Exploiting Group Structure in MAC Protocol Design for Multichannel Ad Hoc Cognitive Radio Networks

Sachin Kadam, Devika Prabhu, Nitish Rathi, Prakash Chaki and Gaurav S. Kasbekar

***Abstract*—The design of an efficient Medium Access Control (MAC) protocol for multichannel ad hoc Cognitive Radio Networks is an important problem and has been the topic of extensive recent research. In this paper, we present the design and performance evaluation of a protocol, Group MAC (GMAC), which is customized for a situation that commonly arises in ad hoc networks: the network consists of multiple groups of nodes such that a large fraction of the traffic of each node needs to be sent to other nodes of its own group. Some examples are: (a) units (e.g., platoons) in a military ad hoc network, (b) divisions in an emergency or disaster relief network, (c) departments in a corporate or university network. Our protocol requires each secondary node to have only one narrowband transceiver, does not rely on a control channel and incorporates a novel technique for dynamically balancing the traffic load of secondary nodes across the set of free channels. We formulate the problem of partitioning the network nodes into groups based on the volumes of data traffic to be sent between different pairs of nodes, which we call the Group Formation Problem (GFP). We show that the GFP is NP-complete and propose a greedy algorithm to solve it. We analyze the stability region of the GMAC protocol using a queuing theoretic framework. Our extensive simulations show that a large fraction of the bandwidth unoccupied by primary users is utilized by the GMAC protocol for data transmissions.**

I. INTRODUCTION

Traditionally, government agencies have been allocating radio spectrum by assigning exclusive licenses to users to operate their networks in different geographical regions [4]. However, this has led to an artificial spectrum scarcity, wherein most of the usable radio spectrum is allocated, but underutilized [15]. Cognitive Radio Networks (CRNs) are emerging as a promising solution to this dilemma; in these networks, there are two types of spectrum users– primary users (PUs), which have prioritized access to channels, and secondary users (SUs) that detect and use "spectrum holes", i.e., chunks of spectrum that are currently not in use by PUs [4]. The design of an efficient Medium Access Control (MAC) protocol is crucial in order to ensure high utilization of the free spectrum by SUs, its effective sharing among different SUs and the provision of a high Quality of Service (QoS) (e.g., throughput, delay) to SUs [14]. In particular, the design of a MAC protocol for multichannel ad hoc CRNs is an important problem and has been the topic of extensive recent research (a literature review is provided in Section II).

The design of a MAC protocol for multichannel ad hoc CRNs involves a number of challenges, which we now describe [14]: SUs are allowed to use the spectrum only when PUs are not using it. So they need to periodically sense the spectrum to detect spectrum holes. Also, SUs must vacate a channel quickly when a PU appears on it; for this purpose, there must be "quiet periods" on a channel during secondary transmissions, during which all SUs pause their transmissions and sense that channel. Recall that in traditional wireless networks (e.g., 802.11 networks [21]), all the nodes in a given network operate on a single channel at a time. However, modern wireless transceivers are capable of rapidly switching between different channels, e.g., the switching time is approximately 25μs ([38]) for off-the-shelf Wi-Fi transceivers and even lower e.g., 14μs ([34]) for custom transceivers. This capability allows nodes in a multi-channel network to dynamically switch across different channels, resulting in a substantial gain in performance over single channel networks [49], [5], since two or more transmitter-receiver pairs can simultaneously communicate on different channels. For example, in Fig. 1, node pairs (A,B), (C,D) and (E,F) communicate in parallel over three different channels, thus achieving a better performance than in a single channel operation, wherein the three node pairs would have to take turns on a single channel, incurring a large delay.

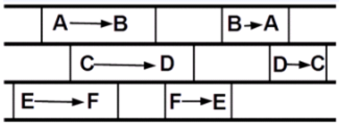

Fig. 1. All six nodes in the figure are in the transmission range of each other. The node pairs (A,B), (C,D) and (E,F) communicate in parallel over three different channels.

Further, in multi-radio networks, a single node may be equipped with two or more wireless transceivers (radios) [40], [41], [24] thus allowing a single node to communicate simultaneously on two or more channels. For example, in Fig. 2, node A has two radios, allowing it to communicate with nodes B and C, and with nodes B and D in parallel.

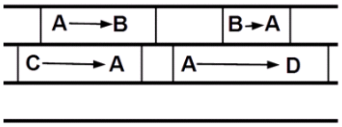

Fig. 2. Node A, which has two radios, first communicates with B and C in parallel, then with B and D in parallel.

Although dynamic channel switching and the use of multiple radios per node can greatly enhance the network performance

---

S. Kadam and G. Kasbekar are with Department of Electrical Engineering, Indian Institute of Technology (IIT) Bombay. D. Prabhu, N. Rathi and P. Chaki are with PepsiCo India, Delhi, Google, Hyderabad and Cloud System Research Laboratories, NEC Corporation, Kanagawa, Japan respectively. Their email addresses are {sachink, gskasbekar}@ee.iitb.ac.in, devika.vijayprabhu@pepsico.com, nitishrathi702@gmail.com, p-chaki@ah.jp.nec.com respectively. Part of this work was done while D. Prabhu and N. Rathi were with IIT Indore and P. Chaki was with IIT Bombay. The contributions of S. Kadam and G. Kasbekar have been supported by SERB grant SB/S3/EECE/157/2016.

A preliminary version of this paper appeared in IEEE WCNC 2016 [27].

in multi-channel multi-radio and Cognitive Radio networks as compared to operation on a single channel at a time, efficiently achieving coordination among different nodes is a key challenge in these networks. In particular, note that for two nodes, say A and B, to be able to exchange data, both must have a radio on a common channel at a time; thus, the MAC protocol used must ensure that shortly after a packet is generated at A with destination B, A and B should have a radio on a common channel. Achieving this efficiently for all the node pairs in an ad hoc network is especially challenging since only a subset of the available radio spectrum can be sensed by a node in real-time due to hardware cost and size considerations [26], [49]. For example, suppose three nodes, say A, B and C, each equipped with one radio, are initially on different channels; also, an application at A generates a packet destined for B, and an application at B generates a packet for C at around the same time. If A switches to B's channel and B switches to C's channel, the packet transfer from A to B fails. The MAC protocol used must take such possibilities into account. The MAC protocol used must also overcome the multichannel hidden terminal problem [49]. Also, the MAC protocol used must effectively balance the traffic load of the SUs, which is often non-uniform across SUs, over the free channels in real-time [12].

Comparisons of various MAC protocols for multichannel ad hoc CRNs are provided in [1], [14], [18], [39], [43], which show that the performance of any given MAC protocol strongly depends on the number of channels, number of nodes, characteristics of the data traffic in the network etc and hence no one MAC protocol necessarily performs well in all situations. Hence, it is important to design MAC protocols that are customized for a given network situation. In this paper, we present the design and performance evaluation of a protocol, Group MAC (GMAC), which is customized for a situation that commonly arises in ad hoc networks: the network consists of multiple groups of nodes such that a large fraction of the traffic of each node in the network needs to be sent to other nodes of its own group. Several examples of such groups in ad hoc networks may be readily envisioned: (a) units (e.g., platoons) in a military ad hoc network (note that typically a large fraction of the traffic of a given node would need to be sent to other nodes of its own unit), (b) divisions in an emergency or disaster relief (e.g., floods, earthquakes) network, (c) departments in a corporate or university network, (d) the sets of nodes belonging to different classes (e.g., nodes exchanging sensor measurements, nodes exchanging smart meter readings, nodes that are part of a vehicular network) in a heterogeneous Machine-to-Machine (M2M) network. Also, note that the above examples are some of the prime applications where CRNs have huge potential [37]. Our GMAC protocol is specifically targeted for such networks: the basic idea is that groups of nodes, which frequently communicate among themselves, are kept on the same channel as far as possible so as to keep the channel switching overhead and vulnerability to the multichannel hidden terminal problem [49] minimal. However, achieving this involves a number of challenges (described in Section IV), which are addressed as explained in Section V, where the GMAC protocol is presented in detail. In addition, the protocol incorporates a novel technique for load balancing, which ensures that the traffic load is uniformly distributed across the free channels even when the traffic requirements of different SUs are highly heterogeneous. In Section VI, we formulate the problem of partitioning the set of nodes into groups based on the volumes of data traffic to be sent between different pairs of nodes. We show that this is an NP-complete problem and provide a greedy algorithm to solve it. We analyze the stability region of the GMAC protocol using a queuing theoretic framework in Section VII, study its performance via extensive simulations in Section VIII and finally conclude the paper in Section IX.

II. RELATED WORK

The IEEE 802.22 [22] and IEEE 802.11af [17] standards specify MAC protocols for CRNs; however, these network architectures are centralized, in which a base station or access point controls its associated client devices, whereas in this paper we seek to design a MAC protocol for ad hoc CRNs.

In prior work, a large number of MAC protocols have been designed for ad hoc CRNs as well as multi-channel multi-radio (MC-MR) networks, which are closely related to CRNs. The MAC protocol design problem in MC-MR networks is similar to that in CRNs, with an important difference being that in CRNs, secondary nodes need to avoid interfering with PU transmissions [31].

We now briefly review some representative protocols for multi-channel wireless networks and CRNs; detailed surveys can be found in [1], [14], [18], [39], [43], [48], [51]. In [60], [32], [3], [58], [19], [5], [40], [48], [49], [54], [57], (respectively, in [12], [35], [53], [61], [2], [8], [9], [33], [46], [52]) MAC protocols for multi-channel wireless networks (respectively, for CRNs) are proposed. In the model described in [40], each node is assumed to be equipped with N radios, where N is the number of available channels. In [35], it is assumed that the secondary network owns a dedicated control channel and each node has two transceivers, one of which is a wideband transceiver. The protocols proposed in [53], [57] use a dedicated control channel and two transceivers per node. The above protocols [35], [40], [53], [57] achieve accurate sensing and/ or high throughput at the expense of high radio hardware cost. In [5], [12], [48], [49], [54], MAC protocols that use a single transceiver per node are proposed. The protocols in [49], [53], [54], [57] are single rendezvous based: control (rendezvous) packets for scheduling data transmissions are sent on a single channel at a time, whereas the protocols in [5], [12], [48] enable multiple rendezvous exchanges to take place in parallel on different channels. A decentralised MAC protocol is designed in [2], in which performance improvement in terms of communication time and throughput is achieved due to reduction in the number of handshaking signals exchanged over a Common Control Channel (CCC) between SUs. An efficient and reliable control channel (CC) is designed in Wi-Fi based CRNs using the point coordination function (PCF) of Wi-Fi networks in [8]; later the same CC is used to design a hybrid MAC protocol. A decentralized MAC protocol called coexistence cognitive radio MAC (CCR-MAC) protocol [9] deals with unfairness problems in heterogeneous CRNs. Every SU in CCR-MAC use two transceivers, one tuned to the CCC and the other to exchange information over data channels. To enhance the throughput of a multichannel CRN, an efficient MAC protocol, called link maintenance MAC

(LM-MAC) protocol is proposed in [33]. In the LM-MAC protocol a method to reestablish the previously existing links between different SUs in every frame is proposed. The MAC protocols proposed in [2], [8], [9], [33], [58] use a fixed CCC in their design. In dynamic CCC (DCCC) MAC protocol [52], a dynamic CC is used. The CC is selected from the set of free channels by using the support-vector-machine (SVM)-based learning technique in every frame. In the Channel-Aggregation Diversity (CAD-MAC) based MAC protocol [46], each node can utilize multiple channels simultaneously using the CAD technique and transmit multiple data packets during each transmission.

Improvement in the spectral efficiency in the Full Duplex Multi-channel MAC (FD-MMAC) protocol [60] is achieved by independent performance of destination discovery and channel assignment by both the source nodes and destination nodes, without using a CCC. A Multi-Channel MAC (MC-MAC) protocol for Wireless Body Area Networks (WBANs) is designed in [32], in which an efficient channel assignment strategy is developed. A novel distributed multi-channel MAC protocol that uses fast and slow hopping sequences with two transceivers per device is developed in [3]. In the reliable channel reservation based multi-channel MAC (RCR-MAC) protocol [58], the problem of congestion on the control channel is addressed. A Markov chain model, which combines the channel-hopping strategy of the cyclic quorum based multichannel (CQM) MAC protocol and the IEEE 802.11 distributed coordination function (DCF), is proposed in [19].

However, unlike this paper, none of the papers in prior work design a MAC protocol specifically customized for the case where the ad hoc network consists of multiple groups, with the majority of the traffic of each node destined for other nodes of its own group. As explained in Section IV, this results in overheads that are avoided under our GMAC protocol through its customized design for such networks. Also, the GMAC protocol requires only one narrowband transceiver per node, resulting in lower hardware cost than several protocols in prior work, e.g., those in [35], [40], [53], [57], [3], [9]. In addition, unlike several protocols proposed in prior work ([2], [8], [9], [33], [58]), the GMAC protocol does not use a CCC; hence, it does not suffer from problems such as saturation of the CCC and deterioration in performance when PUs appear on the CCC or the CCC is attacked by a jammer. Finally, GMAC is a multiple rendezvous protocol and hence achieves improvement in performance over single rendezvous protocols via parallel rendezvous exchanges on multiple channels.

