AN ABSTRACT OF THE THESIS OF

Megha Maiya for the degree of Master of Science in Computer Science presented on July 27, 2010.

Title:

iMAC: Improved Medium Access Control for Multi-channel Multi-hop Wireless Networks

Abstract approved: ______________________________

Bechir Hamdaoui

Trends in wireless networks are increasingly pointing towards a future with multi-hop networks deployed in multi-channel environments. In this thesis, we present the design for *iMAC*—a protocol targeted at medium access control in such environments. *iMAC* uses control packets on a common control channel to faciliate a three-way handshake between the sender and receiver for every packet transmission. This handshake enables the sender and receiver to come to consensus on a channel to use for data transmission and also signals to neighboring nodes about the contention on that channel. *iMAC* then uses a mechanism similar to 802.11 for data communication. Our evaluation of *iMAC* shows that it provides significant gains in throughput in comparison with uninformed channel selection, especially when contention for channel bandwidth is neither too low nor too high; intelligent selection of channels by *iMAC* is necessary to harness available bandwidth resources in the presence of medium levels of contention.

iMAC: Improved Medium Access Control for Multi-channel Multi-hop Wireless Networks

by

Megha Maiya

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented July 27, 2010
Commencement June 2011

Master of Science thesis of Megha Maiya presented on July 27, 2010.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Megha Maiya, Author

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 – Introduction

Over the last few years, wireless networks have become ubiquitous. Wireless devices, services and applications have penetrated nearly every aspect of our lives. We have witnessed an explosive growth in demand for wireless resources to cater to all the new wireless services and applications being churned out every year. Wireless communications, including 3G and 4G information services, have seen remarkable expansion over the last few years, as millions of users buy smart-phones which are not only used for voice communication but also as a device to access the Internet. In addition, the US has introduced the National broadband plan [5] to enable IT-based economic recovery. All of these trends have made the wireless spectrum a limited resource.

Various surveys conducted over the last few years have shown that a significant fraction of the wireless spectrum is either unused or underutilized. Even in cities such as New York and Washington, D. C., less than 20% of the available spectrum is actually being utilized most of the time [6]. These numbers imply that ample spectrum opportunities are available both along the time and frequency dimensions to satisfy newer wireless resource requirements that arise [10]. Such opportunities, often referred to as white spaces, are unused portions of the UHF spectrum. One example is the 180 MHz available bandwidth from channel 21 (512 MHz) to 51 (698 MHz) with the exception of channel 37 [3].

Federal Communications Commission (FCC), National Telecommunication and Information Administration (NITA), and other governing bodies and organizations are actively engaged in combating the looming spectrum crisis. Reclaiming underutilized spectrum, reallocating misallocated spectrum, and redrawing the spectrum map are a few of the measures FCC is currently undertaking or planning to undertake. As a result of these efforts to address spectrum issues, the FCC adopted on November 4th 2008 rules for unlicensed use of the television white spaces, opening them up for Secondary Users (SUs) complying with these rules [8]. These unlicensed devices can utilize the spectrum only when and where they are unused by the licensed or Primary Users (PU). For a detailed report on the rules that safeguard incumbent services against harmful interference, please refer to the Second report and order issued by the FCC [8].

Devices capable of complying with the rules set by FCC typically rely on cognitive radios. These radios are capable of empowering devices to hop from one frequency band to another depending on availability of the spectrum for SU use. However, this new capability in hardware needs to be coupled with next generation software in order to take advantage of the availability of multiple spectral opportunities. Specifically, in case of Medium Access Control (MAC) protocols for wireless networks, the design of traditional IEEE 802.11 needs to be upgraded to be able to co-ordinate spectrum access in this new scenario.

A fundamental capability that 802.11 lacks is the ability to mediate access of multiple frequency bands by wireless devices and nodes in a wireless network. Also, the presence of multiple channels complicates further the well-known problems of hidden and exposed terminals in case of ad-hoc networks. In this thesis, we describe our design of a new MAC protocol that enables the access of multiple channels by nodes in multi-hop ad-hoc networks.

Our Improved Medium Access Control, referred to as *iMAC* through the rest of the thesis, leverages many pieces of prior work in its approach for tackling the challenges associated with medium access control in multi-channel networks. Our goal is to develop a simple protocol which is completely decentralized and does not involve the overhead of synchronizing nodes with respect to time. Every node maintains a snapshot of traffic in its vicinity on all data channels. This information is not perfect; rather, it is maintained solely based on the node's perception of activities in the network. This state is maintained by overhearing nodes communicate on a common control channel. Nodes use this information in deciding on a channel to communicate on. The common channel is ear-marked for communication between senders and receivers to exchange handshake signals and come to consensus on a channel to be used for communication. This channel selection is based on approximate information about channel availability that each node has locally.

To evaluate *iMAC*, we run simulations using our implementation of *iMAC* in NS-2. We compare performance of *iMAC* with a protocol, which we call UCS-MAC(Uninformed Channel Selection-MAC). UCS-MAC is identical to *iMAC* except that nodes pick channels to communicate on at random instead of using channel availability information. Our experiments help identify optimal network parameters, i.e., number of channels, network load, hop length, and node density, to obtain maximum benefit from the additional network resources available. We find that *iMAC* yields up to 30% gain in average throughput in comparison with a non-intelligent multi-channel protocol.

Our primary contributions in developing *iMAC* are as follows:

1. Providing a simple and decentralized solution for the fundamental problem of designing a MAC protocol for multi-channel, multi-hop networks.

2. Significantly improving the performance of ad-hoc networks and wireless mesh networks by enabling them to utilize additional bandwidth available from multiple channels.

The rest of this thesis is organized as follows. In Chapter 2, we lay down our design objectives in developing *iMAC*. Chapter 3 provides a detail description of *iMAC*'s design. In Chapter 4, we present evaluation results of our NS-2 based simulations. Related work is discussed in Chapter 5, and finally, we conclude in Chapter 6.

Chapter 2 – Design goals

Solving the problem of managing access of multiple frequency bands/channels is a fundamental problem within the realms of spectrum agile networks. In developing a protocol for multi-channel networks we set ourselves the following design guidelines.

- **Inexpensive:** *iMAC* should make use of available hardware resources for a wireless device and not require additional hardware support as part of its design. Wireless devices capable of communicating across multiple channels are empowered by cognitive radios to do so. The design of the protocol should be able to harness bandwidth on multiple channels by only making use of these radios. It is common practice for wireless devices to be equipped with a single half-duplex radio transceiver for communication. A few design approaches require devices to use at least two or three of such transceivers instead. Additional radios are needed primarily because channel access decisions require knowledge of usage information across the available set of channels; it is impossible to get perfect usage information without having one radio dedicated for the purpose of monitoring the use of each channel. Our design needs to be able to tackle this problem without incurring additional cost resulting from use of additional hardware.

