# EEMC-MAC: An Energy Efficient Protocol for Multi-Channel Wireless Networks

Thiago Fernandes Neves
Department of Computer Science
University of Brasilia, UnB
Brasilia, Brazil
e-mail: tfn.thiago@cic.unb.br

Jacir Luiz Bordim
Department of Computer Science
University of Brasilia, UnB
Brasilia, Brazil
e-mail: bordim@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 spectrum use. In this context, this work addresses the problems of energy efficient multi-channel assignment and communication scheduling in wireless networks. Considering that the channel allocation is an NP-complete problem, this paper presents a time and energy-efficient protocol. The protocol divides its operation in management and transmission stages. Empirical 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.**

***Keywords*—wireless networks; energy efficient protocols; multi-channel assignment; scheduling.**

## I. Introduction

The quest for uninterrupted, high throughput wireless networks has been highly influenced by the popularisation of mobile devices and social networks. This trend in mobile applications has boosted the research efforts for Medium Access Control (MAC) protocols capable to cope with the demand. One of the major concerns in designing such protocols is to keep energy consumption at acceptable levels as the wireless devices are often powered by batteries. *Topology control* and *duty-cycle* are two energy saving strategies widely adopted in wireless networks [1]. 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 allow wireless devices to alternate between inactive and active mode. When in active mode, the device is able to send or receive data and when in doze mode, the device is in energy conservation mode, where it is not able to send or receive data. This last strategy is particularly challenging as a device in doze mode is not able to receive data packets. Thus, the development of techniques to ensure that communicating devices will be active at the same time when there is data to send or receive are necessary [4].

The available MAC protocols are usually designed for single-channel environments [5]. Such protocols, especially in dense scenarios, have problems with packet collision, thus increasing packet retransmission, end-to-end delay and reducing throughput. Multiple communication channels have been used to increase throughput in wireless networks [6]. Such channels can be obtained via opportunistic spectrum access techniques, thus obtaining temporary access to unused licensed frequencies [7]. With the availability of multiple channels, Frequency Division Multiple Access (FDMA) based techniques, for example, allows to select several communication channels with non-overlapping and non-interfering frequencies. Thus, a pair of nodes can communicate at the same time and without interference since they are allocated to different channels.

A number of works consider the use of multiple channels in wireless networks [8], [9], [10]. Some of these works combined multi-channel MAC protocols with duty-cycle schemes to increase network throughput and decrease energy consumption. Tang *et al.* [11] proposed a multi-channel energy efficient protocol that minimizes energy consumption in wireless sensor networks. The proposed protocol allows the transmitting nodes to estimate the receiving node activation time without the use of a control channel. Incel *et al.* [8] proposed a multi-channel MAC protocol for wireless sensor networks. The proposed scheme works in a distributed fashion and makes communication schedule 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, focus on maximising the throughout rather than minimizing energy consumption.

Zhang *et al.* [9] 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 the 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 [12]. As in [9], the ECOA-BP divides its operation in management and transmission stages and uses a control channel during the management stage. This technique uses efficient transmission assignment and duty-cycle strategy to alternate the nodes between active and inactive modes, thus reducing the power drainage rate.

The previous works show that is possible to reduce energy consumption at the cost of higher communication time. Conversely, one can minimise the communication time at the cost of higher energy consumption [13]. Clearly, there is a challenge in finding a compromise between these conflicting parameters. Both Zhang et al. [9] and Neves et al. [12] focus on balancing these parameters. However, the use of a single control channel in the management stage, independently of the number of available channels, can be a bottleneck, as it increases the communication time [2].

 ISBN: 978-1-61208-305-6 

This paper addresses the problems of multi-channel allocation, transmission scheduling and energy consumption in wireless networks. As in related works, it is assumed 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. The time is assumed to be slotted and its duration to be long enough to ensure a single data packet transmission or reception. 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. The proposed protocol operation is divided in management and transmission stages. Unlike most of the similar proposals, the proposed protocol uses all the available channels in both management and transmission stages. Empirical 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.

The remainder of this paper is organised as follows. Section II describes the considered communication model. Section III presents the channel assignment problem along with an energy-efficient heuristic to tackle it. Section IV presents the proposal, while Section V presents the empirical results. Section VI concludes the work.