III. NETWORK MODEL

We consider a region where the available spectrum is divided into M identical and mutually non-overlapping channels. These channels are intermittently used by PUs (e.g., TV broadcast stations), who have licensed them from the spectrum regulator. An ad hoc network of SUs can use these channels whenever they are not in use by the PUs. Each node is equipped with a single half-duplex transceiver, which is capable of either transmitting or receiving on a single channel at a time. We first consider a single hop scenario, i.e., all nodes are within the transmission range of each other. In Section V-G, we study the multi-hop scenario in which not all nodes may be in the transmission range of a given node.

A node exchanges data packets with all the other nodes in the network. However, the network nodes are divided into groups, such that a significant fraction of the outgoing traffic of any given node is destined for nodes of its own group. Every group is managed by a group leader (GL)– the functions of the GL are explained in Section V. On each channel, when the number of nodes is low or moderate, nodes communicate using the 802.11 Distributed Coordination Function (DCF) [21], which is a variant of CSMA/CA [30]. Under this protocol, nodes that have a packet to send contend for channel access using random backoffs– see [21] for details of the protocol. When the number of nodes is high within a group, a large number of collisions can occur under 802.11 DCF [21]; hence, TDMA can be used for data transmission within that group. The GL can assign time slots to different nodes of the group. The hybrid MAC protocols proposed in [10], [13], [20], [59] can be used for data transmission within each group in case of varying traffic loads.

The group to which each node belongs may be configured manually by the user or system administrator (e.g., using available information such as the department, platoon etc. to which the user belongs) or alternatively, nodes may dynamically divide themselves into groups based on the history of the volumes of traffic transmitted between pairs of nodes and a learning algorithm. For simplicity of exposition, we first describe the GMAC protocol assuming that the nodes in the network have already been partitioned into groups. In Section VI, we formulate the problem of partitioning the nodes into groups based on the volumes of data traffic to be sent between different pairs of nodes, show that it is NP-complete, and provide an algorithm to solve it.

IV. MOTIVATION AND KEY IDEA BEHIND PROPOSED PROTOCOL

We first explain how a typical multichannel MAC protocol operates. For concreteness, we describe a split-phase protocol (see [39], [49], [53]). Also, we ignore the PUs and consider a multichannel ad hoc network for ease of exposition. Out of the M available channels, one channel is designated as the control channel. As shown in the example in Fig. 3, time is divided into alternating periods called the rendezvous phase (RP) and data phase (DP). In the RP, all the nodes of the network tune to the control channel and, by contending as in 802.11 DCF, exchange a series of handshakes to schedule data packet transmissions in the following DP. Each handshake consists of (i) a request packet (RTS) sent by a node, say A, which wants to send a data packet in the following DP to another node B, and (ii) a response packet (CTS) from node B to node A. In a handshake, the participating nodes decide which channel out of the M channels they will tune to, for exchanging data packets, in the following DP. Note that in the DP, there may be more than 2 nodes per channel; nodes contend as in 802.11 DCF on each channel to send the scheduled data packets (see Fig. 3).

A key observation is that in the RP, all the nodes of the network tune to a single channel to schedule data transmissions. In a general network, which is not divided into groups, and in which a given node may potentially have

	Rendezvous	Data	Rendezvous
Ch.3		6,7	
Ch.2		3,4,5	
Ch.1	1,2,3,4,5,6,7	1,2	1,2,3,4,5,6,7

Fig. 3. In this example, channel 1 is the control channel. There are 7 nodes in the network numbered from 1 to 7. In the RP, all the 7 nodes tune to the control channel and exchange handshakes in which node 1 schedules a transmission to node 2 on Ch. 1, node 3 to node 4 on Ch. 2, node 6 to node 7 on Ch. 3 and node 4 to node 5 on Ch. 2. In the following DP, nodes 1 and 2 switch to Ch. 1, nodes 3; 4 and 5 switch to Ch. 2 and nodes 6 and 7 switch to Ch. 3.

data packets to transmit to an arbitrary other node with a high probability at any given time, an approach similar to that employed in split-phase protocols, or multiple rendezvous protocols [5], [48], in which nodes periodically meet pairwise on common channels to schedule data transmissions, seems to be necessary. However, in a network in which group structure information is available, which is the focus of this paper, this information can be exploited to significantly improve the network performance as we now explain. For concreteness, consider a network with 100 nodes, which is divided into 10 groups $G1$, $G2$, $\ldots$, $G10$ such that 90% of the packets of any given node are intended for other nodes of its own group and the rest are intended for nodes of the other groups. Suppose there are $M = 5$ channels. Under a split-phase protocol, all 100 nodes would tune to the control channel in every RP to exchange handshakes. This would result in a large number of collisions during the RP and hence the length of a RP would have to be long to ensure that the necessary handshakes are completed. More importantly, the outcome of a typical RP would be that most nodes of any given group would switch to the same channel in the following DP.

Now, consider an alternative approach in which, each group remains by default on a fixed channel, called its home channel (HC), where each node continuously keeps exchanging data packets with other nodes of its own group, except when a node needs to send one or more data packets to a node of another group with a different HC, in which case it switches to the latter to send the packets and then returns to its own HC. For example, each of the 5 channels in the above example may be the HC for two groups. The advantages of this approach are as follows. First, since most of the packets of any given node (90% in the above example) are destined to other nodes of its own group, which are on its own HC, they can be directly sent, which results in fast packet delivery for most packets. Second, at any time, there are fewer nodes on each channel under this approach than in the RP under the split-phase protocol (about 20 under this approach and 100 in the RP under the split-phase approach in the above example), because of which there are fewer collisions [1]. Third, the overhead incurred in switching to other channels is minimized since each node rarely (for at most 10% of the packets in the above example) has packets to send to nodes not on its own HC. Thus, in this approach, the overhead of nodes repeatedly assembling on a common channel for rendezvous and then dispersing to different channels for load balancing is eliminated. Fourth, the need for a dedicated control channel is eliminated; thus, it is robust to problems such as appearances of PUs on the control channel, control channel congestion, jamming of control channel by miscreants etc [12].

However, several other challenges need to be addressed to make our approach work:

1) First, the traffic volumes of different groups may be non-uniform resulting in some HCs being congested and the others being underutilized. So a scheme for balancing the traffic load across the M channels is needed. Moreover, the traffic volume of a group may vary with time necessitating dynamic assignment of HCs to groups.
2) Second, consider a scenario where the HCs of nodes A, B and C are different. If node A wants to send some packets to node B, it will visit node B's HC. But node B may be away to node C's HC to send some packets to C and A does not know how long it will be before B returns to its HC.
3) The channels on which primaries (incumbents) appear must be immediately vacated by all the groups present on the channels.
4) Since our protocol does not use a common control channel, it is challenging to ensure that different nodes do not simultaneously make changes to the same protocol parameters.

Our GMAC protocol is designed to overcome all of the above challenges. We describe the protocol in detail and explain how it overcomes the above challenges in Section V.

V. THE GMAC PROTOCOL

A. *Overview*

Let $\mathcal{M}$ be the set of all channels. At any time, $\mathcal{M}$ is divided into sets $\mathcal{P}$, $\mathcal{H}$ and $\mathcal{B}$, where $\mathcal{P}$ is the set of channels currently in use by the PUs, $\mathcal{H}$ is the set of channels that are Home Channel (HC) for at least one group, and $\mathcal{B}$ is the set of remaining channels in $\mathcal{M}$ called buffer channels (BCs). Let P, H and B denote the cardinalities of the sets $\mathcal{P}, \mathcal{H}$ and $\mathcal{B}$ respectively. The BCs are used for balancing the traffic load across different channels as explained in Section V-C. Throughout the network operation, each group is aware of the sets $\mathcal{P}$, $\mathcal{H}$ and $\mathcal{B}$, members of the other groups and their current HCs. This information is updated from time to time as explained later. Each group is dynamically assigned a HC and all the nodes in the group remain tuned to this channel by default. Two or more groups may be assigned to the same HC. Every group is managed by a group leader (GL), which sends periodic beacons on the group's HC containing information pertaining to PU occupancy, synchronization and other protocol parameters. When a PU appears on the HC of a group, all the group members must immediately vacate the channel. For this purpose, each group maintains an ordered list of the channels in $\mathcal{H} \cup \mathcal{B}$ other than its own HC, which act as backup channels. This ordered list is stored by each group member; also, whenever a change in this list occurs, the GL informs all the group members of this change by including

---

[1] In the proposed alternative approach, when the number of nodes on a HC is high, TDMA can be used instead of 802.11 DCF to further reduce the number of collisions. In case of varying traffic loads, the hybrid MAC protocols proposed in [10], [13], [20], [59] can be used.

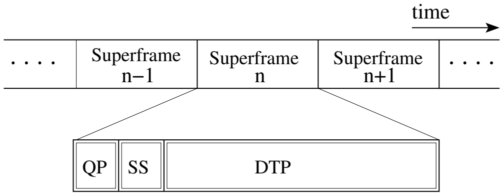

Fig. 4. GMAC Superframe

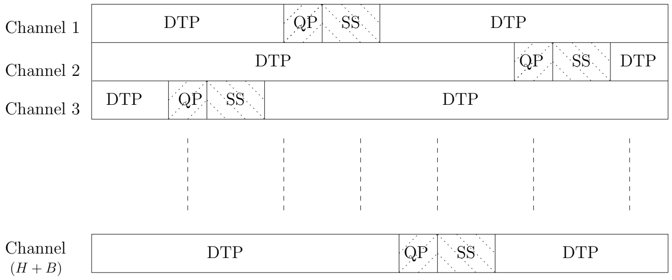

Fig. 5. The figure shows the division of time into superframes for the channels in $\mathcal{H} \cup \mathcal{B}$.

the new list in a few beacons. When an incumbent appears on the home channel of a group, the entire group shifts to backup channels in this order until they find a channel not occupied by incumbents, and use this channel as the new HC. Also, a GL divides the responsibility of sharing updates with nodes of other groups among its group members. Thus, the occurrence of any event in a particular HC (e.g., addition of a new node to a group on the HC) is shared with groups on all other HCs. When a new group forms, it selects one of the channels in $\mathcal{H} \cup \mathcal{B}$ as its HC and broadcasts its channel number on all channels in $\mathcal{H}$. These broadcasts are used by the network nodes to update their sets $\mathcal{H}$ and $\mathcal{B}$. Similarly, if an incumbent leaves a channel in $\mathcal{P}$, the group which was scanning that channel detects this and informs the groups on the other HCs via update broadcasts. Update packets pertaining to changes in PU occupancy, in a group's HC etc are sent as explained in Section V-D.

The task of sensing the channels in $\mathcal{P} \cup \mathcal{B}$ is divided among nodes on different HCs. Also, each node only senses its own HC and a subset of the channels in $\mathcal{P} \cup \mathcal{B}$. Specifically, let $\mathcal{F} = \mathcal{P} \cup \mathcal{B}$. Then $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_H$, where $\mathcal{F}_h$ is the set of channels sensed by the nodes in HC h in addition to channel h ($i.e.$, their own HC). Also the sets $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_H$ are disjoint. The set of all nodes that sense a given channel always belong to the same HC, so it is easy to achieve tight synchronization (which is required for implementing quiet periods (QPs) [2]) among them. Among all nodes assigned to sense a particular channel in $\mathcal{F}$, one node is selected as channel leader (CL) for that channel, and it is responsible for transmitting beacons periodically on the channel if it is a BC [3]. Also, in case of any change in PU occupancy (arrival of PUs on a BC or exit of PUs from a channel in $\mathcal{P}$) on the sensed channel, it informs the GL of this change. There may be errors in sensing by some nodes, e.g., even though a PU is not present on a channel, some nodes may report that it is present (false alarm [4]) or even though a PU is present on a channel, some nodes may report that it is absent (misdetection [4]). Such problems can be mitigated by having the CL of a channel j decide its status (occupied or free) based on the majority of the reports from the nodes that sense channel j.

On each channel in $\mathcal{H} \cup \mathcal{B}$, time is divided into superframes of equal duration. Each superframe consists of a QP, a synchronization slot (SS) and a data transmission period (DTP) as shown in Fig 4. On each HC, in the SS, the GL broadcasts a beacon packet, which contains the current values of various protocol parameters and also ensures that all the nodes on the channel remain tightly synchronized (and thus can implement QPs). The superframes of the channels in $\mathcal{H} \cup \mathcal{B}$ are aligned such that their QPs are non-overlapping (see Fig. 5), which allows a node to sense its own HC as well as some BCs. Recall that each node knows every other node's group and HC. A node X with a packet to send to node Y checks if Y has the same HC as itself; if so, it directly sends the packet to Y in a DTP. If not, X switches to Y's HC in a period that falls within the DTPs of both X and Y and sends the packet to Y; in the latter case, node X must follow the rules specified in Section V-B.