- **Simple in Design:** The protocol should be simple in design. There should be no overhead or complexity of periodically synchronizing nodes across channels with respect to time or information about availability of a channel. Global time synchronization is a design decision taken in many papers discussed in Chapter 5. We seek to avoid this overhead. Given the chaotic nature of wireless networks, ensuring that nodes have a synchronized perception of the network and available spectrum is unlikely to add significant value to channel access decisions.

- **Leverage from 802.11:** Traditional 802.11 has efficient collision avoidance mechanisms. The problem of hidden terminals is handled well in the existing design. However, these problems become more complex in multi-channel environments. Rather than developing new techniques from scratch, we seek to apply the techniques used in traditional 802.11 to suit multi-channel needs.

- **Efficient:** The channel selection scheme employed as part of the design should result in significant gains in achieved throughput when compared to a scheme that randomly assigns channels to nodes. Therefore, it is important to use channel status and availability information in performing channel selection. Further, our solution should be applicable to communication that occurs across multiple hops.

Chapter 3 – *iMAC* protocol

In this chapter, we describe our design of *iMAC* motivated by the the design goals identified in the previous chapter. *iMAC* leverages medium access techniques from IEEE 802.11, which works in a single channel environment. We reuse the collision avoidance techniques and the sequence of control frames exchanged in 802.11. In *iMAC*, we augment these techniques to scale to multiple data channels.

In *iMAC*, nodes listen on a common control channel when not actively participating in data communication. When a neighboring sender and receiver wish to exchange data, they use the common control channel to come to a consensus on which channel to use for the data transmission. Both nodes then switch to the agreed upon channel and use 802.11 to negotiate the data transfer.

Our key insight is that when communicating nodes have the option to choose from a number of channels to exchange DATA and ACK frames or broadcast DATA frames on, they have to do so amidst other nodes communicating over these channels. Choosing a channel already being used for communication by several other nodes will result in throughput close to that achieved in a single channel environment. This would render the additional bandwidth available for communication in the multi-channel environment under-utilized and hence wasted. Therefore, it is vital for nodes to mutually agree upon a channel that will yield the maximum throughput.

Channel	**Nodes**
Channel 1	(A:t1), (B:t2)
Channel 2	(C:t3), (D:t4), (E:t5)
...	...
Channel n	...

Figure 3.1: Structure of Channel Information Table

3.1 Channel selection

To choose from a set of channels, a node needs to be informed about the availability of all channels. This leads to the requirement that every node maintain some kind of status information for every channel. In other words, every node needs to maintain a snapshot of the spectrum in its vicinity. We call this snapshot maintained by every node as the Channel Information Table (CIT).

3.1.1 CIT

CIT is a table maintained and referred to by every node in the network. The primary purpose CIT serves is to enable nodes to get a fair idea of which channel has the highest availability.

Figure 3.1 shows an example CIT with some sample table entries. For every channel in the network, the CIT contains a pair of values for every neighboring node that is known to be using the channel—a node identifier (nodeID) and the time (t) at which that node switched to the said channel. A channel ID associated with no node-time tuples implies that the node is not aware of any of its neighboring nodes occupying the corresponding channel. This information is not completely accurate; we explain later in this chapter why and how to cope with problems that might arise due to this.

When two nodes have to select the most suitable channel for their communication they employ the channel selection algorithm over two stages at either end. First, the sender initiates channel selection by querying its CIT for a ranked list of channels based on channel availability. The sender node queries its CIT for an ordered list of channelIDs; channelIDs with shorter lists, i.e., with fewer Node-Time tuples are assigned higher priority over IDs with longer lists. The sender sends this ordered list of channels over to the receiver.

At the receiver's end, it queries its own CIT to determine the most suitable channel for communication given the sender's ranking of channels. The receiver matches its ordered channel list with the sender's list and chooses the first channel in the sender's list having the highest ranking as per its own rank assignments. In case of a set of channels having the same and least number of Node-Time tuples, the channelID in the sender's list with the highest rank amongst these set of channels is chosen. Essentially, if the receiver has no preference over a set of channels, the sender's ranking of the channels is used to attach priority between these channels.

At the end of this channel selection phase, the communicating nodes have mutually agreed upon a data channel. They will now switch to the chosen data channel to begin data exchange.

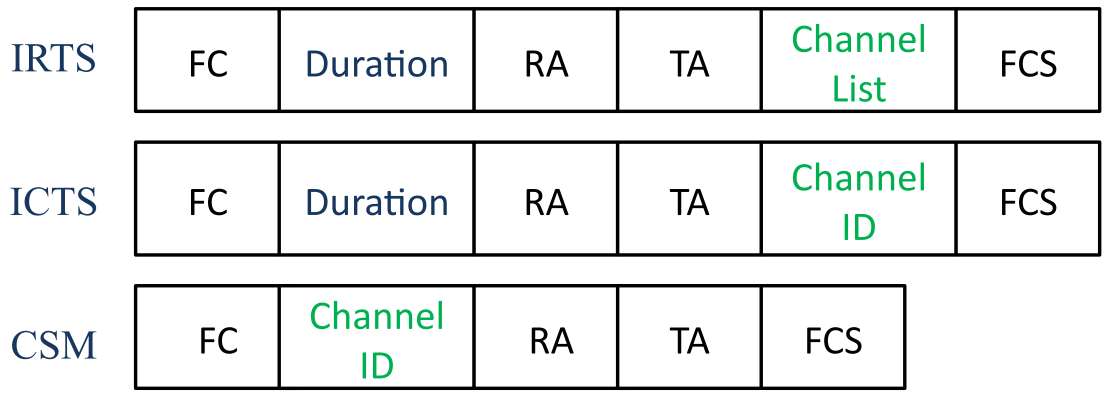


Figure 3.2: Structure of control frames used in *iMAC* protocol.

3.1.2 Control Frames

To enable the channel selection stage, *iMAC* uses the exchange of new control frames between the communicating nodes while on the common channel. The basic idea of ensuring the sender and receiver nodes mutually agree on a common channel is similar to the way communication parameters are negotiated during the establishment of a TCP connection. Here too we rely partly on a three-way handshake stage on the control channel to let the sender and the receiver come to a consensus on the channel to switch to for communication. This three-way handshake also serves the purpose of informing the neighbors of the sender and receiver of the channel chosen for communication, so that they can update their respective CITs.

