An Energy-Efficient Multichannel Packet Transmission Scheduling for Ad Hoc Networks

Thiago Fernandes Neves, Felipe de Moraes Modesto, and Jacir Luiz Bordim
Department of Computer Science
University of Brasilia, UnB
Brasilia, Brazil.
Email: {tfn.thiago, felipe, bordim}@cic.unb.br

***Abstract*—The popularization of wireless network technologies has driven the quest for efficient solutions in the use of the available resources. In particular, there is an increasing demand for solutions to reduce energy consumption and improve channel use. This work addresses the problems of multi-channel assignment and communication scheduling in wireless networks. Considering that channel allocation is a NP-complete problem, this paper presents a time and energy-efficient heuristic to tackle the multi-channel assignment problem. Once channel assignment is performed, an energy-efficient protocol allows the stations to complete their data transfers using minimum resources. The protocol divides its operation in management and transmission stages. The main contribution of this work is to present a multi-channel communication protocol that efficiently reduces communication time by exploring multiple channels even for control messages. Empirical results show that the management stage takes, in average, less than 9% from the protocol total time while the transmission stage, in average, takes only 5% more time than the optimum time.**

Keywords–energy efficient protocols, multi-channel assignment, scheduling, wireless networks.

I. INTRODUCTION

The quest for uninterrupted wireless connectivity has been highly influenced by the popularization of mobile devices and social networks. This trend in mobile applications has motivated the proposal of Medium Access Control (MAC) protocols capable of coping with a varying number of application demands and devices characteristics. Despite of these advances, one of the major concern regarding the design of such protocols is the need to reduce energy consuption [1]. As wireless devices usually operate on battery power, and recharging them may not be an option while on the move, means to preserve and extend nodal lifespan is of interest.

Among existing energy-saving strategies, *topology control* and *duty-cycle* have been widely employed in the context of wireless networks [2]. Topology control techniques typically allow wireless devices to adjust their transmission power in order to conserve energy without affecting network connectivity [3]. Duty-cycle schemes, on the other hand, allow wireless devices to alternate between inactive and active mode. When in active mode, devices are able to send or receive data; while in doze mode devices remain in energy conservation mode and are not able to send or receive data. This last strategy is particularly challenging as devices in doze mode are not able to receive data packets. There are research opportunities regarding the development of techniques that ensure communicating devices are only active when they have data to send or receive [4]. In [5], the authors show that energy consumption can be reduced by increasing the time needed to complete a given task and vice-versa. The authors have shown that these parameters are usually conflicting and finding a compromise between them is not trivial.

Regardless the fact that most wireless devices are capable to tune to different frequencies to send and/or receive data packets, existing MAC protocols are usually designed to operate on a single-channel, where all the nodes are confined [6]. Channel assignment in wireless networks is usually performed during the deployment phase. The reason behind this is that channel assignment is a complex and time consuming task that may not produce the desirable results when naive approaches are employed. Indeed, the Channel Assignment Problem (CAP) satisfies the interference constraints by maximizing throughput. In its general form, the CAP problem is equivalent to the Generalized Graph-coloring Problem (GCP), which has been proved to be an NP-complete problem [7]. This work explores duty-cycle techniques and propose a multi-channel assignment heuristic that enables the transmission scheduling of data items to be carried out in an energy-efficient manner.

The remainder of this paper is organized as follows. Section II describes related works. Section III describes the communication model considered in this work. Section IV presents the channel assignment problem along with an energy-efficient heuristic to tackle it. The EEMC-MAC protocol details is described in Section V. The simulation environment and results are presented in Section VI. The building blocks to adapt the proposed EEMC-MAC to multi-hop environment is presented is Section VII. Finally, Section VIII concludes the work and presents future direction.

II. STATE OF THE ART

To reduce packet collision, protocols such as the IEEE 802.11 are available and can reduce interference in ongoing communications. The control mechanism applied in IEEE 802.11 is the well-known CSMA/CA protocol [8]. By listening before transmitting data, nodes can determine whether a channel is busy or available. However, this mechanism does not avoid the overexploitation of spectrum resources. Indeed, scenarios with excessive competition may drastically reduce network throughput. The CSMA/CA protocol relies on random *backoff* and cannot prevent communications from starting simultaneously. Channel assignment in wireless network environments is typically static. This means that, while channel selection can be based on spectrum conditions during network initialization, channel degradation does not cause the data channel to be changed. Thus, MAC protocols that are tailored for single-channel settings have difficulties copping with heavy network loads and fail to provide means for networks to switch channels depending on spectrum occupancy.

Access to multiple communication channels is an alternative to increase throughput in wireless networks [9]. For example, by employing opportunistic spectrum access techniques, users can temporary access unused licensed frequencies [10]. With access to multiple channels, Frequency Division Multiple Access (FDMA) based techniques allows the selection of several communication channels with non-overlapping and non-interfering frequencies. Therefore, multiple pairs of nodes can communicate at the same time without interference given they have been allocated to different channels. Indeed, a number of works consider the use of multiple channels in wireless networks [11], [12], [13], [14], [15], [16]. Hamdaoui *et al.* [11] proposed a protocol where the channels are assigned to groups based on "transmission intentions". In this scheme, each group elects a leader to study channel conditions and select the best channel for itself. All data channels operate independently and intergroup coordination is performed by channel leaders using a dedicated control channel. Alternatively, Hsu *et al.* [12] propose a contention model based on channel aggregation. The protocol considers that multiple data channels can be used simultaneously for data transfers. Transmission pairs select the channels used for transmission based on average occupancy and *backoff* necessary to access these channels. These protocols focus in increasing network throughput and do not consider the energy costs involved with their communication cycles. An example of an early energy efficient protocol is defined in [5], where a randomized time- and energy-optimal routing protocol is proposed. To achieve this goal, users learn their roles in packet routing and wait for their turn by deferring spectrum access to either receive or send data packets, thus reducing energy costs. The protocol, however, requires that users know information about the network during initialization and is applied to a single-channel network context.

The use of multi-channel MAC protocols with duty-cycle schemes to increase network throughput and decrease energy consumption is proposed in [14], [17]. These works focused on multi-channel energy-efficient protocol tailored for wireless sensor networks. Incel *et al.* [14] proposed a scheme that works in a distributed fashion and schedules communications based on Time Division Multiple Access (TDMA) algorithms. This approach has been shown to reduce packet collision by informing the nodes what periods of time they need to be active. The proposed scheme, however, focuses on maximizing throughout while energy consumption is a secondary goal. Tang *et al.* [17] proposed a protocol that allows transmitting nodes to estimate the receiving nodes' activation time without the use of a control channel. Zhang *et al.* [15] proposed a multi-channel MAC protocol for ad hoc networks. The proposed scheme works by dividing its operation in management and transmission stages. At the beginning of the management stage, all the nodes wishing to communicate turn to the control channel. The management stage dynamically adjusts its duration based on the traffic and it is used to allow the nodes to reserve data channels using a dedicated, common, control channel. During the transmission window, nodes communicate using several channels, while non-communicating nodes stay in doze mode.

In previous work, we proposed an energy efficient protocol for multi-channel allocation and transmission scheduling in wireless networks, termed ECOA-BP [18]. As in [15], the ECOA-BP protocol divides its operation in management and transmission stages and uses a control channel during the management stage. The technique proposed uses efficient transmission assignment and duty-cycle strategy to alternate the nodes between active and inactive modes, thus reducing the power drainage rate. Previous works show that is possible to reduce energy consumption at the cost of higher communication time [5]. Both Zhang *et al.* [15] and Neves *et al.* [18] focus on balancing these parameters. Additionally, both works consider that network coordination is performed in a single control channel. Independently of the number of available channels, the use of a single control channel to manage channel access can be a bottleneck, as it increases the communication time [19]. Concerned with coordination costs, Cordeiro *et al.* [20] propose that the management stage, known as Beacon Period (BP), takes place in the data channels. The authors suggest that channel access is structured into recurring super-frames synchronized globally so that users can migrate between channels to communicate with different nodes. While the model proposed achieved promising results, it does not consider the energy costs required to implement and maintain global coordination.