B. Receiver Not Present on its Home Channel

We use the following scheme to address the second challenge described in the last paragraph of Section IV. Let $T_{h,min}$ and $T_{f,max}$ be two parameters such that $T_{f,max}$ is much larger than a packet transmission time. We require that when a node visits a foreign channel to send some data or distribute updates, it must return to its HC within time $T_{f,max}$. Also, after a node returns to its HC, it must remain there for at least a time $T_{h,min}$ before it can switch again to some foreign channel. Now returning to the scenario described in the above challenge, when A visits B's HC to send some data to it, A repeatedly sends a request packet (RTS) to B until it receives a response (CTS) from B. With a high probability, A will receive a CTS from B within a time $T_{f,max}$. One worst case scenario is that B switched to C's HC at almost the same time as A switched to B's HC, and B returns to its HC after spending a time $T_{f,max}$ on C's HC. A will also have to return to its HC at approximately the same time and it will miss B. When this scenario occurs, A subsequently reattempts to deliver the packets to B by switching again to B's HC; thus, the packets are delivered to B after an increased delay. But as stated above, the parameter $T_{f,max}$ is much larger than a packet transmission time. Thus, the probability of such an occurrence is small [4]. Fig. 6 illustrates an example of receiver not present in its HC.

C. Load Balancing

The traffic requirements of different groups may differ significantly at a given time. We use the following mechanism

---

[2] Recall that a QP is a period in which all secondary nodes pause their transmissions and sense the channel to check for PU transmissions [4].

[3] These beacons are required for synchronization of transmitter-receiver pairs that switch to BCs for load balancing.

[4] Consider an example where nodes D and E are in different HCs. D has a packet destined for E, and simultaneously E has a packet for D. D switches to E's HC and E to D's HC at the same time and they miss each other. This keeps repeating until both of them give up and drop the packet. To prevent such situations from occurring, a node that has a packet for a node in a different HC waits for a random amount of time before switching to the destination's HC in the GMAC protocol.

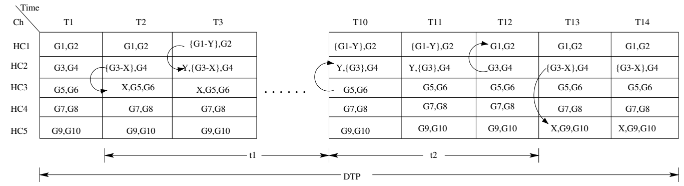

Fig. 6. Consider 5 HCs (BCs and PU occupied channels have not been shown in this figure), each hosting 2 groups. The figure shows a section of a DTP, time being sliced into slots T_1, T_2, T_3, $\ldots$ for ease of exposition. At the beginning of slot T_2, node *X* from group $G3$ leaves its HC and switches to $HC3$. However, unaware of the fact that node *X* is not present in $HC2$, node *Y*, which has a packet to send to node *X* switches to $HC2$ and starts sending RTS packets to *X*. After a duration t_1, node *X* switches back to its HC and *Y* gets a response to its RTS packets in $HC2$. Node *Y* then sends its data packets to *X*. Here, $t_1 \leq T_{f,max}$. After completion of its transmission to *X*, node *Y* returns to $HC1$. Now, node *X*, which has its next data packet destined for a node on $HC5$, cannot leave its HC for a minimum duration of $T_{h,min}$. Thus, it waits for time $t_2 \geq T_{h,min}$ and then switches to $HC5$.

for balancing the traffic load across the set of free channels (the channels in $\mathcal{H} \cup \mathcal{B}$). Each node u continuously keeps track of the congestion level on its HC by sensing the HC. In particular, let $CL_u(t) \in [0, 1]$ be the congestion level measured by node u on its HC at time t. It is defined [5] as the fraction of time that u's HC was busy with packet transmissions in a window of fixed length, say T_{CL}, preceding time instant t. If a node u with HC h needs to send some data packets to another node v on HC h and finds $CL_u(t)$ to be above a predetermined threshold level, $CLTH_u$, then the pair of nodes u and v switch to one of the channels $b \in \mathcal{B}$, exchange the data packets on channel b, and then switch back to channel h. This must be done subject to the limits $T_{f,max}$ and $T_{h,min}$, which are explained in Section V-B. The channel b to switch to is agreed upon by u and v prior to the switch via a handshake on channel h. Note that nodes from multiple HCs may simultaneously be present on a given channel $b \in \mathcal{B}$. On the channel b, nodes contend as in 802.11 DCF to gain access to the channel. In this way, some of the load from congested HCs is offloaded to the BCs [6].

D. Initialization and Distribution of Updates

When a node starts up, it scans all channels for beacons, from which it obtains information about active groups and their HCs. If the group to which it is assigned is active, it sends a request packet for joining the group on that group's HC, which is followed by a response packet by the GL. Subsequently, the GL updates the group parameters (e.g., the list of group members that are responsible for sensing various channels in $\mathcal{F} = \mathcal{P} \cup \mathcal{B}$, distributing updates on various channels in $\mathcal{H}$) to include the new node, and distributes these updates to all the other groups using the process in the next paragraph. If a node is the first node of its group to start up, it becomes the GL, selects a channel as HC and starts sending beacons [7].

A group also needs to distribute updates to groups on other HCs: when a PU appears on or leaves a channel that was being sensed by it, and when the group's HC or backup channel changes. These updates are distributed as follows. The GL maintains a list of the HCs on which different members of that group are responsible for sending the above updates; also, the GL keeps all the group members informed of this list by including it in a few beacons whenever a change in the list occurs. Each update that needs to be distributed is included by the GL in a few beacons, and the group members, upon their receipt, broadcast them on the other HCs. On each of these HCs, this update packet is acknowledged with a response packet by the respective GL [8], which then includes the update in a few subsequent beacons.

We now analyze the number of control messages (e.g., beacons and update packets) exchanged among different nodes under the GMAC protocol, which affects the efficiency of the protocol. First, note that in each superframe (SF) exactly one beacon packet is sent on every channel in $\mathcal{H}$ and $\mathcal{B}$ by the corresponding GL or channel leader (CL). However, the transmission duration of a beacon packet is a very small fraction of the length of a SF; hence the overhead due to beacon packets is negligible. Apart from beacon packets, some update packets are exchanged among the nodes executing the GMAC protocol whenever one of the following events occurs: (i) a PU appears on a channel in $\mathcal{H}$, (ii) a PU appears on a channel in $\mathcal{B}$ or a PU leaves a channel in $\mathcal{P}$, (iii) a new node joins one of the groups, (iv) a node departs from the network. The events in (i), (ii), (iii) and (iv) may result in changes in the HCs and backup channel lists of some groups, the sets $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_H$ etc, and may result in the exchange of some update packets. The number of such update packets exchanged would depend on how frequently the events (i), (ii), (iii) and (iv) occur. Typically in practice, nodes would join or depart the network at a much slower time-scale (typically several minutes [42]) than the duration of a SF (e.g., 160ms [23]); also, changes in the PU activity (appearance and exit of PUs from channels) would occur at a very slow time-scale (typically of the order of minutes to hours [11]) in several types of networks, e.g., when the PUs are television broadcasters. In such cases, the overhead due to update packets would be small and hence the efficiency of the GMAC protocol would be high.

We now present a simplified numerical analysis of the efficiency of the GMAC protocol, i.e., the fraction of the bandwidth of the free channels that is available for the transmission of data packets under the protocol. Suppose the sum of the transmission durations of the update packets that are

---

[5] The time taken by the SS, QP slots and time periods during which node u is not present on its HC are not included in the above window used in the calculation of the quantity $CL_u(t)$.

[6] The channels in the set $\mathcal{B}$ are used only by nodes on HCs with a high amount of congestion, specifically, those HCs for which $CL_u(t)$ is above the threshold $CLTH_u$ for some of the nodes u on the channel. Even on these HCs, only the active nodes that currently have data packets to send to other nodes on the same HC, may use the channels in $\mathcal{B}$. Typically in practice, at any given time, only a subset of the HCs would be congested and on these HCs, only a subset of the set of nodes would be active. Hence, only a limited amount of traffic would be off-loaded to the channels in $\mathcal{B}$.

Further, if the values of the thresholds $CLTH_u$ were the same for all the nodes, a large number of nodes on the same HC h may find their $CL_u(t)$ variables exceeding the threshold value at about the same time and switch to BCs, thus drastically reducing the load on channel h and increasing the congestion on BCs. To prevent this from happening, we let $CLTH_u = TH_{base} + rand_u$, where TH_{base} is a base value common to all nodes and $rand_u$ is a small random offset, which is locally generated at node u.

[7] If two nodes of a group become the GL (e.g., if they start up around the same time), then when they discover each other for the first time, the node with the smaller node ID (identifier) continues to be the GL and the other node ceases to be the GL.

[8] If multiple groups are present on the HC, the GL responsible for sending beacons on the HC sends the response packet (see Section V-E).

exchanged when event (i) occurs is upper bounded by W_{A_h}, and let n_{A_h} be the number of times a PU appears on a HC in a given SF. Similarly, suppose the sum of the transmission durations of the update packets that are exchanged when event (ii) occurs is upper bounded by W_P, and let n_P be the number of times a PU appears on a BC or departs from a channel in $\mathcal{P}$ in the SF. Let the sum of the transmission durations of the control packets (e.g., request by the new node to join a group, response from a GL, update packets) that are exchanged when event (iii) occurs be upper bounded by W_J, and let n_J be the number of times a new node joins the network in the SF. Let the sum of the transmission durations of the control packets that are exchanged when event (iv) occurs be upper bounded by W_D, and let n_D be the number of times a node departs from the network in the SF. Let S_F, Q and S_S be the durations of a SF, quiet period and synchronization slot respectively, and let F be the number of free channels in the SF, i.e., the number of channels in the set $\mathcal{H} \cup \mathcal{B}$ in the SF. Then the efficiency of the GMAC protocol is lower bounded by $\left(1 - E\left[\frac{(n_J W_J + n_D W_D + n_P W_P + n_{A_h} W_{A_h}) + F(Q+S_S)}{F S_F}\right]\right)$.

E. Two or More Groups with the same HC

Under the GMAC protocol, multiple groups may be assigned the same HC. However, this happens only when very few channels are free and the rest are occupied by primaries, i.e., are in the set $\mathcal{P}$. Note that when very few channels are free, in order to accommodate the set of all nodes on the set of free channels, a large number of secondary nodes must be assigned per free channel on average under any Cognitive MAC protocol. In this case, collisions are avoided using a contention resolution protocol, e.g., the 802.11 Distributed Coordination Function (DCF), TDMA or a hybrid protocol, on each free channel.

Under the GMAC protocol, when two or more groups have the same HC, the GL of only one of these groups sends beacons on the HC. The HCs of different groups are merged and split as follows. When a group, say $G1$, switches to the HC, say h_2, of another group, $G1$'s GL transmits its group parameters to the GL responsible for sending beacons on h_2, which includes them in subsequent beacons. When some channels in $\mathcal{P}$ become free, some of the groups on a given HC h_1 with two or more groups may change their HC. Suppose $G1$'s GL was sending beacons on channel h_1. Before some other group, say $G2$, leaves the channel, $G2$'s GL sends a packet to intimate $G1$'s GL of its departure, after which $G1$'s GL stops including $G2$'s parameters in its beacons. If group $G1$ itself wants to leave h_1, then before leaving, its GL sends a similar intimation to the GL of one of the other groups on channel h_1, say $G2$, after which $G2$'s GL starts sending beacons on channel h_1.

F. Sensing Channel and HC Allotment Algorithm

Let B_g and P_g denote the subsets of $\mathcal{B}$ and $\mathcal{P}$ respectively that are sensed by group g. Group g includes the lists B_g and P_g in each beacon and update packet that it sends. Also, each group is dynamically assigned a HC and all the nodes in the group remain tuned to this channel by default. HC allotment in GMAC is performed as follows. When the first group in the network starts operation, it senses all the channels and initializes the sets $\mathcal{P}$, $\mathcal{H}$ and $\mathcal{B}$. This group selects a channel, say h, to be its HC from the set of free channels. At this point in time, note that the set $\mathcal{H}$ contains a single channel h, which is the HC of the first group. During the subsequent network operation, the sets B_g and P_g and the HCs for different groups are updated in response to different events as follows: a) When a new group, say g_{new}, joins the network, it scans each channel and finds out the sets B_g and P_g for all the existing groups and the HC of every existing group using beacons. It then computes its HC and possibly also modifies the HCs of some other groups so as to balance the data transmission loads of different groups as well as possible across the set of free channels. Also, it computes the sets $B_{g_{new}}$ and $P_{g_{new}}$ to balance the sensing load as well as possible across different groups. In particular, it moves some of the channels out of the channels in $\mathcal{B}$ and $\mathcal{P}$ that are currently being sensed by other groups to $B_{g_{new}}$ and $P_{g_{new}}$ respectively. The group g_{new} then distributes the update to all the channels using the process described in Section V-D and the recipient groups record the changes and change their own HCs and sets B_g and P_g if necessary. (b) When an incumbent leaves a channel in $\mathcal{P}$, the group that was sensing the channel computes updates to the HCs of all the groups and the sets B_g and P_g, so as to balance the data transmission loads of different groups as well as possible across the set of free channels and to balance the sensing load as well as possible across different groups, and distributes the updates to all the groups as in (a). (c) When an incumbent appears on a channel in $\mathcal{B}$, the group that was sensing the channel computes and distributes the updates. (d) When an incumbent appears on a channel in $\mathcal{H}$, the group that was sending beacons on the channel computes and distributes the updates. (e) Before a group leaves the network, it intimates one of the other groups (selected at random), which then computes and distributes the required updates.