## II. Communication Model

An ad hoc network consists of a set of $n$ nodes. A single-hop network setting can be represented by a complete graph $G'_n$, where each node in this network has a single transceiver and a unique identifier, that is known by the other nodes. The communication scenario of this network, on the other hand, can be represented by a directed graph $G = (V, E)$ (communication graph), where $V = \{v_1, v_2, ..., v_n\}$ is a set of nodes (vertices) and $E \subseteq V^2$ is a set of communications (edges). 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 $G'_n$. Each edge $e_h = (v_s, v_d) \in E$ represents a communication between a source node $v_s$ and a destination node $v_d$. There are no parallel edges between any two nodes. Consider $s_i$ as the transmission set of a node $v_i$, which contains all the nodes that $v_i$ ($v_i \in V$), has data packets to send, and $d_i$ as the reception set of a node $v_i$, which contains all the nodes that have data packets to $v_i$. This way, each node $v_i$ has $\tau_i = |s_i| + |d_i|$ data packets to send and receive.

As an example, Figure 1 represents a possible communication graph for a network topology $G'_n$. 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, the node $v_1$ has data to send to nodes $v_2$ and $v_4$ , thus, $s_1 = \{v_2, v_4\}$, and no data to receive, thus $d_1 = \emptyset$. 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 [9], 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

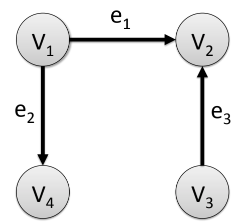


Figure 1: Communication graph example with 4 nodes.

active, a node can send or receive data. Otherwise, the node can save power in the idle mode. 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.

## III. The Channel Assignment Problem (CAP)

In a network environment, where many frequency channels are available, the task of channel assignment that satisfies the interference constraints and maximizes the throughput is known as the Channel Assignment Problem (CAP). To prevent interference between communications, the same channel cannot be allocated for two pairs of neighbouring nodes simultaneously. In its general form, the CAP problem is equivalent to the Generalised Graph-coloring Problem (GCP), which is known as a NP-complete problem [14]. 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 serialised. However, in the general case scenario, optimum solutions are complex to obtain.

Because of the NP-completeness of the CAP, many researchers proposed heuristics and approximation algorithms for the problem, which, however, can not guarantee optimum solutions. Proposed alternatives vary from neural networks, to genetic and graph theory based heuristics [14]. Next, an heuristic based on graph theory to solve the CAP problem is presented.

### A. ECOH: An Edge Coloring Heuristic

Figure 2 presents an Edge Coloring Heuristic, termed ECOH. The proposed heuristic takes as input a communication graph $G = (V, E)$ and a number $k$ of available channels. As output, the algorithm returns 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 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. The selection criterion is the choice of an edge belonging to

 

**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 proposed edge colouring heuristic (ECOH).

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 the condition $E = \emptyset$ is reached, line 2. In this example, o 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 $CS_i$ to communicate concurrently using the $k$ channels in the same time slot.

### B. ECOH: Involved Complexities

The ECOH heuristic has two main loops aligned: one that runs up to $E = \emptyset$ and another that compares vertices of edges in $E$ with vertices in $CS_i$, looking for edges without dependencies. Thus, ECOH runs in $O(p^2)$ time, where $|E| = p$. Note that this complexity considers the worst case scenario where the nodes are not able to communicate in parallel or $k = 1$. For latter reference, consider the following result:

*Lemma 1:* Given a number of channels $k$ and a communication graph $G = (V, E)$, the ECOH heuristic computes a list of communication sets $CS = \{CS_1, CS_2, ..., CS_r\}$ such that the edges in each set $CS_i$, $1 \leq i \leq r$ have no dependency with one another and $k \leq |CS_i|$. The ECOH computes the $r$ lists in $O(p^2)$ operations.

## IV. 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. In addition, the protocol performs these tasks in order to minimise both energy consumption and the time required to transmit data. First, it is presented some routines that are used in the protocol. Then, the protocol details are presented, followed by the protocol complexities.

### A. Transmission Set Grouping Routines

Recall that each node $v_i \in V$ contains a set $s_i$, which identifies the destination nodes to which node $v_i$ has data to send. In this subsection, the objective is combining such sets in 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, line 2. Let $v_l$ be the node with the highest ID in $g_i$. The loop in lines $3-8$ combines the transmission sets $s_j$, $1 \leq j \leq l$ such that the local node $v_l$ knows $s_l \cup s_{l-1} \cup ... \cup s_1$ in the end of the algorithm. Note that the above routine is very efficient in terms of energy consumption, once each node stays is active mode for just 2 time slots: one to send its transmission set and other to receive the transmission set from another node. Now, suppose that $|C| = k$, $k > 1$, channels are available, where $C$ is the set of channels $C = \{c_1, c_2, ..., c_k\}$. In this case, the $CombineGroup$ routine could be improved to take advantage of several channels.

