An Improved IEEE 802.11 MAC Protocol for Wireless Ad-Hoc Networks with Multi-Channel Access Capabilities

Megha Maiya and Bechir Hamdaoui
Oregon State University
meghamaiya@gmail.com;hamdaoui@eecs.oregonstate.edu

ABSTRACT

We propose a MAC protocol for multi-hop wireless networks that are capable of multiple channel access. The proposed protocol relies on a common channel to allow the exchange of control messages between the sender and the receiver so as to agree on a channel to use for data transmission. The sender and the receiver then switch to the agreed-upon data channel, and use a mechanism similar to 802.11 to share the channel during data transmission. Using NS-2, we show that the proposed MAC protocol provides significant throughput gains, especially when contention for channel bandwidth is medium.

KEYWORDS: *MAC protocols, load balancing and resource sharing, multichannel access, multihop networks.*

1. INTRODUCTION

FCC and other governing bodies and organizations are actively engaged in combating the looming crisis in spectrum supply [1, 2, 4, 5]. As a result of these efforts, FCC adopted rules for unlicensed use of the television white spaces, opening them up for Secondary Users (SUs) complying with these rules [6]. These unlicensed devices can utilize the spectrum only when and where they are unused by the licensed or Primary Users (PUs). 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 [6].

Devices capable of complying with the rules set by FCC typically rely on cognitive radios [9]. These radios are capable of empowering devices to hop from one frequency band to another depending on availability of the spectrum for SU use [13]. 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 [3] needs to be upgraded to be able to co-ordinate spectrum access in this new scenario.

During the past few years, many MAC protocol designs have been proposed for wireless networks with multi-channel access capabilities [7, 8, 10–12, 14–16]. Generally, most of the reported protocols set aside a channel for common traffic control and use the other channels for data communications. For example, DCA-MAC [15] assumes that each wireless device is equipped with two half-duplex transceivers, where one of them is always tuned onto the control channel, and the other is tuned onto a data channel. Unlike DCA-MAC, MC-MAC [12], on the other hand, relies on one half-duplex transceiver only, and hence, it requires that all devices be periodically tuned onto the control channel for an interval of time during which source-destination pairs negotiate and select their new data channel. The authors in [10] propose DOSS which, like DCA-MAC, functions on a packet-by-packet basis. Under DOSS, the spectrum is divided into one control channel, and many pairs of (data, busy tone) channels, i.e., for each data channel, there is a busy tone channel mapped to it. While the control channel is used for negotiating which data channel to be used, busy tone channels are used by receivers to prevent nearby transmitters from interfering with them. Like DCA-MAC, DOSS requires that each device have one extra half-duplex transceiver to be able to sense busy tones concurrently with data communications. DOSS also requires extra power, which is needed for transmitting the busy tones.

In [7], an IEEE 802.11-based MAC protocol for frequency-agile networks has been proposed. 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 also overcomes the problem of gaps in information about the status of data channels by using two half-duplex radio transceivers: One transceiver is permanently tuned onto the common control channel while the other is tuned onto the assigned data channel. Like DCA-MAC and DOSS, this protocol necessitates hardware changes and additional hardware requirements for communicating nodes. The authors in [8] propose OS-MAC also for spectrum-agile networks, which enables multichannel users to adaptively and dynamically seek and exploit spectrum opportunities. Although OS-MAC is shown to improve spectrum utilization through IEEE 802.11 like mechanisms, it is suitable for single-hop networks only.

In this paper, we propose an improved IEEE 802.11 MAC protocol, referred to as *iMAC*, for multi-hop wireless networks that empower wireless devices to effectively access multiple frequency bands in multi-hop wireless networks. *iMAC* is simple, suits multi-hop networks, completely decentralized, does not incur node synchronization overhead, and does not require any extra hardware. We evaluate the performances of *iMAC* using NS-2, and compare them with those of UCS-MAC (Uninformed Channel Selection-MAC) protocol. We find that *iMAC* yields up to 30% gain in average throughput due to its intelligent channel selection method.

The rest of the paper is organized as follows. Section describes *iMAC*. In Section , we present evaluation results of our NS-2 based simulations. Finally, we conclude the paper in Section .

2. MULTICHANNEL MAC PROTOCOL

In this section, we describe the design of our proposed MAC. *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 renders 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 1. Structure of Channel Information Table

2.1. Channel Selection

To choose from a set of channels, a node needs to be informed about the availability of all channels. This requires 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).