G. Operation of the GMAC Protocol in the Multihop Case

So far, we have considered the single-hop scenario, i.e., we assumed that all nodes are within the transmission range of each other. In this subsection, we study the multi-hop scenario, in which not all nodes may be in the transmission range of a given node. In the multi-hop case, data is transferred from a source node u to a destination node v over multiple hops; routes are computed using a routing algorithm for ad hoc networks, e.g., Destination-Sequenced Distance-Vector Routing (DSDV [44]), Dynamic Source Routing (DSR [25]), Ad-hoc On-Demand Distance Vector Routing (AODV [45]) etc.

The challenges of receiver not being present on its HC and load balancing are addressed in the multi-hop case similar to the single hop case (see Sections V-B and V-C).

There is one more important issue to be addressed, viz., multi-channel hidden terminal problem [49] which is as follows (see Fig. 7(a)). Let us consider a node u which wants to communicate with another node v. Whenever node u finds node v's HC (say j) free, it sends a request message (REQ) to node v on HC j. Node v responds with an acknowledgment (ACK) packet on channel j and the channel is reserved for their data transfer. Suppose there is another node x whose HC is not channel j but which has data packets destined to a node

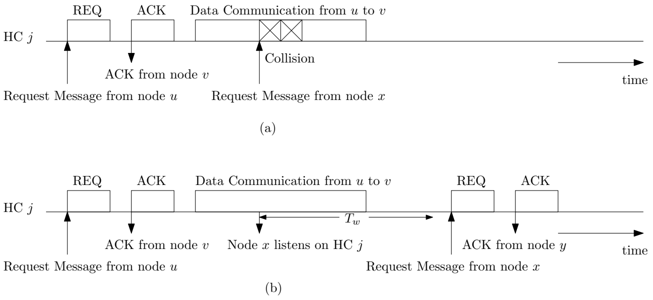

Fig. 7. In Fig. (a), the multichannel hidden terminal problem is explained, and in Fig. (b), how this problem is addressed in the GMAC protocol is shown.

y whose HC is channel j. Assume that node x is not in the transmission range of node u, but is in the range of node v. Hence, when node x senses HC j, it is not able to detect the ongoing data packet transmission from u to v. Unaware of the fact that already data communication is occurring on channel j, node x sends a request message (REQ) to node y on channel j, which results in a collision at node v. This problem can be addressed in the GMAC protocol as follows (see Fig. 7(b)). Whenever a node visits a channel which is not its HC, it must listen on that channel for a duration, say T_w, that is slightly more than one maximum length packet transmission duration. At any time during this duration if it finds that the channel is busy then it must wait for the ongoing communication to finish; at the end of the duration T_w, whenever the channel is found to be free, the node may send a REQ message on the channel. In this way we can overcome the multi-channel hidden terminal problem.

Distribution of updates in the multihop case can be done using any algorithm for efficiently broadcasting information in a multi-hop ad hoc network, e.g., the one proposed in [56].

VI. PARTITION OF THE NETWORK NODES INTO GROUPS

We assume that the set of nodes of the network executing the GMAC protocol is divided into multiple groups at the start of network operation, e.g., by the system administrator, possibly using estimates of the traffic volumes that need to be exchanged among different nodes. However, the traffic volumes that need to be exchanged among different pairs of nodes may change with time and hence the grouping of nodes needs to be periodically modified, e.g., modified once every few hours. Whenever groups are formed, they are numbered as $1, 2, 3, \ldots$. When re-grouping of nodes is done, it is done by the GL of the group that is currently group 1, using the greedy algorithm described in Section VI-C. That is, when the grouping that will be used in period t is found, it is computed [9] by the GL of group 1 in period $t-1$. In this section we formulate the problem of partitioning nodes into groups[10] based on the volumes of data traffic to be sent between different pairs of nodes, show that it is NP-complete and propose an algorithm to solve it. Before solving the optimization problem defined in Section VI-A, some information needs to be exchanged between nodes (e.g., traffic volumes between different pairs of nodes). This information is transmitted on the HCs of the previous period using the procedure for distribution of updates described in Section V-D. That is, the information that needs to be exchanged for the formation of groups in period t is transmitted on the HCs of period $t-1$.

A. Problem Formulation

Suppose the network is represented by an undirected graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. There is an edge between two nodes iff they are in the transmission range of each other. For $u, v \in \mathcal{N}$, let $\lambda_{u,v}$ be the rate (in bps) at which data needs to be transmitted from node u to node v. The rates $\{\lambda_{u,v} : u, v \in \mathcal{N}\}$ can be estimated using the history of data packet transmissions between different pairs of nodes. Let $\mathcal{M}_f, \mathcal{H}$, and $\mathcal{B}$ denote the set of free channels, HCs, and BCs respectively. The channels in $\mathcal{M}_f$ are partitioned into $\mathcal{H}$ and $\mathcal{B}$ and we assume that [11] $|\mathcal{B}| = \lceil \alpha |\mathcal{M}_f| \rceil$, where $0 < \alpha < 1$, i.e., a fixed fraction α of the free channels are used as BCs.

For each pair of nodes $u, v \in \mathcal{N}$, data is sent from u to v along a route that is computed using a routing algorithm for ad hoc networks, e.g., DSDV [44], DSR [25], or AODV [45]. If u and v are neighbours, then let $\lambda'_{u,v}$ be the total rate at which data needs to be exchanged between node u and node v– this includes the data originated at u (respectively, v) with destination v (respectively, u) as well as data originated by and/ or destined to nodes other than u and v, which is forwarded from u to v or from v to u. Let

$$X_{u,j} = \begin{cases} 1, & \text{if node } u \text{ is assigned to HC } j, \\ 0, & \text{else.} \end{cases}$$

Let

$$Y_{u,v} = \sum_{j \in \mathcal{H}} X_{u,j} X_{v,j}. \tag{1}$$

Then:

$$Y_{u,v} = \begin{cases} 1, & \text{if } u \text{ and } v \text{ belong to the same HC,} \\ 0, & \text{else.} \end{cases}$$

The problem is:

$$\max \sum_{u,v \in \mathcal{N} : (u,v) \in \mathcal{E}} \lambda'_{u,v} Y_{u,v} \tag{2}$$

subject to (1) and the following constraints:

$$\sum_{u \in \mathcal{N}} X_{u,j} \leq C, \forall j \in \mathcal{H}. \tag{3}$$

$$\sum_{j \in \mathcal{H}} X_{u,j} = 1, \forall u \in \mathcal{N}. \tag{4}$$

The constraint (3) ensures that at most C nodes can be assigned a channel j as HC. The constraint (4) ensures that each node is assigned exactly one HC. We refer to the above problem as the Group Formation Problem (GFP). Intuitively, an optimal or near-optimal solution to the GFP is expected to result in efficient operation of the GMAC protocol since,

---

[9] The GL of group 1 in period $t-1$ also selects the nodes that will act as the GLs of different groups in period t, possibly based on criteria such as the reliabilities of different nodes, proximities of different nodes to other nodes of their groups etc.

[10] We assume that there is exactly one group per HC. Hence, assigning nodes to groups is equivalent to assigning nodes to HCs.

[11] $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$ and $\lceil x \rceil$ denotes the ceiling of x.

by (2), under such a solution, $Y_{u,v}$ would likely be 1 for nodes u, v such that $\lambda'_{u,v}$ is high, i.e., a lot of data is exchanged between nodes u and v. Hence, such nodes u and v would likely be assigned the same HC, resulting in reduction of the data transfer delay since nodes u and v would not need to visit foreign channels to send data to each other.

Remark 1: The constraint in (3) indirectly limits the amount of interference on each HC j by ensuring that at most C nodes can be assigned channel j as HC. The interference can be more directly limited by adding another constraint, which ensures that the total number of pairs of neighbouring (and hence potentially mutually interfering) nodes on each HC is less than or equal to a threshold, say C', i.e., $\sum_{u,v\in\mathcal{N}:(u,v)\in\mathcal{E}} X_{u,j}X_{v,j} \leq C', \forall j \in \mathcal{H}$. The problem of analyzing the complexity of, and designing algorithms to solve, the GFP with the latter constraint added is an open problem for future research.

B. NP-completeness of the GFP

Theorem 1: The GFP is NP-complete.

Proof: The decision version of the GFP is as follows: "Given a number L, do there exist $\{X_{u,j} : u \in \mathcal{N}, j \in \mathcal{H}\}$ that satisfy the constraints (1), (3) and (4) such that $\sum_{u,v\in\mathcal{N}:(u,v)\in\mathcal{E}} \lambda'_{u,v}Y_{u,v} \geq L$?"

Given $\{X_{u,j} : u \in \mathcal{N}, j \in \mathcal{H}\}$, we can check in polynomial-time whether they satisfy (1), (3) and (4), and whether $\sum_{u,v\in\mathcal{N}:(u,v)\in\mathcal{E}} \lambda'_{u,v}Y_{u,v} \geq L$, where $Y_{u,v}, u, v \in \mathcal{N}$ are given by (1). Thus the GFP is in class NP [29]. We now show that the GFP is NP-complete by reducing the m-Dimensional Matching (mDM) problem, which has been shown to be NP-complete in [16], to it.

The decision version of the mDM problem is as follows: "Let $G = (\mathcal{N}, \mathcal{E})$ be a weighted undirected graph, w_e be the weight of edge $e \in \mathcal{E}$, and $|\mathcal{N}| = km$, where k and m are positive integers and $m \geq 3$. Given a number L, does there exist a partition of $\mathcal{N}$ into disjoint subsets $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_k$ of m vertices each such that the sum of the weights of the edges that have both endpoints in the same set $\mathcal{N}_i$ $(i = 1, 2, \ldots, k)$ is greater than or equal to L?"
Let

$$X'_{l,i} = \begin{cases} 1, & \text{if the vertex } l \text{ belongs to the subset } \mathcal{N}_i, \\ 0, & \text{else.} \end{cases}$$

and

$$Y'_{l_1,l_2} = \sum_{i=1}^{k} X'_{l_1,i}X'_{l_2,i}. \tag{5}$$

Then:

$$Y'_{l_1,l_2} = \begin{cases} 1, & \text{if the vertices } l_1, l_2 \text{ belong to the same subset,} \\ 0, & \text{else.} \end{cases}$$

Consider the following constraints:

$$\sum_{l\in\mathcal{N}} X'_{l,i} = m, i = 1, 2, \ldots, k. \tag{6}$$

$$\sum_{i=1}^{k} X'_{l,i} = 1, \forall l \in \mathcal{N}. \tag{7}$$

The decision version of the mDM problem can be written as:[12] "Given a number L, do there exist $X'_{l,i}$, $l \in \mathcal{N}, i \in \mathcal{H}$ such that $\sum_{l_1,l_2\in\mathcal{N}:(l_1,l_2)\in\mathcal{E}} w_{(l_1,l_2)}Y'_{l_1,l_2} \geq L$ and the constraints in (6) and (7) are satisfied?" The constraint (6) indicates that the cardinality of every subset $\mathcal{N}_i, i = 1, 2, \ldots, k$ is exactly m. The constraint (7) ensures that each vertex belongs to exactly one subset.

Now, we show that the mDM problem is polynomial-time reducible to GFP, i.e., mDM $<_p$ GFP. Consider the instance of the mDM problem stated in the previous paragraph. From this instance, we construct the following instance of the GFP. Let the network be represented by the graph $G = (\mathcal{N}, \mathcal{E})$, for each edge $e = (u, v) \in \mathcal{E}$, let $\lambda'_{u,v} = w_e$, suppose there are $|\mathcal{H}| = k$ HCs and let $C = m$.

Given this instance, we ask: do there exist $\{X_{u,j} : u \in \mathcal{N}, j \in \mathcal{H}\}$, such that the constraints (1), (3) and (4) are satisfied and $\sum_{u,v\in\mathcal{N}:(u,v)\in\mathcal{E}} \lambda'_{u,v}Y_{u,v} \geq L$? We claim that the answer is yes if and only if the answer to the question in the above mDM instance is yes. The necessity part is proved as follows. If the answer to the above question is yes then there exist $\{X_{u,j} : u \in \mathcal{N}, j \in \mathcal{H}\}$, which satisfy the constraints (1), (3) and (4), and $\sum_{u,v\in\mathcal{N}:(u,v)\in\mathcal{E}} \lambda'_{u,v}Y_{u,v} \geq L$. Let $X'_{l,i} = X_{l,i}$, $\forall l \in \mathcal{N}, i \in \mathcal{H}$ and Y'_{l_1,l_2} be given by (5) $\forall l_1, l_2 \in \mathcal{N}$. Then in the mDM problem, the constraint (7) holds since (4) holds. Also (6) follows from (3) and the fact that $|\mathcal{N}| = km$; finally, $\sum_{l_1,l_2\in\mathcal{N}:(l_1,l_2)\in\mathcal{E}} w_{(l_1,l_2)}Y'_{l_1,l_2} \geq L$ since $\sum_{u,v\in\mathcal{N}:(u,v)\in\mathcal{E}} \lambda'_{u,v}Y_{u,v} \geq L$. This proves necessity.