A. Our Contribution

The aforementioned works focused on exploiting the availability of multiple channels to improve communication time. However, they neglect to analyze energy conservation and the overhead introduced by the proposed coordination schemes. This paper addresses the problems of multi-channel allocation, transmission scheduling and energy consumption in wireless networks. As in related works, it assumes that the devices work on batteries and have a single transceiver, capable of tuning to one of the several available channels and to switch between active (regular energy consumption) and inactive (reduced energy consumption) operation modes. As customary, time is assumed to be slotted with slot durations long enough to ensure that a single data packet can be transmitted or received by any user in the network within a single slot [5], [14]. In this context, this paper proposes a time and energy-efficient protocol capable of performing multi-channel allocation and transmission scheduling in a wireless setting. This paper is an extended version of work published in [1]. The proposed scheme, termed *Energy-Efficient Multi-Channel MAC protocol* (EEMC-MAC) divides its operation into management and transmission stages. Unlike most similar proposals, the proposed protocol uses all the available channels in both management and transmission stages. Experimental results show that the management stage, in average, takes less than 5% from the total protocol execution time, while the transmission stage is optimum in terms of energy consumption.

III. Communication Model

Consider an Ad Hoc network consisting of a group of n nodes each with a single transceiver and unique identifier

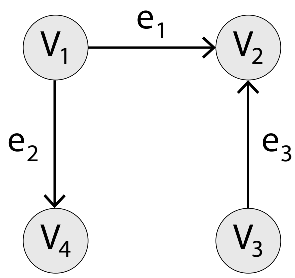


Figure 1: Communication graph example with 4 nodes.

(UID), connected as a single-hop network represented by the complete graph $\mathcal{G}_n$. The communication scenario of this network is represented by a directional graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of nodes (vertices) and E is a set of communications (edges), $E \subseteq V^2$. Consider $E = \{e_1, e_2, ..., e_p\}$, where $e_h = \{(v_s, v_d) | \{v_s, v_d\} \subseteq V, s \neq d\}$, $1 \leq h \leq p$, as a set of edges representing the communication graph of the network $\mathcal{G}_n$. Each edge $e_h = (v_s, v_d) \in E$ represents a communication between a source node v_s and a destination node v_d. Each node is assumed to have at most one packet per destination in the communication graph. Let s_i be the transmission set of a node v_i ($v_i \in V$). That is, s_i represents all the nodes that v_i has data packets to send to. Furthermore, let d_i be the reception set of a node v_i. Hence, d_i represents all the nodes that have data packets to send to v_i. Thus, for a given communication graph $\mathcal{G}_n$, each node v_i has $\tau_i = |s_i| + |d_i|$ data packets to send and receive. In other words, τ represents the amount of time a node needs to be awake to (i) transmit the data packets to its neighbours; and (ii) receive the items destined to it.

As an example, Figure 1 represents a possible communication graph for a network topology $\mathcal{G}_n$. In this figure, $V = \{v_1, v_2, v_3, v_4\}$ and $E = \{e_1, e_2, e_3\}$, where $e_1 = (v_1, v_2)$, $e_2 = (v_1, v_4)$ and $e_3 = (v_3, v_2)$. In this communication graph, node v_1 has data to send to nodes v_2 and v_4 and no data to receive, thus, $s_1 = \{v_2, v_4\}$ and $d_1 = \emptyset$. Similarly, $s_2 = \emptyset$, $d_2 = \{v_1, v_3\}$, $s_3 = \{v_2\}$, $d_3 = \emptyset$, $s_4 = \emptyset$ and $d_4 = \{v_1\}$.

As presented in [15], this paper assumes that data transmission/reception occur in time slots, with each transmission/reception taking exactly one time slot. In each time slot t_j, $j \geq 0$, where t_j is equal to the time interval $[t_j, t_{j+1})$, a node can be in active or inactive operation mode. When active, a node can send or receive data. In case a node is not transmitting or receiving data, the node goes into idle mode so as to save power. That is, energy consumption is associated with the amount of time that the node remains in active mode. Consider $C = \{c_1, c_2, ..., c_k\}$ as the set of available channels for communication. When a channel $c_i, 1 \leq i \leq k$, is used by a pair of nodes in the time slot t_j, it will be unavailable for other nodes in this time slot. In the case that two or more transmitting nodes use the channel c_i during time slot t_j, a collision occurs and the data packets are lost. Hence, the challenge is to find a scheduling that: (i) allows the transmitting nodes to send and receive data without collision; and (ii) minimizes the communication time. Table I summarizes the notations used throughout this work.

IV. The Channel Assignment Problem (CAP)

In a network environment where many data channels are available, the task of channel assignment that satisfies interference constraints and maximizes throughput is known as the Channel Assignment Problem (CAP). To prevent interference between communications, a same channel cannot be allocated for two pairs of neighbouring nodes simultaneously. In its general form, the CAP problem is equivalent to the Generalized Graph-coloring Problem (GCP), which is known as a NP-complete problem [7]. Given the communication graph G and k channels in the presented communication model, the CAP consists in performing the communications using the minimum amount of time and communication channels. Note that if $k = 1$ this problem is simplified, once all the communications must be serialized. However, in the general case scenario, optimum solutions are complex to obtain.

Because the CAP is NP-complete, many researchers have proposed heuristics and approximation algorithms with lower computational costs. These solutions, however, can not guarantee optimum results. The proposed alternatives vary from neural networks, to genetic and graph theory based heuristics [7]. Next, an heuristic based on graph theory to solve the CAP problem is presented.

A. ECOH: An Edge Coloring Heuristic

The proposed heuristic, termed *Edge COloring Heuristic* (ECOH), takes as input a communication graph $G = (V, E)$ and a number k of available channels and produces as output a list of "communication sets", called CS. The list of communication sets is defined by $CS = \{CS_1, CS_2, ..., CS_r\}$, with $CS_i \subseteq E$ and the elements in CS_i are disjoint, $1 \leq i \leq r \leq |E|$. The details of the ECOH is presented in Figure 2. The basic idea behind the proposed heuristic is the distribution of edges belonging to E into r communication sets, so that the edges contained in a set CS_i have no dependencies with each other. In this context, dependencies occur between two or more communication sets that involve a same node v_i. The selection criterion is the choice of an edge belonging to a greater degree vertex in E. This edge will be part of the initial transmission set CS_i and it will be a comparison base for the other edges belonging to E. Only the edges without dependences with other elements in CS_i will be removed from E and incorporated into this set. An edge is considered not dependent on a set of edges when it does not share any vertex with the edges on this set. The procedure is repeated until the r transmission sets are formed and the set E is empty.

To better understand the operations of the ECOH, consider as input the communication graph represented in Figure 1 and the number of available channels to be equal to 2 ($k = 2$). Thus, $E = \{e_1, e_2, e_3\}$, where $e_1 = (v_1, v_2)$, $e_2 = (v_1, v_4)$ and $e_3 = (v_3, v_2)$. Suppose that the edge e_2 is inserted into the first set of edges in CS_1, line 5 (Figure 2). Going through all edges of E, line 6, the algorithm checks that the edge e_3 has no dependence on the set CS_1 and decides to insert it, line 8. As there are no more edges in E without dependencies with the elements of the set CS_1, the algorithm terminates the loop. A new loop is then started, line 2, and the variable r is incremented to 2. In the new loop, the algorithm inserts the edge e_1 in the set CS_2, ending the algorithm, since

TABLE I: TABLE OF NOTATIONS.

Symbol	Definition
$\mathcal{G}_n$	Graph modelling the wireless network topology;
$G = (V, E)$	Graph representing the communication scenario;
$V = \{v_1, v_2, ..., v_n\}$	Set of nodes (or devices);
$E = \{e_1, e_2, ..., e_p\}$	Set od edges (or transmissions);
n	Number of nodes ($n = \|V\|$);
p	Number of edges, ($p = \|E\|$);
$\Delta(G)$	Maximum graph degree;
s_i	Number of items v_i has to send (i.e., v_i's transmission set);
d_i	Number of items destined to v_i (i.e., v_i's reception set);
τ_i	Total number of items v_i sends and receives ($\tau_i = \|s_i\| + \|d_i\|$);
$C = \{c_1, c_2, ..., c_k\}$	Set of channels;
k	Number of available channels ($k = \|C\|$);
g_i	Subset of V ($g_i \subseteq V$);
l	Number of nodes in $\|g_i\|$;
T_m	EEMC-MAC management stage time (in time slots);
T_t	EEMC-MAC transmission stage time (in time slots);
T	Management and transmission stage time ($T = T_m + T_t$);
$T_{t'}$	EEMC-MAC transmission state optimum time (in time slots);

Algorithm ECOH(G,k)

1: $G = (V, E)$, $r \leftarrow 0$;
2: **while** ($E \neq \emptyset$) **do**
3: $r \leftarrow r + 1$;
4: Select an edge e of the vertex with higher degree in E;
5: $CS_r \leftarrow e$, $E \leftarrow E - e$;
6: **for** (each $e_h \in E$) **do**
7: **if** (no vertex in $e_h \in CS_r$) **and** ($|CS_r| \leq k$) **then**
8: $CS_r \leftarrow CS_r \bigcup e_h$;
9: $E \leftarrow E - e_h$;
10: **end if**
11: **end for**
12: **end while**
13: $CS \leftarrow \{CS_1, CS_2, ..., CS_r\}$;

Figure 2: The details of the proposed Edge COloring Heuristic (ECOH).

the condition $E = \emptyset$ is reached, line 2. In this example, the algorithm output would be $CS = \{CS_1, CS_2\}$, where $CS_1 = \{e_2, e_3\}$ and $CS_2 = \{e_1\}$. Note that, according to the algorithm, $|CS_i| \leq k$. That means each communication set has at most $k = 2$ disjoint elements. This construction allows the nodes in each communication set CS_i to communicate concurrently using the k channels in the same time slot.

