrier sensing is done in software on the host computer using raw I and Q samples from the USRP. The I and Q samples are magnitude-squared and a moving average of I^2+Q^2 is compared with a carrier sense threshold to detect the presence of carrier in any given channel. Carrier sense threshold is tuned for each channel used to provide a 100% accuracy in carrier sensing in our testbed. Using this mechanism, we developed an 802.11-like CSMA based MAC protocol without RTS/CTS and ACK.

The time domain I and Q samples from the ADC need to be transferred to the host machine from USRP through the USB interface on the receive side, and vice versa on the sender side. This transfer delay imposes restrictions on carrier sensing. In the GNURadio software, packets are sent to the host computer through USB only when there are a sufficient amount of data collected in the USRP buffer. This delay is dependent of decimation rate in USRP (which determines the channel width). USB block size and number of USB blocks in the buffer also introduce delays. These delays present 'blind spots' for carrier sensing [37, 101], when a potential interferer is transmitting, however a potential sender cannot sense the carrier. The slot time must be carefully chosen so that it exceeds the 'blind spot' delay. In our implementation, we use channel widths varying from 200 KHz to 1 MHz and our measurements show corresponding delay range from 30 ms to 8 ms. We choose a slot time of 32 ms – slightly higher than this maximum time to ensure that samples are available for carrier sensing decision. Authors in [101] have also experienced similar delays with identical USRP hardware.

Backoffs are counted in slots. The minimum contention window size is always chosen as the optimal using a table look-up following the model in Section 5.3. Exponential backoffs are used as before.

Limitations

Before we go forward we caution the reader about the limitation of the experiments: (i) The number of contending nodes are either statically fixed or told by an oracle in our experiments. Population estimation procedure are not implemented. (ii) The Ethernet interface serves as the control channel. It is relatively fast and effectively error-free. (iii) Receivers are told by the oracle when and which channel to receive on. Thus, any cost due to multichannel reception is not evaluated. However, any overhead of channel switching on the sender or receiver is captured. (iv) Of course, the slot time (32ms) is several orders of magnitude larger than 802.11a/b (9μs and 20μs, respectively), which in turn again orders of magnitude larger than the our target slot size (approximately of 1μs). However, this is purely due to the hardware limitation. Thus, our experiments should be viewed as 'scaled-down in speed,' but still with a real limitation of slot size, albeit due to a different artifact.

5.9 Experimental Evaluation

The 6 GNU Radio/USRP nodes in our testbed are grouped into 3 sender-receiver pairs. The nodes are deployed in such a way that for any given channel configuration, (i) a single collision domain is created on every channel, but (ii) there is no adjacent channel interference even with a zero guard band. This topology gives us an opportunity to evaluate the benefit of channelization for different guard band widths. Due to processing limitations in the host machine used in our setup, the maximum usable bandwidth B without any underrun or overrun in USRP is 1 MHz. We choose the center frequency at 2.5 GHz to avoid interference from other wireless devices operating in the unlicensed 2.4 GHz band. Three different channel configurations are used by dividing the 1 MHz channel into 1, 2 and 3 subchannels. (Further division is meaningless as our testbed can have at most 3 transmitters.)

The actual width of the channels (b) depend on the guard band size (g) to be used. As noted before, GMSK modulation is used with spectral efficiency of 1 bps/Hz providing 1 Mbps nominal throughput in the entire band.

In the benchmarking experiments reported below, each sender transmits back-to-back UDP packets (indicating saturated load) for 60 seconds and the throughput is measured at the corresponding receiver. The throughput is normalized to the nominal channel bit rate of 1 Mbps for presentation. Repeating the experiments at different times showed little variation and thus confidence intervals are not shown.

5.9.1 Fixed Channelization

Here we study the benefit of fixed channelization in different emulated high speed networks by varying the packet length. As before, packet length is presented in terms of packet time (when using single channel) counted in slot time, i.e., T_p/σ. Figure 5.7 shows the aggregated normalized throughput when all three senders transmit simultaneously using 1, 2 or 3 channels. With 1 channel, all senders are on the same channel. With 2 channels, one sender is on one channel and the other 2 are on the other channel. With 3 channels, they are all on separate channels. We show results for 0 Hz guard band, a moderate size guard-band of 90 KHz and a large guard band of 180 KHz.

For each channel configuration, throughput improves as expected with increase in packet length. For the zero guard band case, the 3 channel case shows almost 100% channel efficiency as each of the transmitters occupy independent channels. Since the contention windows are optimized based on the number of contending nodes, there is zero overhead leading to a very high efficiency. For any specific packet length, there is a sizable improvement in channel efficiency due to channelization. With zero guard band, note an improvement from 48% to 76% from 1 to 2 channels and about 100% when using 3 channels for the smallest (1 slot) packets.

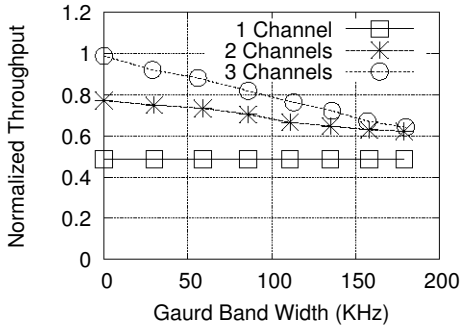


Figure 5.8. Impact of guard band width on channelization. (Packet time = 1 slot time.)

When longer packets the reduction is smaller. With zero guard band, there is monotonic increase in throughput when more channels are used. Figure 5.7(b) and 5.7(c) show however that the rate of improvement decreases with guard band size and finally throughput goes down with more channels when guard band wastage become significant. The experience here qualitatively follows the modeling experience in Sections 5.3 and 5.4.

Next, we study the effect of varying the guard band width on channelizations. Here, we use packet time equal to slot time in all cases. Figure 5.8 shows the throughput for the 1, 2 and 3 channel configurations with varying guard bands. Note that the decrease is more rapid in the 3 channel case compared to 2 channel case as more spectrum is wasted due to the guard bands.

5.9.2 Adaptive Channelization

Now we study the benefits of adaptive channelization compared to fixed channelization as well as using a single channel. We have implemented the AMC protocol using the Ethernet as the control channel. Due to the limitation of the current hardware, the receivers tune to only one channel where the sender will transmit, as told by an oracle. Due to the above limitation, population size is also not estimated, but

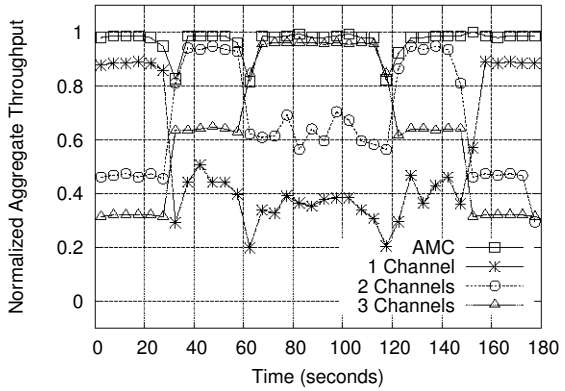


Figure 5.9. Benfits of adaptive channelization compared to fixed channelization. (Packet time = 1 slot time. Zero guard band.)

the nodes learn about the number of contenders via an oracle. These oracles are implemented by a combination of scripting and broadcast communication on the Ethernet.

We use specific traffic pattern to demonstrate the power of adaptive channelization. The first sender starts transmitting at 0s and ends at 180s. Similarly, the second transmitter transmits from 30s to 150s and the third transmitter transmits from 60s to 120s. When transmitting, all senders transmit back-to-back UDP packets as before. Note the above pattern means that traffic ramps up at intervals of 30s, from 1 to 2 to 3 senders, and then ramps down similarly again. Figure 5.9 shows the aggregate normalized throughput computed every 5s along a timeline, when using three different fixed channel configurations and also using the AMC protocol. AMC almost always gives close to 100% throughput as nodes try to occupy individual channels leading to zero backoff (recall again contention window is optimized) and no bandwidth independent overhead. There is indeed some degradation in throughput when channels are split and merged (at 30s,60s,120s and 150s). This happens exactly when the number of contenders change and there is a change

in number of channels. Change in channel configuration takes as much as 500ms in our testbed. Synchronization via the Ethernet (for implementing the oracle and control messages) also adds to this latency. But overall AMC performs significantly better than any fixed channel configuration as it matches the number of contenders to the number of channels.

5.10 Conclusions and Future Work

In wireless networking literature, use of multiple channels to improve throughput performance is not new. However, in this paper we have offered a refreshing viewpoint. In regimes where the bandwidth independent overheads dominate, single channel performance suffers, with efficiency often falling below 50% even with an optimal contention window. Splitting the available channel into multiple smaller channels has the potential to improve performance considerably. We have shown using realistic numbers that high data rate wireless networks (1 Gbps and up) definitely falls in this regime. However, the number of channels to use depends on the number of contending nodes. Thus, the channelization must be adaptive. This issue is further complicated by use of guard bands.

We have developed an Adaptive Multichannel (AMC) protocol that adapts the number of channels to use based on an estimation of the number of contenders. Simulation results show an often factor of 2 performance improvement relative to using a single channel. We have further demonstrated the viability of the AMC approach using a software radio testbed using the GNU Radio/USRP platform. Experiments using 3 links and upto 3 channels show a similar scale of performance improvement without any additional spectrum use.

Our future work will consider extending the AMC protocol for multihop/multicell operation and more realistic evaluations on higher speed platforms,

specifically focusing on addressing the receiver capability issues. We will also explore opportunities for similar multichannel approach in underwater wireless networks that have a large propagation delay and thus also suffers from large bandwidth independent overheads. Finally, the current work considers only uniform channel splitting. Non-uniform channels may provide better load balancing [74] and is worth exploring.

Chapter 6

Detection and Removal of Wormhole Attack

6.1 Introduction

Wireless ad hoc and sensor networks are typically used out in an open, uncontrolled environment, often in hostile territories. In particular, several important applications for such networks come from military and defence arenas. Use of wireless medium and inherent collaborative nature of the network protocols make such network vulnerable to various forms of attacks. In this work our focus is on a particularly devastating form of attack, called *wormhole* attack [48, 84, 98]. Here, the adversary connects two distant points in the network using a direct low-latency link called the *wormhole link*. The wormhole link can be established by a variety of means, e.g., by using a network cable and any form of "wired" link technology or a long-range wireless transmission in a different band. The end-points of this link (*wormhole nodes*) are equipped with radio transceivers compatible with the ad hoc

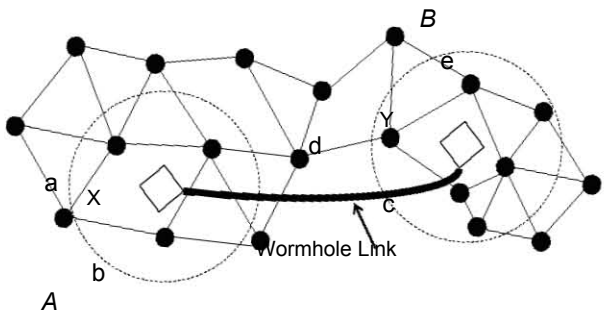


Figure 6.1. Demonstration of a wormhole attack. X and Y denote the wormhole nodes connected through a long wormhole link. As a result of the attack, nodes in Area A consider nodes in Area B their neighbors and vice versa.

or sensor network to be attacked. Once the wormhole link is established, the adversary captures wireless transmissions on one end, sends them through the wormhole link and replays them at the other end.