To prove sufficiency, suppose the answer to the question in the mDM problem is yes. Let $X_{l,i} = X'_{l,i}$, $\forall l \in \mathcal{N}, i \in \mathcal{H}$ and $Y_{u,v}$ for $u, v \in \mathcal{N}$ be given by (1). Then in the GFP, the constraint (3) follows from (6) and (4) follows from (7). Also, $\sum_{u,v\in\mathcal{N}:(u,v)\in\mathcal{E}} \lambda'_{u,v}Y_{u,v} \geq L$ since $\sum_{l_1,l_2\in\mathcal{N}:(l_1,l_2)\in\mathcal{E}} w_{(l_1,l_2)}Y'_{l_1,l_2} \geq L$. This proves sufficiency. The result follows.

■

C. Greedy Algorithm for solving the GFP

We now present a greedy algorithm to solve the GFP. In Section VIII-B, we show via simulations that this greedy algorithm finds a good solution to the GFP. We first provide an overview of the greedy algorithm. Since the objective of the GFP is to maximise the quantity in (2), we sort the edges $(u, v) \in \mathcal{E}$ in decreasing order of $\lambda'_{u,v}$. For each pair of nodes (u, v) in this order, we assign the same HC to u and v whenever possible. We now describe the greedy algorithm in detail. Renumber the values $\{\lambda'_{u,v} : (u, v) \in \mathcal{E}\}$ such that $r(1) \geq r(2) \geq \ldots \geq r(|\mathcal{E}|)$. Here $r(1) = \max\{\lambda'_{u,v} : (u, v) \in \mathcal{E}\}$ and $r(|\mathcal{E}|) = \min\{\lambda'_{u,v} : (u, v) \in \mathcal{E}\}$. Let $f(u)$ be 1 if node u has been assigned a HC and 0 otherwise. Let $N(j)$ denote the number of nodes assigned to HC j. Let j_u represent the HC assigned to node u. Let $\mathcal{H}$ be partitioned into $\mathcal{H}_{free}$ and $\mathcal{H}_{full}$, where $\mathcal{H}_{free}$ denotes the set of HCs that have been assigned to $C - 1$ or fewer nodes (set of free HCs) and $\mathcal{H}_{full}$ denotes the set of HCs that have been assigned to C nodes (set of full HCs).

A pseudocode of the algorithm is presented in Fig. 8. This algorithm executes a loop over all node pairs, i.e., from $k = 1$

---

[12] If the vertices l_1 and l_2 are not connected by an edge, then we assign an edge (l_1, l_2) between them with $w_{(l_1,l_2)} = 0$.

```
Initialization:
  • $\mathcal{H}_{free} = \mathcal{H}, \mathcal{H}_{full} = \phi$.
  • $f(u) = 0; \forall u \in \mathcal{N}$.
  • $X_{u,j} = 0; \forall u \in \mathcal{N}, j \in \mathcal{H}$.
  • $N(j) = 0; \forall j \in \mathcal{H}$.
Begin
1: Renumber the values $\{\lambda'_{u,v} : (u,v) \in \mathcal{E}\}$ such that $r(1) \geq r(2) \geq \ldots \geq r(|\mathcal{E}|)$. Here $r(1) = \max\{\lambda'_{u,v} : (u,v) \in \mathcal{E}\}$ and $r(|\mathcal{E}|) = \min\{\lambda'_{u,v} : (u,v) \in \mathcal{E}\}$
2: for $k$ = 1 to $|\mathcal{E}|$ do
3:     $\mathcal{H}_{free}$ = FREECHANNELS($N, \mathcal{H}_{free}$)
4:     $\mathcal{H}_{full} = \mathcal{H} \setminus \mathcal{H}_{free}$
5:     Identify the endpoints $u$ and $v$ of edge $r(k)$.
6:     if $(f(u) = 0) \wedge (f(v) = 0)$ then
7:         Select a HC $j$ at random from $\mathcal{H}_{free}$ such that $C - 2$ or fewer nodes are assigned to $j$. If no such HC $j$ exists, then jump to the next iteration at 2.
8:         $X_{u,j} = 1, X_{v,j} = 1$
9:         $j_u = j, j_v = j$
10:        $f(u) = 1, f(v) = 1$
11:        $N(j) \leftarrow N(j) + 2$
12:    else if $(f(u) = 1) \wedge (f(v) = 0)$ then
13:        if $j_u \in \mathcal{H}_{free}$ then
14:            $j_v = j_u, f(v) = 1$
15:            $N(j_v) \leftarrow N(j_v) + 1$
16:        else
17:            Jump to the next iteration at 2.
18:        end if
19:    else if $(f(u) = 0) \wedge (f(v) = 1)$ then
20:        if $j_v \in \mathcal{H}_{free}$ then
21:            $j_u = j_v, f(u) = 1$
22:            $N(j_u) \leftarrow N(j_u) + 1$
23:        else
24:            Jump to the next iteration at 2.
25:        end if
26:    end if
27: end for
28: For each node that is not yet assigned a HC, assign a HC that is randomly selected from $\mathcal{H}_{free}$.

1: function FREECHANNELS($N, \mathcal{H}_{free}$)
2:     $h_{free} = \mathcal{H}_{free}$
3:     for $i$ = 1 to $|\mathcal{H}_{free}|$ do
4:         $j = \mathcal{H}_{free}(i)$
5:         if $N(j) = C$ then
6:             $h_{free} = h_{free} \setminus j$
7:         end if
8:     end for
9:     return $h_{free}$
10: end function
End
```

Fig. 8. Greedy Algorithm for solving the GFP.

to $|\mathcal{E}|$. The values of $N(j)$ for all $j \in \mathcal{H}_{free}$ are checked. If for any HC, say j', $N(j') = C$, then j' is added to the set of full HCs and removed from the set of free HCs, i.e., $\mathcal{H}_{full} = \mathcal{H}_{full} \cup j'$ and $\mathcal{H}_{free} = \mathcal{H}_{free} \setminus j'$ (see the function $FREECHANNELS$ described in Fig. 8). The endpoints u and v of edge $r(k)$ are identified (see line 5). These nodes u, v are checked to see whether they are already assigned a HC or not; i.e., the values of $f(u)$ and $f(v)$ are checked. If both the nodes are not assigned a HC yet (i.e., $f(u) = 0$ and $f(v) = 0$), then a channel, say j, such that $(C - 2)$ or fewer nodes have been assigned to it, is chosen uniformly at random from among the set of free channels $\mathcal{H}_{free}$ and these nodes are assigned to it, i.e., $X_{u,j} = 1$, $X_{v,j} = 1$. The HC value assigned to these nodes is stored: $j_u = j$, $j_v = j$. The values of $f(u)$ and $f(v)$ are updated to 1 and the value of $N(j)$ is incremented by 2 (see lines 6 to 11). Suppose any one of them is zero and the other is one. If $f(u) = 1$ and $f(v) = 0$, it implies that node u is already assigned a HC $j = j_u$, i.e., $X_{u,j} = 1$. If $j \in \mathcal{H}_{free}$, then the node v is assigned to the HC j, i.e., $X_{v,j} = 1$; the value of j is stored in j_v and the value of $N(j)$ is incremented by 1; else no action is taken and we jump to line 2. The case $f(u) = 0$ and $f(v) = 1$ is dealt with similarly. In case both $f(u) = 1$ and $f(v) = 1$, it implies that both the nodes are already assigned HCs. So move to the next iteration without taking any action. The value of k is incremented, i.e., $k = k + 1$ and the algorithm is repeated till k reaches $|\mathcal{E}|$. It is possible that, even after the completion of all iterations some nodes might not have been assigned a HC yet. Such nodes are assigned a HC by randomly selecting a HC from $\mathcal{H}_{free}$.

The computational complexity of the proposed greedy algorithm is as follows. In the beginning, we arrange the $|\mathcal{E}|$ edges (u, v) in decreasing order of the values $\lambda'_{u,v}$. The complexity of this step, e.g., using the Merge sort algorithm [29], is $\mathcal{O}(|\mathcal{E}| \log |\mathcal{E}|)$. Then, for each edge (u, v) in $\mathcal{E}$, we assign each of the nodes u and v to one channel selected from the H HCs; the complexity of this step is $\mathcal{O}(H|\mathcal{E}|)$. So the total computational complexity is $\mathcal{O}(|\mathcal{E}|(H + \log |\mathcal{E}|))$.

VII. ANALYSIS OF GMAC

In this section, we analyze a simplified version of the GMAC protocol that was presented in Section V. The analysis provides insight into: (i) the difference in the performance of the GMAC protocol with and without BCs, (ii) how various protocol parameters (e.g., $CLTH_u$) should be chosen so as to maximize the protocol's stability region.

A. *Simplified Model*

We assume that each node is in the transmission range of every other node. Let $\mathcal{G} = \{1, \ldots, G\}$ be the set of groups. Let n_g be the number of nodes in group g. When PUs appear on and leave channels at a time scale that is much slower than that of the operation of the GMAC protocol (e.g., when the PUs are TV stations), the number of free channels varies slowly. As an approximation for this scenario, we assume that there are a constant number, say M', of free channels, of which $H > 0$ (respectively, $B = M' - H$) channels are HCs (respectively, BCs). Let $\{\mathcal{G}_1, \ldots, \mathcal{G}_H\}$ be a partition of $\mathcal{G}$ such that groups in the set $\mathcal{G}_h$ have HC h. So the number of nodes on HC h is $N_h = \sum_{g \in \mathcal{G}_h} n_g$. Let the total arrival rate of packets destined to nodes in group g be λ_g– this includes the packets sent to nodes of group g by other nodes of group g as well as nodes of other groups. So the total arrival rate to nodes on HC h is $\Lambda_h = \sum_{g \in \mathcal{G}_h} \lambda_g$. We assume that the arrival processes of packets are Poisson with the above rates. Recall that on each channel, nodes contend for channel access using 802.11 DCF and that the total packet throughput on a channel is a function of the number of contending nodes [7]. To model this, let the total packet service rate on HC h be $\mu(N_h)$. For tractability, this ignores the facts that: (i) nodes with HC h leave channel h from time to time to send packets to nodes of other HCs and to visit BCs, and nodes with other HCs also intermittently visit h, (ii) the number of nodes contending for access on channel h varies with time. Similarly, let the service rate on each BC be

μ_0, again assumed to be a constant for simplicity. We assume the packet service times to be exponentially distributed random variables.

Recall from Section V-C that the traffic from congested HCs is off-loaded to BCs. To model this, we assume that when C_h packets are present on HC h (this includes the packet whose transmission is ongoing and those queued for transmission on the channel), new arriving packets to channel h are redirected to a BC (chosen randomly from the B BCs). Here, C_h is a protocol parameter analogous to the parameter $CLTH_u$ defined in Section V-C. For simplicity we assume that all the packets off-loaded to a BC b are served on channel b itself, although in the original GMAC protocol some of them may be redirected back to their HC.

The stability region of the protocol is the set of all arrival rate vectors $(\lambda_1, \ldots, \lambda_G)$ for which the network is stable, i.e., the packet queue lengths do not go to infinity [13].

B. Results

First, suppose $B \geq 1$. Under the assumptions in Section VII-A, HC h is an $M/M/1/C_h$ queue [14] with arrival rate Λ_h and service rate $\mu(N_h)$. It can be modeled as a Continuous Time Markov Chain (CTMC [55]) with states $0, 1, \ldots, C_h$, where the channel is in state i iff there are i packets on the channel. The stationary probability, p_i, of state i is given by $p_i = \left(\frac{\Lambda_h}{\mu(N_h)}\right)^i \frac{\left(\frac{\Lambda_h}{\mu(N_h)}\right)-1}{\left(\frac{\Lambda_h}{\mu(N_h)}\right)^{C_h+1}-1}$ (see [55]). The probability, say $p_B(h)$, that a packet is off-loaded to a BC is the probability that the incoming packet finds C_h packets present on the channel, which by the PASTA [15] property is:

$$p_B(h) = p_{C_h} = \left(\frac{\Lambda_h}{\mu(N_h)}\right)^{C_h} \frac{\left(\frac{\Lambda_h}{\mu(N_h)}\right) - 1}{\left(\frac{\Lambda_h}{\mu(N_h)}\right)^{C_h+1} - 1}. \tag{8}$$

The arrival process to each BC b is the superposition of the processes of packets off-loaded from all the HCs. The rate of the arrival process to BC b is given by:

$$\Lambda_b = \frac{1}{B} \sum_{h=1}^{H} p_B(h) \Lambda_h. \tag{9}$$

Now, suppose we allocate HCs to groups such that each HC h is overloaded, i.e., $\frac{\Lambda_h}{\mu(N_h)} > 1$. Then for C_h large enough: $\left(\frac{\Lambda_h}{\mu(N_h)}\right)^{C_h+1} - 1 \approx \left(\frac{\Lambda_h}{\mu(N_h)}\right)^{C_h+1}$. Under this approximation, by (8), we get: $p_B(h) = 1 - \frac{\mu(N_h)}{\Lambda_h}$. Substituting this into (9), we get: $\Lambda_b = \frac{1}{B}\left[\sum_{h=1}^{H} \Lambda_h - \sum_{h=1}^{H} \mu(N_h)\right]$. Now, note that: $\sum_{h=1}^{H} \Lambda_h = \sum_{h=1}^{H} \sum_{g \in \mathcal{G}_h} \lambda_g = \sum_{g=1}^{G} \lambda_g = \lambda_{total}$, where λ_{total} is the total arrival rate of traffic into the system. Hence: $\Lambda_b = \frac{1}{B}\left[\lambda_{total} - \sum_{h=1}^{H} \mu(N_h)\right]$.

