Electronics **2015**, *4*, 359-379; doi:10.3390/electronics4020359

OPEN ACCESS

electronics

ISSN 2079-9292

www.mdpi.com/journal/electronics

Article

HMC-MAC Protocol for High Data Rate Wireless Sensor Networks

Gerard Chalhoub *, Rana Diab and Michel Misson

Clermont University, University of Auvergne, LIMOS, BP 10448, F-63000, Clermont-Ferrand, France
CNRS, UMR 6158, LIMOS, F-63173 Aubiere, France

* Author to whom correspondence should be addressed; E-Mail: gerard.chalhoub@udamail.fr;
Tel.: +33-473-177-048; Fax: +33-473-177-111.

Academic Editor: Trung Dung Ngo

Received: 23 April 2015 / Accepted: 9 June 2015 / Published: 15 June 2015

Abstract: Using multiple channels in wireless sensor networks helps increase the overall throughput and avoid interferences. In addition, introducing multi-interface nodes further helps in increasing the packet delivery rate for those specific nodes. In this paper, we evaluate a channel allocation method based on neighborhood discovery up to 3 hops and a hybrid MAC protocol designed for high data rate wireless sensor networks. We propose a network segmentation that takes into consideration the existence of multi-interface sink in order to further increase to packet delivery ratio. Our protocol, called HMC-MAC, uses Time Devision Multiple Access (TDMA) for sequencing nodes activity, and Carrier Sense Multiple Access (CSMA)/CA with Frequency Devision Multiple Access (FDMA) for simultaneous data exchange. We evaluated our method using NS2 simulator and results show that HMC-MAC protocol improves the overall network performance compared to other protocols especially with high data rate and burst traffic.

Keywords: wireless sensor networks, MAC protocol, multi-channel, multi-hop

1. Introduction

Multi-channel MAC protocols have recently obtained considerable attention in wireless sensor networks (WSNs) because they help increase the network performance. Operating on multiple channels can ensure robustness against internal and external interference and allow collision free parallel communications.

Most existing MAC protocols proposed for WSNs use a single channel for data transmission. This is essentially due to the fact that energy efficiency is considered to be the most important issue in WSNs. On the other hand, in dense deployment of sensor networks, a single channel utilization generates a high level of interference and limit the use of the bandwidth. This is especially the case when a large burst of packets is generated simultaneously in different nodes due to a change in the monitored conditions, or when running multiple applications and accumulating the traffic of each application leads to higher packet density in the network. In these cases, the use of multiple channel can help optimize the bandwidth usage as it has been shown in the literature.

The hardware of most popular platforms and those compliant with the IEEE 802.15.4 and Zigbee standards provides a radio chip capable of switching its communication channel, and is therefore not limited to a single channel operation. In the literature, several surveys on multi-channel MAC protocols with different objectives have been proposed for WSNs [1–3]. These surveys classify multi-channel MAC protocols according to several criteria: periodicity of channel assignment, centralized or distributed manner for channel assignment, schedule-based or contention based MAC protocols.

In a multi-hop topology, packets must be transmitted several times in order to reach the final destination. This overloads the radio channel and induces interference. It also increases the risk of collision due to the hidden terminal problem for example [4], which results in degrading the medium access protocol performance. The use of multiple channels helps mitigate the effect of interference and enhances the network performance under high traffic load. The availability of multiple channels increases the number of simultaneous transmissions between nodes which leads to an increase in the global network throughput.

HMC-MAC was first proposed in [5], a Hybrid Multi-Channel MAC protocol that uses a combination between TDMA (Time Division Multiple Access), CSMA (Carrier Sense Multiple Access) and FDMA (Frequency Division Multiple Access) techniques. It is based on CSMA/CA which has the advantage of being a distributed and a lightweight algorithm that does not require a strict synchronization to exchange data frames and which enables the network to be easily scalable. Simulation results using NS2 simulator showed that HMC-MAC provides a smaller interference level compared to other allocation methods based on random choice and 2-hop neighborhood information.