The routine $CombineTS$, depicted in Figure 4, shows how transmission sets can be combined, using multiple channels simultaneously. Similarly to the $CombineGroup$ routine, $CombineTS$ takes two input parameters: 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, 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. In the end of the algorithm, the local node $v_1$ will have all the transmission sets $s_l \cup s_{l-1} \cup ... \cup s_1$. For latter reference, consider the following result:

*Lemma 2:* The $CombineGroup$ routine combines the transmissions sets in $g_i$ in $|g_i| - 1$ time slots using a single channel with each node in active mode for 2 time slots. The $CombineTS$ routine combines the transmission sets in $g_l$ in $\log k + 1$ time slots, using $|C| = k$ channels and with each node in active mode for at most $\log k + 1$ time slots, where $k \geq \lfloor \frac{l}{2} \rfloor$ and $|g_l| = l$. For both routines, it is assumed that each node can send at most 1 data packet to any other node in the network and it has a local buffer of $l^2$ bits.

**Algorithm** CombineGroup($g_i$,$c_i$)

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

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

**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.

### *B. EEMC-MAC Details*

Next, the details of the EEMC-MAC protocol is presented. The EEMC-MAC is divided in two stages: management and transmission, which are described in the next subsections.

*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. In 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 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.

**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.

Depending on the input, the $ManagementStage$ may call the $CombineGroup$ routine in parallel for $k$ groups, taking $\frac{n}{k} - 1$ time slots for execution with each node staying in active mode for 2 time slots. The $CombineTS$ routine is called once and takes $\log k + 1$ time slots to be executed with each node $v_i$ staying in active mode for $\tau_i$ time slots. Considering the results in Lemma 2, the following result is presented:

*Lemma 3:* Given a set of nodes $V$ and a set of channels $C$, where $|C| = k$ and $|V| = n$, the $ManagementStage$ combines the transmission sets in $O(\frac{n}{k} + \log k)$ time slots with each node in active mode for $O(\log k)$ time slots.

*2) EEMC-MAC: Transmission Stage:* The transmission stage of the EEMC-MAC protocol begins immediately after the management stage. In this stage, the leader node $v_m$ already computed the communication graph $G$. Figure 6 presents the $TransmissionStage$ details. In the beginning of the algorithm, the leader node $v_m$ has the communication graph of the entire network $G$. To solve the communication dependences, the leader node $v_m$ executes the ECOH heuristic and gets 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 a next step, all the nodes enter in active mode and tune into 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, goes from 1 to $r$ (the number of communication sets) and the second loop goes from 1 to the number of elements in the communication set indicated in the previous loop, lines 5-11. It begins by selecting an unused edge from the set $CS_i$. The nodes in this set enter in active mode, line 7, and tune in the indicated channel and perform the data transmission, line 8, returning to inactive mode after the transmission, line 9. This process continues until all the nodes in each communication set exchange their data sets.

The EEMC-MAC transmission stage runtime depends on how the ECOH heuristic creates the list of communication sets $CS = \{CS_1, CS_2, ..., CS_r\}$. Analysing the $TransmissionStage$ (Figure 6), note that $r$ time slots are necessary to perform all the transmissions represented in $CS$. The transmissions in $CS_i$ are disjoint and can be performed concurrently using multiple channels. An additional time slot is used for the $CS$ broadcast. Thus, in the worst case scenario, where every transmission has to be serialised, $r$ would be equal to the number of edges in the communication graph, that is, $r = p$. However, in the best case scenario, all the transmission in $G$ could be spread over the $k$ available channels, that is, $r = \frac{p}{k}$. This way, the transmission stage total runtime is between $\Omega(\frac{p}{k})$ and $O(p)$ time slots per execution. It is considered that the ECOH heuristic execution and $CS$ diffusion can be performed in the same time slot. It should be noted that every node in this stage, except when the nodes received the $CS$ broadcast, enters in active mode only to send or receive data. This way, the energy consumption for each node $v_i \in V$ in this stage is equal to $\tau_i + 1$. The EEMC-MAC transmission stage complexities are summarised below:

*Lemma 4:* The $TransmissionStage$ takes $\Omega(\frac{p}{k})$ and $O(p)$ time slots per execution with each network node $v_i \in V$ in active mode for no more than $\tau_i + 1$ time slots.