The sender initiates the channel selection process by sending a list of channels over to the receiver. We introduce a control frame IRTS—*iMAC*'s Request To Send—for this communication. The receiver responds by sending the identifier of the chosen channel on the control frame ICTS—*iMAC*'s Clear To Send. Lastly, the sending node completes the three-way handshake by sending across to the receiver, and in the process to all its neighbors, a message summarizing the channel chosen for communication. We introduce a control frame called CSM—Channel Selection Message—for this purpose.

Figure 3.2 shows the structure of an IRTS frame. Majority of the fields have the same name and serve the same purpose as in the Request To Send (RTS) control frame of 802.11 [1]. The Frame Control (FC) field remains unchanged. It contains information such as the protocol version, type of message being sent, fragmentation details, and Wired Equivalent Privacy (WEP) information. Duration field is equal to the time to transmit one ICTS, one CSM, and two Short Interframe Space (SIFS) intervals. The Receiver Address (RA) and Transmitter Address (TA) fields remain unchanged. Channel List is a ranked list of probable channels the transmitter proposes to communicate with the receiver. Frame Check Sequence also remains unaltered.

The structure of an ICTS frame, as shown in Figure 3.2 is similarly based on the Clear To Send (CTS) frame of 802.11 [1]. Here too, Frame Control, Receiver Address, and Frame check Sequence fields have not been modified. Duration field is the time required to send one CSM frame and one SIFS interval. ChannelID is the identifier of the channel selected by the receiver at the end of stage two of channel selection process. Also included is the Transmitter Address field which contains the address of the receiver node. This is done in order to accommodate the second purpose of the three-way handshake serves, which is to inform neighbors of the receiver about the decision of choosing a particular channel for communication by the transmitter (TA)–receiver (RA) pair. Later in this chapter, we explain how these neighboring nodes make use of an ICTS frame to update their CITs.

Lastly, Figure 3.2 shows the structure of a CSM frame. It consists of Transmitter and Receiver Addresses (TA, RA) and the chosen channel (ChannelID) for communication between them. Again the Frame Control and Frame Check Sequence fields are similar to ones in IRTS and ICTS frames. When the transmitter node receives the ICTS frame sent by the receiver, it completes the three-way hand-shake by sending a CSM frame containing the channelID present in the ICTS frame. CSM, apart from informing the receiver node that its ICTS successfully reached the sender, serves as a way to inform the sender's neighbors about the decision of the pair of nodes to use a certain channel for communication.

3.1.3 Updating CIT

Nodes are tuned in to the common control channel when they are not involved in DATA-ACK exchange on a chosen data channel. This is when nodes hear IRTS, ICTS, or CSM messages addressed for a neighboring node. The information contained in these messages lead to updates or inserts to the node's CIT. A node inserts Node-Time tuples into the list corresponding to the ChannelID received as part of either an ICTS or a CSM message.

When a receiver node decides on a data channel as part of the channel selection process, it sends an ICTS message to the sender node conveying its decision. At this point, it is established that the pair of nodes wanting to communicate will use a particular channel for their communication. Therefore, a node at single-hop distance from the receiver node can make note of the fact that the node pair specified in the ICTS it received decided to switch to the channel whose ID is specified in the message. The receiver's neighbors insert an entry for both the transmitter and receiver nodes in to their respective CITs against the chosen ChannelID.

A node needs to know the current load on a channel to be able to pick a channel that is least loaded for its own communication, for which it uses the snapshot of channel usage it maintains in its CIT. It is therefore very important for a node to ensure its snapshot is current. In order to time out Node-Time tuples from the CIT, we use a system-wide constant T as the maximum permissible time for a pair of nodes to spend on their selected channel for exchanging DATA and ACK. The value for T in our current specification in *iMAC* is informed by our observations from several simulation runs. Whenever a node performs an update on its CIT, it weeds out all node entries associated with timestamps older than T in comparison with the current time. This helps maintain the freshness of the channel usage snapshot stored in CIT.

How often should a node update its CIT? It suffices to do so every time a node queries its table for a list of channels that are lightly loaded. In other words, when the sender node starts the channel selection process, it should clean up its CIT before generating its list of channel preferences which will be sent in the IRTS to the receiver. Also, the receiver node should update its table just before accessing its contents to come up with a single channel preference for communication after incorporating the sender's preferences. When a sender node receives an ICTS message, it is all set to switch to the chosen channel for communication. However, the sender's neighbors, not within vicinity of the receiver, do not know yet about the channel chosen for data exchange. When the sender sends a CSM message to complete the three-way handshake, neighbors of the sender other than the intended receiver save necessary information contained in the message. They insert a Node-Time tuple for either of the nodes sent in the CSM message, against the entry for the channelID included in the message. Similar to the handling of an ICTS message, nodes save the time at which they received information about a pair of nodes wanting to communicate on a channel.

3.2 Common Control Channel

When a node receives a packet to be transmitted to another node, it initiates the channel selection process on the common channel, before actually exchanging DATA and ACK with the intended receiver. Whether or not the actual communication happens depends on whether the IRTS–ICTS–CSM exchange on the common control channel succeeds. If the three-way handshake fails, no DATA–ACK exchange happens on the chosen data channel. This being the case, employing suitable carrier sense and collision avoidance techniques in accessing the common control channel becomes imperative to ensure that the maximum possible performance is achieved.

We leverage from 802.11 mechanisms of virtual carrier sense to solve the hidden terminal problem. The control frames IRTS and ICTS function as ways to control access of the common channel medium by nodes, apart from serving as ways to communicate channel preferences to all nodes. Just as in RTS and CTS frames of 802.11, a neighboring node receiving an IRTS or ICTS will set its Network Allocation Vector (NAV) to the duration field in the frames. They will defer accessing the common control channel for the time needed to exchange ICTS and CSM messages as specified in the duration fields of an IRTS or ICTS packet. Like in 802.11, we use binary exponential backoff times chosen randomly from a collision window having the same minimum and maximum window limit sizes as 802.11. One important difference is the time for which the common channel is reserved by sending an IRTS and ICTS packet; time needed only for the channel selection process so that the transmitter and receiver nodes mutually decide on a channel. The duration fields do not include the time required for exchanging DATA and ACK as in the case of traditional 802.11. Also, similar to 802.11, in case of failure to receive an ICTS, a sender attempts to retransmit the IRTS seven times before the data packet is discarded.

3.3 Data Channel

Figure 3.3 shows the sequence of exchange of the control frames on the common control channel. The diagram also shows exchange of RTS and CTS packets between the transmitting and receiving nodes, on the chosen channel before exchange of DATA and ACK.