In this paper, we extend the state of the art, we give more details about the algorithms of our contribution and we evaluate additional network metrics. We consider high data rates and burst traffic in multi-hop WSNs where a large burst of packets needs to be transported to the final destination. We consider one single destination also known as sink. We are interested in evaluating the performance of HMC-MAC and compare it to other channel allocation schemes and MAC protocols. We introduce a multi-interface sink in order to further enhance the network throughput when all the traffic is destined to the sink. In order to optimize the reception of the sink, we modified the channel allocation method in order to keep the interfaces of the sink in reception mode and at least one neighbor of each interface in transmission mode. Thus, the channel allocation scheme allows the sink to continuously receive data frames.

The paper is organized as follows. In Section 2, we give a quick survey of the different techniques used in multi-channel MAC protocols for WSNs. In Section 3, we describe how HMC-MAC works and explain the neighbor discovery process, the new channel allocation method and data exchange method that take into account the multi-interface sink. Section 4 is dedicated to the performance evaluation of HMC-MAC using the NS-2 simulator. Finally, we conclude the paper and we discuss perspectives in Section 5.

2. State of the Art

In [6] authors propose MASN, a new multi-channel allocation protocol for hierarchical ZigBee/IEEE 802.15.4 networks corresponding to realistic situations where the number of nodes is less than 50 with a hierarchical routing and many-to-one transmissions. The main goal is to improve the bandwidth. The unslotted CSMA/CA access method is used. In MASN, assignment of the different channels is centralized and distributed by the coordinator by using a function designed to calculate the channel offset between two successive children. The proposed channel assignment algorithm is inspired by the hierarchical address assignment process used in ZigBee. The performance of MASN is evaluated through simulations using NS2 simulator. MASN is compared with mono-channel, random allocation, and subtree allocation. Results show that the proposed protocol improves the global throughput by a factor between 2 and 5 compared to the single channel, and by an average factor of 2 compared to a random channel allocation or to a simple subtree distribution. The main advantage of this solution is its simplicity of integration in IEEE 802.15.4 devices with light modifications on the MAC layer. On the other hand, authors used a single topology for the simulations which might be the most convenient topology for MASN. Also if a node is performing transmission and reception simultaneously, then it leads to conflict. This case is not studied in MASN.

In [7] authors proposed EE-MAC (Energy Efficient hybrid MAC for WSN), a hybrid of TDMA and FDMA MAC protocol with priority technique. It is a centralized protocol that uses a semi-dynamic channel assignment approach for channel allocation. The goal of this protocol is to improve energy efficiency, delay and packet delivery ratio. EE-MAC operates in two phases, a setup phase and a transmission phase. During the first phase, the following operations take place: neighbor discovery, TDMA slot assignment, FDMA slot assignment, local framing, and global synchronization. These operations run only during the set-up phase and every time a change in the topology occurs. During the transmission phase, the period is divided into time intervals. Each frame is equally divided into several fixed time slots. Each slot is divided into schedule subslots and contention subslots. Each cycle starts with scheduled slots followed by contention slots. Nodes use LPL (Low Power Listening) [8] during contention slots and send HELLO messages to the base station. When a node hears a HELLO message sent to the base station by one of its neighbors, it adds that nodes to its neighbor list. The updated neighbor list is included in the next Hello message and sent by that node. Based on Hello messages received from all nodes, the base station performs a breadth first search algorithm in order to assign for each node an appropriate channel as well as a specific time slot. Scheduled node can communicate in a collision free manner. EE-MAC introduces an adaptive priority scheme for channel access. The priority scheme is inspired from IEEE 802.11e [9]. The priority is based on some statistics collected by these nodes during their lifetime which includes forwarding loads, the distance to the base station, and the remaining battery charge. Authors compared the protocol to HyMAC [10] protocol in terms of energy efficiency, packet delivery ratio and end to end delay. The simulation is carried out by varying the number of packets, the number of nodes and by changing the packet size. Simulation results using NS2 simulator showed that EE-MAC provides a significant improvement compared to HyMAC, in energy efficiency, packet delivery ratio, and delay. On the other hand, EE-MAC suffers from the overhead of Hello messages that are exchanged and sent to the base station.