From the above, it is clear that the ECOH processes all edges of the graph (while-loop), selecting, at each iteration, a set of at most k independent edges (i.e., no two edges share a common vertex). As the set operations can be performed in $O(1)$ time, the ECOH takes $O(p \cdot k) \leq O(p^2)$ time to complete its execution. For latter reference, we state the above results in the following Lemma:

Lemma 1: The task of computing r disjoint communication sets CS_i, $(1 \leq i \leq r)$, where each CS_i comprises of at most k independent edges, can be computed in $O(p^2)$ time, where $p = |E|$.

V. Proposed Protocol

This section presents the details of the proposed protocol, named *Energy Efficient Multi-Channel MAC Protocol* (EEMC-MAC Protocol). This protocol aims to perform multi-channel allocation and scheduling to enable data communication. The protocol performs these tasks in order to minimize both energy consumption and the time required to transmit data. Thus, the goodness of the protocol, in terms of energy consumption, is assessed by evaluating the number of transmitting time slots a node is awake during the protocol execution. This evaluation does not consider the processing time of a given task, which is assumed to be lower than the cost of tramissing or receving a data packet [5]. First, the overall routines performed by the protocol are presented. Then, the protocol is described in detail, followed the analysis of its complexity.

A. Transmission Set Grouping Routines

Recall that each node $v_j \in V$ contains a set s_j identifying the destination nodes that v_j has data to send. In this subsection, the objective is to combine such sets for a given node. The $CombineGroup$ routine, presented in Figure 3, aims to achieve this goal using a single communication channel. The routine takes as input a set of nodes g_i, $g_i \subseteq V$, and a communication channel c_i. In the first step of the algorithm, each node in g_i computes a consecutive local ID in the range $[1, .., l]$, in line 2. That is, $|g_i| = l$. This task can be accomplish by employing fast, energy-efficient, leader election algorithms such as those presented in [21], [22]. Clearly, after this step, node v_l representes the node with the highest ID in g_i. The loop in lines $3 - 8$ combines the transmission sets s_j, $1 \leq j \leq l$ so that node v_l knows $s_l \cup s_{l-1} \cup ... \cup s_1$ at the end of the algorithm. Note that the routine above is very efficient in terms of energy consumption given that each node stays in active mode for 2 time slots: one to receive the transmission set and another to sent the combined transmission set. For latter reference, we state the following result:

Lemma 2: The task of combining l transmissions sets $s_l \cup s_{l-1} \cup ... \cup s_1$ can be performed on a single channel in $l - 1$ time slots with each node v_j, $(1 \leq j \leq l)$, awake for at most 2 time slots.

Consider a set of channels $C = \{c_1, c_2, ..., c_k\}$ where $|C| = k$, $k > 1$ are available. In this example, the $CombineGroup$ routine can be improved to take advantage of several channels. The routine $CombineTS$, as depicted

Algorithm CombineGroup(g_i,c_i)

Let $|g_i| = l$;
Each node computes its local ID within the range $[1, ..., l]$ such that $g_i = \{v_1, v_2, ..., v_l\}$;
for $j \leftarrow 1$ **to** $l - 1$ **do**
  Nodes v_j and v_{j+1} enter in active mode;
  v_j sends its transmission set s_j to v_{j+1} using channel c_i;
  Node v_{j+1} attaches s_j to s_{j+1};
  Node v_j enters in inactive mode;
end for

Figure 3: Algorithm that combines the transmission sets in a group.

in Figure 4, shows how transmission sets can be combined, using multiple channels simultaneously. Similarly to the $CombineGroup$ routine, $CombineTS$ takes two parameters as input: a group of nodes g_l, $g_l \subseteq V$, and a set of channels C, where $|g_l| = l$ and $|C| = k$. The routine is only executed if $k \geq \lfloor \frac{l}{2} \rfloor$, this way, all the transmissions in g_l can be parallelized in the k channels. At the beginning of the algorithm, all the active nodes compute their local ID in the range $[1, ..., l]$, line 4. The procedure grows a binary tree, combining the leaf nodes and working its way to the root using the k available channels, lines 5-13. At the end of the algorithm, the local node v_1 will have all the transmission sets $s_l \cup s_{l-1} \cup ... \cup s_1$.

The $CombineTS$ routing algorithm consists of two nested loops (line 5 and 6). The inner loop is executed in parallel for all available channels, taking a single time slot for each $l/2$ channels while the outer loop is executed for $\log l{+}1$ iterations. As $k \geq \lfloor \frac{l}{2} \rfloor$, the $CombineTS$ takes at most $\log k{+}1$ time slots to combine the transmitting sets of a group of l nodes. The above discussion is summarized into the following Lemma:

Lemma 3: The task of combining l transmissions sets $s_l \cup s_{l-1} \cup ... \cup s_1$ on a k-channel setting can be accomplished in $\log k + 1$ time slots with each node v_i, $(1 \leq i \leq l)$, being awake for at most $\log k + 1$ time slots.

B. EEMC-MAC Details

This subsection presents the details of the EEMC-MAC protocol, which aims to explore the availability of multiple channels to allow nodes to send and receive data packets using as few time slots as possible. As it will be shown latter, the EEMC-MAC performs this task in an energy-efficient manner. The proposed protocol consists of a management and a transmission stages. The first stage builds the communication graph using the $CombineGroup$ and $CombineTS$ routines. Then, the communication graph is used to compute the communication sets during the transmission stage with the help of the ECOH heuristic. With this information at hand, each node learns when it must be awake to transmit and to receive its share of items. The details of the management and transmission states are presented next.

1) EEMC-MAC: Management Stage: The management stage main idea is to ensure that a leader node gets all the s_i transmission sets from all the nodes $v_i \in V$. This process must occur in a energy efficient way and use the maximum number of available channels. Then, the leader node can join all the communication sets and create the communication graph $G = (V, E)$. Figure 5 shows the management stage steps. At the beginning of the algorithm all the nodes are in inactive mode. If $k < \frac{n}{2}$, the n nodes in the set $V = \{v_1, v_2, ..., v_n\}$ are divided in k groups of nodes g_1, g_2, ..., g_k, lines 2-3. Once each node knows the values of k, n and its local ID, it has the condition to identify the group it belongs to. The goal is to reduce the number of active stations down to k. In the next step, k calls of the routine $CombineGroup$ are performed, line 5. As described above, the routine $CombineGroup$ will combine the transmission sets in each group g_i to just one node per group and the other nodes involved are set to inactive mode. The routine $CombineTS$ is called for all the active nodes. This routine will guarantee that all the transmission sets

Algorithm CombineTS(g_i,C)

1: Let $|g_i| = l$ e $|C| = k$;
2: **if** ($k \geq \lfloor \frac{l}{2} \rfloor$) **then**
3:   Let $C = \{c_1, c_2, ..., c_k\}$;
4:   Each node computes its local ID within the range $[1, ..., l]$ such that $g_i = \{v_1, v_2, ..., v_l\}$;
5:   **while** ($l > 1$) **do**
6:     **for** ($i \leftarrow 0$ **to** $(\frac{l}{2} - 1)$) in parallel **do**
7:       Assign channel c_{i+1} to pair (v_{i+1},v_{l-i});
8:       v_{l-i} sends its transmission set s_{l-i} to v_{i+1};
9:       v_{i+1} makes $s_{i+1} = s_{i+1} \bigcup s_{l-i}$;
10:       v_{l-i} goes into inactive mode;
11:     **end for**
12:     $l \leftarrow l/2$;
13:   **end while**
14: **end if**

Figure 4: Algorithm that combines the transmission on all groups.