This brings us to the other important design challenge; that of nodes not being able to listen on the common control channel at all times. Our assumption of a single interface per node results in nodes being absent from the common channel for significant lengths of time, when performing data communication on other channels. This results in nodes not updating their CITs when and if their neighbors completed the IRTS–ICTS–CSM three-way handshake and decided on choosing a particular channel for communication, during their absence from the control channel. Their CITs will not be up-to-date and

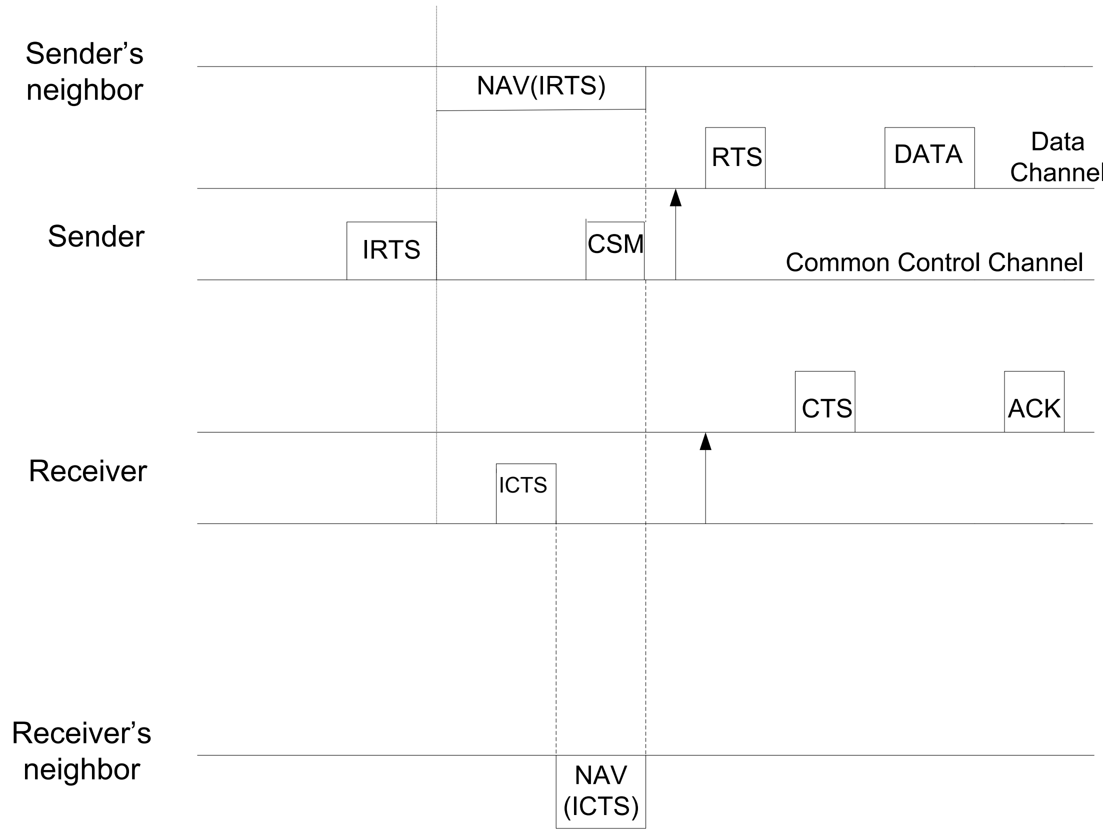


Figure 3.3: Sequence diagram for *iMAC* protocol.

fail to represent the most current snapshot of channel loads. When the nodes initiate communication with one of their neighbors at a future point in time, their channel selection algorithm will have to make use of inaccurate data from their respective CITs. This naturally leads to probable selection of a channel which is perhaps more loaded than perceived as per information in the CITs.

Such inaccuracies in CITs occur more often than not when nodes are constantly involved in data communication, and thus, routinely absent from the common channel. Hence, nodes have to deal with a second level of medium access contention with other nodes using the channel they switch to. We handle this situation by making the nodes follow the traditional 802.11 medium access mechanism to access the chosen channel once they switch to the channel of their choice, as shown in the sequence diagram (Figure 3.3).

The nodes, while on the chosen channel, keep counting down the system wide time constant T, in order to ensure that they do not stay longer than that on the chosen channel. After completing exchange of DATA and ACK on the chosen channel, or running out of number of retransmit attempts for DATA or RTS, or counting down to zero time on the channel, whichever happens earlier, the nodes switch back to and continue to listen on the common control channel.

Chapter 4 – Evaluation

In this chapter, we evaluate the *iMAC* protocol along various dimensions—number of channels, network load, node density, and hop length. In each case, we compare the throughput obtained using *iMAC* with an alternative protocol UCS-MAC, which is identical to *iMAC* except that the receiver returns a channel chosen at random in the ICTS message. This is in contrast to *iMAC*, in which nodes use their local impressions of channel availability in making channel selection decisions.

4.1 Simulation setting

We evaluate *iMAC* based on simulations with ns version 2.31. We augment the basic distribution of ns with the contributed codebase that helps simulate a multi-channel environment. In each of our experiments, we set the workload to be a certain number of

Notation	**Parameter**
N	Number of nodes
A	Area of environment
m	Number of channels
η	Network load
F	Number of flows
R_i	Bandwidth rate of i^{th} flow
h_i	Hop length of i^{th} flow
T	Transmission range

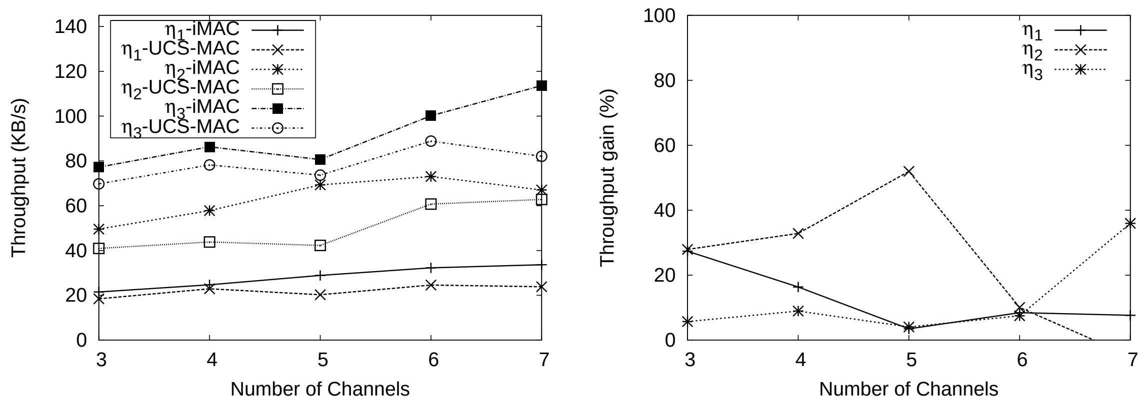
Table 4.1: Variable parameters of each simulation run.

constant bitrate (CBR) UDP flows that all start roughly in the beginning of the simulation. We then run this simulation for a period of 1500 seconds and measure the aggregate throughput observed across all flows over the entire duration of the simulation. In all simulations, we keep the bandwidth of each channel constant at 1 Mbps. The various parameters in each simulation run are explained in Table 4.1.