In [11], authors proposed MMSN protocol which is a distributed protocol that combines FDMA and CSMA mechanisms. It uses a semi-dynamic channel assignment approach for channel allocation. MMSN uses different strategies for channel assignment. Strategies differ according to the level of overhead and the effectiveness of the channel allocation. These strategies are the following:

(i) The exclusive frequency assignment strategy, the number of frequencies should be equal to or greater than the number of nodes within two hops. Nodes exchange their identities (IDs) in their 2-hop neighborhood. Nodes choose frequencies according to the increasing order of their ID values. If a node presents the smallest ID in its 2-hop neighborhood, it chooses the smallest frequency among those available and then it sends the chosen frequency to its 2-hop neighbors. This strategy ensures that nodes in any 2-hop neighborhood are assigned different frequencies, but it induces high communication overhead due to the ID and frequency exchange between 2-hop neighbors using CSMA/CA.

(ii) The even-selection strategy which is more suitable for dense WSNs where the number of frequencies is smaller than the number of nodes within two hops. It is similar to exclusive frequency strategy, but when all frequencies are already chosen in a 2-hop neighborhood, a node randomly chooses one of the least chosen frequencies.

(iii) The eavesdropping strategy where each node takes a random backoff during which it waits for its 1-hop neighbors to choose their frequencies. When its backoff expires, it randomly chooses one of the least chosen frequencies within one hop. This strategy has less communication overhead compared to the first two strategies, but it only collects frequencies information within one hop neighborhood and during a limited random time interval which results in more interference.

(iv) The implicit-consensus strategy where each node calculates a random number for itself and a random number for each node in its 2-hop neighborhood using the same pseudo-random number generator. A node wins current frequency if its random number is the highest among all the other random numbers. It then broadcasts its frequency to the 2-hop neighbors. This strategy guarantees smaller overhead but assumes a high number of available frequencies.

At the beginning of each time slot, nodes contend for the medium to broadcast control traffic on a common broadcast channel. When a node transmits a packet, it switches between its own channel and the destination channel during the preamble sending time, which results in increasing the protocol overhead and the number of repetitions due to frequent channel switching. The performance of MMSN is evaluated through simulations using GloMoSim simulator and compared with CSMA/CA.

In [12], authors propose MC-LMAC (Multi-Channel Lightweight MAC) protocol which is a distributed schedule-based multi-channel MAC protocol based on the single channel energy-efficient MAC protocol LMAC [13]. MC-LMAC guarantees that the same slot/channel pair is not simultaneously used within a two-hop neighborhood. Each node manages an “occupied slots” vector per channel with a length equal to the number of time slots. This vector is used for storing the information about the slots occupied by the neighbors. Each bit in the “occupied slots” vector is set if the time slot at the same position is occupied. This vector is then sent to 1-hop neighbors. In order for a node to choose a channel, it executes an OR operation over all the "occupied slots" vectors received on a certain channel in order to find a free time slot. The authors claim that there are no collision and no interference in a 2-hop neighboring allocation, but we will show this is not the right strategy when data frame acknowledgments are used. On the other hand, MC-LMAC suffers from the overhead of the control messages that are exchanged in order to discover the channels used in the different TMDA slots in 2-hop neighborhood. The problem increases as the network density increases.

In [14], authors propose MuChMAC (Multi-Channel MAC) protocol, a distributed low-overhead Multi-Channel MAC protocol. It uses a dynamic channel assignment approach. It combines TDMA and asynchronous MAC techniques. Time is divided into slots. A node is able to independently choose its receiving channel switching sequences based on its ID and the current slot number using a pseudo-random generator. A broadcast slot is inserted every n slots. These common broadcast slots also follow a pseudo-random channel hopping sequence. A sender is thus able to calculate the channel of the receiver. Each node switches its radio transceiver to the receiver channel and starts to send small preamble messages. When the receiver wakes up, it hears a preamble and it acknowledges the sender that it is awake in order to start sending data frames. MuChMAC ensures waking up the receiver and the sender within the same slot using loose global time synchronization and a fixed upper bound on clock drift. MuChMAC is tested using Sentilla JCreate motes in order to evaluate its performance. Authors compare the protocol with X-MAC [15]. Results show that the protocol is energy efficient and improves reliability. On the other hand, the channel allocation is based on a random mechanism that does not take into consideration the channel usage in the neighborhood which might lead to a high collision rate.