### C. EEMC-MAC: Main Procedure and Complexities

The main procedure of the EEMC-MAC protocol executes the two aforementioned stages in sequence. Thus, the EEMC-MAC total runtime is $\Omega(\log k + \lceil \frac{p}{k} \rceil)$ and $O(\lceil \frac{n}{k} \rceil + \log k + p)$ time slots. Theorem 1 summarises the protocol complexities, considering the Lemmas 3 and 4.

*Theorem 1:* The EEMC-MAC protocol solves the multi-channel medium access control and transmission scheduling in a time slotted, synchronized, single hop wireless settings, represented by a communication graph $G = (V, E)$, in $O(\lceil \frac{n}{k} \rceil + \log k + p)$ time slots, with each node $v_i \in V$ in active mode for $O(\log k + \tau_i)$ time slots, where $|V| = n$, $|E| = p$, $|C| = k$ and $\tau_i$ is the number of data packets that node $v_i$ has to send and receive.

### D. EEMC-MAC: A working example

To exemplify the protocol application, consider the communication graph represented by Figure 7a. 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.

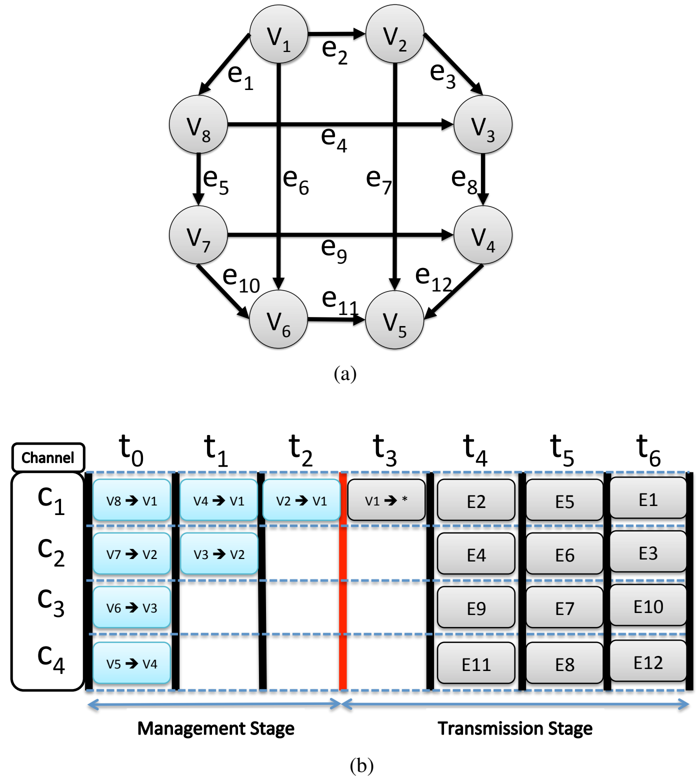


Figure 7: (a) Communication graph example with 8 nodes. (b) Channel representation for the EEMC-MAC protocol.

Figure 7b 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 7b. 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.

## V. SIMULATION

The evaluation of the proposed protocol has been performed through simulation. For this purpose, a simulator has been developed in Matlab environment [15]. The simulator incorporates the characteristics of the EEMC-MAC protocol, described in Section IV. To verify the goodness of the proposed solution, the simulation results are compared with the optimum solutions. This section describes the simulation parameters, the evaluation metrics and then presents the obtained results.

### *A. Simulation Parameters and Evaluation Metrics*

The simulation has been conduced for a varying number of nodes, data packets per node and data channels. The number $n$ of nodes assume the following values: $8, 16, 32, 64, 128, 256$. The number of data packets per node assume values in one of the five different ranges: $0\%$ to $20\%$, $21\%$ to $40\%$, $41\%$ to $60\%$, $61\%$ to $80\%$ and $81\%$ to $100\%$. Each range represents a percentage of the maximum number of transmissions per node. Recall, from the communication model, that each node can have a maximum of $n-1$ outcoming edges, that is, a node can send 0 or 1 packets to any destination in the communication graph per EEMC-MAC execution cycle. For example, in a setting with 16 nodes using the first range ($0\%$ to $20\%$), each node would have from 0 to $20\% * (16-1) = 3$ data packets to send. The number of channels is defined as $2^i$, with $i$ going from 0 to $\lfloor \frac{n}{2} \rfloor$. The simulation results are drawn from the average of 200 simulation runs for each setting.