Algorithm ManagementStage(n, k)

1: All the nodes in $V = \{v_1, v_2, ..., v_n\}$ start in inactive mode;
2: **if** ($k < \lfloor \frac{n}{2} \rfloor$) **then**
3:   Divide the nodes in V into k groups: $g_1, g_2, ..., g_k$;
4:   **for** $i \leftarrow 1$ **to** k in parallel **do**
5:     Execute *CombineGroup*(g_i,c_i);
6:   **end for**
7: **end if**
8: Let g_l denote de set of active stations;
9: The active stations execute *CombineTS*(g_l,C);
10: Let v_m be the last active station from the previous step;
11: Node v_m uses the transmission sets information to build the communication graph G;

Figure 5: Building the communication graph from the obtained transmission sets.

Algorithm TransmissionStage

1: Let v_m be the network node leader (from the previous stage) with the communication graph G;
2: Node v_m executes $ECOH(G, k)$ and gets the communication sets $CS = \{CS_1, CS_2, ..., CS_r\}$;
3: All the nodes in V enter in active mode and tunes into channel c_1. Node v_m broadcasts CS in channel c_1. All the nodes in V receives the CS broadcast and enters in inactive mode;
4: **for** $i \leftarrow 1$ **to** r **do**
5:   **for** $j \leftarrow 1$ **to** $|CS_i|$ in parallel **do**
6:     Select an unused edge $e_h = \{v_s, v_d\}$ from CS_i;
7:     Nodes v_s and v_d enter in active mode;
8:     Node v_s sends a packet to v_d using channel c_j;
9:     Nodes v_s and v_d enter in inactive mode;
10:     Mark the edge e_h from CS_i as used;
11:   **end for**
12: **end for**

Figure 6: Each node proceeds to the assigned channel to transmit and receive data packets.

will be combined and forwarded to a single node $v_m \in V$, lines 9-10. Node v_m will hold all the network transmission sets. At the end, node v_m uses the transmission sets information to build the communication graph $G = (V, E)$ (line 11).

In the worst case scenario, when $k < \lfloor \frac{n}{2} \rfloor$ (line 2), the Algorithm $ManagementStage$ makes k parallel calls to the $CombineGroup$ routine. As each group has n/k stations, the routine $CombineGroup$ takes $\lceil \frac{n}{k} \rceil - 1$ time slots according to Lemma 2 with each node awake for at most 2 time slots. Next, the $CombineTS$ (line 9) is executed for all k active stations. According to Lemma 4, $\log k + 1$ time slots are required to combine the transmission sets of k nodes with each node being awake for at most $\log k + 1$ time slots. Thus, overall, the $ManagementStage$ takes $O(\lceil \frac{n}{k} \rceil + \log k)$ time slots to compute the communication sets of all nodes in G with no station begin awake for more than $O(\log k)$ time slots. For latter reference, we state the following result:

Lemma 4: The $ManagementStage$ routine takes at most $O(\lceil \frac{n}{k} \rceil + \log k)$ time slots to combine the transmission sets of n stations using k channels with no station being awake for more than $\log k + 3$ time slots.

Clearly, this stage works based on the capability of the nodes to compute locals IDs in a given range. As mentioned in Section V-A, this task can be accomplished by employing fast and energy-efficient leader election algorithms, such as those proposed in [21], [22], [23]. As the number of nodes increases, it may be necessary to restrict the number of transmission sets in a given round of the management phase. This would prevent the combined transmission set to grow without bound. Again, a leader election algorithm could be employed to select a reasonable number of nodes to a particular round, thus limiting the number of data transfers in a given management phase. Another alternative is to allow the nodes to only combine sets up to a certain threshold, after that, the nodes would only relay the data sets. These nodes would, however, be able to receive data, but not transmit date in this particular round. Nodes that participated in the management phase, without completing their transmissions, could be given higher priority in the subsequente rounds. The aforementioned strategies would suffice to guarantee that the combined transmission set would be transferred within a single slot of time. Hence, in this work, we assume that the transmission set is such that is can be transferred within a single slot time.

2) EEMC-MAC: Transmission Stage: The transmission stage of the EEMC-MAC protocol begins immediately after the management stage. At the beginning of this stage, the leader node v_m has already computed the communication graph G. Figure 6 presents the $TransmissionStage$ details. To solve the communication dependences, the leader node v_m executes the ECOH heuristic and generates the list of communication sets $CS = \{CS_1, CS_2, ..., CS_r\}$, lines 1-2. The ECOH ensures that $|CS_i| \leq k$, that is, each set has at most the number of available channels and all the elements in each set CS_i are disjoint. In the following step, all the nodes enter in active mode and tune to channel c_1 to receive the CS broadcast from the leader node v_m and then return into inactive mode (line 3). The first loop, line 4-12, iterates from 1 to r (the number of communication sets) while the second loop iterates from 1 to the number of elements in the communication set indicated by the previous loop (lines 5-11). The inner loop begins by selecting an unused edge from the set CS_i. The nodes in this set enter in active mode (line 7), tune to the indicated channel and perform the data transmission (line 8). After transmitting their packets, these nodes return to inactive mode (line 9). This process continues until all the nodes in each communication set exchange their data sets.

It should be clear that the duration of the transmission stage depends on dependencies of the communication sets computed by the $ECOH$ heuristic. We consider that the execution of the $ECOH$ and the broadcast of the CS (lines 2-3) can be completed in a single time slot. As each communication set CS_i has at most k elements, the $TransmissionStage$ allows the concurrent transmission of all the elements in a given communication set CS_i in a single iteration of the outer loop. As there are r communications sets, r iterations of the outer loop are required. Clearly, in the worst case, none of the p transmissions can be performed in parallel. In this case, p transmissions are required. On the other hand, when all transmissions can be performed in parallel, $TransmissionStage$ takes n/k time slots. Thus, the amount of time that the $TransmissionStage$ needs to complete all data transfers is between $\Omega(\lceil \frac{p}{k} \rceil)$ and $O(p)$. The following Lemma summarizes the discussion above:

Lemma 5: The task of transferring p data items in a k-channel setting, where each node v_i, $1 \leq i \leq n$, has τ_i items to send and receive, can be completed by the $TransmissionStage$ algorithm in $\Omega(\lceil \frac{p}{k} \rceil)$ and $O(p)$ in the best and worst case scenarios, respectively, with no station being awake for more than $\tau_i + 1$ time slots.

C. EEMC-MAC: Main Procedure and Complexities

The main procedure of the EEMC-MAC protocol consists in the sequential execution of the management and transmis-

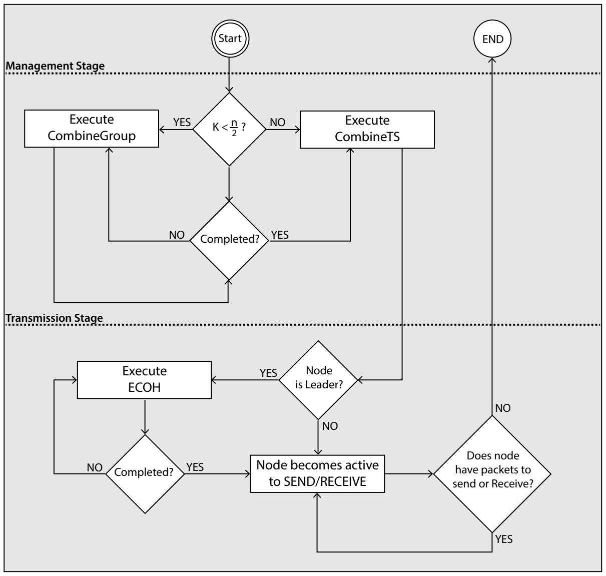
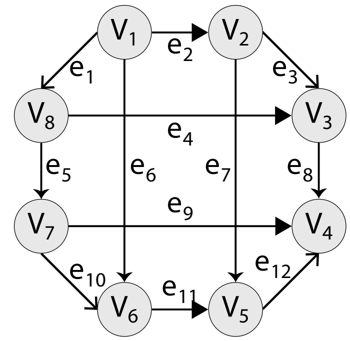
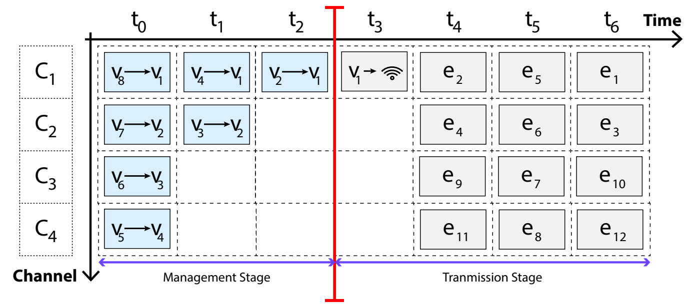


Figure 7: The EEMC-MAC protocol main tasks.