In [16], authors describe TMCP a centralized Tree-based Multi-Channel Protocol for data collection applications based on CSMA/CA. It uses a fixed channel assignment approach for channel allocation. The whole network is partitioned into multiple sub-trees having the base station (BS) as a common root where each sub-tree is allocated a different channel. TMCP can work with a small number of channels and without the need for time synchronization. It uses a greedy algorithm which decreases the radio interference. TMCP has three components, Channel Detection (CD), Channel Assignment (CA), and Data Communication (DC). CD finds available orthogonal channels. CA partitions the whole network into sub-trees and allocates a different channel to each sub-tree. DC manages the data collection through each sub-tree. TMCP is tested using MicaZ motes and simulated using GloMoSim in order to evaluate its performance. Results show that TMCP improves the throughput while keeping high packet delivery ratio and low latency. On the other hand, TMCP blocks the direct communications between nodes belonging to different sub-trees which is an inconvenient restriction. In addition, communications inside the sub-tree are contention based and use CSMA/CA algorithm.

Y-MAC [17] is a TDMA-based multi-channel MAC protocol that aims at reducing latency. Authors adapted a distributed method based on algorithms proposed in LMAC and MMSN. Time slots are assigned to the receivers. Y-MAC uses a slot allocation vector in order to achieve time slot assignment. It allows new nodes to join the network and to assign time slots in a dynamic way. This vector is used for storing information about the slots occupied by the neighbors. This vector is then broadcast to 1-hop neighbors. In order for a node to allocate a free time slot in its 2-hop neighborhood, it executes an OR operation over all the slot allocation vectors received from its neighbors. If there are many available time slots, one of them is randomly selected; otherwise several nodes should share the same time slot. At the beginning of each time slot, potential senders to the same receiver compete in order to access the medium using a CSMA/CA algorithm. Multiple packets are sent successively on different channels, the receiver and the sender hop to a new channel according to a predetermined sequence. Y-MAC reduces latency by offering the possibility for nodes that were not able to send their traffic to compete in a consecutive slot. All potential senders that have pending frames destined to the same receiver also hop to the same channel and compete again.

Y-MAC was implemented using the RTOS operating system on TmoteSky motes. Authors compared Y-MAC with Crankshaft in multi-hop environments. Results show that the duty cycle of Y-MAC is slightly higher than Crankshaft. Also the average per hop delivery latency of Crankshaft rises faster than that of Y-MAC as the traffic load increases and Y-MAC outperforms Crankshaft in terms of data reception rates. Y-MAC exploits multiple channels in order to handle burst messages. Authors proved that Y-MAC achieves effective transmission of burst messages under high traffic conditions (1 frame/node/sec). On the other hand, the channel allocation method is not detailed in the paper, authors only insist on not using the same channel in 1-hop neighborhood which leads in high interference due to 2-hop interference (interference caused by transmissions received from nodes that are 2-hop away from the receiver).

We argue that 2-hop neighborhood is not enough to avoid collisions and interference, and that 3-hop neighborhood should be considered when allocating channels. In addition, neighborhood discovery causes traffic congestion when the network is dense, and thus exchanging control traffic results in a lot of collisions using CSMA/CA. In order to avoid this, we propose to use bitmaps and TDMA for representing and exchanging neighboring information. Very few protocols use multi-interface nodes and thus most protocols reach network saturation very fast. In our protocol we propose to use multi-interface nodes for the sink and show how our protocol copes better with high data rates. This paper is an extension to the short paper presented in [18]. In the short paper, the state of the art was very brief and did not exceed one column in double column format. In this paper, we extended the state of the art and discussed the performance results of cited papers. We also explained in more details the HMC protocol. In addition, only aggregate throughput and packet loss were evaluated in the short paper. In this paper, more significant results are discussed such as packet delivery ratio and number of frame repetitions.

3. HMC-MAC Protocol

In this section, we explain how the network is created and neighbor discovery is done according to HMC-MAC. A time segmentation approach is used by HMC-MAC and is decided by a special multi-interface node, the Network Coordinator (NC). We explain how TDMA is used to ensure a multi-hop beacon propagation for synchronization and neighbor discovery. Then, we explain how the neighborhood information is used to allocate channels.