The protocol execution time will be evaluated considering: $(i)$ the percentage of time the protocol spend in the transmission stage ($R_{ts}$); and $(ii)$ the ratio between the protocol transmission stage time and the optimum transmission stage time ($R_{opt}$). The $R_{ts}$ is defined as follows:

$$R_{ts} = \frac{T_t}{T_t + T_m}, \quad (1)$$

where $T_m$ is the number of time slots the protocol needed in the management stage, $T_t$ is the number of time slots needed in the transmission stage. Note that lower $R_{ts}$ value indicates that the protocol incurs in a lower message overhead to transmit the data items. The $R_{opt}$ is defined similarly:

$$R_{opt} = \frac{T_t}{T'_t}, \quad (2)$$

where $T'_t$ is the optimum transmission stage time (in time slots). The $R_{opt}$ values indicates the gap between the current transmission stage time and the optimum time. Clearly, when $R_{opt} = 1$, the EEMC-MAC protocol achieved the minimum necessary time to complete the transmission stage.

### *B. Simulation Results*

Figures 8a and 8c present the simulation results for $R_{ts}$, considering $n = 16, 32, 64$ nodes and 5 different transmission configurations, that is, a communication graph with 0% to 20% of maximum number of edges, 21% to 40%, and so on. The $x$-axis shows the number of channels while in the $y$-axis presents the $R_{ts}$ values.

It can be observed that the $R_{ts}$ values decrease with an increase in the number of channels. This was expected as an increase in the number of channels allows for a larger number of parallel transmissions, decreasing the time needed for the transmission stage. As the number of nodes increase, the management stage time decreases. This can be observed in Figure 8c, where the $R_{ts}$ is close to 100%. That is, the protocol spends most of its time in the transmission stage.

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 was, in average, less than 5% from the total protocol execution time. In what follows, a closer look is taken at the time needed for the transmission stage.

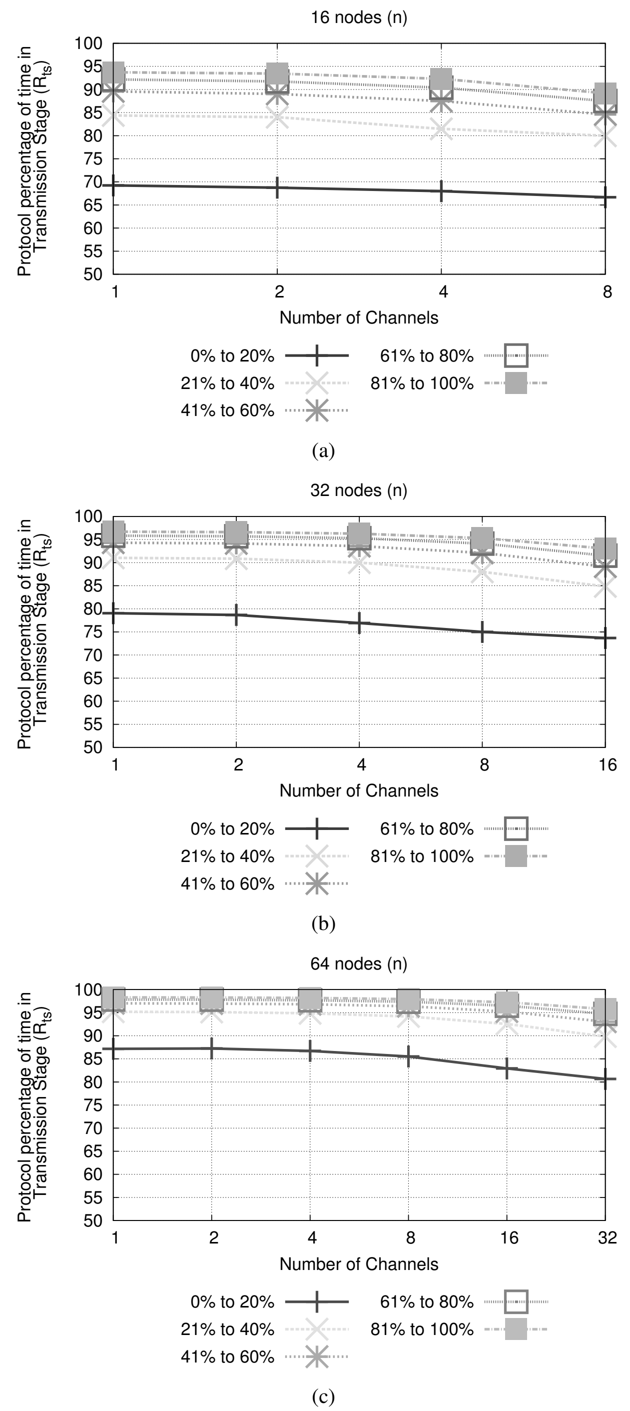


Figure 8: Simulation results for $n = 16, 32, 64$ nodes and metric $R_{ts}$.