An example is shown in Figure 6.1. Here X and Y are the two end-points of the wormhole link. As the signals received on one end of the wormhole link are repeated at the other end, any transmission generated by a node in the neighborhood of X will also be heard by any node in the neighborhood of Y and vice versa. The net effect is that all the nodes in region A assume that nodes in region B are their neighbors and vice versa. For example, traffic between nodes like a and e can now take a one-hop path via the wormhole instead of a multi-hop path. If the wormhole is placed carefully by the attacker and is long enough, it is easy to see that this link can attract a lot of routes. Note that if the wormhole link is short, it may not attract much traffic, and hence will not be of much use to the adversary. Thus, throughout the chapter we consider only such attacks in which the wormhole link is long enough so that regions A and B do not overlap.

6.1.1 Significance of Wormhole Attack

While wormhole could be a useful networking service as this simply presents a long network link to the link layer and up, the attacker may use this link to its advantage. After the attacker attracts a lot of data traffic through the wormhole, it can disrupt the data flow by selectively dropping or modifying data packets, generating unnecessary routing activities by turning off the wormhole link periodically, etc. The attacker can also simply record the traffic for later analysis. Using wormholes an attacker can also break any protocol that directly or indirectly relies on geographic proximity. For example, target tracking applications in sensor networks can be easily confused in the presence of wormholes. Similarly, wormholes will affect connectivity-based localization algorithms, as two neighboring nodes are localized nearby and the wormhole links essentially 'fold' the entire network. This can have a major impact as location is a useful service in many protocols and application, and often out-of-band location systems such as GPS are considered expensive or unusable because of the environment.

A wormhole attack is considered dangerous as it is independent of MAC layer protocols and immune to cryptographic techniques. Strictly speaking, the attacker does not need to understand the MAC protocol or be able to decode encrypted packets to be able to replay them. In its most sophisticated form, the wormhole can be launched *at the bit level* or at the *physical layer* [38]. In the former, the replay is done bit-by-bit even before the entire packet is received (similar to cut-through routing [79]). In the latter, the actual physical layer signal is replayed (similar to a physical layer relay [100]). These forms of wormholes are even harder to detect. This is because such replays can happen quite fast and thus they cannot be detected easily by timing analysis. To distinguish these attacks from the simpler form of attack, where the wormhole nodes copy the entire packet before transmittal through the wormhole link, we will refer to this simpler form of attack as *store-and-forward*

attack following the terminology used in [38].

6.1.2 Limitations of Prior Work and Our Contributions

The current solutions for wormhole are limited particularly in connection with large sensor networks, where sensor nodes carry low-cost, relatively unsophisticated hardware and scalability is an important design goal. This rules out use of additional hardware artifact that several reported techniques use – such as directional antennas [47], GPS [48], ultrasound [99], guard nodes with correct location [86]. This also rules out fine grain timing analysis used in several techniques [38, 48]. Also, physical-layer attacks may be immune to timing analysis [38]. Finally, the scalability requirements rule out global clock synchronization [48] or any form of global computations [117].

In the current work, we develop a localized algorithm for detecting wormhole attacks that is purely based on local connectivity information. Such information is often collected any way by various upper layer protocols such as routing, thus may not present any additional overhead. No additional hardware artifact is needed making the approach universally applicable. No timing analysis is done ensuring that we can detect even physical layer attacks. Our technique does not use location information and is able to detect attacks that are launched even before the network is set up, that may influence localization. We expect that our technique is particularly useful for sensor networks as the existing techniques are quite limited there. Also, connectivity is not expected to change frequently in sensor networks, making our connectivity-based approach quite practical.

The detection algorithm essentially looks for *forbidden substructures* in the connectivity graphs that should not be present in a legal connectivity graph. Understanding of the wireless communication model (i.e., a model that describes with some given confidence whether a link between two nodes should exist) helps the

detection algorithm substantially, but is not strictly required. The models we require can be very general and we will demonstrate the capability of the detection using several realistic models such as quasi-unit disk graphs [60] and link models for Berkeley motes as modeled in the TOSSIM simulator [6].

6.2 Related Work

Several papers in literature have developed countermeasures for wormhole attacks. We discuss them in two categories.

6.2.1 Approaches that Bound Distance or Time

In [48] authors have considered packet leashes – geographic and temporal. In geographic leashes, node location information is used to bound the distance a packet can traverse. Since wormhole attacks can affect localization, the location information must be obtained via an out-of-band mechanism such as GPS. Further, the "legal" distance a packet can traverse is not always easy to determine. In temporal leashes, extremely accurate globally synchronized clocks are used to bound the propagation time of packets that could be hard to obtain particularly in low-cost sensor hardware. Even when available, such timing analysis may not be able to detect cut-through or physical layer wormhole attacks.

In [27], an authenticated distance bounding technique called MAD is used. The approach is similar to packet leashes at a high level, but does not require location information or clock synchronization. But it still suffers from other limitations of the packet leashes technique. In the Echo protocol [99], ultrasound is used to bound the distance for a secure location verification. Use of ultrasound instead of RF signals as before helps in relaxing the timing requirements; but needs an additional hardware. In a recent work [38], authors have focused on practical methods of

detecting wormholes. This technique uses timing constraints and authentication to verify whether a node is a true neighbor. The authors develop a protocol that can be implemented in 802.11 capable hardware with minor modifications. Still it remains unclear how realistic such timing analysis could be in low-cost sensor hardware.

In [24], the authors propose statistical approaches to detect increase in number of neighbors and decrease in lengths of shortest paths between all pairs of nodes due to wormhole presence. While probabilistic like ours, their approach cannot remove wormholes and is also centralized.

6.2.2 Graph Theoretic and Geometric Approaches

LiteWorp [55] uses a combination of one-time authenticated neighbor discovery and use of guard nodes that attest the source of each transmission. The neighbor discovery process, however, can be vulnerable to wormhole attacks, if the attack is launched prior to such discovery. A followup paper from the same authors attempts to remove this inefficiency [56], however assumes availability of location information. As mentioned before, this itself could be suspect. In [86] a graph-theoretic framework is used to prevent wormhole attacks. The protocol assumes the existence of special-purpose guard nodes that know their "correct" locations, have higher transmit power and have different antenna characteristics. Use of such special-purpose guard nodes make this approach impractical.

In one approach, directional antennas are used to prevent wormhole attacks [47]. The authors develop a cooperative protocol where nodes share directional information to prevent wormhole endpoints from masquerading as false neighbors. that needs to be certified free from wormhole attack. However, use of directional antennas limits use of such protocols.

In another approach [117] somewhat related, distance estimates between sensors that hear each other is used to determine a "network layout" using multi-dimensional scaling (MDS) technique. The technique is similar to localization of the network nodes in a metric space. Without any wormhole the network layout should be relatively flat. But the layout could be warped in presence of wormholes. The technique is purely centralized and is considerably susceptible to distance estimation errors.

Finally, purely physical layer mechanisms can prevent wormhole attacks such as those involving authentication in packet modulation and demodulation [48]. Such techniques require special RF hardware.

6.3 Wormhole Detection Algorithm

The placement of wormhole influences the network connectivity by creating long links between two sets of nodes located potentially far away. The resulting connectivity graph thus deviates from the true connectivity graph. Our detection algorithm essentially looks for *forbidden substructures* in the connectivity graph that *should not* be present in a legal connectivity graph.

Knowledge of the wireless communication model between the nodes helps our detection algorithm. This is because a communication model can help define what substructures observed in the connectivity graph could be forbidden. However, our approach is still applicable when the communication model is unknown. In this case we need to run an extra search procedure to determine a critical parameter for the detection algorithm. This parameter will be made clear later in this section.

We first develop our wormhole detection algorithm, starting from the unit disk

graph model and then general (known or unknown) communication models, and finally discuss how to automatically remove links created by wormhole once a wormhole is detected.

6.3.1 Unit Disk Graph Model

In unit disk graphs (UDG) each node is modeled as a disk of unit radius in the plane, modeling the communication range of the node with omni-directional antenna. Each node is a neighbor of all nodes located within its disk. UDGs have long been used to create an idealized model of multi-hop wireless networks. We start with this model and formulate our approach of wormhole detection.

6.3.1.1 Hardness of wormhole detection

We first note that under the UDG model, the problem of detecting wormhole attacks with connectivity information is NP-hard. This is observed from the equivalence of wormhole detection with UDG embedding. If the observed connectivity graph has no valid UDG embedding in the plane, it can be deduced that there must be a wormhole present in the network. This can happen when wormhole attack creates long-distance links (longer than unity) which should not exist in a UDG. Conversely, if the observed connectivity graph does admit a valid UDG embedding, then *any* algorithm based on connectivity information only will have to output 'no wormhole'. In such a case, wormhole link, even present, is not distinguishable from a valid link in the embedded UDG. In the absence of any other information, this embedding has to be taken as the ground truth. This can happen, for example, when wormhole links are short and thus appear no different than a link in UDG. This can also happen when the link is indeed long, but lack of sufficient node density prevents detection. This issue will be clearer as we move forward in the chapter. In such

cases, wormhole detection has to use information other than the connectivity graph.

It is known that finding a UDG embedding in 2D is a NP-hard problem [23]. Thus, it is equally hard to detect a wormhole attack using connectivity information alone. A similar relationship between wormhole detection and network localization is also exploited in [117].

The basic idea in our detection algorithm is to look for graph substructures that *do not* allow a unit disk graph embedding, thus *can not* be present in a legal connectivity graph. Due to the hardness result mentioned above, our algorithm will not guarantee the detection of wormhole in all cases. Rather, we aim to design a simple localized algorithm that provides a sufficiently high detection probability in connected networks. We will demonstrate the performance of the algorithm empirically in the next section.

6.3.1.2 Disk packing

The key notion we exploit is a packing argument – inside a fixed region, one cannot pack too many nodes without having edges in between. The forbidden substructures we look for are actually those that violate this packing argument. To be rigorous, we start with some definitions.

Denote by $p(\mathcal{S}, r)$ the *packing number*, which is the maximum number of points inside a region $\mathcal{S}$ such that every pair of points is *strictly* more than distance r away from each other. We assume that no two network nodes are located at the same point. Denote by $\mathcal{D}_R(u)$ a disk of radius R centered at u. $\mathcal{D}$ denotes just a unit disk to simplify notations. As a well-known fact [33], in a unit disk there can be at most 5 nodes whose pair-wise distances are strictly more than 1. Thus $p(\mathcal{D}, 1) = 5$.

Given two disks of radius R centered at u, v with distance r away, define by *lune* the intersection of the two disks, $\mathcal{L}(r, R) = \mathcal{D}_R(u) \cap \mathcal{D}_R(v)$. When $R = r = 1$, we sometimes omit the radii and denote by $\mathcal{L}$ the lune of unit disks set at unit distance

apart.

Lemma 6.3.1. $p(\mathcal{L}, 1) = 2$.

Proof. Refer to Figure 6.2 for an illustration of a lune $\mathcal{L}$. The line segment uv divides the lune into two parts, the upper and lower ones. The two intersections of the two unit circles centered at u, v are denoted p, q respectively. Denote by w the midpoint of segment uv. $|pw| = \sqrt{3}/2 < 1$. It is not hard to see that inside the

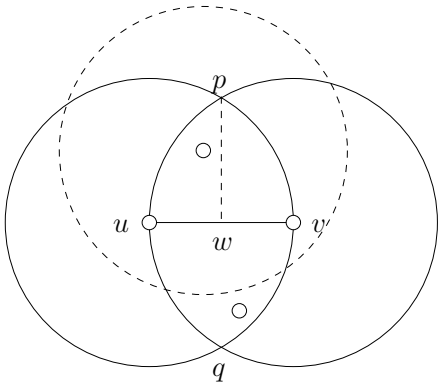


Figure 6.2. One can only pack at most two nodes inside a lune with inter-distance more than 1.

upper half of the lune one can not place two nodes with their distance strictly larger than 1. Indeed, for any node x in the upper half of $\mathcal{L}$, $|xv| \leq 1$, $|xu| \leq 1$, $|xp| \leq 1$. Thus there can only be two nodes inside $\mathcal{L}$ with inter distance larger than 1. □

We can generalize the result for packing of disks of radius β, with the proof appearing in the appendix.