(a)



(b)

Figure 8: (a) Communication graph example with 8 nodes. (b) Channel representation for the EEMC-MAC protocol.

sion stages. The sequence of steps executed by the EEMC-MAC protocol is depicted in Figure 7. The protocol initiates in the management stage, where all communication sets are computed and combined with the aid of the $CombineGroup$ and $CombineTS$ routines. Once the communication sets are computed, the $TransmissionStage$ calls the $ECOH$ heuristic to compose the independent edges set. According to the arrangement of the independent edges set, the available channels are explored to reduce the overall communication time. Based on its ID, which is know to each node, they area able to determine the exact time to awake to send or receive their data items. Hence, each node stays in doze mode as long as possible so as to preserve battery power.

The time complexity and number of awake time slots for each station running the EEMC-MAC protocol can be obtained by combining the results in the Lemma 4 and Lemma 5.

Theorem 1: The tasks of channel assignment and transmission scheduling in a k-channel, single hop, wireless network represented by a communication graph $G = (V, E)$, can be solved by the EEMC-MAC protocol in $O(\lceil \frac{n}{k} \rceil + \log k + p)$ time slots with each node $v_i \in V$ being awake for at most $O(\log k + \tau_i)$ time slots, where $|V| = n$, $|E| = p$, $|C| = k$ and τ_i is the total number of items a node v_i has to send or receive.

D. EEMC-MAC: A working example

To exemplify the protocol application, consider the communication graph represented by Figure 8a. This graph has 8 vertices, $V = \{v_1, v_2, ..., v_8\}$, and 12 edges, $E = \{e_1, e_2, ..., e_{12}\}$. Consider the presence of $k = 4$ communication channels.

Figure 8b represents a possible data transmissions using 4 channels, the proposed communication graph and the EEMC-MAC protocol. The protocol main procedure begins with the execution of the management stage (shown in Figure 5). Once the number of channels is large enough ($k \geq \lfloor \frac{n}{2} \rfloor$), the routine $CombineTS$ is called. This routine will group all the transmission sets s_i of nodes in V, using the $k = 4$ channels, until the leader node v_1 gets all the communication sets, represented in time slots t_0 to t_2 in Figure 8b. This procedure of grouping transmission sets ends the management stage. The transmission stage (shown in Figure 6) starts immediately after the management stage ends. In this stage, the leader node v_1 uses the ECOH heuristic (Figure 2) to solve the graph communication dependencies and to obtain the list of communication sets CS. This list allows to perform the transmission scheduling, containing the channel and time slot each node must tune to send or receive data. Note that the ECOH heuristic ensures that parallel transmission does not share vertices in common. The leader node, then, broadcasts CS to all the other nodes in time slot t_3. Time slots t_4 to t_6 represent the scheduled packet transmissions.

VI. SIMULATION

The evaluation of the proposed protocol has been performed through simulation. To this end, the communication model presented in Section III was implemented in Matlab environment [24]. The simulator incorporates the characteristics of the EEMC-MAC protocol, described in Section V. To verify the goodness of the proposed solution, the simulation results are compared with the theoretical optimum solutions. This section begins by describing the simulation parameters and evaluation metrics followed by the simulation results and analysis.

A. Simulation Parameters and Evaluation Metrics

To analyze the EEMC-MAC, simulations have been conduced for a varying number of nodes, data packets per node and data channels. The number of nodes assume the following values: $n = 16, 32, 48, 64, 80$. Recall, from the communication model, that each node can have a maximum a degree of at most $n-1$ edges, that is, a node can send 0 or 1 packet to any destination in the communication graph per EEMC-MAC execution cycle. Thus, for a given set of nodes, the data items range from a few data items to send up to $n-1$ data items. The number of data packets per node assume values in one of the five different ranges: $R_1 = 10\%$ to 20%, $R_2 = 30\%$ to 40%, $R_3 = 50\%$ to 60%, $R_4 = 70\%$ to 80% and $R_5 = 90\%$ to 100%. Each range represents a percentage of the maximum number of transmissions per node. For example, in a setting with 16 nodes using the first range (R_1), each node would have from $10\% * (16-1) = 1.5$ (say 1) to $20\% * (16-1) = 3$ data packets to send. The number of channels assume the following values $k = 1, 2, ..., \lfloor \frac{n}{2} \rfloor$. The simulation results are drawn from the average of 200 simulation runs for each setting.

The following metrics are used to assess the goodness of the EEMC-MAC protocol:

- **Total execution time (M1)**: The amount of time (in the slots) that EEMC-MAC protocol needed to complete its operation for a given scenario;
- **Effective channel use (M2)**: The percentage of communication channel throughput that was used for effective data transmission (goodput) during the EEMC-MAC operation;
- **Protocol time in transmission stage (M3)**: The percentage of time that EEMC-MAC protocol spend at the transmission stage during its operation;
- **Ratio between EEMC-MAC transmission stage and and the optimum transmission stage time (M4)**;
- **Energy consumption estimation (M5)**.

Metric M1 aims to evaluate the reduction of the protocol operation time with the increase in the number of communication channels. As the number of available channels increases, more parallel transmissions can share the same time slot, reducing the overall completing time. Obviously, this limit dependes on the ECOH arrangement. Hence, M1 is an indicator of both, the impact of the communication channels and the goodness of the ECOH heuristic. Metric M2 evaluates the effective channel use, that is, the ratio between the number of packets that were transmitted by the number of packets that could have been transmitted. For example, consider that EEMC-MAC transmitted $|E|$ packets in T time slots using k channels. Once it is assumed that one packet is transmitted in one time slot, the maximum number of packets would be $T \cdot k$ and the effective channel use is the ratio $\frac{|E|}{T \cdot k}$. Metric M3 evaluates the percentage of time the EEMC-MAC remained in the transmission stage. This metric gives a direct indication of the EEMC-MAC overhead for transmitting data packets. For example, consider that the EEMC-MAC needed T_m time slots for the management stage and T_t time slots for the transmission stage. This metric calculates the percentage of time spent in the transmission stage, that is, the ratio $\frac{T_t}{T}$, where $T = T_m + T_t$. Metric M4 evaluates how far the EEMC-MAC schedule scheme is from the optimum one. It consists on the ratio between the EEMC-MAC transmission stage time and what would be the optimum time. For example, consider the EEMC-MAC needed T_t time slots for the transmission stage and the optimum time would be T'_t time slots. This metric calculates the ratio $\frac{T_t}{T'_t}$. Clearly, when $\frac{T_t}{T'_t} = 1$, the EEMC-MAC protocol achieved the minimum time to complete the transmission stage. Note that, in every case, $\frac{T_t}{T'_t} \geq 1$. Finally, Metric M5 shows energy cost of the proposed scheme considering current devices.

B. Simulation Results

The simulation results for metric M1 are presented in Figures 9a to 9d, where the number of nodes are fixed in 16, 32, 64 and 80 nodes, respectively. These figures present the results for all defined ranges (R_1, R_2, ..., R_5). The x-axis shows the variation in the number of channels while the y-axis presents the number of time slots required to complete the protocol execution. Two main characteristics can be observed in these graphics: (i) when the number os channels increases, the time required to complete the transmissions decrease, and; (ii) this reduction in time tends to stabilize. Table II, line 1, summarizes the average reduction in communication time for each range when compared with the serialized solution ($k = 1$). The average reduction for range R_1 is equal to 13.5922 times, for range R_2 is equal to 16.6856 times. When all the simulations are considered, we have that EEMC-MAC is able to reduce the average communication time in over than 17 times.

Figures 10a and 10d present the simulation results for metric M2 using the same parameters as in M1. As before, the x-axis shows the variation in the number of channels while in the y-axis presents the values for effective channel use. It can be observed that the effective channel use is higher when fewer channels are available. This occurs because with fewer channels the execution time takes longer and the time required for management tends to impact less in the total transmission time. As the number of channels increases, the total time tends to decrease, as can be seen in the results for metric M1, and the impact of the management increases. However, after a certain point, the management tends to become more efficient once lesser groups are created in the management stage. Table II, line 2, summarizes the average values for metric M2 for each range. The average ratio for range R_1 is equal to 68.4786%, for range R_2 is equal to 79.5270%. When the average of all the communication settings is computed, EEMC-MAC achieved a effective channel use of more than 80%.