Now, clearly the system is stable iff each BC is stable. Under the assumptions in Section VII-A, a BC is a G/M/1 queue [55] with arrival rate Λ_b and service rate μ_0. So the system is stable iff $\Lambda_b < \mu_0$. By the expression for Λ_b derived above, this is equivalent to: $\lambda_{total} < B\mu_0 + \sum_{h=1}^{H} \mu(N_h) = \lambda_{stable}$ (say). Hence, the stability region of the network is:

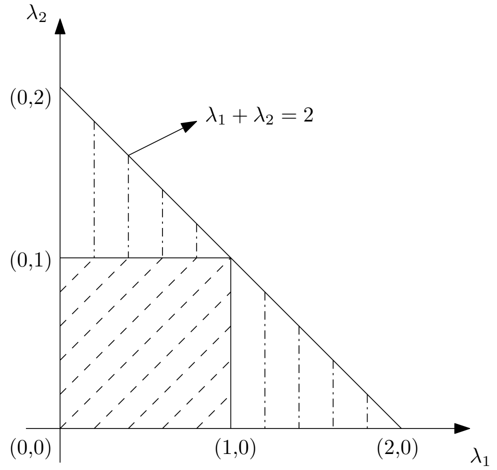

Fig. 9. Consider $G = 2$, $M' = 2$, $\mu(n) = 1$ for $n = 1, 2$, λ_1 and λ_2 on the x and y axis respectively. The stability region with $B = 0$ is the crossed dash lines region, whereas that with $B \geq 1$ is the union of the crossed dash lines region and the dash-dotted lines region.

$$\left\{(\lambda_1, \ldots, \lambda_G) : \lambda_{total} = \sum_{g=1}^{G} \lambda_g < \lambda_{stable}\right\}. \tag{10}$$

Next, suppose $B = 0$. Then packets cannot be off-loaded from HCs to BCs, and hence each HC must serve all the packets arriving on it. Hence, HC h is an M/M/1 queue with arrival rate Λ_h and service rate $\mu(N_h)$. Recall that HC h is stable iff $\Lambda_h < \mu(N_h)$ [55]. The network is stable iff every HC is stable. Thus, the stability region is given by:

$$\left\{(\lambda_1, \ldots, \lambda_G) : \Lambda_h = \sum_{g \in \mathcal{G}_h} \lambda_g < \mu(N_h),\ h = 1, \ldots, H\right\}. \tag{11}$$

Example: Suppose the service rates are constant: $\mu(n) = \mu, \forall n$ and $\mu_0 = \mu$. Then: $\lambda_{stable} = B\mu + \sum_{h=1}^{H} \mu = (B + H)\mu = M'\mu$. So by (10), the stability region with $B \geq 1$ is:

$$\left\{(\lambda_1, \ldots, \lambda_G) : \lambda_{total} = \sum_{g=1}^{G} \lambda_g < M'\mu\right\}. \tag{12}$$

The above holds irrespective of the choice of the partition $\{\mathcal{G}_1, \ldots, \mathcal{G}_H\}$. Note that the stability region in (12) is the maximum stability region that any policy can achieve, since all arrival rate vectors whose total arrival rate is less than the total available service rate in the system are in the region. Thus, the GMAC protocol is optimal when $B \geq 1$. The stability region with $B = 0$ is that in (11), where $\mu(N_h) = \mu$ and the partition $\{\mathcal{G}_1, \ldots, \mathcal{G}_H\}$ is selected so as to minimize the quantity $\max_{h \in \{1, \ldots, H\}} \Lambda_h$. In particular, note that the stability region with $B = 0$ is a subset of the following set:

$$\{(\lambda_1, \ldots, \lambda_G) : \lambda_g < \mu,\ g = 1, \ldots, G\}. \tag{13}$$

---

[13] Formally, the network is stable if the Continuous Time Markov Chain (CTMC) corresponding to the queue lengths at nodes is positive recurrent [55].

[14] Recall that an $M/M/1/C_h$ queue is a queue with Poisson arrivals, exponential service times, 1 server and limit C_h on the number of customers in the system [55].

[15] "Poisson Arrivals See Time Averages" [55].

Clearly, the stability region in (13) is much smaller than that in (12)– Fig 9 shows an example. Equations (10) and (11) and the above example show that the use of BCs can significantly enhance the stability region. Intuitively this is because if no BCs are used, the system becomes unstable if even one HC is unstable, whereas BCs enable the even distribution of the traffic load across channels and hence full utilization of the available resources. Next, as explained above, the parameter C_h is analogous to the parameter $CLTH_u$. Recall that we chose the parameter C_h to be a large value (see the paragraph after (9)), which allowed us to achieve a large stability region– in particular, the optimal stability region in the constant service rates case. This suggests that it is beneficial to select a high value for the GMAC protocol parameter $CLTH_u$. Intuitively, this is because when $CLTH_u$ is high, nearly the entire available capacity of the HCs is utilized.

VIII. SIMULATIONS

We now evaluate the performance of the GMAC protocol via simulations.

A. Single Hop Case

Throughout this subsection, we consider a CRN with $G = 10$ groups and n nodes (SUs) in each group. Let $N = Gn$ (respectively, M) denote the total number of nodes (respectively, channels). In this subsection, we consider the single hop case in which all N nodes are in the transmission range of each other. On a given channel and in a given superframe (SF), a PU is present (respectively, absent) throughout the SF with probability r (respectively, $1 - r$), independently of other channels and SFs. At each node in group k, at the beginning of each SF, with probability $\frac{\lambda_k}{n \times burst\_s}$ (respectively, $1 - \frac{\lambda_k}{n \times burst\_s}$), $burst\_s$ (respectively, 0) data packets arrive, where $burst\_s = 20$. Thus, the expected total arrival rate at group k is λ_k packets per SF. Each SF is divided into 100 slots, and the time required for transmission of a data packet is 2 slots. Also, the length of a QP (respectively, SS) is 1 (respectively, 2) slot(s). In DTPs, nodes contend as in Slotted ALOHA [6] to acquire access to the channel; also, upon acquiring access, a node is allowed to continuously transmit at most nh, nf and nb data packets on HCs, foreign channels and BCs respectively [16]. *Mean_CLTH* denotes the average value of the parameter $CLTH_u$ defined in Section V-C. Let p denote the fraction of data packets of a node whose destination is another node of its own group and α be the fraction of free channels that are used as BCs.

First, we let $\lambda_k = k\lambda_0$, for k = 1, 2, ..., 10. The left (respectively, right) plot in Fig. 10 shows the maximum value, say λ_0^*, of λ_0 for which the network is stable (respectively, the average delay, say Avg_Delay_Stable, for a fixed arrival rate that belongs to the stability region) versus α. These plots show that: (i) use of BCs significantly enhances the performance over the no BCs (α = 0) case, and (ii) the best network performance (i.e., highest λ_0^* and lowest Avg_Delay_Stable) is obtained for medium values of α. This is because, for low values of α, only a few BCs are available for offloading traffic from HCs, and hence the HC of group 10, whose arrival rate is the highest ($10\lambda_0$), becomes unstable for a small value of λ_0. On the other hand, if α is larger than a threshold, there are a large number of groups on each HC, resulting in a lot of collisions, and hence degraded network performance. Next, we let $\lambda_k = \lambda$ for all $k = 1, 2, \ldots, 10$. The left (respectively, right) plot in Fig. 11 shows the maximum value, say λ^*, of λ for which the network is stable (respectively, Avg_Delay_Stable) versus *Mean_CLTH* for $\alpha = 0.2, 0.5$ and 0.8. Note that for a fixed α, λ^* and Avg_Delay_Stable are best for a medium value of *Mean_CLTH*. This is because for very low values of *Mean_CLTH*, the HCs are underutilized, whereas for very high values, a large number of collisions occur on HCs, thus degrading the network performance. The plots in Fig. 11 also show, similar to the trends in Fig. 10, that for each fixed value of *Mean_CLTH*, the best λ^* and Avg_Delay_Stable are observed at a medium value of α ($\alpha = 0.5$).

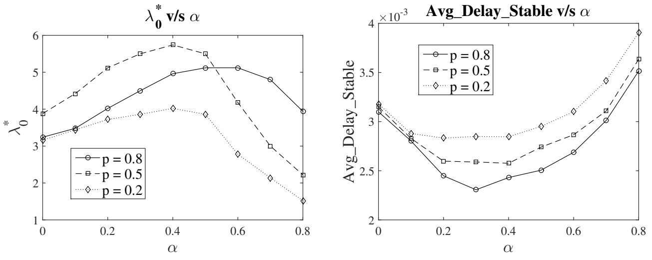

Fig. 10. The following parameters are used in these plots: $M = 20, N = 100, r = 0.5, nf = 1, nh = nb = 10, T_{h,min} = 5, T_{f,max} = 60, Mean\_CLTH = 0.3$. Also, $\lambda_0 = 1$ for the plot on the right.

The left (respectively, right) plot in Fig. 12 shows λ^* (respectively, Avg_Delay_Stable) versus p for three values of N. Also, the curve labeled "Ideal" in the left plot is the value of λ^* under an ideal MAC protocol [17] in which at each point in time, all the bandwidth unoccupied by PUs is used for data transmissions. The left (respectively, right) plot in Fig. 12 shows that λ^* increases (respectively, Avg_Delay_Stable decreases) in p. This is because for high values of p, most of the traffic of a node is intended for other nodes from its own HC; so the overheads involved in switching to foreign channels are reduced. The plots also show that for each fixed p, the performance degrades as N increases, which is because the number of collisions increases. Also, the left plot shows that for high values of p, the λ^* achieved by GMAC is around 62% of that for the ideal case.

The left (respectively, right) plot in Fig. 13 shows λ^* (respectively, Avg_Delay_Stable) versus M for three values of N. The left (respectively, right) plot in Fig. 13 shows that λ^* increases (respectively, Avg_Delay_Stable decreases) in M, which is because more channels are available for data transmission. The performance degrades as N increases as in the plots of Fig. 12. Finally, for the left plot of Fig. 13, the GMAC protocol achieves a λ^* that is between 40% and 45% of that for the ideal case. Thus, a large fraction (close to 1/2) of the bandwidth unoccupied by PUs is utilized by the GMAC protocol for data transmissions.

---

[16] Note that transmission of multiple data packets upon acquiring access results in a higher efficiency than if only one data packet were transmitted. The limits nh, nf and nb are imposed to ensure short-term fairness in the transmission opportunities that different nodes get.

[17] Note that this protocol cannot be practically implemented in a distributed manner and is considered only for comparison with the GMAC protocol.

We have compared the performance of the GMAC protocol with those of two existing multichannel MAC protocols for CRNs, namely, Hopping based MAC (HMAC) protocol [50] and Multichannel MAC (MMAC) protocol [49] combined with Single-Radio Adaptive Channel (SRAC) algorithm [36]. The MMAC protocol [49] has been designed for multichannel wireless networks, which are closely related to CRNs, with a difference being that in CRNs, secondary nodes need to avoid interfering with PU transmissions. On the other hand, the SRAC algorithm [36] incorporates sensing by secondary nodes to avoid interference to PU transmissions, and is a flexible algorithm that can be used in conjunction with one from among multiple MAC protocols for multichannel wireless networks. So the combination SRAC-MMAC is a multichannel MAC protocol for CRNs. The left (respectively, right) plot in Fig. 14 shows λ^* (respectively, Avg_Delay_Stable) versus p for the GMAC, HMAC and SRAC-MMAC protocols. From the plots in Fig. 14, we can infer that for sufficiently large values of p, the GMAC protocol significantly outperforms the HMAC and SRAC-MMAC protocols. This is consistent with intuition since the design of the GMAC protocol has been customised for networks with large values of p.

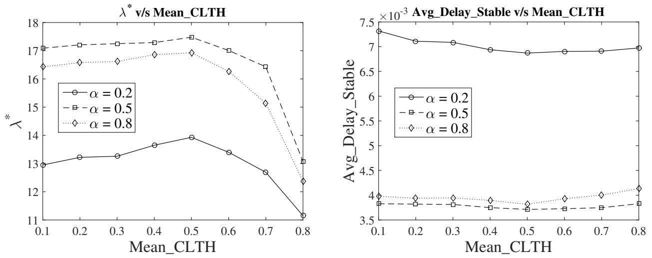

Fig. 11. The following parameters are used in these plots: $M = 15, N = 100, p = 0.8, r = 0.4, nf = 5, nh = 10, nb = 30, T_{h,min} = 10, T_{f,max} = 70$. Also, $\lambda = 10$ for the plot on the right.