Lemma 6.3.2. $p(\mathcal{L}(r, R), \beta) \leq \lfloor \frac{8}{\pi}(R/\beta + 1/2)^2 \cdot \arccos(r/(2R + \beta)) - \frac{4r}{\pi\beta^2}\sqrt{(R + \beta/2)^2 - r^2/4} \rfloor$ *for* $r \leq 2R$.

Proof. See the appendix. □

Remark. Lemma 6.3.2 only gives a loose bound for $p(\mathcal{L}, \beta)$. When $\beta = 1$, Lemma 6.3.2 gives $p(\mathcal{L}, 1) \leq 5$, which is worse than the bound in Lemma 6.3.1. This motivates us to find a practical bound for $p(\mathcal{L}, \beta)$ by other techniques as will be shown later.

6.3.1.3 Forbidden substructure for wormhole detection

The packing results are used to define forbidden substructures for unit disk graphs. The wormhole connects all nodes in region A with all the nodes in region B (Figure 6.1). Thus we can have two independent (i.e., non-neighbor) nodes in region A, say, a, b, that share three common neighbors c, d, e in region B that are independent. This constitutes a forbidden structure, since in any valid UDG embedding of the connectivity graph the three common neighbors must be within the intersection of disks centering a, b. Since they are independent, their pairwise distance must be more than 1. By Lemma 6.3.1 we know that this can not happen. Thus the discovery of this forbidden substructure reveals the existence of a wormhole.

However, this technique of finding forbidden substructure cannot always *guarantee* detection of wormholes because the existence of nodes like c, d, e in region B is dependent on the density of nodes in the network. The technique will fail when region B has only 2 nodes, for example. For such low density cases, we need to go beyond 1-hop and look for similar forbidden substructures among k-hop neighbors. Here, we will look for f_k common independent k-hop neighbors of two non-neighboring nodes. f_k is a parameter to be discussed momentarily. To summarize, the forbidden substructures we will use in our algorithm are the following.

- **3 independent common 1-hop neighbors:** Two non-neighboring nodes having 3 independent common neighbors; In general, we have
- f_k **independent common** k**-hop neighbors:** Two non-neighboring nodes

having f_k independent common k-hop neighbors.

We call f_k the *forbidden parameter* of the wormhole detection algorithm. f_k must be more than the packing number for unit distance inside the lune of two disks of radii k (modeling the k-hop neighborhood) placed at distance 1 (modeling the lower bound for the distance between non-neighbors). Thus, $f_k = p(\mathcal{L}(1,k),1) + 1$, with $p(\mathcal{L}(1,k),1)$ as the corresponding packing number to be determined by Lemma 6.3.2 or other methods. Also, from Lemma 6.3.1, for k =1, $f_1 = 3$. For a communication model that is not unit disk graph, the determination of f_k will be discussed in subsection 6.3.3.

If a network has one of these forbidden substructures, we know *for sure* that there is a wormhole. For a given node density, if there is wormhole present, the possibility of finding it improves with increasing k. This is because larger neighborhoods simply provide more nodes to work with, thus increasing the possibility of finding forbidden substructures. Our evaluations in the next section show that testing for 1-hop is often sufficient to provide a very high detection rate requiring 2-hops only for very sparse, disconnected or irregular networks. This makes the approach quite practical.

6.3.2 Algorithm Description

Recall that the wormhole detection algorithm is to search by each node a forbidden structure in its neighborhood. The algorithm is localized and distributed. Each node searches for forbidden structures in its k-hop neighborhood. We will explain the algorithm for the general k-hop detection. In our empirical studies $k \leq 2$ was found sufficient for most of the cases.

Each node u maintains the list of $2k$-hop neighbors $N_{2k}(u)$. Node u finds a non-neighboring node, v, from $N_{2k}(u)$ and checks their k-hop neighbor lists to compute

their common k-hop neighbors $C_k(u, v)$. Note that to find a non-empty $C_k(u, v)$ set, node u need not look for v beyond $2k$ hops. We now need to look for the existence of the forbidden substructure (i.e., f_k independent nodes) in $C_k(u, v)$. One way to do this would be to compute the maximum independent set among $C_k(u, v)$ and comparing the size of this set with f_k. But computing the maximum independent set is a NP-hard problem, even for unit disk graphs [41, 42]. Thus we relax the detection rule by finding a *maximal* independent set (a set of independent nodes such that no other node can be included), which can be done by a simple greedy algorithm: we start from an empty set, pick an arbitrary node and include it in the independent set, remove its neighbors, and continue until we run out of nodes in $C_k(u, v)$. The resulting set is a maximal independent set.

We compare the size of the maximal independent set thus obtained with the forbidden parameter f_k. If it is equal or larger than f_k, then we output 'wormhole detected'. The outline of the algorithm is as follows.

1. In a preprocessing stage, find the forbidden parameter f_k, based on the node distribution and communication model. (For UDGs, the bound on f_k can be derived from Lemmas 6.3.1 and 6.3.2. We discuss other techniques of finding f_k in practice in the next subsection, which also generalize to non-UDGs.)

2. Each node u determines its $2k$-hop neighbor list, $N_{2k}(u)$, and executes the following steps for each non-neighboring node v in $N_{2k}(u)$.

3. Node u determines the set of common k-hop neighbors with v from their k-hop neighbor lists. This is $C_k(u, v) = N_k(u) \cap N_k(v)$. This can be determined by simply exchanging neighbor lists.

4. Node u determines the maximal independent set of the sub-graph on vertices $C_k(u, v)$, by using the greedy algorithm presented above.

5. If the maximal independent set size is equal or larger than f_k, node u declares the presence of a wormhole.

The way the algorithm is presented makes it appear as if some work is duplicated (nodes u and v are doing the same computation by symmetry). These can be easily resolved by using some priority rules based on node ids.

The algorithm presented above depends only on the $2k$ and k-hop neighbor lists of each node. If the wormhole attacks are required to be detected as soon as they are in place, ideally our algorithm can be run everytime there is a change in topology. Since it is a local algorithm, only the nodes affected by the change in topology need to re-run it. In practice, the requirement to run it immediately after the attack is placed is not so strict. In such cases, the algorithm can be run periodically depending on the security requirements and the network condition. For example, in mobile networks it is probably more sensible to run it periodically, while in static networks, it should be triggered by changes in topology.

The message and time complexity of the algorithm is dependent on k. As we mentioned, for all cases we considered in our simulations, including fairly low density cases, $k \leq 2$ has been sufficient. In cases where the network in fact has enough density to be connected and is fairly uniform (like in most practical cases), $k = 1$ has been found to be sufficient. The computational cost for $k = 1$ is roughly $O(d^3)$, where d is the average degree of the nodes. Essentially a node checks each of $O(d^2)$ non-neighboring nodes in its 2-hop neighborhood, and pays a cost of $O(d)$ for finding the maximal independent set size in the intersection list. For any practical network, d is typically a small constant. So the detection algorithm is quite efficient.

6.3.3 Consideration of Node Distribution and General Communication Model

Consideration of node distribution is important in the performance of our algorithm. The packing number $f_k - 1$ used above, i.e., the maximum number of independent common k-hop neighbors of two independent nodes, is the theoretical worst case bound for an arbitrary distribution. If the sensors are deployed with a known distribution, then the forbidden parameter f_k we use in the forbidden substructure can be much smaller than the theoretical worst case. For example, for the 2-hop detection case, $p(\mathcal{L}(1,2),1) \leq 18$ by lemma 6.3.2, providing $f_2 = 19$. Unless the node density is very high, it is unlikely that we will be able to find that many common independent 2-hop neighbors between two non-neighboring nodes to be able to detect a wormhole attack. This observation prompts us to tune this critical parameter f_k according to the specific node distribution and not relate it directly to the packing number that models an absolute bound. In general, the smaller f_k is, the higher the detection rate. When f_k is too small, we may have false positives as some legal configuration may be identified as wormhole.

The second important consideration is the communication model. The unit disk graph model considered so far is an overly simplified model for wireless communications. Experiments show that packet reception range is not a perfect disk [39]. Our approach can be generalized to any communication model, and even to situations where communication model is unknown. The algorithm indeed remains the same. But the preprocessing step involving the determination of the forbidden parameter f_k in the first step of the algorithm differs.

In following we describe a number of techniques to obtain the forbidden parameter f_k in practice.

6.3.3.1 Known models

For any practical node deployment we typically know the radio propagation characteristics for the specific hardware used subject to the deployment environment, as well as the spatial distribution of nodes. We could try to find f_k directly using mathematical or geometrical constructs. For example, a quasi-unit disk graph model [60] assumes that two nodes have a link if their distance is within $\alpha \leq 1$ and do not have a link if their distance is larger than 1. If two non-neighboring nodes have f_1 independent common neighbors, these nodes must be within the lune $\mathcal{L}(\alpha, 1)$ and are pairwise distance α away. Thus the packing number is $f_1 = p(\mathcal{L}(\alpha, 1), \alpha) + 1$. In general, we have $f_k = p(\mathcal{L}(\alpha, k), \alpha) + 1$.

For all communication models, it may not be always possible to evaluate such expressions, or even write such mathematical constructs. In such cases, we can run simulations with the targeted distribution to obtain an estimated connectivity graph, with which we can estimate the forbidden parameter f_k. For example, for any pair of non-neighboring nodes we can find the maximal independent set among their common k-hop neighbors and take the maximum as $f_k - 1$. Our simulation results in this work actually use this method and obtain tight bounds for f_k. Notice that when the communication model is probabilistic, the maximum number of independent neighbors of two non-neighboring nodes, $f_1 - 1$, is also probabilistic. Thus false positives are possible in theory under our detection algorithm.

6.3.3.2 Unknown models

When nothing is known about the node distribution and/or communication model, it becomes harder to estimate f_k. In this case, we run the detection algorithm with a standard parametric search for the unknown parameter f_k. We start with a large initial value for f_k, and run the algorithm as presented before. If no wormhole

is detected, we halve f_k and rerun the algorithm. Notice that when f_k is small enough, false positives will show up. We choose f_k to be the value when only a very small fraction of nodes report wormholes, or the minimum number of tolerable false positives. One good mechanism would be to run this parametric search in a safe part of the network, guaranteed to be free from wormhole, before deploying it in the entire network. We can then estimate the parameter such that there is no false positive detection in the safe part and apply the parameter for the entire network

If no such safe part can be ascertained, the search must run in the network that has potentially been inflicted with wormholes already. In that case, a "threat level" must be assumed. The threat level is to be used as a guidance for what fraction of nodes must report wormholes before f_k will not be reduced any further.

6.4 Wormhole removal

Once a forbidden structure is discovered, it is usually expected that user should manually intervene and remove the wormhole nodes. Here, we devise a simple approach to remove the wormhole link without manual intervention. Since the wormhole link is 'invisible' to the network, it is not possible for the network to automatically remove it. But the true impact of a wormhole attack is due to the many *illegal* links it creates between two sets of far-away nodes. Thus it is enough to remove these illegal links to achieve wormhole removal. We devise a simple distributed algorithm which does so without manual intervention. Here, we outline the approach for the 1-hop detection case for UDGs. It can be easily extended for other cases.