Figures 11a and 11d present the simulation results for metric M3 using the same parameters as in the previous metrics. The x-axis shows the variation in the number of channels while in the y-axis presents the values for the percentage of protocol time in the transmission stage. Note that the percentage of protocol time in transmission stage tends to decrease once the protocol transmission stage time decrease with an increase in the number of channels. There is a small increase when the number of channels is close to the maximum once the management stage becomes more efficient. These results are in agreement with those in metric M2. Table II line 3 summarizes the average values for metric M2 for each range. The average

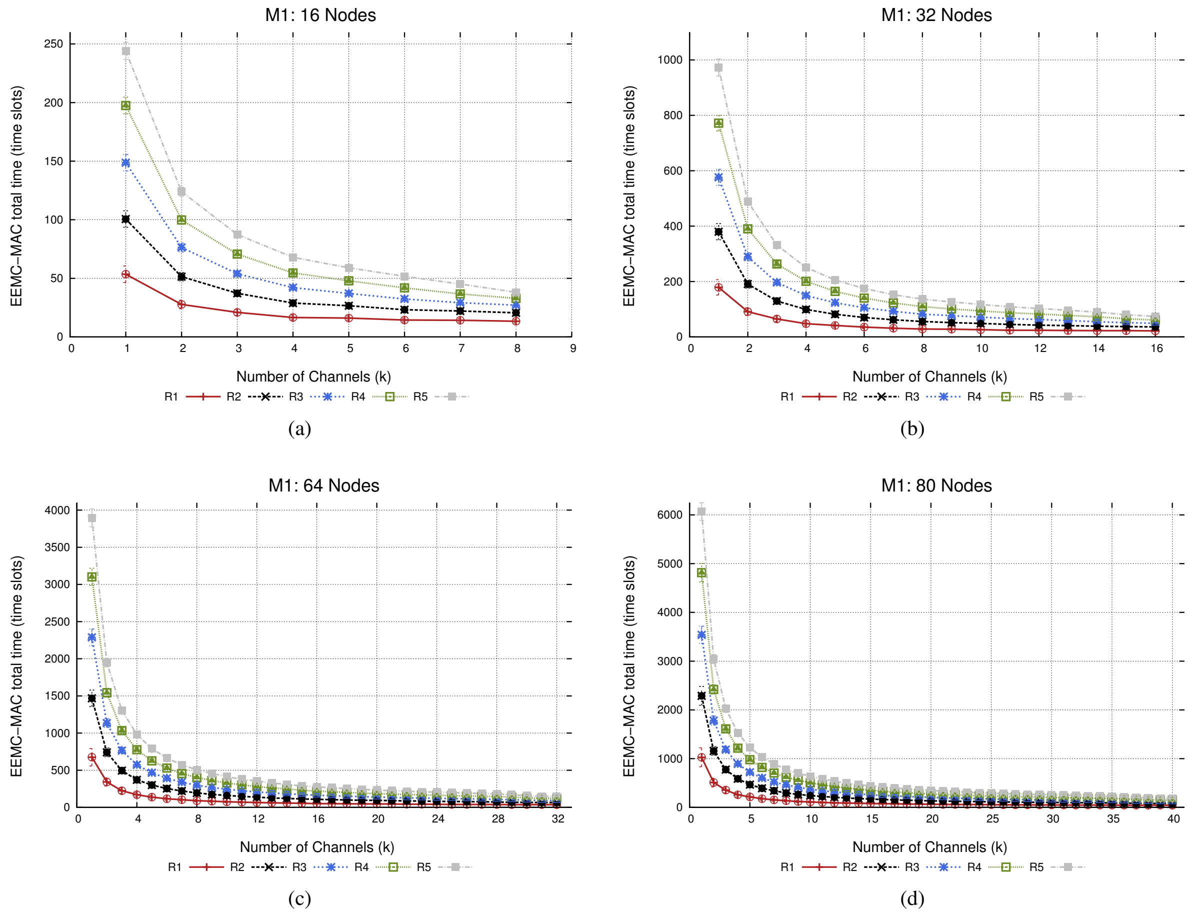


Figure 9: Simulation results for metric M1.

ratio for range R_1 is equal to 82.6813%, for range R_2 is equal to 91.1936% and so on. As can be observed, the percentage of time the protocol needs for management is minimal when compared with the total protocol execution time. In fact, this time is, on average, less than 9% from the total protocol execution time. In should be noted that for dense graphs, as in range R_5, the average time for management was less than 4% from the total protocol execution time.

Figure 12 presents the simulation results for the metric M4. From the *Vizing Theorem* [25], it is a valid lower bound to assume that the optimum channel assignment execution time, when there is no channel restriction, is equal to $\Delta(G)$, where $\Delta(G)$ is the graph maximum degree. Thus, for comparison purposes, it is assumed that $T_{t'} = \Delta(G)$ and that $k = \lfloor \frac{n}{2} \rfloor$. This channel restriction is necessary so that the EEMC can be compared to the optimum values. In the x-axis, in Figure 12, shows the number of nodes in the communication graph while the y-axis presents the values for the T_t/T'_t ratio. The number of data packets per node follows the previously defined ranges.

It can be observed in Figure 12 that $T_t/T'_t \approx 1$ when lower packet loads are presented ($R1$ and $R2$). The values obtained for T_t/T'_t increase with the number the of nodes and transmissions per communication graph. However, even in such cases, the EEMC-MAC transmission stage execution time was always less than 14% higher when compared with the optimum transmission stage time. Clearly, a larger communication graph increases the number of similar choices in the selection criterion of the protocol transmission scheduling. This, in turn, increases the chance of producing an unfavourable scheduling, thus increasing the communication time. Note that the choice of an inappropriate transmission scheduling at a given step W impacts in the choice of other transmissions at step $W+1$. Table II, line 4, summarizes the average values for metric M4 for each range. The average ratio for range R_1 is equal to 1.0184, for range R_2 is equal to 1.0211. From the results for metric M4 it is concluded that the EEMC-MAC achieved performance close to the optimum in many cases. When the average of all the communication settings are taken into consideration, the EEMC-MAC is less than 5% from the optimum time.

Table III shows the amount of power for different operation modes of three popular devices [26]. To assess the energy consumption of the proposed scheme, metric M5, the Cisco Aironet is considered. Recall that, according to Lemma (4) and (5), a node v_i is awake for at most $\log k + \tau_i + 4$ time slots during the Management and Transmission stages. In what follows, $k = 4$ and 1Mbps channels are assumed. Each node v_i is supposed to hold s_i data packets of 512 bytes that must be transferred to the corresponding destination.