To define network load η, we seek to use a metric that is comparable across different network topologies and environments with different number of channels. Rather than using the number of flows or the aggregate bandwidth across all flows as the metric, we define the normalized network load as follows. When the area of the topology is A and the transmission range is T, there can be at most $\frac{A}{(\pi \cdot T^2)}$ number of hops active at any point in time on one channel; within any circle of radius of T, at most one communication can be active. Therefore, the maximum throughput a network can support is $(m \cdot C \cdot \frac{A}{(\pi \cdot T^2)})$, where C is the capacity of a single data channel. On the other hand, the load imposed on the network by F flows, where the i^{th} flow has a bandwidth rate of R_i and uses a path of h_i hops is $(\sum_{i=1}^{F} h_i \cdot R_i)$. Combining both of these, we define our metric for normalized network load as follows.

$$\eta = \frac{\sum_{i=1}^{F} h_i \cdot R_i}{m \cdot C \cdot \frac{A}{(\pi \cdot T^2)}} \tag{4.1}$$

We next evaluate the benefits of *iMAC* by comparing the aggregate throughput with *iMAC* with the corresponding result with UCS-MAC. For any given simulation, if the aggregate throughputs measured with *iMAC* and UCS-MAC are I and R, we compute the *throughput gain* with *iMAC* as $\frac{(I-R)}{R}$. We compute throughput or throughput gain



(a) (b)

Figure 4.1: (a) Throughput with *iMAC* and UCS-MAC, and (b) throughput gain of *iMAC* over UCS-MAC, as a function of m for different values of η. N = 50, A = 1500m × 1500m, average hop length is between 4 and 5.

for a particular combination of parameters as the average over 30 samples obtained from 30 different settings which satisfy that combination of parameters. Throughout our evaluation, we keep transmission range constant at 250m.

4.2 Impact of number of channels

First, we study *iMAC*'s benefit over random channel selection as a function of number of channels. We fix N at 50 and A at 1500m × 1500m. We then vary m from 3 to 7. In each case, we consider three load values(approx.): $\eta_1 = 0.2$, $\eta_2 = 0.5$, and $\eta_3 = 0.8$.

Figure 4.1 plots throughput and throughput gain as a function of m. First, in Figure 4.1(a), we observe that in all cases *iMAC* yields significantly higher average throughput than UCS-MAC. Second, in Figure 4.1(b), we observe that as the number of channels in the network increases, correspondingly the network load at which the highest value of

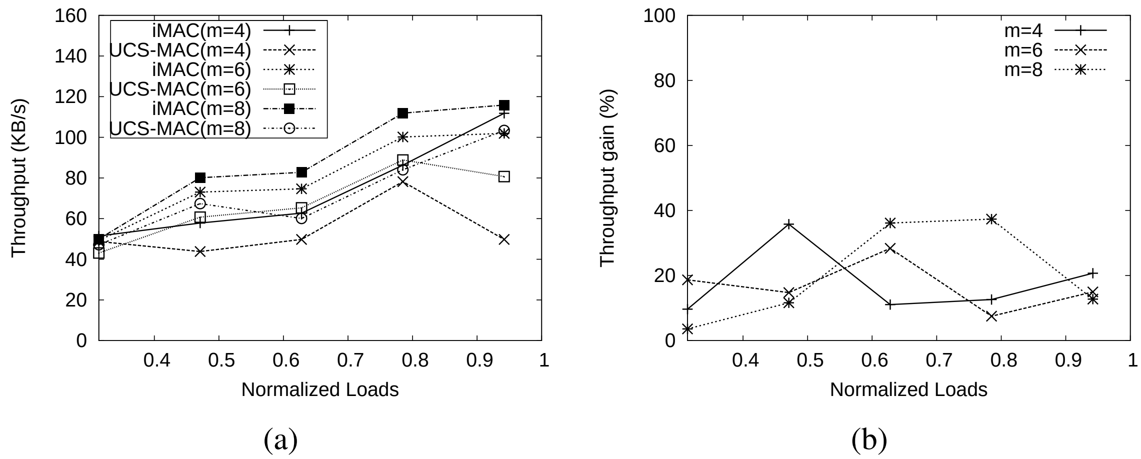
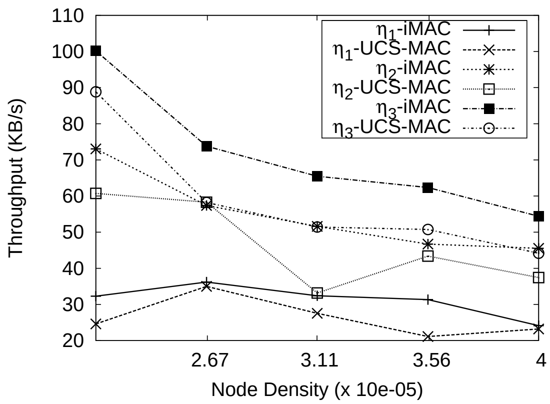


Figure 4.2: (a) Throughput with *iMAC* and UCS-MAC, and (b) throughput gain of *iMAC* over UCS-MAC, as a function of η for different values of m. N = 50, A = 1500m $\times$ 1500m, average hop length is between 4 and 5.

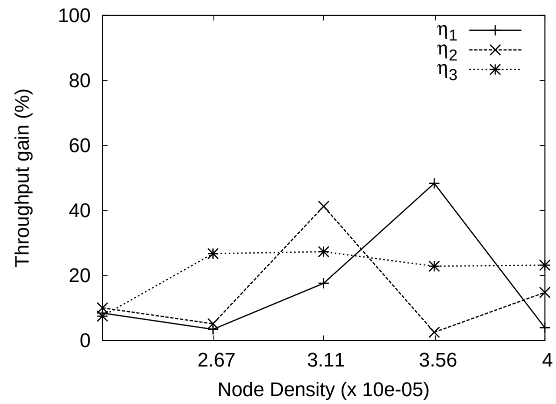
throughput gain is measured increases. This is because when there is high contention for bandwidth in the network, good utilization of available bandwidth is obtained even with random channel selection. High contention occurs either when the number of channels is low or the network load is high. As a result, there is not much throughput gain to be obtained using *iMAC* under high contention. Therefore, for every value of number of channels, there exists a different value for network load at which intelligent channel selection by *iMAC* is able to harness the optimal bandwidth for use by the flows in the network.