After a successful 1-hop detection in UDGs, we have two non-neighboring nodes a, b with 3 common independent neighbors c, d and e. Figure 6.1 shows one possible placement of these nodes to form the forbidden substructure, such that

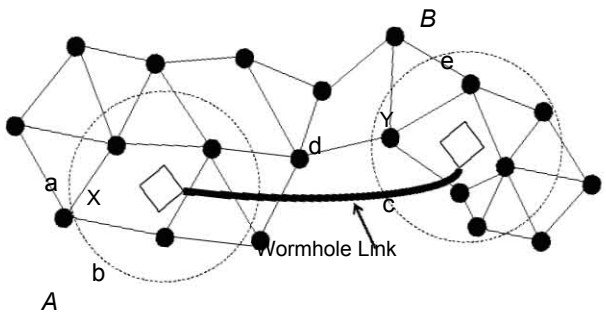


Figure 6.3. Example of second possible placement of the forbidden substructure.

a and b are placed in one region (lets call it region A, without loss of generality) and c, d and e are placed in another region, B. Another possible placement is shown in Figure 6.3. Here, a, b are located in region A; d, e are located in region B; but c is located just outside A neighboring a and b. It can be verified that these are the only two placements possible for the 1-hop detection in UDGs.

One can define two types of nodes neighboring the wormhole region – *corrupted* and *uncorrupted* nodes. Corrupted nodes are the nodes inside A and B which can hear transmissions from the wormhole nodes X and Y. These are the nodes with illegal links. In other words they have their neighbor lists corrupted – they have far away nodes also as neighbors – due to the presence of the wormhole link. Nodes outside A and B are uncorrupted. Our wormhole removal algorithm tries to identify, and blacklist, all nodes that are possibly corrupted (the rest are surely uncorrupted nodes). Once identified, each corrupted node tries to fix its neighbor list. It does so by comparing its neighbor list with the neighbor lists of its uncorrupted neighbors. Note that even one link due to wormhole placement left out un-removed can potentially cause a huge damage to the network. Thus our removal scheme allows error on the aggressive side and removal of legals links, as long as all the illegal links are definitely removed.

Inferring from the two placements discussed above, one can say that nodes

which satisfy any of these two conditions must include all corrupted nodes:

- The node is a neighbor of both a and b, or,
- The node is a neighbor of at least 2 nodes out of c, d and e.

The first condition will identify all nodes in area B. All nodes in B will be neighbors of a and b due to the wormhole link. Similarly, the second condition identifies all nodes in area A.

We call the nodes identified using the above method *suspicious nodes*. The set of suspicious nodes will include *all* corrupted nodes and may include some uncorrupted nodes. On the other hand, all nodes not identified by this method, are definitely uncorrupted nodes.

Once all suspicious nodes are identified, our wormhole removal algorithm works in two stages – first it tires to remove all *possibly* illegal neighbors from the neighbor list of each suspicious node. This might remove many legal neighbors also. In the second stage, the algorithm tries to add as many legal neighbors as possible back to the neighbor list.

6.4.1 Stage 1 - Blacklisting

To remove the illegal links, each suspicious node, u, takes the intersection of its neighbor set, $N(u)$, with the neighbor sets of other non-suspicious (uncorrupted) nodes. While these uncorrupted nodes can be any number of hops away, we consider the case when they are just one hop away. Thus, u takes intersection of $N(u)$ with $N(v)$ for all its neighboring uncorrupted nodes v. Any neighbor which is part of such an intersection must be a legal neighbor.

Any node $w \in N(u)$ which is not part of any such intersections and is also a suspicious node, is blacklisted by u and added to its illegal neighbor list $N_{il}(u)$. All

future transmissions from nodes in $N_{il}(u)$ will be ignored by node u making the wormhole attack ineffective. When all suspicious nodes finish blacklisting nodes from their neighbor lists, this completes the first stage of wormhole removal. We note that at this stage the removal is a bit aggressive to guarantee that all illegal links due to wormhole will be removed, however, some legal links may be removed as well. In particular, nodes near the center of the wormhole regions A and B might end up getting most of their neighbor list blacklisted. This can happen as they may not have many uncorrupted neighbors with other common legal neighbors. This is not desirable as it affects the network connectivity. In the second stage, we try to tackle this problem.

6.4.2 Stage 2 - Revival

In the second stage of our wormhole removal algorithm, we try to alleviate the problem mentioned above. Once the first stage is completed, each suspicious node u has two sets of nodes in its neighbor list $N(u)$ – possibly illegal neighbors ($N_{il}(u)$) and legal neighbors ($N_l(u)$). In the second stage, we let these suspicious nodes compare their neighbor set with the neighbor set of other suspicious nodes also. By doing so, they try to identify such blacklisted neighbors which may be their real neighbors. Specifically, each suspicious node u, for each of its neighbor $v \in N_l(u)$, takes the intersection of illegal neighbor set, $N_{il}(u)$ with v's legal neighbor set, $N_l(v)$. Each node $w \in N_{il}(u) \cap N_l(v)$, is moved by u from $N_{il}(u)$ to its $N_l(u)$. The second stage of the algorithm is complete when each node u compares its neighbor list with all its legal neighbors v.

Since the sets N_{il} and N_l change after the second stage, repeating the second stage can result in smaller blacklist sets. Thus, we repeat the second stage till a stable state is reached. This completes our wormhole removal algorithm.

6.4.3 Non-UDG cases

While we have explained the removal algorithm in the case of UDGs, it can be easily extended to non-UDGs using the same techniques as used in detection. In particular, when the forbidden parameter is f_k, the the method for detecting suspicious nodes will change. It will become as follows:

- The node is a neighbor of both a and b, or,
- The node is a neighbor of at least $f_k - 1$ nodes out of the f_k independent neighbors of a and b.

6.5 Simulation Results

In this section, we present simulation results demonstrating the effectiveness of our algorithm in detecting and removing wormhole attacks. In the first subsection, we present our results of evaluating the probability of successful detection for networks with various node distributions and connectivity models. We consider three different connectivity models in our simulations: a) unit disk graph, b) quasi-unit disk graph and c) the model used in the TOSSIM simulator [6], which is based on real empirical data from a motes testbed. We evaluate the algorithm with two different node distributions: i) grid distribution with some perturbations (modeling a planned sensor deployment) and ii) random distribution. In the second subsection, we present the results of simulations demonstrating the effectiveness of our wormhole removal algorithm by using unit disk graphs as an example connectivity model.

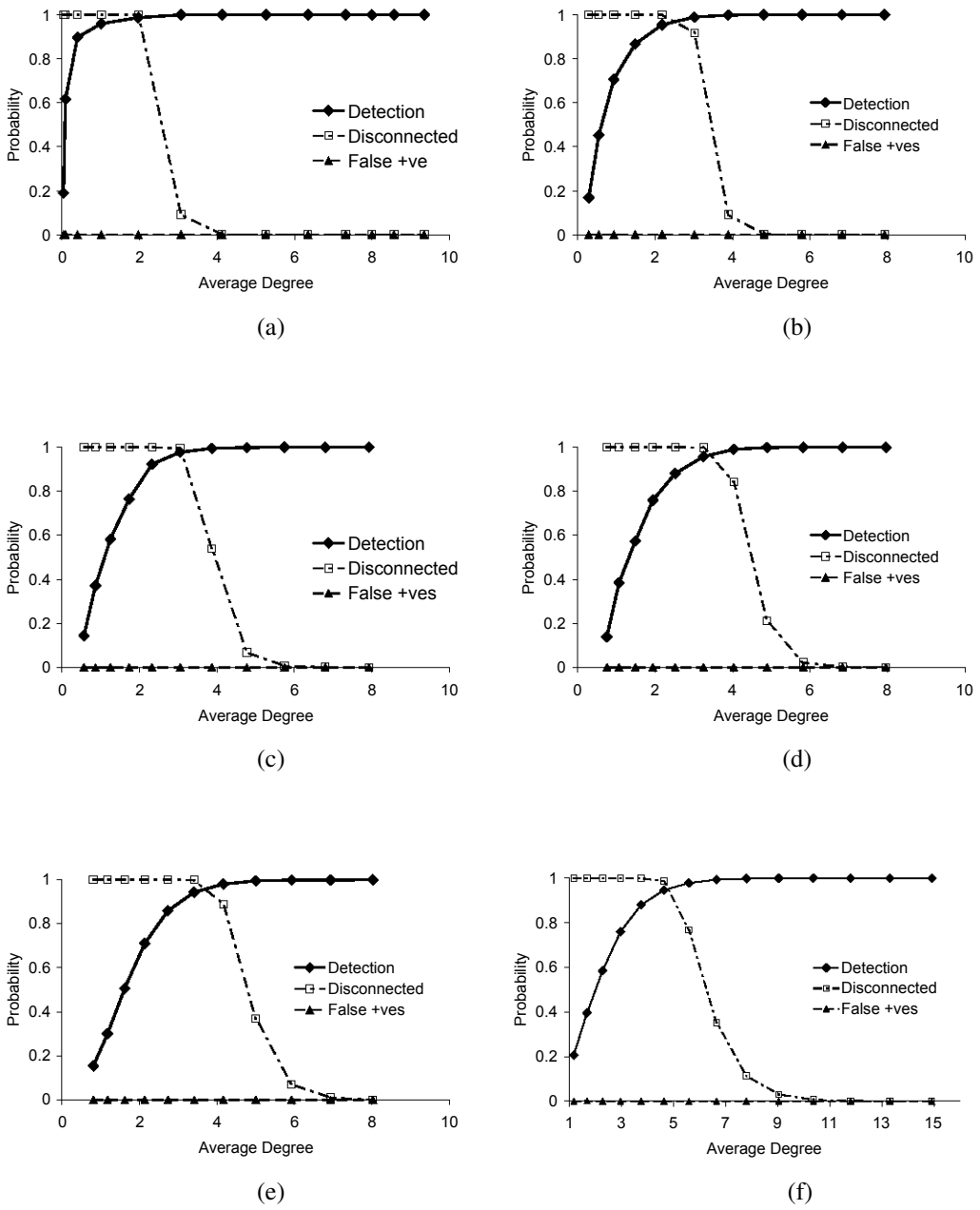


Figure 6.4. Probability of wormhole detection, graph disconnection and false positives for UDG connectivity, Perturbed Grid and Random node distributions.

6.5.1 Details of Models and Evaluation Approach

In the quasi-UDG model, if the transmission radius of the nodes in the network is R and the quasi-UDG factor is α (where, $0 \leq \alpha \leq 1$), then there exists a link between every pair of nodes within distance αR. If the distance is greater than R, then there is no link. If the distance d between a node pair is within $[\alpha R, R]$, we assume presence of a link with probability $\frac{d}{R-\alpha R}$. In the TOSSIM model, the provided `LossyBuilder` tool is used to generate bit error probabilities (say, P_b) between node pairs. In order to build the connectivity graph, it is assumed that the link exists with probability $(1 - P_b)$. Note that the TOSSIM model does not assume that the links are bi-directional. Our algorithm works irrespective of whether the links are directional or bi-directional.

Each simulation is run with 225 nodes. Since our technique is localized (we use only 1-hop and 2-hop detections in our experiments) and the simulations so far concentrate on detecting only a single wormhole, simulating a very large networks is not required to determine the performance of our approach. For the grid-like topologies the nodes are placed in a 15×15 grid. Then their x and y coordinates are changed to a randomly chosen value between $[x-px, x+px]$ and $[y-py, y+py]$ respectively, where p is the perturbation parameter. Values of p from 0.0 to 1.0 have been used. For the random case, x and y coordinates are chosen randomly. As noted before node density is an important factor in our algorithm. Node density is varied in different experiments by changing the geographic area containing the nodes (for TOSSIM) or by changing the transmission radius of the nodes (for UDG and Quasi-UDG cases).