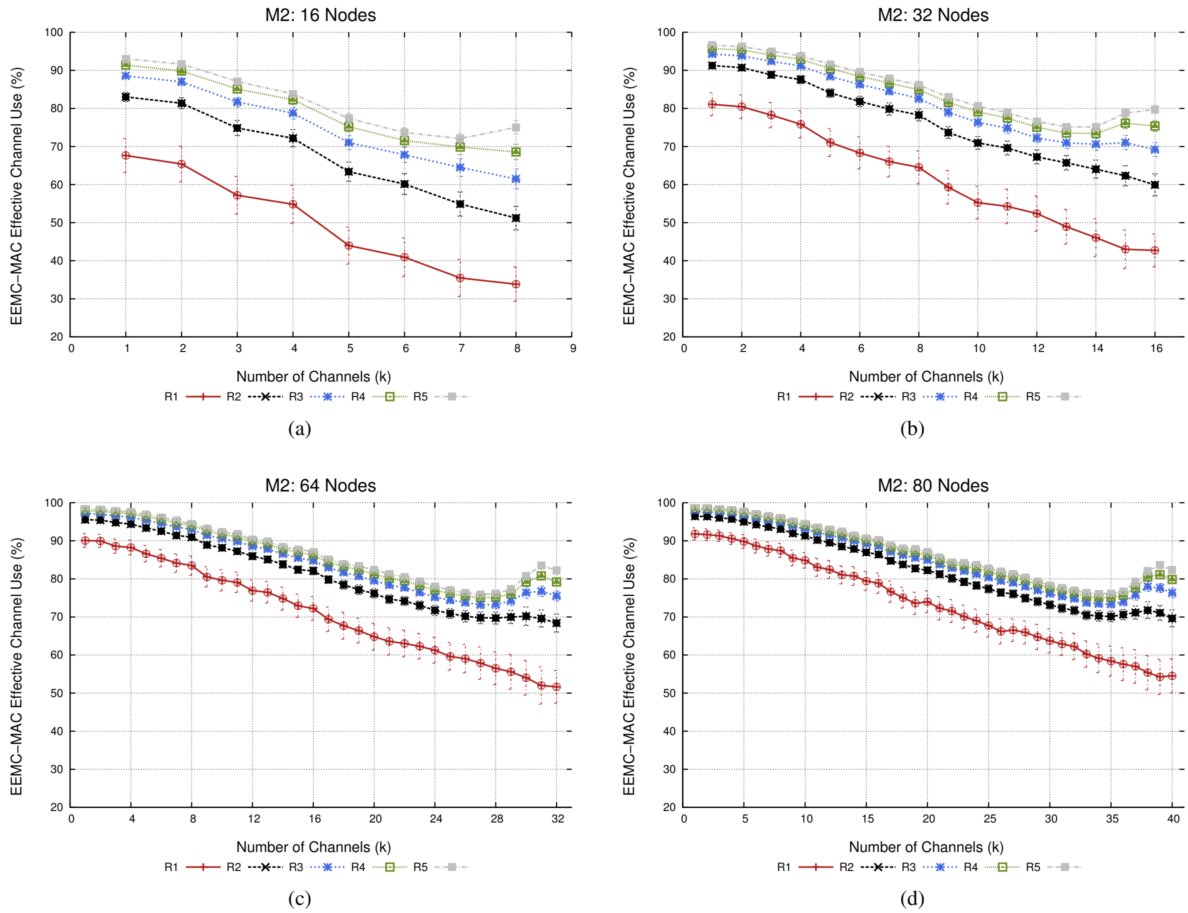


Figure 10: Simulation results for metric M2.

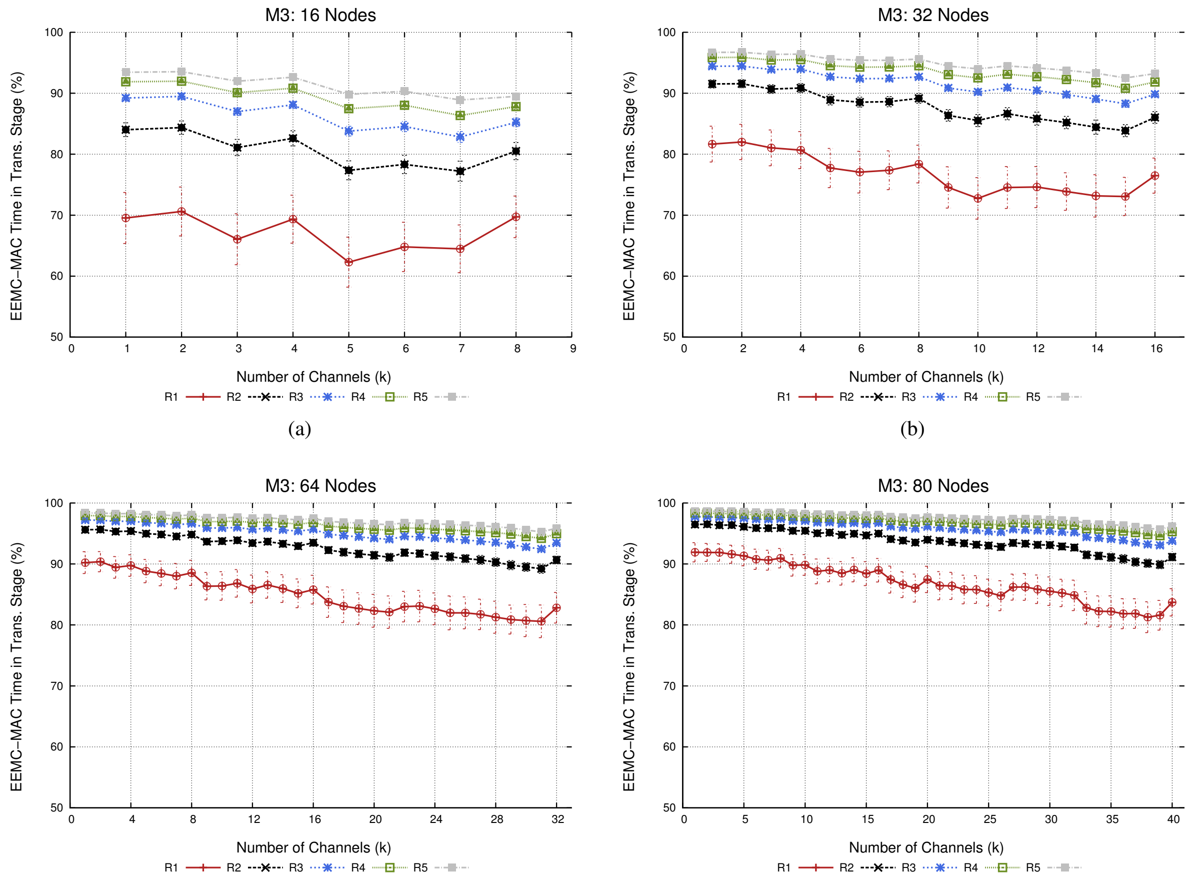
TABLE II: SUMMARY OF RESULTS.

Metric	R_1	R_2	R_3	R_4	R_5	Total
M1	13.5922	16.6856	18.3645	19.1940	19.9054	**17.5483**
M2	68.4786	79.5270	83.2576	85.1464	86.2663	**80.5352**
M3	82.6813	91.1936	94.0510	95.5074	96.3886	**91.9644**
M4	1.0184	1.0211	1.0348	1.0602	1.1102	**1.0489**

Similarly, d_i packets are expected to be received by node v_i. The amount of packets, τ_i, is computed based on the higher values of each range (R_1 to R_5). The energy consumption of a node v_i considers only the worse case scenario for both management and transmission stages. For comparison purpose, a single channel (SC), slotted time, protocol is considered. This latter protocol, hereafter referred to as SC, works in a similar fashion as the slotted Aloha protocol [27]. Note that, without a suitable scheduling algorithm, nodes must compete for channel resources. Thus, in order to make a fair comparison, the SC energy consumption is computed based on the amount of time a node v_i, in the worst case, expends to send and receive, respectively, s_i and d_i, data items. In other words, contention time to access the common channel and idle time is not considered. Table IV shows the energy consumption (E_j), in Joules, for both protocols. As can be seen in the table, with an increase in the number of packets each node has to exchange, the proposed scheme provides higher energy savings. Note that, in the SC protocol, each node has to constantly monitor the channel to verify whether a packet is destined to itself or not. Hence, in the worst case, a node must wait for all transmissions (that is, receive all transmitted packets) to correctly obtain its share of items. In the EEMC, on the other hand, each nodes awakes only to send and receive data.

TABLE III: POWER CONSUMPTION TO TRANSMIT AND RECEIVE FOR DIFERENTE DEVICES [26]

Device	Transmit	Receive	Idle	Sleep
Cisco Aironet	1.48W	1.0W	830mW	75mW
ORiNOCO 11b	1.43W	925mW	925mW	45mW
Mica mote	36mW	13.5mW	13.5mW	< 1 μA



(a) (b) (c) (d)

Figure 11: Simulation results for metric M3.

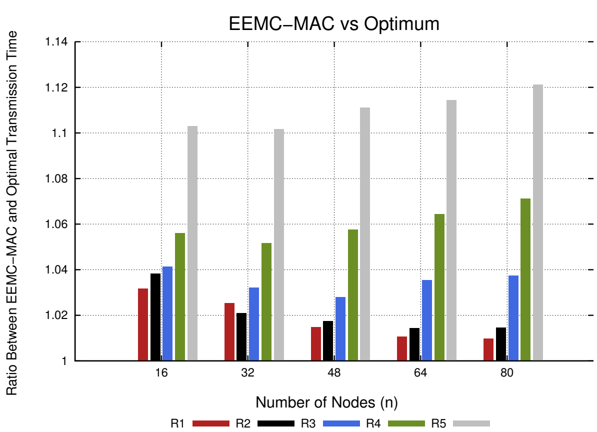


Figure 12: Simulation results for metric M4.