3.1. Network Creation and Beacon Propagation

The first node to be activated in the network is the node that will create the network and start broadcasting periodical beacons. When a new node is activated and wants to join the network, it starts by scanning for beacons. If no beacon is detected, the node will consider that it is the first node in the network and create a new network. This node is known as the Network Coordinator (NC) and is on depth 0. In case the new node receives at least one beacon, it chooses the most suitable one and sends an association request to the node that is broadcasting it. The choice of the beacon can be based on different criteria such as the RSSI and the LQI for robust links, or the number of hops separating the source of the beacon from the destination node for minimum end-to-end delay, or a combination of both depending on the application needs. In our case we consider the current number of children of the source in order to achieve load balancing at the association phase.

In the case where two nodes create new networks at the same time and are able to detect each other, the NC with a bigger ID quits its own network and joins the network of the NC with the smaller ID. The case where the NCs are not in range is out of the scope of this paper and will be studied in future works.

Time is divided into cycles (Figure 1). Each cycle is divided into intervals during which communications are organized in a specific manner. Each cycle begins with a TDMA beacon propagation period $[T_0; T_1]$ that is followed by a data exchange period $[T_1; T_2]$ and an inactive period $[T_2; T_0]$. Depending on the application needs, the duration of these intervals can be adjusted to absorb the generated traffic.

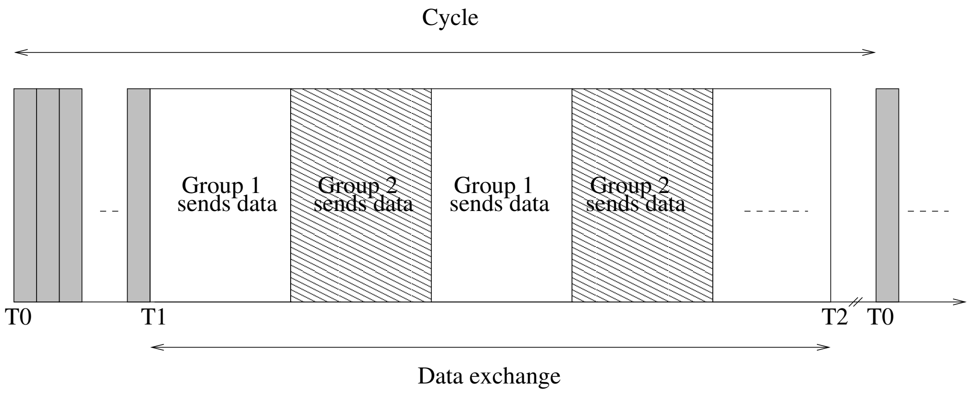

Figure 1. Global cycle. $[T_0; T_1]$ is a synchronization period, $[T_1; T_2]$ is dedicated for data exchange.

During $[T_0; T_1]$ the NC broadcasts a beacon that is propagated in a multi-hop manner to reach all the nodes of the network based on the MaCARI protocol [19]. Each node in the network has a unique beacon transmission slot during which it broadcasts its beacon. The beacon contains information that enables each node to know in which slot it should propagate its beacon. When a new node joins the network, its address is sent to the NC in order to update the propagation order. The NC will then include this address in the beacon for m consecutive cycles. Similarly, in case of a node departure, a leave request is sent to the NC (it can be sent by the node that is willing to leave the network, or sent by neighboring nodes that detected that a node has failed. The latter aspect is out of the scope of this paper). This allows nodes to update their local propagation order. The propagation order is maintained in each node and updated using the beacon. The order of the addresses defines the order in which the beacon is propagated: if the address of a node has the index 3 in the propagation order, it means that it is the third node to send the beacon during $[T_0; T_1]$ (more details on this aspect can be found in [5,20]). This guarantees that beacons are not sent at the same time and thus avoids collisions. The beacon is propagated using a known common channel. This way new nodes scan only one channel to discover the network.

3.2. Neighbor discovery

Neighbor discovery enables us to know which nodes are prone to interfere and which nodes are authorized to transmit simultaneously. In order to avoid overloading the network with long control messages to exchange neighborhood information, we use bitmaps to represent neighbors. Using the local propagation order (which is a list of all the node addresses of the network), every node is able to build and manage a bitmap that represents all the nodes in the network. Each index of the bitmap corresponds to the node address with the same index in the propagation order. This way, any node is able to represent its neighbors using one bit for each neighbor. Thus, when the network is dense, neighborhood information will not overload the nodes with control traffic and neighbor list tables using addresses or identifications.