Channel Information Table (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 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 section 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

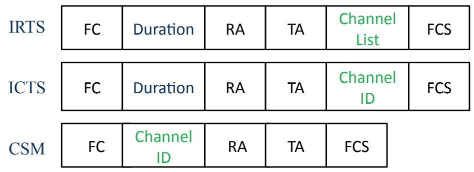

Figure 2. Structure of control frames used in *iMAC* protocol.

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.

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 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 [3]. 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 2 is similarly based on the Clear To Send (CTS) frame of 802.11 [3]. 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 section, we explain how these neighboring nodes make use of an ICTS frame to update their CITs.

Lastly, Figure 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.

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.

2.2. 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.

2.3. Data Channel

Figure 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 perform-

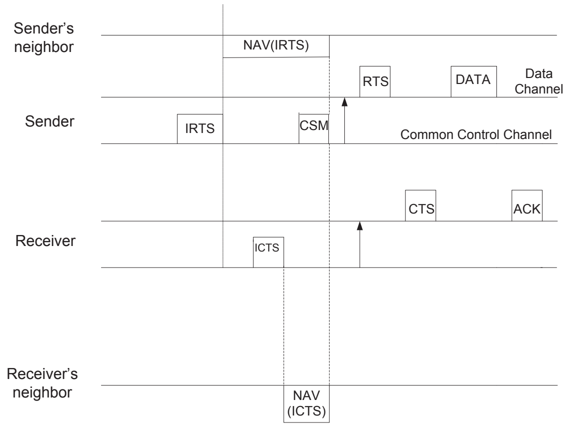

Figure 3. Sequence diagram for *iMAC* protocol.

ing 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 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).

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.

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 1. Variable parameters of each simulation run.

3. EVALUATION

In this section, we evaluate and compare the throughput obtained using *iMAC* with that obtained under UCS-MAC.

3.1. Simulation Setting

We evaluate *iMAC* based on simulations with NS-2 version 2.31. We augment the basic distribution of NS-2 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 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 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 (roughly) 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 (roughly) 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 data 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.

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

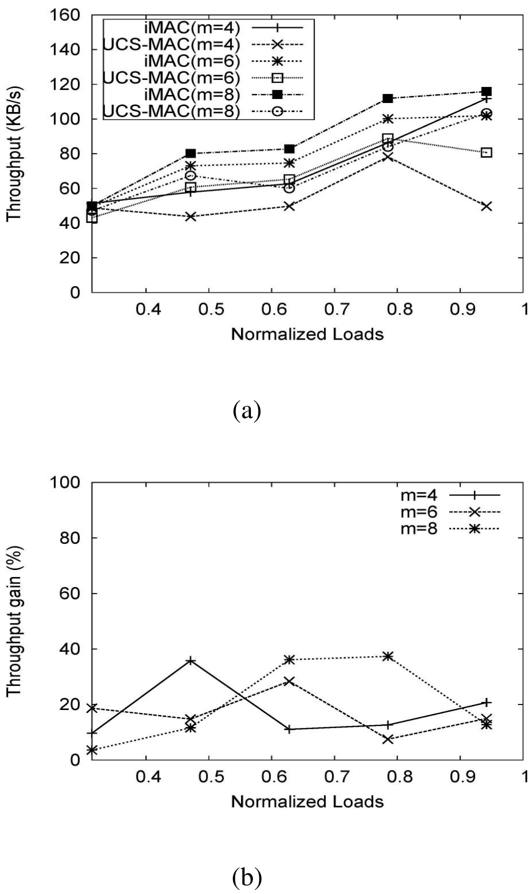

Figure 4. (a) Throughput with *iMAC* and UCS-MAC, and (b) throughput gain of *iMAC* over UCS-MAC, as a function of η for different values of m. N = 50, A = 1500m $\times$ 1500m, average hop length is between 4 and 5.