4.3 Impact of network load

Next, we seek to understand *iMAC*'s performance under different load regimes. In this case, we again fix N at 50 and A at 1500m $\times$ 1500m. We consider five different values



(a)



(b)

Figure 4.3: (a) Throughput with *iMAC* and UCS-MAC, and (b) throughput gain of *iMAC* over UCS-MAC, as a function of N and hence the node density for different values of η. m = 6, A = 1500m $\times$ 1500m, average hop length is between 4 and 5.

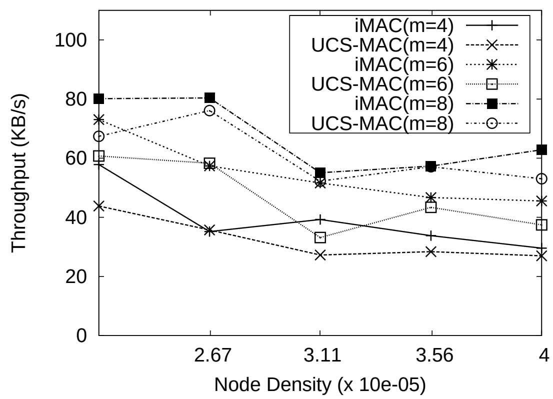
of load in the range $(0, 1)$, and in each case consider three different values for m—4, 6, and 8.

Figure 4.2 shows the variation of throughput and throughput gain across different load values. Again, in Figure 4.2(a), we see that *iMAC* consistently provides better throughput than UCS-MAC. Second, in Figure 4.2(b), we see that the value of m at which the highest throughput gain is measured varies with the value of network load; higher the load, greater the number of channels at which the best gain is obtained. As before, this is because there is not much scope for channel selection when the network is under high contention. For each value of network load, there is a particular value for the number of channels at which the benefits of intelligent channel selection are best seen.

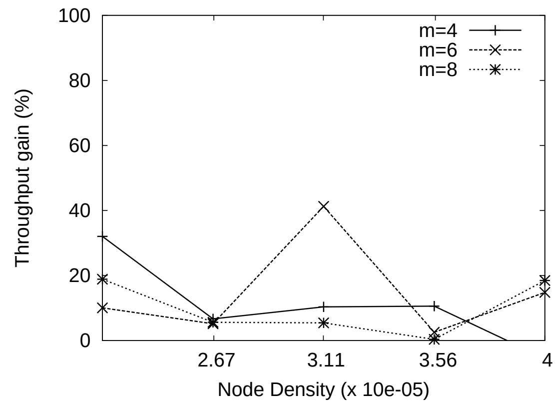
4.4 Impact of node density

As we have seen in the above experiments, the throughput gains obtained with *iMAC* are dependent on the level of contention in the network. Apart from number of channels and network load, another parameter that determines the level of contention is node density ($\frac{N}{A}$), i.e., the number of nodes per unit area of the topology. We vary node density by keeping the area A constant at 1500m $\times$ 1500m and vary the number of nodes in the range 50–90. For each combination of A and N, we consider three different values for η and three different values for m.

Figure 4.3 plots throughput and throughput gain as a function of the node density. First, in Figure 4.3(a), we observe that in all cases *iMAC* produces higher average throughput than UCS-MAC. We observe that the throughput achieved decreases with increasing node density; higher node density causes increased interference for every node. Second, in Figure 4.3(b), we observe that as the node density increases, the value of the network load at which gain in throughput measured is highest, decreases. This is because contention in the network is a function of network load and interference caused by node density. Both add to the contention factor as they increase. The optimal node density for each value of network load for a given number of channels varies with the load. At low network loads, the use of intelligent channel selection provided by *iMAC* becomes apparent when there is high interference at which point the gain is maximum. At lower interference levels, intelligent channel selection becomes insignificant as the network load is already low and hence there is not much gain observed. Similarly, for higher network loads, the peak in gain is witnessed at lower node densities. At higher



(a)

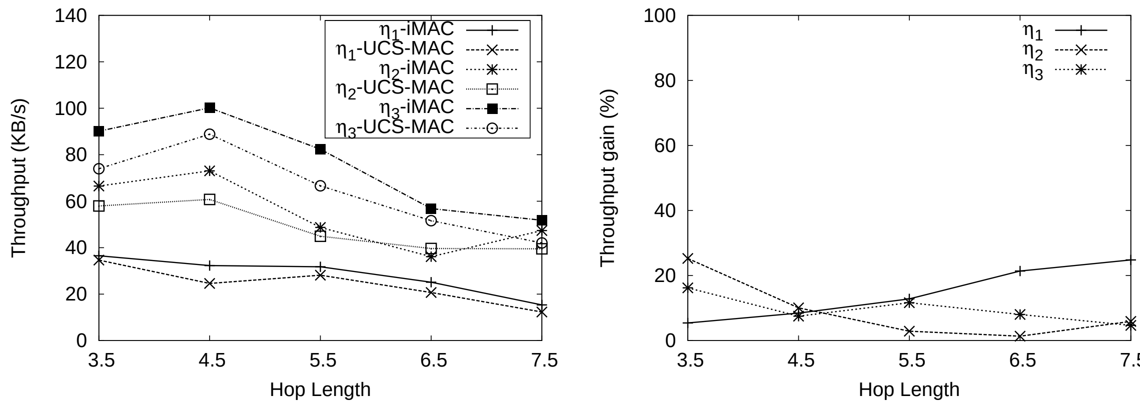


(b)

Figure 4.4: (a) Throughput with *iMAC* and UCS-MAC, and (b) throughput gain of *iMAC* over UCS-MAC, as a function of N and hence the node density for different values of m. η = 0.471, A = 1500m × 1500m, average hop length is between 4 and 5.

node densities, due to heavy contention both because of higher load and increased interference, UCS-MAC performs almost as well as *iMAC*.