After the topology is created, the nodes are connected using the given connectivity model. For detection, once the connectivity graph is established, the following experiments are performed:

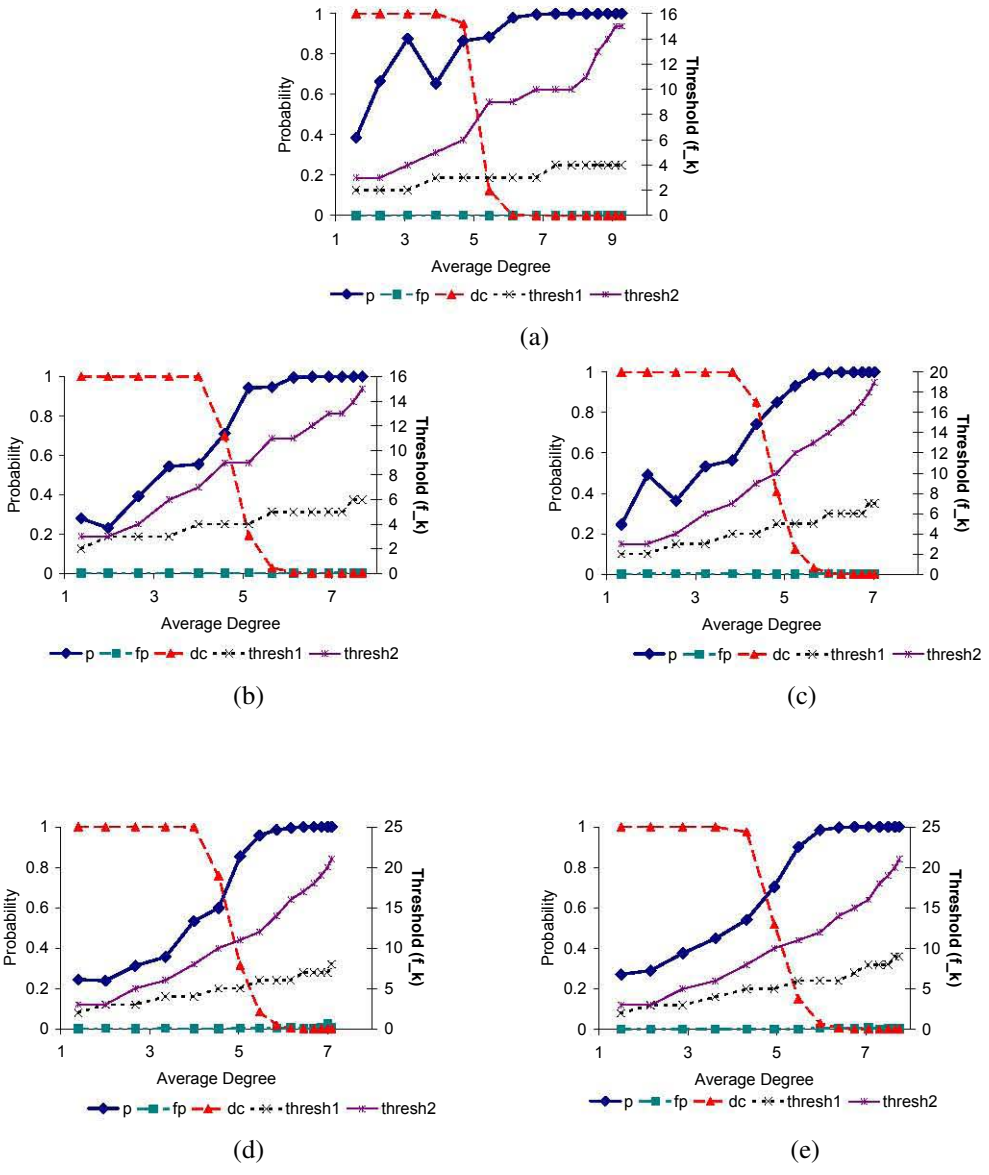


Figure 6.5. Probability of wormhole detection, graph disconnection and false positives for Quasi-UDG connectivity and Perturbed Grid node distribution with perturbation parameter=0.5

- Connectivity in the entire network is checked. The network is assumed disconnected if any two nodes do not have a path to each other.[1]
- The wormhole detection algorithm is run to see whether there is a false positive. (At this time, there is no wormhole attack)
- A wormhole attack is established between two randomly chosen locations. The algorithm is run again to see whether it detects the wormhole.

The algorithm was run with $k \leq 2$ only. We will see momentarily that this already gives very good results for most practical scenarios. We have repeated each experiment for $10,000$ times with randomly generated topologies and attacks, but with the same node distribution model and connectivity model, and then reported various probabilities for different node densities. Three probabilities are computed: (i) probability of detection, (ii) probability of false positive and, (iii) probability of network disconnection. To avoid boundary effects, we have not considered the boundary nodes when calculating the degree, testing for disconnected networks, etc.

6.5.2 Results for Wormhole Detection

We wanted to study the effect of varying node distribution and varying connectivity model on the performance of our algorithm. Thus we performed two sets of experiments to illustrate the difference.

Figure 6.5 shows the results for the case when the connectivity model was varied slowly from more regular to more random while keeping the node distribution constant at perturbed grid with $p = 0.5$. The subfigures show results for connectivity

[1] While our technique is independent of whether the entire network is connected or not, connected networks are more useful from a practical standpoint.

model varying from quasi-UDG with $\alpha = 0.0$ (modeling UDG) case to quasi-UDG with $\alpha = 1.0$ (modeling very random connectivity) case.

Similarly, figure 6.4 shows the results for the case when the node distribution was varied from a perfect grid to a random node distribution while keeping the connectivity model constant as UDG. To vary the node distribution randomness, we varied the perturbation parameter p from 0.0 (modeling perfect grid) to 1.0 (modeling highly random grid). We also present the case of random node distribution (non-grid like) for comparision. And lastly, figure 6.6 shows the results for the case with TOSSIM as connectivity model and grid and random node placement.

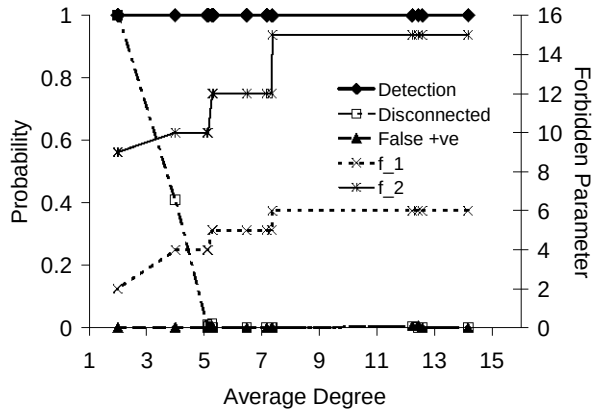
Recall that the forbidden parameter f_k is an input parameter to our algorithm and is evaluated separately in a pre-processing step as shown in subsection 6.3.3. The results also shows f_k values for different experiments. For UDG cases, it is observed that only 1-hop detection is enough for all cases except at very low densities (average degree ≤ 1), and f_1 is constant at 3. Thus, we do not show the f_k curves for UDG graphs (figure 6.4). In general, the following observations can be made from the results:

- *Our algorithm provides very good results (no false alarms and 100% detection) when the network disconnection probability is 0.* This observation is independent of communication or node distribution model used.

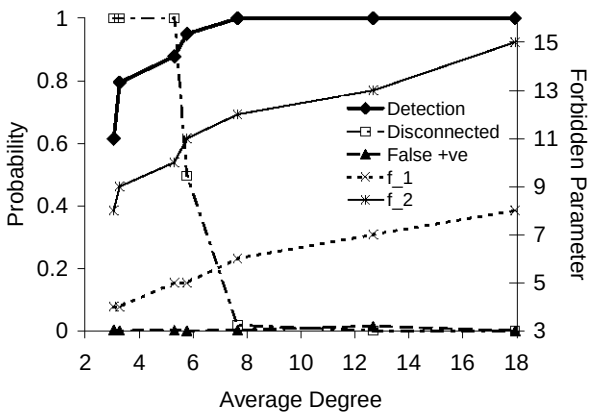
- Detection probability does drop for low density cases; however, in such cases the likelihood that the network is disconnected also increases (hence the usefulness of the network also drops).

- Almost always, even with a 50% chance of the network being disconnected, our algorithm has detected the wormhole attack in 90% or more cases.

- For the same average node density, detection performance gets worse as the randomness of deployment (in terms of node distribution or communication



(a) Perturbed Grid node distribution with p=0.2



(b) Random node distribution

Figure 6.6. Probability of wormhole detection, graph disconnection and false positives with TOSSIM connectivity model.

model) increases. For example, the detection rate is better in UDG Perturbed Grid scenario (Figure 6.4a) than UDG Random scenario (Figure 6.4d) or Quasi-UDG Perturbed Grid scenario (Figure 6.4b) and so on. This phenomenon is expected because the estimation of f_k is more accurate in less random scenarios.

- The detection probability goes down as the randomness in either node distribution or connectivity model is increased.
- The disconnection probability goes up as the randomness in either node distribution or connectivity model is increased.
- Thus, it can be deduced in general that our algorithm's performance depends a lot on the disconnection probability in the network.

As mentioned earlier, 1-hop only detection does not perform well in non-UDG cases. Figure 6.7 compares 1-hop detection probability with the case when both 1 and 2-hop detection algorithm were used (2-hop detection runs only when 1-hop fails), under the setup of Figure 6.4e with Random node distribution and Quasi-UDG connectivity model. Note that as the value of parameter f_1 increases, the 1-hop detection fails to detect wormhole attacks in some cases, and hence 2-hop detection kicks in.

Finally, we run a set of simulations demonstrating how the forbidden parameter f_k can be estimated for a situation where the communication model and/or the node distribution are unknown. The given scenario uses $k = 1$, quasi-UDG model and nodes placed in grid with perturbation of 0.2 and average degree of 6. Both the false positive probability (in the absence of wormhole) and detection probability (in presence of wormhole) are shown for different values of f_1 in Figure 6.8.

There exist critical values of f_1 (4 in the plot) where the detection probability

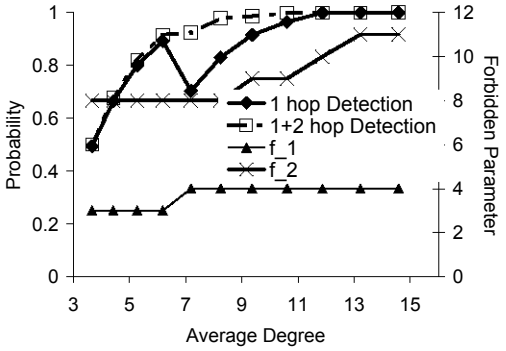


Figure 6.7. Comparison of 1-hop vs 1 and 2-hop detection.

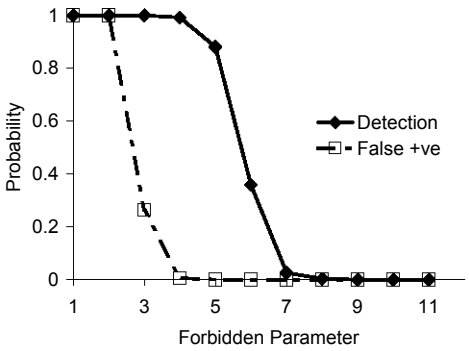


Figure 6.8. Estimation of the forbidden parameter in a quasi-UDG model.

is close to 100%, but the false positive probability is close to 0%. This demonstrates that if the parametric search is used in a safe network, a suitable value for f_1 can be estimated by simply observing the false positive probabilities. When f_1 is reduced from a large value, the detection of real wormholes shows up first before false positives.