VII. EEMC-MAC for Multiple Hops

In this section, it is proposed a possible extension of the EEMC-MAC for multiple hops using a cluster scheme. Younis *et al.* [28] proposes HEED, an energy efficient clustering approach for distributed ad hoc networks. This approach fits with the deterministic nature of the EEMC-MAC as the clustering process is completed within a constant number of iterations (regardless the network diameter) and the control overhead is linear in the number of nodes. Every node uses just local information in the clustering process. For this purpose, at the beginning of the EEMC-MAC for multiple hops, all the networks nodes are organized into clusters, following the HEED scheme. Each cluster has the following features: synchronous time; single hop communication; a list of communication channels; and a Cluster Head (CH). The CH has the following roles: to act like the leader node of the EEMC-MAC protocol, being the responsible for grouping all the transmission sets, create and deliver the data scheduling to the other cluster nodes; to create and maintain an inter-cluster routing table; and

TABLE IV: ENERGY EXPENDITURE OF THE PROPOSED PROTOCOL FOR DIFFERENT NODE DENSITY AND τ VALUES.

EEMC					
	16	32	48	64	80
R1	0.0319	0.0666	0.1040	0.1443	0.1976
R2	0.0624	0.1275	0.2056	0.2764	0.3265
R3	0.0928	0.1986	0.2971	0.4085	0.4890
R4	0.1233	0.2596	0.3986	0.5405	0.6515
R5	0.1538	0.3205	0.4901	0.6624	0.8039
SC					
R1	0.2148	0.8228	1.8240	3.2185	5.3399
R2	0.4296	1.6456	3.8507	6.7052	10.6797
R3	0.6444	2.6055	5.6748	10.1918	16.0196
R4	0.8592	3.4284	7.7015	13.6785	21.3595
R5	1.0740	4.2512	9.5255	16.8970	26.3656

to perform the inter-cluster communication. Nodes in different clusters are not necessarily synchronized. There is a common channel for inter-cluster communication.

The EEMC-MAC can be executed within each cluster with just a few modifications in the original algorithm. If a node v_s in cluster A wants to send a packet to node v_d in cluster B, it sends the packet to the CH of cluster A, which stores the packet in its local buffer. Observe that, from the EEMC-MAC protocol, the CH knows exactly the time slots in both management and transmission stages it has no data to send or receive. This way, the CH uses these available time slots to perform the inter-cluster communication in the common channel. When the CH of cluster B receives the relayed packet from the CH of cluster A, it will add this packet to its transmission set, delivering the packet to v_d in the next EEMC-MAC cycle.

VIII. Conclusion

The increasing popularization of mobile devices and the emergence of high content applications, increased the need for high throughput and energy efficient protocols for wireless networks. In this context, this work proposes an energy efficient protocol, named EEMC-MAC, for multi-channel allocation and transmission scheduling in wireless networks. The EEMC-MAC protocol divides its operation in management and transmission stages. The energy expenditure in the management stage is reduced and empirical results shows that this stage represents less than 9% of the total protocol operation time. The transmission stage is optimum in energy consumption and, when compared with the optimum transmission stage time, the protocol needs, in average, 5% more time. It is also proposed a possible extension of the protocol for multiple hops. In future works, it is intended to address fault tolerance and to improve the communication model.

References

[1] T. F. Neves and J. L. Bordim, “EEMC-MAC: An energy efficient protocol for multi-channel wireless networks,” in The Eighth International Conference on Systems and Networks Communications (ICSNC2013), 2013, pp. 6–12.

[2] P. Mohapatra and S. V. Krishnamurthy, Ad Hoc Networks - Technologies and Protocols. Springer Sciente + Business Media, Inc. chapters 1, 3, 6, 2005.

[3] Y. Zhu, M. Huang, S. Chen, and Y. Wang, “Energy-efficient topology control in cooperative ad hoc networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 8, 2012, pp. 1480–1491.

[4] K. Chowdhury, N. Nandiraju, D. Cavalcanti, and D. Agrawal, “CMAC - A multi-channel energy efficient MAC for wireless sensor networks,” in Wireless Communications and Networking Conference, vol. 2. IEEE, 2006, pp. 1172–1177.

[5] J. L. Bordim, J. Cui, and K. Nakano, “Randomized time-and energy-optimal routing in single-hop, single-channel radio networks,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 86, no. 5, May 2003, pp. 1103–1112.

[6] S. Tsao and C. Huang, “A survey of energy efficient mac protocols for IEEE 802.11 wlan,” Computer Communications, vol. 34, no. 1, 2011, pp. 54–67.

[7] G. Audhya, K. Sinha, S. Ghosh, and B. Sinha, “A survey on the channel assignment problem in wireless networks,” Wireless Communications and Mobile Computing, vol. 11, no. 5, 2011, pp. 583–609.

[8] I. C. Society, “802.11-2012 - IEEE standard for information technology-telecommunications 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, Tech. Rep., July 2011.

[9] Q. Zhao and B. M. Sadler, “A survey of dynamic spectrum access,” Signal Processing Magazine, vol. 24, no. 3, 2007, pp. 79–89.

[10] C. Stevenson, G. Chouinard, Z. Lei, W. Hu, S. Shellhammer, and W. Caldwell, “Ieee 802.22: The first cognitive radio wireless regional area network standard,” Communications Magazine, vol. 47, no. 1, January 2009, pp. 130–138.

[11] B. Hamdaoui and K. Shin, “OS-MAC: An efficient MAC protocol for spectrum-agile wireless networks,” IEEE Transactions on Mobile Computing, vol. 7, no. 8, August 2008, pp. 915–930.

[12] A. Hsu., D. Wei, and C. Kuo, “A cognitive MAC protocol using statistical channel allocation for wireless ad-hoc networks,” in Wireless Communications and Networking Conference, March 2007, pp. 105–110.

[13] J. Jia, Q. Zhang, and X. Shen, “HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management,” IEEE Journal on Selected Areas in Communications, vol. 26, no. 1, January 2008, pp. 106–117.

[14] O. Incel, L. Van Hoesel, P. Jansen, and P. Havinga, “MC-LMAC: A multi-channel mac protocol for wireless sensor networks,” Ad Hoc Networks, vol. 9, no. 1, 2011, pp. 73–94.

[15] J. Zhang, G. Zhou, C. Huang, S. Son, and J. Stankovic, “Tmmac: An energy efficient multi-channel mac protocol for ad hoc networks,” in IEEE International Conference on Communications, 2007, pp. 3554–3561.

[16] A. Raniwala, K. Gopalan, and T. Chiueh, “Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 8, no. 2, 2004, pp. 50–65.

[17] L. Tang, Y. Sun, O. Gurewitz, and D. Johnson, “Em-mac: a dynamic multichannel energy-efficient mac protocol for wireless sensor networks,” in Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, 2011, p. 23.

[18] T. F. Neves, M. F. Caetano, and J. L. Bordim, “An energy-optimum and communication-time efficient protocol for allocation, scheduling and routing in wireless networks,” in 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, 2012, pp. 848–854.

[19] M. F. Caetano, B. F. Lourenço, and J. L. Bordim, “On the performance of the IEEE 802.11 in a multi-channel environment,” in 22nd International Conference on Computer Communications and Networks, July 2013, pp. 1–7.

[20] C. Cordeiro and K. Challapali, “C-MAC: A cognitive MAC protocol for multi-channel wireless networks,” in IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, April 2007, pp. 147–157.

[21] C. Lavault, J.-F. Marckert, and V. Ravelomanana, “Quasi-optimal leader election algorithms in radio networks with log-logarithmic awake time-slots,” in Telecommunications, 2003. ICT 2003. 10th International Conference on, vol. 2, Feb 2003, pp. 1113–1119 vol.2.

[22] M. Kardas, M. Klonowski, and D. Pajak, “Energy-efficient leader election protocols for single-hop radio networks,” in Parallel Processing (ICPP), 2013 42nd International Conference on, Oct 2013, pp. 399–408.

[23] Energy-Efficient Initialization Protocols for Ad-Hoc Radio Networks, vol. Vol.E83-A, 2000.

[24] M. Grant and S. Boyd, “Cvx: Matlab software for disciplined convex programming,” http://cvxr.com/cvx, March 2014.

[25] V. G. Vizing, “On an estimate of the chromatic class of a p-graph,” Diskret. Analiz, vol. 3, no. 7, 1964, pp. 25–30.

[26] J. C. Zheng, Rong Hou and N. Li, Power management and power control in wireless networks, Y. Pan and X. Yang, Eds. Nova Science Publishers, 2004, vol. 2.

[27] A. S. Tanenbaum, Redes de Computadores. Campus, chapter 4, 2003.

[28] O. Younis and S. Fahmy, “Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” IEEE Transactions on Mobile Computing, vol. 3, no. 4, 2004, pp. 366–379.