In Figure 4.4, we plot throughput with either protocol as a function of node density across three different values for the number of channels, with a fixed network load. Figure 4.4(a) again shows that in a network with multiple channels *iMAC* is capable of achieving higher average throughputs when compared to UCS-MAC. We also see that the throughput gained increases with the increase in number of channels. However, there is a steady decline in the throughput achieved as the number of nodes increase. In Figure 4.4(b), we observe the impact of node density and number of channels on the gain in throughput achieved by *iMAC* over UCS-MAC. Again, as in the previous graph, the peak in throughput gain is at different node densities for different number channels. The highest gain in each case is seen at medium network loads. In addition, we observe that networks provisioned with a large number of channels tend to also display significant



(a) (b)

Figure 4.5: (a) Throughput with *iMAC* and UCS-MAC, and (b) throughput gain of *iMAC* over UCS-MAC, as a function of average Hop Length for different values of η. m = 6, A and N are varied for every hop length.

gains at higher interference. As is seen with 6 and 8 channels, the gain improves again as the interference increases. This is because, at higher interference there is increased impact of conversations in the vicinity of the node on the node's communication; there is scarcity of bandwidth. If more channels are provided to offset this effect, intelligent channel selection provided by *iMAC* helps achieve more efficient sharing of the channel resource.

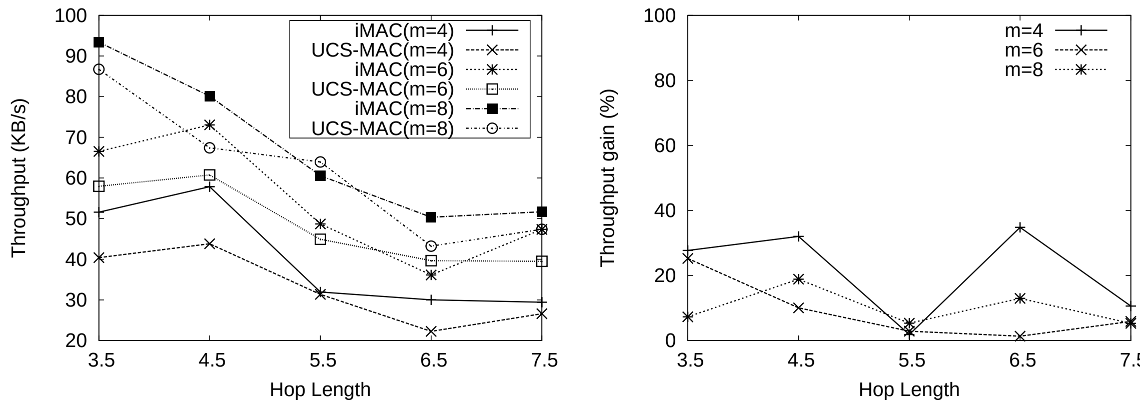
4.5 Impact of hop length

We are also interested in studying the impact of hop length on achievable performance. We ensure that the node density remains constant while testing the impact of hop length, so as not to cause any change in the level of interference every node faces in its transmission. We vary the hop length by increasing the area and keeping the node density constant. For this, we vary area A in the range 1250m $\times$ 1250m to 2500m $\times$ 2500m and correspondingly vary the number of nodes in the range 35–140. This results in variation of average hop length from 3.5 to 8.5. For each hop length, we consider three different values for η and three different values for m.

Figure 4.5 is a plot of throughput and throughput gain over average hop lengths for different values of network load ranging from low to high. First, Figure 4.5(a) shows that in all network load conditions and topologies with flows of different hop lengths, *iMAC*-enabled networks achieve significantly higher throughput in comparison with UCS-MAC. We also see that increase in hop length impacts the achieved throughput inversely.

Figure 4.5(b) again studies the gain in throughput *iMAC* can achieve over UCS-MAC with variation in hop lengths. Increase in hop length contributes to increase in packet loss rate resulting in decrease in achievable throughput. This, as in the case of interference, is inversely proportional to the throughput. We see that at lower loads the gain achieved is higher at larger hop lengths. The benefit of employing intelligent channel selection is emphasized at larger hop lengths. On the other hand, medium and high network loads peak sooner in the graph. At high loads intelligent channel selection does not add much value in case of larger hop lengths.

Figure 4.6 shows the variation of throughput and throughput gain across different values of average hop lengths. The network load is kept constant and the measurements are done for three different number of channels. Figure 4.6(a) shows that the throughput achieved by *iMAC* is greater than that achieved by UCS-MAC at almost all combinations of number of channels and average hop lengths. There is a decline in the throughput



(a) (b)

Figure 4.6: (a) Throughput with *iMAC* and UCS-MAC, and (b) throughput gain of *iMAC* over UCS-MAC, as a function of average hop lengths for different values of m. η = 0.471, A and N are varied for every hop length.

achieved at a particular number channels with increase in average hop length. With increase in hop length UCS-MAC sometimes performs comparable to *iMAC* showing inability of channel selection to add value in such conditions.

Figure 4.6(b) is a plot of the gain in throughput of *iMAC* over UCS-MAC across different combinations of average hop length and number of channels. The achieved peaks in gains are at different hop lengths for different number of channels showing there is a optimal value of hop length for each case of number of channels at which gain is significantly high. At larger hop lengths, intelligent channel selection is not capable of harnessing additional resources provided in the form of increased number of channels. We see that 6 and 8 channel cases peak earlier on in the graph.

4.6 Summary

Our evaluation of *iMAC* shows that the use of *iMAC*, in comparison with UCS-MAC, yields significant throughput gains. *iMAC*'s channel selection based on the contention and channel use of neighbors in the topology helps harness the available bandwidth resources for the flows in the network. The throughput gains yielded by *iMAC* are highest when medium network load needs to be spread across a medium number of channels subject to a medium level of contention. When there is low or high contention for resources, random channel selection suffices.

Chapter 5 – Related work

The problem of designing a Medium Access Control protocol for spectrum agile networks encompasses a more fundamental problem—that of arbitrating channel access by entities communicating across multiple channels. Here, we discuss some of the various prior efforts towards addressing this problem and contrast these prior approaches with our solution.

In [12], the authors design a MAC protocol for the multiple channel scenario with the objective of justifying the use of multiple channels over single channel networks. They do so by showing that the use of multiple channels results in improved network performance in ad-hoc networks. The protocol they propose uses IEEE 802.11 to create spatial reuse of the available spectrum so as to increase overall system capacity. The authors further use their simulation results to propose multichannel throughput scaling laws. In contrast to this design, iMAC takes into account the fact that when nodes are equipped with a single network interface, the channel status information maintained by each node is mostly inaccurate due to periods of absence of the nodes from the control channel. Accordingly iMAC is better equipped to deal with contention on traffic channels and hence provides better performance.

The authors of [2] also design a MAC protocol for multichannel use. They design a robust, distributed asynchronous protocol which has a flat architecture with respect to the roles played by nodes in an ad-hoc network. The channel assignment decision for communication between a sender and receiver is made by the receiver node. The protocol is designed with the objective of delivering good performance when the network is under high load. Since every node switches back and forth between its roles of being a transmitter and that of a potential receiver for some other transmitter based on a randomly chosen timeout period for trying to reach its intended receiver, the protocol achieves self-balance of the saturated network.