6.5.3 Results for Wormhole Removal

We performed simulations to quantify the performance of our wormhole removal algorithm. In general, we wanted to a) evaluate the probability of our algorithm successfully removing all illegal links in the network and b) understand the impact of the removal algorithm on the connectivity of the network. We performed simulations with unit disk graph connectivity model and varying node distributions in a setup similar to the one described in the previous section. 225 nodes were placed in a 1000×1000 area and their transmission radius was varied to vary the average node degree.

Once the nodes are placed using a particular node distribution and the connectivity established using UDG model, the following experiments were performed:

- Connectivity in the entire network is checked. The net- work is assumed disconnected if any two nodes do not have a path to each other.
- The wormhole attack is placed between randomly chosen points in the network.
- The 1-hop wormhole detection algorithm was executed to check if the attack can be detected. If detected, the wormhole removal algorithm was used to remove illegal links. Once the removal is complete, we check if the algorithm has removed all illegal links.

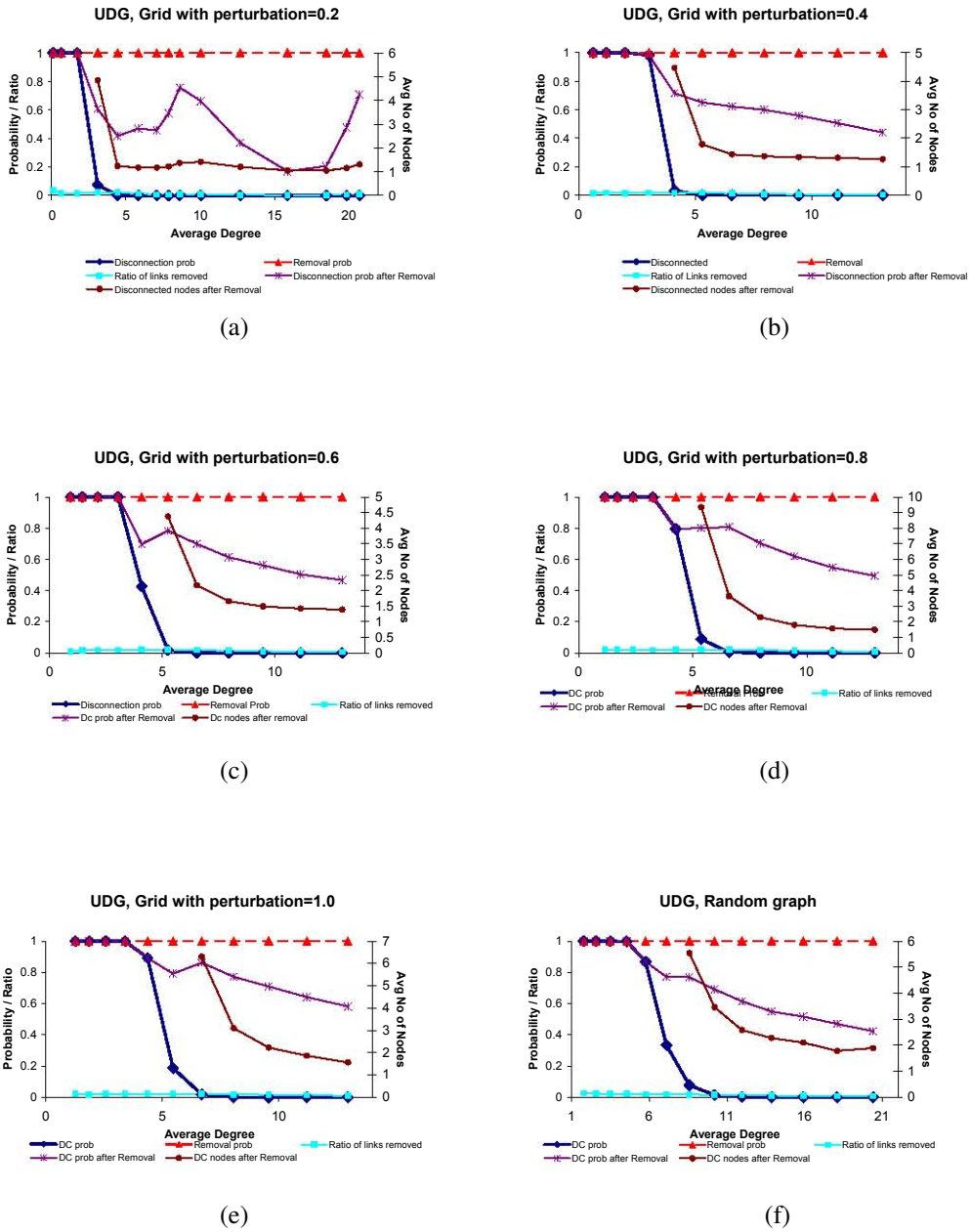


Figure 6.9. Probability of wormhole removal, graph disconnection, and removal penalty for UDG connectivity, Perturbed Grid and Random node distributions.

- If the removal was successful, we record the ratio of legal links removed to the total number of legal links present in the network. This ratio gives us an idea about how many legal links were sacrificed for removing the wormhole.

- If the network was not disconnected before starting the wormhole, then we again check the connectivity in the network to measure the disconnection probability after removal.

- If network is found disconnected, we find the number of nodes that are disconnected. To do this, we just find the largest connected component size in the network. The network size (225) minus the largest connected component size gives us the number of disconnected nodes.

10000 runs of each experiment was performed and the average results are plotted in Figure 6.9. The results show that our wormhole removal algorithm is very effective. It removes all illegal links with a 100% probability while keeping the ratio of links removed well below 1%. We note that the network is more likely to be disconnected after running the removal algorithm. But the number of nodes which remain isolated due to some legal links getting removed is on an average less than 2 (out of 225 nodes) in all cases when the network was not previously disconnected.

Similar to the detection algorithm, the performance of our wormhole removal algorithm slightly degrades with more randomness in node distribution. But it should be noted that removal probability still remains 100%. Thus, even in very random scenarios, our algorithm can guarantee successful wormhole removal. The penalty for such high success rate is paid in terms of a slight increase in the number of nodes getting disconnected. For random node distributions and highly perturbed grid networks, the number of disconnected nodes after removal is only 3. Thus, even in highly random scenarios, the penalty is not very high.

Again, while we have shown the results of our removal algorithm only for

UDGs, it can be easily extended to non-UDGs using the same techniques as used in detection.

6.6 Extensions

In this section we discuss some implicit assumptions made in our algorithm design about the wormhole attack or the victim network and how our wormhole detection and removal algorithms would perform in absence of those assumptions.

We discussed wormhole attacks in networks where nodes are assumed to be placed in two dimensional Euclidean space. The algorithms presented here can be extended for higher dimensional space with slight modifications. For example, instead of the disk packing argument used in section 6.3.1 for two dimensions, a sphere packing argument can be used for three dimensional Euclidean space and similar bounds on packing numbers can be obtained. Once those bounds are obtained either by analysis or simulations, forbidden substructures can be defined accordingly and a similar distributed algorithm can be used to find such forbidden substructures. Wormhole removal algorithm can be modified similarly to adapt for higher dimensions.

Also, while we discussed our algorithms assuming the presence of only one wormhole link, an adversary can place more than one wormhole links (with more than two wormhole nodes) in the network in order to inflict more damage. Our algorithms work in presence of multiple wormholes as well. The performance of our wormhole detection algorithm depends only on the presence of forbidden substructures in the network. In case of multiple wormholes, the probability of finding such forbidden substructures will only increase.

We have also assumed till now that the wormhole nodes are not placed too close to each other. In particular, we assumed that the wormhole regions like A and B

in Figure 6.1 do not overlap. In this case detecting the wormhole attack by looking for forbidden substructures is harder since even under high node density conditions, the existence of forbidden substructures cannot be not guaranteed. For example, in Figure 6.1 (a UDG case), if areas A and B overlap in such a way that node d was part of both A and B, then d will be a neighbor of a and b as well as c and e. Thus nodes c, d and e will not be independent and a forbidden substructure involving these 5 nodes will not exist. It can be verified that if A and B heavily overlap, no forbidden substructure will exist.

However, wormholes in the above case can be detected by another simple check. If the wormhole nodes overlap in such way that a forbidden substructure could not be found, then there must be some nodes which are present in the region $A \cap B$. When any such node u broadcasts a message, all nodes v neighboring u and $v \notin A \cap B$ will hear that message twice. This is because v will receive one copy of the packet directly from u and another copy indirectly through a wormhole node (from X if $v \in A$ or from Y if $v \in B$). Depending on the medium access control mechanism, if the latency introduced by wormhole nodes is not very large, these packets can also collide, in which case the link to neighbor u is considered as low quality and thus blacklisted. If the latency introduced by wormhole link is non-zero and the packets are small enough, collisions can be avoided and a wormhole is detected. Thus our detection algorithm will have the following extension. All nodes in the network will periodically send a small 'wormhole-check' packet and if any neighboring node receives two copies of the packet, a wormhole attack can be declared detected.

The same extension can also be used to safeguard our algorithm against an adversary who is aware of our detection mechanism. An adversary who knows our detection algorithm can try to defeat it by making the wormhole node X (Y) rebroadcast the packets it hears from the nodes in A (B). By doing so, X (Y) will

make all nodes in A (B) neighbors of each other. Thus, our algorithm will not be able to find non-neighboring nodes like a and b (c, d and e) and hence wormhole attack will not be detected. If nodes periodically broadcast a 'wormhole check' packet and their neighbors hear it twice, such a rebroadcast mechanism by the adversary can also be detected.

Until now in our work we assumed that the transmission power used by the wormhole nodes is the same as the power used by nodes in the victim network. Thus, the transmission ranges of wormhole nodes are similar to that of the victim nodes. Interesting issues arise when we consider the wormhole transmission power as a tunable parameter. Does the adversary really need to know the transmission power used by victim network? Will a different choice of transmission power for wormhole nodes benefit the adversary? For simplicity, we consider the disk graph model and the transmission range used by network nodes is R. If the wormhole nodes use a transmission range greater than R, it will afflict more damage to the network. This is so because more nodes will be covered by the broadcasting range of the wormhole nodes and more shortest paths in the network will go through the wormhole link. On the other hand, our detection algorithm will also perform better. The larger the size of the wormhole regions, the higher the chances of finding forbidden substructures.

If the adversary chooses a wormhole transmission range smaller than R, it reduces the chance of detection by our algorithm but it also reduces the amount of harm the attack can cause to the network. The damage can still be significant and so we extend our algorithm to tackle this case. Note that if the wormhole nodes use a range less than $R/2$, our algorithm will not be able to detect the attack at all. This is because any two nodes in a wormhole region will be at most R distance away. Thus they will be neighbors of each other. This will deny the existence of any forbidden substructure since no independent set of nodes will exist in a wormhole

region.

We extend our algorithm to overcome this challenge in the following way. If the maximum transmission power used by the nodes is such that the transmission range is R, then each node u will maintain $\log R$ neighbor lists. Each neighbor list here will correspond to a particular value of transmission range. The transmission ranges considered will be R, $R/2$, $R/4$ and so on. Thus, the neighbor list $N^k(u)$ will contain all neighbors of u when the transmission range of all nodes in the network is $R/2^k$. When detection begins, each node u will first run the detection algorithm assuming a transmission range of R and using the neighbor list $N^1(u)$. If no wormhole attack is detected, u will run the detection algorithm again, but this time assuming a range of $R/2$ and with neighbor list $N^2(u)$. This process will continue till either the wormhole attack is detected or the algorithm is executed with all $\log R$ sets of neighbor lists. This completes the wormhole detection. Removal algorithm can be extended in a similar way.