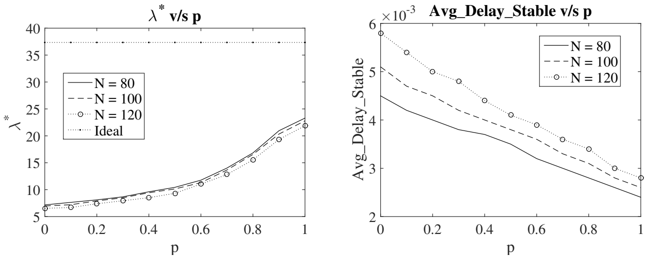

Fig. 12. The following parameters are used in these plots: $M = 15, \alpha = 0.5, r = 0.4, nf = 5, nh = 20, nb = 30, T_{h,min} = 10, T_{f,max} = 70$. Also, $\lambda = 5$ for the plot on the right.

B. Multihop Case

In this subsection we evaluate the performance of the proposed greedy algorithm (see Section VI-C) for solving the GFP and also evaluate the performance of the GMAC protocol in the multihop case.

Let us consider a CRN as an undirected graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Let N = $|\mathcal{N}|$ be the total number of nodes. For the simulations, such a graph is generated in a square of size 1×1. Nodes are placed inside this square at locations which are chosen uniformly at random from $((0, 1) \times (0, 1))$. There is an edge between two nodes iff they are separated by a distance less than or equal to d, where d is a parameter. If nodes u and v are connected by an edge, then we assign the edge weight between them as 1, else we assign ∞. For this network we compute the shortest routes between all the node pairs using Dijkstra's algorithm [30]. For every node pair (u, v), assume that the rate at which packets need to be sent from node u to v is $\lambda_{u,v}$. For the simulations, $\lambda_{u,v}$ is chosen uniformly at random from $(0.5\lambda_{GFP}, 1.5\lambda_{GFP})$, where λ_{GFP} is a parameter.

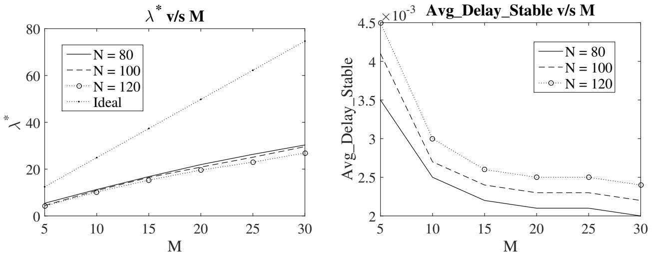

Fig. 13. The following parameters are used in these plots: $p = 0.8, \alpha = 0.5, r = 0.4, nf = 5, nh = 20, nb = 30, T_{h,min} = 10, T_{f,max} = 70$. Also, $\lambda = 4$ for the plot on the right.

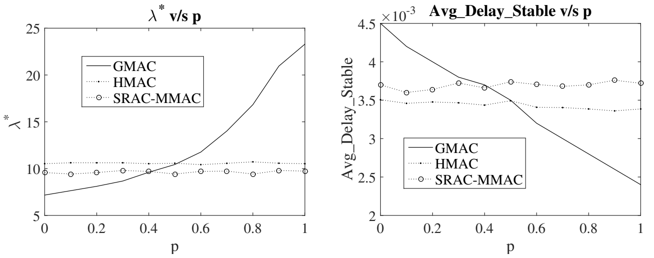

Fig. 14. The following parameters are used in these plots: $N = 80, M = 15, \alpha = 0.5, r = 0.4, nf = 5, nh = 20, nb = 30, T_{h,min} = 10, T_{f,max} = 70$. Also, $\lambda = 5$ for the plot on the right.

The left (respectively, right) plot in Fig. 15 shows the sum of data rates, $\sum_{u,v\in\mathcal{N}:(u,v)\in\mathcal{E}} \lambda'_{u,v} Y_{u,v}$, in (2) versus N (respectively, versus H) for five different group formation schemes, namely, the greedy algorithm (proposed in Section VI-C), Random Edge scheme, Random Node scheme, Tabu Search [28] method and Genetic Algorithm [28] method. In the Random Edge scheme, node pairs are picked in a random order, instead of in decreasing order of rates as in the greedy algorithm, and assigned HCs as in Fig. 8. In the Random Node scheme, nodes are picked in a random order and assigned HCs randomly. Also, the Tabu Search and Genetic Algorithm methods are heuristic algorithms that have been extensively used in prior work to find solutions to various combinatorial optimization problems [28]. From Fig. 15, we can conclude that the proposed greedy algorithm significantly outperforms the other four group formation schemes.

We also compared the performance of the proposed greedy algorithm with that of the exhaustive search method for small parameter values. The left (respectively, right) plot in Fig. 16 shows the sum of data rates versus N for $H = 2$ and $C = 5$ (respectively, versus H for $N = 10$ and $C = 2$). From the left plot, it can be seen that for small values of N, the performances of the greedy algorithm and the exhaustive search method are identical, and even for high values of N, the performance of the greedy algorithm is close to that of the exhaustive search method. From the right plot, too, it can be seen that the performance of the greedy algorithm is close to that of the exhaustive search method.

We consider that, at a node u at the beginning of each SF, with probability $\frac{\lambda_{u,v}}{burst\_s}$ (respectively, $1 - \frac{\lambda_{u,v}}{burst\_s}$), $burst\_s$ (respectively, 0) data packets destined to node v arrive, where $burst\_s = 20$. Thus, the expected arrival rate of packets at node u destined to node v is $\lambda_{u,v}$ packets per SF.

The left (respectively, right) plot in Fig. 17 shows the maximum value, say λ^*_{GFP}, of λ_{GFP} for which the network is stable (respectively, the average delay, say Avg_Delay_Stable, for a fixed arrival rate that belongs to the stability region) versus α for the values $M = 15, 20, 25$. These plots show that the best network performance (i.e., highest λ^*_{GFP} and lowest Avg_Delay_Stable) is obtained for medium values of α. This is because, for low values of α, only a few BCs are available for offloading traffic from HCs. On the other hand, if α is larger than a threshold, then the sizes of groups assigned to each HC become larger due to reduction in the number of HCs, resulting in a lot of collisions on every HC, and hence degraded network performance.

The left (respectively, right) plot in Fig. 18 shows the λ^*_{GFP} (respectively, Avg_Delay_Stable) versus *Mean_CLTH* for $\alpha = 0.2, 0.5$ and 0.8. The trends in these figures are similar to the trends in Fig. 11.

The left (respectively, right) plot in Fig. 19 shows λ^*_{GFP} (respectively, Avg_Delay_Stable) versus M for three values of N. The left (respectively, right) plot in Fig. 19 shows that λ^*_{GFP} increases (respectively, Avg_Delay_Stable decreases) as M increases, which is because more channels are available for data transmission. The plots also show that for each fixed M, the performance degrades as N increases, which is because the number of collisions increases.

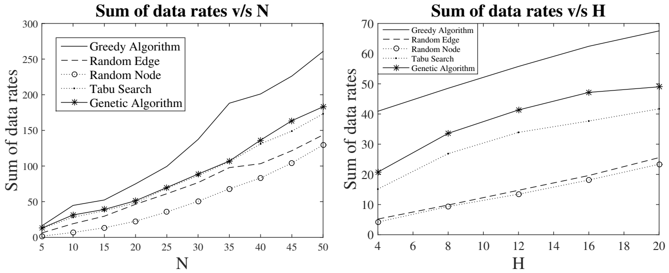

Fig. 15. The following parameters are used in these plots: $d = 0.7, \lambda_{GFP} = 0.5$. In the left plot $H = 10$, $C = 5$ and in the right plot $N = 40$, $C = 2$ are used.

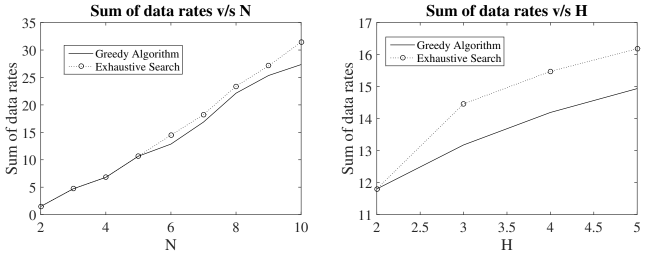

Fig. 16. The following parameters are used in these plots: $d = 0.7, \lambda_{GFP} = 0.5$. In the left plot $H = 2$, $C = 5$ and in the right plot $N = 10$, $C = 2$ are used.

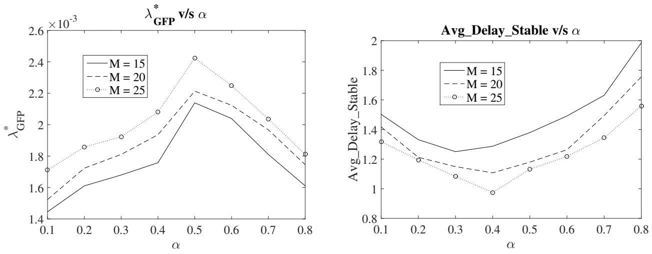

Fig. 17. The following parameters are used in these plots: $N = 30, r = 0.4, nf = 10, nh = nb = 20, T_{h,min} = 10, T_{f,max} = 40, Mean\_CLTH = 0.3, d = 0.7$. Also, $\lambda_{GFP} = 10^{-3}$ for the plot on the right.

IX. CONCLUSIONS

We presented the design and performance evaluation of the GMAC protocol, which achieves high performance by exploiting group structure information to reduce channel switching overhead. The protocol requires each secondary node to have only one narrowband transceiver and does not rely on a control channel. We showed that the problem of partitioning the set of nodes into groups is NP-complete and provided a greedy algorithm to solve it. Our analysis of the GMAC protocol via a queuing theoretic framework and extensive simulations show that the use of buffer channels can significantly enhance the stability region and reduce the average delay by evenly distributing the traffic load across the free channels. Our simulations also show that a large fraction of the bandwidth unoccupied by primary users is utilized by the GMAC protocol for data transmissions.

REFERENCES

[1] C. M. G. Algora, V. A. Reguera, N. Deligiannis, and K. Steenhaut, "Review and Classification of Multichannel MAC Protocols for Low-Power and Lossy Networks," *IEEE Access*, vol. 5, pp. 19536-19561, Sept. 2017.

[2] W. Alhakami, A. Mansour, and G. A. Safdar, "Performance Analysis of a Novel Decentralised MAC Protocol for Cognitive Radio Networks," in *Proc. IEEE APWiMob*, pp. 85-91, Sept. 2016.

[3] K. H. Almotairi and X. Shen, "A Distributed Multi-Channel MAC Protocol for Ad Hoc Wireless Networks," *IEEE Trans. Mob. Comput.*, vol. 14, no. 1, pp. 1-13, Jan. 2015.

[4] I. Akyildiz, W.-Y. Lee, M. Vuran, and S. Mohanty, "NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey," *Comp. Netw.*, vol. 50, no. 13, pp. 2127-2159, Sep. 2006.

[5] 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. ACM MobiCom*, pp. 216-230, Sept. 2004.

[6] D. Bertsekas and R. Gallager, *"Data Networks,"* 2nd ed, PHI, 1992.

[7] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.

[8] H.-L. Chao, T.-L. Li, C.-C. Chung, and S.-H. Wu, "Throughput Analysis of a Hybrid MAC Protocol for WiFi-Based Heterogeneous Cognitive Radio Networks," in *Proc. IEEE VTC*, pp. 1-5, Sept. 2015.

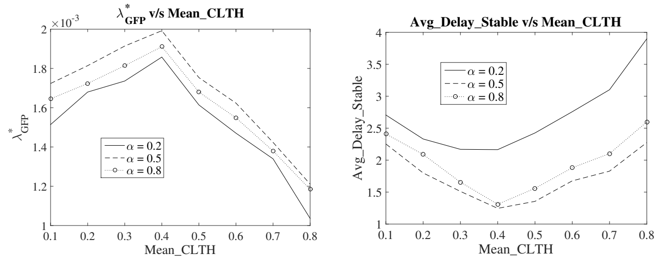

Fig. 18. The following parameters are used in these plots: $M = 20, N = 30, r = 0.4, nf = 10, nh = nb = 20, T_{h,min} = 10, T_{f,max} = 40, d = 0.7$. Also, $\lambda_{GFP} = 10^{-3}$ for the plot on the right.

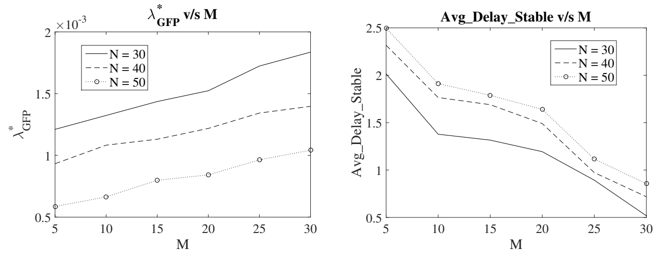

Fig. 19. The following parameters are used in these plots: $r = 0.4, nf = 10, nh = nb = 20, T_{h,min} = 10, T_{f,max} = 40, \alpha = 0.5, Mean\_CLTH = 0.2, d = 0.7$. Also, $\lambda_{GFP} = 5 \times 10^{-4}$ for the plot on the right.