The fact that additional resources in the form of multiple channels enable better performance led researchers to investigate practical applications of this knowledge. Specifically, more work was done to enable such multichannel networks to function in a cognitive network environment abiding by all the rules laid down by the FCC for use of whitespaces by SUs in the spectrum bands licensed for use by PUs. This capability of responding to Primary user traffic and dealing with it has been identified as future work for iMAC.

One of the early designs for cognitive wireless mesh networks was proposed in [9]. The authors design a distributed frequency-agile MAC as an extension to IEEE 802.11s amendment for next generation WMNs. This MAC has been designed to be backward compatible with legacy 802.11 networks. The protocol overcomes the need for coming to consensus on a common control channel and instead makes the communicating nodes aware of this information by using the well known ISM-frequency bands for control traffic. The protocol overcomes the problem of gaps in information about the status of data channels by using two half-duplex radio transceivers. One of these transceivers is permanently tuned to the common control channel while the other is tuned to the assigned data channel, which too changes rarely. However, this would necessitate hardware changes and additional hardware requirements for communicating nodes, which iMAC does not require.

Another design which requires use of multiple radio transceivers per SU node is proposed in [13]. This is one of the early designs proposed for a MAC that enables real-time dynamic spectrum allocation and high spectrum utilization in ad-hoc networks. This protocol uses three bands assigned with specific roles: 1) exchange control messages 2) exchange data, and 3) send a busy tone to address hidden and exposed terminal problems.

Yet another approach [7] is to divide the available spectrum into logical frames and set aside slots or beaconing periods for spectrum management and data communication. However, such a scheme involves overhead of periodic **global synchronization** between nodes of various parameters such as time and channel information. A similar approach is taken in [10] where time is divided into Opportunistic Spectrum Periods. Channel information is exchanged by a single Secondary user representing a group of users. This protocol provides significant improvement in network performance in single hop wireless networks.

HC-MAC[11] takes into account hardware constraints of cognitive radio in the design of MAC. It identifies and addresses two hardware constraints: 1) sensing constraint, and 2) transmission constraint. Taking into account the overhead involved in sensing of spectrum by SUs, the design formulates the sensing problem as the optimal stopping problem. However, the design decision of reserving the common control channel for the period of sensing and accessing part of the spectrum for data communication degrades performance. This is due to the preemption of other nodes from accessing the control channel and parallelizing spectrum sensing. A very similar approach is followed in [14]. HC-MAC[11] differs in considering sensing overhead for multichannel opportunity in its design.

Another distinguishing characteristic of [14] is in the absence of a common control channel. Each of the protocols already discussed set aside a common control channel dedicated to conduct spectrum management related communication. While one approach mandates that all control messages be exchanged only on the common control channel with only DATA and ACK packets on the data channels [11, 13, 12], another approach sends most of the spectrum management messages on the common control channel but exchanges a few control messages on the data channel as well [2, 10, 7, 9]. In contrast to all of these proposals, the authors of [14] design a decentralized cognitive MAC for opportunistic spectrum access in ad-hoc networks. The SUs independently search for spectrum opportunities and there is no co-operation amongst them. The authors develop an analytical framework for opportunistic spectrum access based on results from Partially observable Markov decision processes. However, the design utilizes spectrum usage statics of primary user network for modeling the framework and assumes these statistics remain unchanged for a given time period. iMAC relies on no such assumptions accounting for a more realistic use case in its design and simulation.

Lastly, the authors of [4] pursue a design approach similar to that of [14]. An optimal spectrum sensing and access policy is derived under energy constraints on the secondary users.

Chapter 6 – Conclusions

Enabling wireless networks communicate across multiple channels is a crucial first step in solving the problem of shortage in spectrum. We design a novel Medium Access Control protocol which can manage access of multiple frequency bands by nodes spread in a multi-hop topology. Our protocol, *iMAC*, has been evaluated thoroughly to show that it provides significant improvement in performance under medium load conditions when compared with a protocol using random channel selection technique. *iMAC* is a lightweight protocol having a simple design and requiring no global synchronization or additional hardware support. We use the NS2 implementation of the protocol to study the impacts of network load, node density, hop length, number of channels and other network parameters on achievable throughput for *iMAC*. We further go on to suggest optimal network conditions to achieve maximum gain in throughput based on our observations. Our evaluation of the protocol justifies the intelligent channel selection technique used.

Bibliography

[1] IEEE 802.11 standard specification. `http://standards.ieee.org/getieee802/download/802.11-2007.pdfi`.

[2] Anouar and C. H.Bonnet. A self-balanced receiver-oriented MAC protocol for multiple channels multihop ad-hoc networks. In *IEEE Vehicular Technology Conference*, 2005.

[3] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with wi-fi like connectivity. In *SIGCOMM*, 2009.

[4] Y. Chen, Q. Zhao, and A. Swami. Distributed cognitive MAC for energy-constrained opportunistic spectrum access. In *IEEE Military Communication Conference*, 2006.

[5] New year, new policy push for universal broadband. `http://news.cnet.com/8301-1035_3-10438365-94.html?tag=mncol;txt`.

[6] At stake in broadband push: Wireless spectrum. `http://news.cnet.com/8301-1035_3-10439173-94.html`.

[7] C. Cordeiro and K. Challapali. C-MAC: A cognitive MAC protocol for multi-channel wireless networks. In *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, 2007.

[8] FCC adopts rules for unlicensed use of television white spaces. `http://hraunfoss.fcc.gov/edocs_public/attachmatch/DOC-286566A1.pdf`.

[9] K. Ghaboosi, M. Latva-aho, and Y. Xiao. A distributed multi-channel cognitive MAC protocol for IEEE 802.11s wireless mesh networks. In *Proceedings of The Third International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, 2008.

[10] B. Hamdaoui and K. G. Shin. OS-MAC: An efficient MAC protocol for spectrum-agile wireless networks. In *IEEE Transactions on Mobile Computing*, 2008.

[11] J. Jia, Q. Zhang, and X. S. Shen. HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management. In *IEEE Journal on selected areas in communications*, 2008.

[12] J. Li, Z. J. Haas, M. Sheng, and Y. Chen. Performance evaluation of modified IEEE 802.11 MAC for multi-channel multi-hop ad hoc network. In *17th International Conference on Advanced Information Networking and Applications (AINA'03)*, 2003.

[13] L. Ma, X. Han, and C.-C. Shen. Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks. In *IEEE DySPAN*, 2005.

[14] Q. Zhao, L. Tong, A. Swami, and Y. Chen. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework. In *IEEE Journal on selected areas in communications*, 2007.