It is easy to see that the above algorithm will work when the adversary is trying to defeat our algorithm by varying the wormhole transmission range. If the adversary chooses a transmission range greater than or equal to R, our algorithm will detect it in its first run when assuming a transmission range of R. If the adversary chooses a transmission range between $R/2$ and R, our algorithm will detect it in the second run when assuming a transmission range of $R/2$ and so on. Thus, the adversary does not benefit by varying the wormhole transmission range.

6.7 Conclusion

In this work we propose practical algorithms for wormhole detection and removal. The algorithms are simple, localized, and universal to node distributions and communication models. Our simulation results have confirmed a near perfect detection

performance whenever the network is connected with a high enough probability, for common connectivity and node distribution models. They also showed that our wormhole removal algorithm can guarantee wormhole removal in very random scenarios also. We expect that these algorithms will have a practical use in real-world deployments to enhance the robustness of wireless networks against wormhole attacks.

Appendix

Lemma 6.3.2. We use a packing argument.

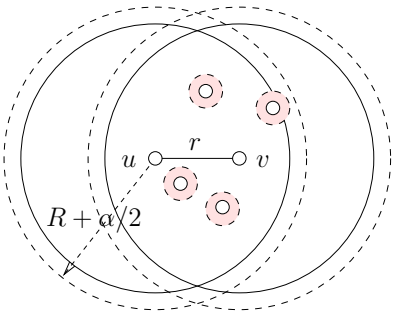


Figure 6.10. Packing in a lune $\mathcal{L}(r, R)$.

Suppose we place a set of nodes P inside $\mathcal{L}(r, R)$ with their inter distances more than β. Thus we place disks of radius $\beta/2$ on each node in P. All the disks are disjoint. Further, all the disks are inside a slightly larger lune $\mathcal{L}(r, R+\beta/2)$, which has an area of $2(R+\beta/2)^2 \arccos(r/(2R+\beta)) - r\sqrt{(R+\beta/2)^2 - r^2/4}$. Thus $p(\mathcal{L}, \beta)$ is no more than the maximum number of non-overlapping disks of radius $\beta/2$ packed inside the lune $\mathcal{L}(r, R+\beta/2)$. The total area of the disks centered on P, $\pi(\beta/2)^2 \cdot |P| \leq 2(R+\beta/2)^2 \arccos(r/(2R+\beta)) - r\sqrt{(R+\beta/2)^2 - r^2/4}$. Thus $p(\mathcal{L}, \beta) \leq |P| \leq \lfloor \frac{8}{\pi}(R/\beta + 1/2)^2 \arccos(r/(2R+\beta)) - \frac{4r}{\pi\beta^2}\sqrt{(R+\beta/2)^2 - r^2/4} \rfloor$ as claimed. □

Chapter 7

Conclusion

In this dissertation, we have studied and proposed multiple ways to improve the capacity of wireless single hop and multi-hop networks. For single hop networks like IEEE 802.11 wireless LANs, we have studied the CSMA/CA MAC layer inefficiencies and proposed improvements. This work is especially important for very high speed wireless networks and underwater wireless networks. For general single hop and multi-hop networks, we have proposed techniques to improve capacity by operating in multiple channels with single data radio and by using TDMA with accurate interference models. Finally, we have also proposed a way to detect and remove Wormhole attacks, a serious security threat to any wireless multi-hop network. Our techniques are general and are applicable to mesh, ad-hoc and sensor networks.

We have proposed two novel MAC protocols that let a network operate with multiple channels. Our multichannel protocols – xRDT and LCM MAC – use only a single data radio and operate without the need of tight time synchronization across the network. xRDT or eXtended Receiver Directed Transmission protocol uses a

second small bandwidth busy tone radio, while LCM MAC or Local Coordination-based Multichannel MAC protocol, performs per-cycle coordination between nodes to solve hidden terminal and deafness problems associated with multichannel protocols. Our ns2 simulations show these protocols improving the capacity of a single-channel IEEE 802.11 like protocol multi-fold.

Characterizing interference in wireless networks is critical in improving spatial reuse and realizing improved network capacity. Various interference models have been proposed in the literature, but no comprehensive study of their comparison exists. In this dissertation, using measurements on IEEE 802.15.4 radio based mesh networks of 20 TelosB motes, we have compared various widely used classical Interference models like the protocol, hop-based and range-based models with the more realistic SINR model (or physical interference model). Our results give new insights on using accurate interference models for TDMA scheduling. We have further evaluated two different incarnations of the SINR model on a mesh testbed of 22 commodity 802.11 based nodes and showed that the widely used 'thresholded' version is inefficient for transmission scheduling purposes and scheduling algorithms using the more realistic 'graded' version should be investigated.

Given a chunk of spectrum, an interesting question to be asked is whether channelization is needed at all. Intuitively, it might make sense to not channelize and use the complete spectrum for high bandwidth transmissions, which in turn should improve performance (at least in terms of delay). We have shown through analysis that, contrary to intuition, channelization actually improves network capacity, especially in high-speed networks. We have developed a novel CSMA protocol which adaptively splits the spectrum into channels based on the current traffic. We implemented a simpler version of the protocol on the software defined radio platform called GNU Radio using the USRP hardware and evaluated the proposed protocol through simulations.

Finally, by using only connectivity information, we have proposed an algorithm to detect and remove the easy-to-deploy and devastating wormhole attack in wireless networks. This is in contrast with the other previous works which have used various hardware artifacts and/or location information etc. Our technique is completely localized and detects wormhole attacks with an almost perfect detection rate for connected graphs.

Bibliography

[1] "Moteiv," http://www.moteiv.com.

[2] Project IEEE 802.11 VHT study group. http://www.ieee802.org/11/Reports/vht_update.htm.

[3] "Soekris Engineering," http://www.soekris.com/.

[4] "The GNU Radio Project," in *http://www.gnu.org/software/gnuradio/*.

[5] "TinyOS community forum," http://www.tinyos.net.

[6] "TOSSIM: A simulator for tinyos networks," User's manual in TinyOS documentation.

[7] "USRP - The Universal Software Radio Peripheral," http://www.ettus.com/.

[8] Wireless HD technology. http://www.wirelesshd.org/technology.html.

[9] "HFA3863 Data Sheet: Direct Sequence Spread Spectrum Baseband Processor with Rake Receiver and Equalizer," Intersil Corporation, 2000.

[10] A. Acharya, A. Misra, and S. Bansal, "MACA-P : a MAC for concurrent transmissions in multi-hop wireless networks," in *Proc. of IEEE PerCom*, March 2003, pp. 505–508.

[11] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *Proc. of Broadnets*, 2004.

[12] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proc. SIGCOMM 2004*, 2004.

[13] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *MobiCom '05*.

[14] M. Alichery, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for throughput optimization in Multi-Radio wireless mesh networks," in *ACM MobiCom*, 2005.

[15] E. Aryafar, O. Gurewitz, and E. Knightly, "Distance-1 Constrained Channel Assignment in Single Radio Wireless Mesh Networks," in *Proc. IEEE INFOCOM*, 2008.

[16] A. R. S. Bahai, B. R. Saltzberg, and M. Ergen, *Multi Carrier Digital Communications: Theory and Applications of OFDM*. Springer, 2004.

[17] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *Proc. of ACM MobiCom*, 2004.

[18] H. Balakrishnan, C. L. Barrett, V. S. A. Kumar, M. V. Marathe, and S. Thite, "The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc networks," *IEEE J. Selected Areas of Communication*, pp. 1069–1079, 2004.

[19] G. Bianchi, "Performance analysis of the IEEE 802.11 Distributed Coordination Function," *JSAC*, 2000.

[20] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," in *Proc. IEEE Infocom Conference*, 2003.

[21] D. Blough, S. R. Das, G. Resta, and P. Santi, "A framework for joint scheduling and diversity exploitation under physical interference in wireless mesh networks," in *IEEE MASS*, Atlanta, 2008.

[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 *MobiCom '06*, 2006.

[23] H. Breu and D. G. Kirkpatrick, "Unit disk graph recognition is NP-hard," *Computational Geometry. Theory and Applications*, vol. 9, no. 1-2, pp. 3–24, 1998. [Online]. Available: citeseer.ist.psu.edu/breu93unit.html

[24] L. Buttyn, L. Dra, and I. Vajda, "Statistical wormhole detection in sensor networks," in *Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS)*, 2005.

[25] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 785–799, 2000.

[26] J. Camp, J. Robinson, C. Steger, and E. Knightly, "Measurement driven deployment of a two-tier urban mesh access network," in *Proc. ACM MobiSys*, 2006, pp. 96–109.

[27] S. Capkun, L. Buttyn, and J. P. Hubaux, "SECTOR: Secure tracking of node encounters in multi-hop wireless networks," in *1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, October 2003.

[28] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl, "A case for adapting channel width in wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 135–146, 2008.

[29] H. Chang, V. Misra, and D. Rubenstein, "A general model and analysis of physical layer capture in 802.11 networks," in *Proc. IEEE Infocom*, 2006.

[30] K. Chebrolu, B. Raman, and S. Sen, "Long-distance 802.11b links: Performance measurements and experience," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2006, pp. 74–85.

[31] R. R. Choudhury and N. Vaidya, "Deafness: A mac problem in ad hoc networks when using directional antennas," in *ICNP*, Berlin, October 2004.

[32] A. Coja-Oghlan, C. Moore, and V. Sanwalani, "Max k-cut and approximating the chromatic number of random graphs," in *Proc Thirtieth Int Coll Automata, Languages, Programming, Lecture Notes in Computer Science 2719, Springer*, 2003, pp. 200–211.

[33] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 2nd ed. New York, NY: Springer-Verlag, 1993.

[34] S. Das, D. Koutsonikolas, Y. Hu, and D. Peroulis, "Characterizing multi-way interference in wireless mesh networks." in *ACM Wintech*, 2005.

[35] J. Deng and Z. J. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," in *Proceedings of IEEE ICUPS'98*, October 1998.

[36] I. S. Department, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11–1997," 1997.

[37] R. Dhar, G. George, A. Malani, and P. Steenkiste, "Supporting Integrated MAC and PHY Software Development for the USRP SDR," *Networking Technologies for Software Defined Radio Networks, 2006. SDR '06.1st IEEE Workshop on*, pp. 68–77, Sept. 2006.

[38] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, "Truelink: A practical countermeasure to the wormhole attack," in *ICNP*, 2006.

[39] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks," UCLA, Tech. Rep. UCLA/CSD-TR 02-0013, 2002.

[40] M. Garetto, T. Salonidis, and E. W. Knightly, "Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks," in *Proc. of IEEE INFOCOM*, Barcelona, April 2006.

[41] M. R. Garey, R. L. Graham, and D. S. Johnson, "Some NP-complete geometric problems," in *Proc. 8th Annu. ACM Sympos. Theory Comput.*, 1976, pp. 10–22.

[42] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman, 1979.

[43] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric SINR," in *ACM MobiHoc'07*, 2007, pp. 100–109.

[44] C. Grassmann, A. Troya, M. Sauermann, M. Richter, and U. Ramacher, "Mapping Waveforms to Mobile Parallel Processor Architectures," in *Proc. SDR Technical Conference*, 2005.

[45] J. Gronkvist and A. Hansson, "Comparison between graph-based and interference-based STDMA scheduling," in *MobiHoc*, 2001, pp. 255–258.