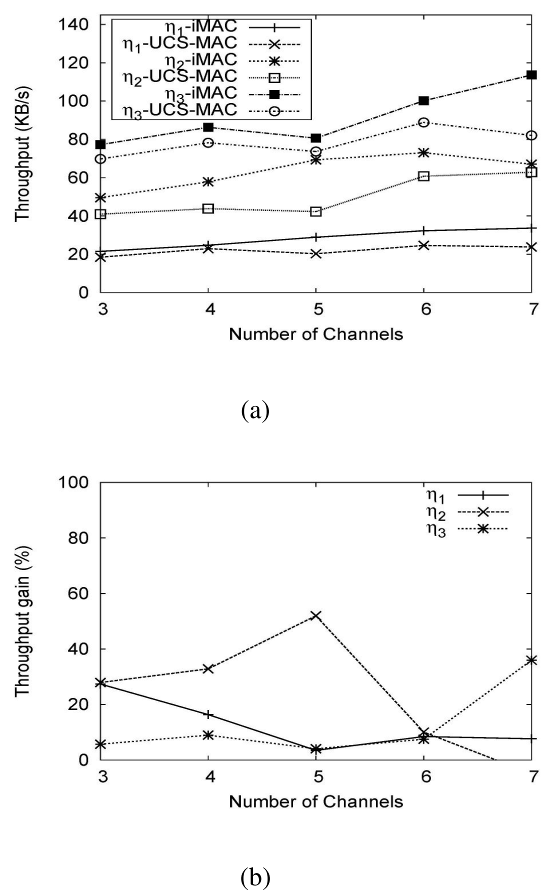

Figure 5. (a) Throughput with *iMAC* and UCS-MAC, and (b) throughput gain of *iMAC* over UCS-MAC, as a function of m for different values of η. N = 50, A = 1500m $\times$ 1500m, average hop length is between 4 and 5.

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 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.

3.2. Impact of Network Load

First, we seek to understand *iMAC*'s performance under different network load regimes. In this case, we fix N at 50 and A at 1500m $\times$ 1500m. We consider five different values of network load in the range $(0, 1)$, and in each case, we consider three different values for m—4, 6, and 8.

Figure 4 shows the variation of throughput and throughput gain across different load values. Figure 4(a) shows that *iMAC* consistently provides better throughput than UCS-MAC. In Figure 4(b), we see that the value of m at which the highest throughput gain is measured varies with the value of network load; the higher the load, the greater the number of channels at which the best gain is obtained. 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.

3.3. Impact of Number of Channels

Next, we study *iMAC*'s throughput benefits as a function of number of channels. We again fix N at 50 and A at 1500m $\times$ 1500m. We then vary m from 3 to 7. In each case, we consider three network load values: $\eta_1 = 0.2$, $\eta_2 = 0.5$, and $\eta_3 = 0.8$.

Figure 5 plots throughput and throughput gain as a function of m. First, in Figure 5(a), we observe that in all cases *iMAC* yields significantly higher average throughput than UCS-MAC. Second, in Figure 5(b), we observe that as the number of channels in the network increases, correspondingly the network load at which the highest value of throughput gain is measured increases. As before, 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. CONCLUSIONS

We design a MAC 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 NS-2 implementation of the protocol to study the impacts of network load and number of channels on achievable throughput under *iMAC*.

REFERENCES

[1] *FCC, Spectrum Policy Task Force (SPTF), Report ET Docet no. 02-135, November*, 2002.

[2] *FCC, Spectrum Policy Task Force (SPTF), Report of the Spectrum Efficiency WG, November*, 2002.

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

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

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

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

[7] 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.

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

[9] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, February 2005.

[10] L. Ma, X. Han, and C. C. Shen. Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks. In *IEEE Inter. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, pages 203–213, 2005.

[11] S. Sankaranarayanan, P. Papadimitratos, A. Mishra, and S. Hershey. A bandwidth sharing approach to improve licensed spectrum utilization. In *IEEE Inter. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, pages 279–288, 2005.

[12] J. So and N. Vaidya. Multi-channel MAC for ad-hoc networks: Handling multi-channel hidden terminals using a single transceiver. In *ACM Proceedings of MOBIHOC*, pages 222–233, 2004.

[13] G. Staple and K. Werbach. The end of spectrum scarcity. *IEEE Spectrum*, March 2004.

[14] M. Timmers, S. Pollin, A. Dejonghe, L. V. der Perre, and F. Catthoor. A distributed multichannel MAC protocol for multihop cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 59(1), 2010.

[15] S. Wu, C. Lin, Y. Tseng, and J. Sheu. A new multi-channel MAC protocol with on demmand channel assignment for multi-hop mobile ad hoc networks. In *Proceedings of Inter. Symp. on Parallel Architectures, Algorithms and Networks*, pages 232–237, 2000.

[16] C. Zou and C. Chigan. On game theoretic DSA-driven MAC for cognitive radio networks. *Computer Communications*, 32(18), 2009.