[9] Y.-C. Cheng, E. H. Wu, and G.-H. Chen, "A Decentralized MAC Protocol for Unfairness Problems in Coexistent Heterogeneous Cognitive Radio Networks Scenarios With Collision-Based Primary Users," *IEEE Sys. Journ.*, vol. 10, no. 1, pp. 346-357, Mar. 2016.

[10] I. Chlamtac, A. Farago, A. D. Myers, V. R. Syrotiuk, and G. Zaruba, "ADAPT: A Dynamically Self-adjusting Media Access Control Protocol for Ad Hoc Networks," in *Proc. IEEE GLOBECOM*, pp. 11-15, Dec. 1999.

[11] K. W. Choi, W. S. Jeon, and D. G. Jeong, "Sequential Detection of Cyclostationary Signal for Cognitive Radio Systems," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4480-4485, Nov. 2009.

[12] C. Cordeiro and K. Challapali, "C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks," in *Proc. IEEE DySPAN*, pp. 147-157, Apr. 2007.

[13] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. C. Sicker, and D. Grunwald, "MultiMAC - An Adaptive MAC Framework for Dynamic Radio Networking," in *Proc. IEEE DySPAN*, pp. 548-555, Nov. 2005.

[14] A. D. Domenico, E. C. Strinati, and M. D. Benedetto, "A Survey on MAC Strategies for Cognitive Radio Networks," *IEEE Commun. Survey Tuts.,* vol 14, no. 1, pp. 21-44, First Quart. 2012.

[15] Federal Communications Commission, "FCC, ET. Docket No 03-222. Notice of proposed rule making and order," Tech. Rep., 2003.

[16] T. A. Feo and M. Khellaf, "A Class of Bounded Approximation Algorithms for Graph Partitioning," *Networks,* vol. 20, no. 2, pp. 181-195, Mar. 1990.

[17] A. B. Flores, R. E. Guerra, E. W. Knightly, P. Ecclesine, and S. Pandey, "IEEE 802.11AF: A Standard for TV White Space Spectrum Sharing," *IEEE Commun. Mag.,* vol. 51, no. 10, pp. 92-100, Oct. 2013.

[18] L. Gavrilovska, D. Denkovski, V. Rakovic, and M. Angjelichinoski, "Medium Access Control Protocols in Cognitive Radio Networks: Overview and General Classification," *IEEE Commun. Survey Tuts.,* vol 16, no. 4, pp. 2092-2124, Fourth Quart. 2014.

[19] X. Hu, L. Ma, S. Huang, T. Huang, and S. Liu, "Dynamic Channel Slot Allocation Scheme and Performance Analysis of Cyclic Quorum Multichannel MAC Protocol," *Math. Prob. in Eng.,* vol. 2017, Sept. 2017.

[20] W. Hu, H. Yousefizadeh, and X. Li, "Load Adaptive MAC: A Hybrid MAC Protocol for MIMO SDR MANETs," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3924-3933, Nov. 2011.

[21] IEEE Standard for IT, Telecom and Information Exchange between Systems, Local and Metropolitan Area Networks, Specific Requirements, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, Mar. 2007.

[22] "IEEE 802.22 Standard for Wireless Regional Area Networks," Jul. 2011, Website: http://grouper.ieee.org/groups/802/22/.

[23] "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Wireless Regional Area Networks (WRAN)–Specific Requirements–Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands," July 2011.

[24] N. Jain, S. R. Das, and A. Nasipuri, "A Multichannel CSMA MAC Protocol with Receiver-based Channel Selection for Multi-hop Wireless Networks," in *Proc. IEEE ICCCN*, pp. 432-439, Oct. 2001.

[25] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Proc. ACM SIGCOMM*, pp. 153-181, Aug. 1996.

[26] J. Jia, Q. Zhang, and X. Shen, "HC-MAC: A Hardware-Constrained Cognitive MAC for Efficient Spectrum Management," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 106-117, Jan. 2008.

[27] S. Kadam, D. Prabhu, N. Rathi, P. Chaki, and G. S. Kasbekar, "Exploiting Group Structure in MAC Protocol Design for Multichannel Ad Hoc Cognitive Radio Networks," in *Proc. IEEE WCNC*, pp. 1-7, Apr. 2016.

[28] P. Kadluczka and K. Wala, "Tabu Search and Genetic Algorithms for the Generalized Graph Partitioning Problem," in *Control and Cybernetics*, vol. 24, pp. 459-476, Jan. 1995.

[29] J. Kleinberg and E. Tardos, *"Algorithm Design,"* Addison Wesley, 2005.

[30] J. F. Kurose and W. W. Ross, *"Computer Networking: A Top - Down Approach,"* Pearson, 6th ed, 2013.

[31] P. Kyasanur and N. Vaidya, "Protocol Design Challenges for Multi-hop Dynamic Spectrum Access Networks," in *Proc. IEEE DySPAN*, pp. 645-648, Nov. 2005.

[32] C. Li, B. Zhang, X. Yuan, S. Ullah, and A. V. Vasilakos, "MC-MAC: a Multi-Channel based MAC Scheme for Interference Mitigation in WBANs," *Wireless Netw.*, vol. 24, no. 3, pp. 719-733, Sep. 2016.

[33] J. Li, T. Luo, J. Gao, and G. Yue, "A MAC Protocol for Link Maintenance in Multichannel Cognitive Radio Ad Hoc Networks," *J. Commun. and Netw.*, vol. 17, no. 2, pp. 172-183, Apr. 2015.

[34] W. Lou, X. Liu, P. Feng, and N. Wu, "An Integrated 0.38-6 GHz, 9-12 GHz Fully Differential Fractional-N frequency Synthesizer for Multi-Standard Reconfigurable MIMO Communication Application," *Ana. Integr. Circ. and Sign. Proces.*, vol. 78, no. 3, pp 807-817, Mar. 2014.

[35] L. Ma, X. Han, and C. -C. Shen, "Dynamic Open Spectrum Sharing MAC Protocol for Wireless Ad Hoc Networks," in *Proc. IEEE DySPAN*, pp. 203-213, Nov. 2005.

[36] L. Ma, C. -C. Shen, and B. Ryu, "Single-Radio Adaptive Channel Algorithm for Spectrum Agile Wireless Ad Hoc Networks," in *Proc. IEEE DySPAN*, pp. 547-558, Apr. 2007.

[37] D. Maldonado, B. Le, A. Hugine, T. W. Rondeau, and C. W. Bostian, "Cognitive Radio Applications to Dynamic Spectrum Allocation: A Discussion and an Illustrative Example," in *Proc. IEEE DySPAN*, pp. 597-600, Nov. 2005.

[38] MAXIM MAX2830 2.4GHz to 2.5GHz 802.11g/b RF Transceiver, PA, and Rx/Tx/Antenna Diversity Switch (March 2011), Website: http://datasheets.maximintegrated.com/en/ds/MAX2830.pdf.

[39] J. Mo, H. -S. W. So, and J. Walrand, "Comparison of Multichannel MAC Protocols," *IEEE Trans. Mob. Comput.*, vol. 7, no. 1, pp. 50-65, Jan. 2008.

[40] A. Nasipuri, J. Zhuang, and S. R. Das, "A Multichannel CSMA MAC Protocol for Multihop Wireless Networks," in *Proc. IEEE WCNC*, pp. 1402-1406, Sep. 1999.

[41] A. Nasipuri and S. R. Das, "Multi-Channel CSMA with Signal Power-based Channel Selection for Multi-hop Wireless Networks," in *Proc. IEEE VTC*, pp. 211-218, Sept. 2000.

[42] D. Niyato, E. Hossain, and P. Wang, "Optimal Channel Access Management with QoS Support for Cognitive Vehicular Networks," *IEEE Trans. Mob. Comput.*, vol. 10, no. 5, pp. 573-591, Apr. 2011.

[43] P. Pawelczak, S. Pollin, H.-S. So, A. Bahai, V. Prasad, and R. Hekmat, "Performance Analysis of Multichannel Medium Access Control Algorithms for Opportunistic Spectrum Access," *IEEE Trans. Veh. Technol.*, vol. 58, no. 6, pp. 3014-3031, Jul. 2008.

[44] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *Proc. ACM SIGCOMM*, pp. 234-244, Sep. 1994.

[45] C. E. Perkins, E. B. -Royer, and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," Jul. 2003, Website: https://www.rfc-editor.org/info/rfc3561.

[46] P. Ren, Y. Wang, and Q. Du, "CAD-MAC: A Channel-Aggregation Diversity Based MAC Protocol for Spectrum and Energy Efficient Cognitive Ad Hoc Networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 2, pp. 237-250, Feb. 2014.

[47] A. Sahai, R. Tandra, S. M. Mishra, and N. Hoven, "Fundamental Design Tradeoffs in Cognitive Radio Systems," in *Proc. ACM First Intern. Worksh. on Tech. and Poli. for Acces. Spect.,* Article 2, Aug. 2006.

[48] H. -S. W. So, J. Walrand, and J. Mo, "McMAC: A Parallel Rendezvous Multi-Channel MAC Protocol," in *Proc. IEEE WCNC*, pp. 334-339, Mar. 2007.

[49] 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*, pp. 222-233, May 2004.

[50] H. Su and X. Zhang, "Channel-Hopping based Single Transceiver MAC for Cognitive Radio Networks," in *Proc. IEEE CISS*, pp. 197-202, Mar. 2008.

[51] A. Sultana, X. Fernando, and L. Zhao, "An Overview of Medium Access Control Strategies for Opportunistic Spectrum Access in Cognitive Radio Networks," *Peer-to-Peer Netw. and Applic.*, vol. 10, no. 5, pp. 1113-1141, Sep. 2017.

[52] K. G. M. Thilina, E. Hossain, and D. I. Kim, "DCCC-MAC: A Dynamic Common-Control-Channel-Based MAC Protocol for Cellular Cognitive Radio Networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3597-3613, May 2016.

[53] M. Timmers, S. Pollin, A. Dejonghe, L. V. Perre, and F. Catthoor, "A Distributed Multichannel MAC Protocol for Multihop Cognitive Radio Networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 446-459, Jan. 2010.

[54] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "Channel-Hopping Multiple Access," in *Proc. IEEE ICC*, pp. 415-419, Jun. 2000.

[55] R. W. Wolff, *"Stochastic Modeling and the Theory of Queues,"* Prentice Hall, 1989.

[56] J. Wu and F. Dai, "Broadcasting in Ad Hoc Networks Based on Self-Pruning," *Int. J. Foundat. of Comp. Sci.*, vol. 14, no. 2, pp. 201-221, Apr. 2003.

[57] 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. I-SPAN*, pp. 232-237, Dec. 2000.

[58] B. Yang, B. Li, Z. Yan, M. Yang, and X. Zuo, "A Reliable Channel Reservation based Multi-Channel MAC Protocol with a Single Transceiver," in *Proc. QSHINE*, pp. 265-271, Aug. 2015.

[59] Q. Ye, W. Zhuang, L. Li, and P. Vigneron, "Traffic-Load-Adaptive Medium Access Control for Fully Connected Mobile Ad Hoc Networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9358-9371, Nov. 2016.

[60] Y. Zhang, L. Lazos, K. Chen, B. Hu, and S. Shivaramaiah, "Multi-Channel Medium Access without Control Channels: A Full Duplex MAC Design," *IEEE Trans. Mob. Comput.*, vol. 16, no. 4, pp. 1032-1046, Apr. 2017.

[61] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 589-600, Apr. 2007.

Sachin Kadam received his Bachelor of Engineering (BE) in Electronics and Communication (E&C) Engg. from People's Education Society Institute of Technology (PESIT), Bangalore in 2007 and M.Tech. in Electrical Engg. from Indian Institute of Technology (IIT) Kanpur in 2012. He is currently a PhD student in the Dept. of Electrical Engg., IIT Bombay. His research interests are M2M Networks, Mobile Ad hoc Networks and Spectrum Sensing. He was a recipient of scholarship Foundation for Excellence (FFE), California, USA during his BE.

Devika Prabhu graduated in Electrical Engineering from IIT Indore and pursued her post-graduation in Management Studies from Indian Institute of Management (IIM) Lucknow. She is currently working with PepsiCo India, handling Delhi operations for the Foods division.

Nitish Rathi received his B.Tech in Electrical Engineering from IIT Indore in 2013 and PGDM from IIM Kozhikode in 2017. He is currently working for Google as Campaign Manager in Hyderabad, India.

Prakash Chaki received his Master's degree in Electrical Engineering from IIT Bombay, Mumbai, India, in 2013. He is currently a Researcher with the Central Research Laboratories of NEC Corporation, Japan. His research interests include wireless communication, networks and channel coding, particularly in polar codes. He was a recipient of IEICE Young Researcher's Award 2016.

Gaurav S. Kasbekar received B.Tech. in Electrical Engg. from Indian Institute of Technology (IIT), Bombay in 2004, M.Tech. in Electronics Design and Technology (EDT) from Indian Institute of Science (IISc), Bangalore in 2006 and Ph.D from University of Pennsylvania, USA in 2011. He is currently an Assistant Professor with the Department of Electrical Engineering, IIT Bombay. His research interests are in the modeling, design and analysis of wireless networks. He received the CEDT Design Medal for being adjudged the best Masters student in EDT at IISc.