[46] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.

[47] L. Hu and D. Evans, "Using directional antennas to prevent wormhole attacks," in *Network and Distributed System Security Symposium (NDSS)*, 2004.

[48] Y. C. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *INFOCOM*, 2003.

[49] IEEE Computer Society LAN/MAN Standards Committee, "802.15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANS)," 2003.

[50] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *MobiCom*, 2003, pp. 66–80.

[51] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel MAC protocol with receiver-based channel selection for multihop wireless networks," in *Proceedings of the 9th Int. Conf. on Computer Communications and Networks (IC3N)*, Phoenix, Oct. 2001.

[52] K. Jamieson, B. Hull, A. K. Miu, and H. Balakrishnan, "Understanding the Real-World Performance of Carrier Sense," in *E-WIND*, Philadelphia, PA, August 2005.

[53] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding congestion in ieee 802.11b wireless networks," in *ACM IMC*, 2005.

[54] A. Kashyap, S. Ganguly, and S. R. Das, "A Measurement-Based Approach to Modeling Link Capacity in 802.11-based Wireless Networks," in *ACM MobiCom*, 2007.

[55] I. Khalil, S. Bagchi, and N. B. Shroff, "LITEWORP: A Lightweight Countermeasure for the Wormhole attack in multihop wireless network," in *International Conference on Dependable Systems and Networks (DSN)*, Yokohama, Japan, 2005.

[56] I. Khalil, S. Bagchi, and N. Shroff, "MOBIWORP: Mitigation of the wormhole attack in mobile multihop wireless networks," in *Second International Conference on Security and Privacy in Communication Networks (SecureComm 2006)*, 2006.

[57] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part-I - carrier sense multiple access modes and their throughput-dely characteristics," *IEEE Transactions in Communications*, vol. COM-23, no. 12, pp. 1400–1416, 1975.

[58] A. Kochut, A. Vasan, A. Shankar, and A. Agrawala, "Sniffng out the correct physical layer capture model in 802.11b," in *Proc. IEEE ICNP*, Berlin, 2004.

[59] M. Kodialam and T. Nandagopal, "Characterizing the Capacity Region in Multi-Radio, Multi-Channel Wireless Mesh Networks," in *ACM MobiCom*, 2005.

[60] F. Kuhn and A. Zollinger, “Ad-hoc networks beyond unit disk graphs,” in *Proc. 2003 Joint Workshop on Foundations of mobile computing*, 2003, pp. 69–78.

[61] P. Kyasanur and N. Vaidya, “Capacity of multi-channel wireless networks: Impact of number of channels and interfaces,” in *Mobicom*, 2005.

[62] ——, “Routing and interface assignment in multi-channel multi-interface wireless networks,” in *WCNC*, 2005.

[63] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi, “An experimental study on the capture effect in 802.11a networks,” in *WINTECH*, 2007, pp. 19–26.

[64] ——, “An experimental study on the capture effect in 802.11a networks,” in *Proc. ACM WiNTECH*, 2007.

[65] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, “SODA: A Low-power Architecture For Software Radio,” in *Proc. Intl. Symposium on Computer Architecture*, 2006.

[66] R. Maheshwari, H. Gupta, and S. R. Das, “Multichannel MAC protocols for wireless networks,” in *Proc. IEEE SECON 2006*, Reston, VA, Sept 2006.

[67] R. Maheshwari, S. Jain, and S. R. Das, “A measurement study of interference modeling and scheduling in low-power wireless networks,” in *Proc. ACM SenSys*, 2008.

[68] ——, “On estimating joint interference for concurrent packet transmissions in low power wireless networks,” in *Proc. ACM WiNTECH*, 2008.

[69] M. K. Marina and S. R. Das, "A topology control approach to channel assignment in multi-radio wireless mesh networks," in *Proc. Broadnets Symposium*, 2005.

[70] M. A. Marsan and D. Roffinella, "Multichannel local area network protocols," *IEEE J. Selected Areas in Communications*, pp. 885–897, Nov. 1983.

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

[72] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh, "Partially-overlapped channels not considered harmful," in *Proc. ACM Sigmetrics*, 2006.

[73] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks," in *IEEE INFOCOM*, 2006.

[74] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan, "Load-aware spectrum distribution in wireless LANs," in *Proc. IEEE ICNP Conference*, 2008.

[75] Moteiv Corporation, *tmote sky: Ultra low power IEEE 802.15.4 compliant wireless sensor module*, San Fransisco, CA, November 2006.

[76] A. Muqattash and M. Krunz, "POWMAC: A single-channel power-control protocol for throughput enhancement in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, May 2005.

[77] R. Nelson and L. Kleinrock, "Spatial-TDMA: A collison-free multihop channel access protocol," *IEEE Transactions on Communication*, vol. 33, pp. 934–944, Sept. 1985.

[78] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "Softmac-flexible wireless research platform," in *In Proc. HotNets-IV*, Nov 2005.

[79] L. M. Ni and P. K. McKinley, "A survey of wormhole routing techniques in direct networks," *Computer*, vol. 26, no. 2, pp. 62–76, 1993.

[80] D. Niculescu, "Interference map for 802.11 networks," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2007, pp. 339–350.

[81] ——, "Interference map for 802.11 networks," in *Proc. IMC*, 2007.

[82] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *Proc. Internet Measurement Conference (IMC)*, 2005.

[83] ——, "Estimation of link interference in static multi-hop wireless networks," in *IMC*, 2005.

[84] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 2002.

[85] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer, "WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks," *NSDI*, 2007.

[86] R. Poovendran and L. Lazos, "A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks," *ACM Journal of Wireless Networks (WINET)*, 2005.

[87] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, "A general model of wireless interference," in *ACM MobiCom*, 2007.

[88] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, 2004.

[89] R. Raniwala and T. Chiueh, "Architechture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *INFOCOM*, 2005.

[90] A. Rao and I. Stoica, "An overlay MAC layer for 802.11 networks," in *MobiSys'05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, 2005, pp. 135–148.

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

[92] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," in *SIGCOMM*, 2006.

[93] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks," in *MobiHoc*, 2006, pp. 190–201.

[94] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-mac: a hybrid mac for wireless sensor networks," in *SenSys*, 2005.

[95] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-based characterization of 802.11 in a hotspot setting," in *ACM E-WIND*, 2005.

[96] J. Ryu, J. Lee, S.-J. Lee, and T. Kwon, "Revamping the ieee 802.11a phy simulation models," in *Proceedings of ACM MSWiM*, Vancouver, 2008.

[97] S.-L.Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Int'l Symp. on Parallel Architectures, Algorithms and Networks (I-SPAN)*, 2000.

[98] K. Sanzgiri, B. Dahill, B. Levine, and E. Belding-Royer, "A secure routing protocol for ad hoc networks," in *International Conference on Network Protocols (ICNP)*, November 2002.

[99] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *ACM Workshop on Wireless Security (WiSe 2003)*, September 2003.

[100] A. Scaglione and Y. W. Hong, "Opportunistic large arrays: Cooperative transmission in wireless multihop ad hoc networks to reach far distances," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, 2003.

[101] T. Schmid, O. Sekkat, and M. B. Srivastava, "An Experimental Study of Network Performance Impact of Increased Latency in Software Defined Radios," in *Proc. Intl. Symposium on Computer Architecture*, 2006.

[102] N. Shacham and P. King, "Architectures and performance of multichannel multihop packet radio networks," *IEEE Journal on Selected Areas of Communication*, vol. SAC-5, no. 6, pp. 1013–1025, 1987.

[103] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. ACM MobiCom*, 2006.

[104] J. Shi, T. Salonidis, and E. Knightly, "Starvation Mitigation Through Multi-Channel Coordination in CSMA Multi-hop Wireless Networks," in *Proc. ACM MobiHoc*, 2006.

[105] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proc. ACM MobiHoc*, 2004.

[106] J. So and N. H. Vaidya, "Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *Proc. ACM MobiHoc*, 2004.

[107] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," in *SenSys '06*, 2006, pp. 237–250.

[108] K. Srinivasan and P. Levis, "RSSI is Under Appreciated," in *EmNets 2006*, 2006.

[109] A. P. Subramanian, H. Gupta, S. R. Das, and M. M. Buddhikot, "Fast spectrum allocation in coordinated dynamic spectrum access based cellular networks," in *Proc. IEEE DySPAN Symposium*, 2007.

[110] Z. Tang and J. J. Garcia-Luna-Aceves, "Hop-reservation multiple access (hrma) for multichannel packet radio networks," in *Proceedings of the IEEE IC3N'98, Seventh International Conference on Computer Communications and Networks*, October 1998.

[111] *CC2420 Radio Datasheet*, 1st ed., Texas Instruments, October 2005.

[112] 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*, vol. COM-23, no. 12, pp. 1417–1433, 1975.

[113] T. Todd and J. Mark, "Capacity Allocation in Multiple Access Networks," Communications," *IEEE Transactions on Communications*, vol. 33, no. 11, 1985.

[114] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *IEEE Infocom*, 2001.

[115] B. S. V. Gambiroza and E. Knightly., "End-to-end performance and fairness in multihop wireless backhaul networks," in *Proc. of ACM Mobicom*, 2004.

[116] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *MobiCom*, 2006, pp. 262–273.

[117] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, New York, NY, USA, 2004, pp. 51–60.

[118] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler, "Exploiting the capture effect for collision detection and recovery," in *IEEE EmNetS-II*.

[119] C. Wu and V. Li, "Receiver-initiated busy-tone multiple access in packet radio networks," in *SIGCOMM '87: Proceedings of the ACM workshop on Frontiers in computer communications technology*. New York, NY, USA: ACM Press, 1988, pp. 336–342.

[120] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile

Ad Hoc Networks," in *Proc. International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN)*, 2000.

[121] X. Wu and R. Srikant, "Bounds on the capacity region of multi-hop wireless networks under distributed greedy scheduling," in *Proc. IEEE INFOCOM*, 2006.

[122] L. Yang, L. Cao, H. Zheng, and E. Belding, "Traffic-aware dynamic spectrum access," in *Proc. The Fourth International Wireless Internet Conference (WICON)*, 2008.

[123] X. Yang, N. Vaidya, and P. Ravichandran, "Split-Channel Pipelined Packet Scheduling for Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 3, 2006.

[124] Y. Yang, J. Wang, and R. Kravets, "Distributed optimal contention window control for elastic traffic in single-cell wireless LANs," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1373–1386, 2007.

[125] H. Yin and S. Alamouti, "OFDMA: A Broadband Wireless Access Technology," in *Proc. IEEE Sarnoff Symposium*, 2006.

[126] Y. Yuan, P. Bahl, R. Chandra, P. A. Chou, I. Ferrell, T. Moscibroda, S. Narlanka, and Y. Wu, "Kognitiv networking over white spaces," in *Proc. IEEE DySPAN*, 2007.

[127] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu, "Allocating dynamic time-spectrum blocks for cognitive radio networks," in *Proc. ACM MobiHoc Symposium*, 2007.

[128] G. Zhou, T. He, J. Stankovic, and T. Abdelzaher, "RID: Radio Interference Detection in Wireless Sensor Networks," in *IEEE Infocom*, 2005.

[129] X. Zhou, S. Gandhi, S. Suri, and H. Zheng, "eBay in the Sky: Strategy-Proof Wireless Spectrum Auctions," in *Proc. ACM MobiCom*, 2008.

[130] X. Zhou and H. Zheng, "TRUST: A General Framework for Truthful Double Spectrum Auctions," in *Proc. IEEE INFOCOM*, 2009.

[131] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *IEEE SECON 2004*, 2004, pp. 517–526.