Figure 9 presents the simulation results for the metric $R_{opt}$. From the Vizing theorem [16], it is a valid lower bound to assume that the optimum channel assignment execution time is equal to $\Delta(G)$, where $\Delta(G)$ is the maximum graph degree. Thus, for comparison purpose, it is assumed that $T_{t'} = \Delta(G)$.

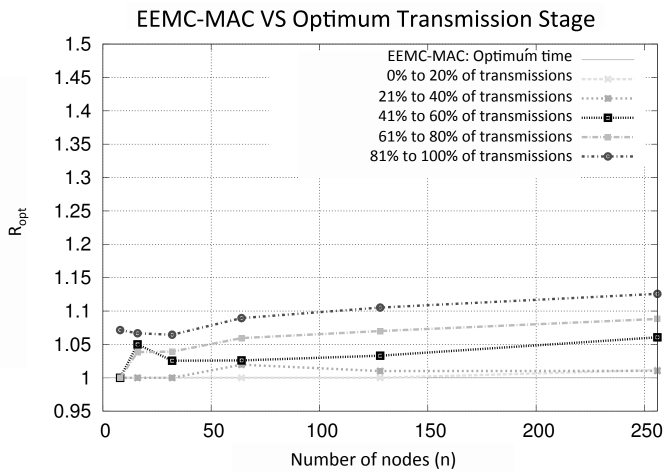


Figure 9: Simulation results for metric $R_{opt}$.

In the $x$-axis in Figure 9 shows the number of nodes in the communication graph while in the $y$-axis presents the $R_{opt}$ values. The number of data items per node follows the ranges defined previously.

It can be observed in Figure 9 that $R_{opt} \approx 1$ when a lower packet load (first two ranges) is presented. The $R_{opt}$ values 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 15% higher when compared with the optimum transmission stage time. A larger communication graph increases the number of similar choices in the selection criterion of the protocol transmission scheduling. Which, in turn, increases the chance of producing unfavourable scheduling, increasing the communication time. Note that the choice of an inappropriate transmission scheduling at a given step $S$ impacts in the choice of other transmissions at step $S+1$. From the results for metric $R_{opt}$ it is concluded that the EEMC-MAC achieved performance close to the optimum in many cases. When the average of all the communication setting is computed, the EEMC-MAC is less than 5% from the optimum time.

## VI. 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 minimum and empirical results shows that this stage represents less than 5% 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, 4% more time. In future works, it is intended to address fault tolerance and to improve the communication model.

## References


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

[2] M. F. Caetano ; Lourenço, B. F. ; Bordim, J.L.,"On Performance Of the IEEE 802.11 in a Multi-channel Environment". In: International Conference on Computer Communications and Networks, 2013, Nassau.

[3] Y. Zhu, M. Huang, S. Chen, and Y. Wang, "Energy-efficient topology control in cooperative ad hoc networks," Parallel and Distributed Systems, IEEE Transactions on, 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, 2006. WCNC 2006. IEEE, vol. 2., 2006, pp. 1172–1177.

[5] 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.

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

[7] C. R. Stevenson, G. Chouinard, Z. Lei, W. Hu, S. J. Shellhammer, and W. Caldwell, "IEEE 802.22: The first cognitive radio wireless regional area network standard," IEEE Stands in Communications and Networking, January 2009, pp. 130 – 138.

[8] 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.

[9] J. Zhang, G. Zhou, C. Huang, S. Son, and J. Stankovic, "Tmmac: An energy efficient multi-channel mac protocol for ad hoc networks," in Communications, 2007. ICC'07. IEEE International Conference on. IEEE, 2007, pp. 3554–3561.

[10] 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.

[11] 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.

[12] 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 Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International. IEEE, 2012, pp. 848–854.

[13] J. 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, 2003, pp. 1103–1112.

[14] 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.

[15] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

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

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