In order to avoid interference when acknowledgments are used, 3-hop neighborhood needs to be discovered. In [20], we discussed why that the reuse of channels should be considered up to 3-hop neighborhood to avoid collisions and interference when immediate MAC layer acknowledgments are used. Figure 2 shows an example of how the use of the same frequency channel by a 3-hop neighbor might cause a collision. Nodes A and D are 3-hop neighbors using the same channel. A collision can occur on C when B is sending data to A and D is sending an acknowledgment to C. Thus, the same channel cannot be used in a 3-hop neighborhood.

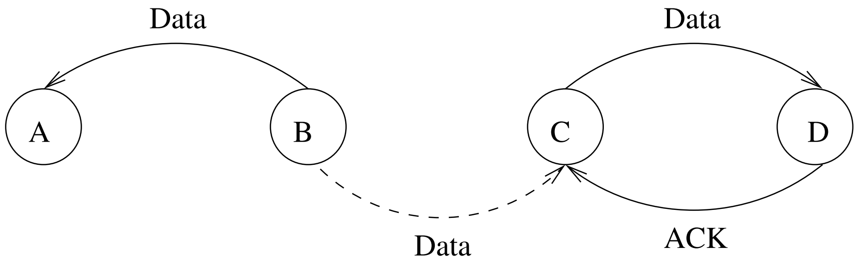

Figure 2. 3-hop neighborhood interference in channel allocation with acknowledgments. A and D are 3-hop away nodes, if they are allowed to receive data during the same time slots, the ACK sent from D to C will collide with the data sent from B to A.

One hop neighbor list is built using the source address of received beacons. When a node receives a beacon from another node, it considers that this node is a neighbor. This does not guarantee that the link is bi-directional, but we assume that it is an interfering node. In order to build 2-hop neighbor lists, each node includes the bitmap of its 1-hop neighbors in the beacon. Hence, when a node receives all the beacons from its neighbors, it is then able to build the list of its 2-hop neighbors. When a node figures in the bitmap of a neighbor, it can consider that a bi-directional link exists with that neighbor. In order to build 3-hop neighbor lists, each node should include the bitmap of its 1-hop neighbors and the bitmap of its 2-hop neighbors in the beacon it sends.

By using the beacon and the propagation order to build 1-hop, 2-hop and 3-hop interference neighborhood, we avoid using additional control traffic such as HELLO messages that are traditionally used by the routing protocols. By using bitmap codification and sending beacons in a TDMA manner, neighboring information are exchanged efficiently between nodes without collision and with light overhead. It should be noted that only new arriving nodes are included in the propagation order to avoid sending the complete list in the beacon and overload the frame payload.

3.3. Channel Allocation Scheme and Node Activity

The role of the NC is to divide time into intervals and inform all nodes of this time segmentation using the beacon frame. In this paper, we assume that each node has only one radio interface (nodes should be low cost and use energy efficient hardware components), the sink can have several radio interfaces. We consider the availability of 16 orthogonal communication channels as allowed by the IEEE 802.15.4 standard [21].

All nodes willing to communicate with the same node compete on the same channel in order to access the medium and send information using CSMA/CA algorithm. Topology is organized into several depths based on the ZigBee Custer-Tree topology [22]. We consider that all the traffic is destined to the NC, for that, exchanges can be sequenced according to the depth of the nodes in the topology and channels can be allocated taking into account the traffic orientation.

In order to increase the throughput, the NC should remain in reception mode and the nodes in depth 1 should should alternate between sending mode and reception mode in order to keep part of them in transmission mode. Thus, we divided the network into two groups (Figure 3), $Group$ 1 includes odd depth nodes that are descendants of an even child of an interface (B, D, F are even children of an interface), and also includes even depth nodes that are descendants of an odd child of an interface. All other nodes are included in $Group$ 2. For example, in Figure 3 $Group$ 1 includes nodes G, H, B, M, D, O, P, F. All the other nodes are part of the second group. When nodes of $Group$ 1 are in transmission mode, nodes of $Group$ 2 are in reception mode, and vice versa. This insures that interfaces of the NC always have nodes in transmission mode forwarding them traffic (this can be ensured using methods of topology control, but it is out of the scope of this paper).

Figure 4 depicts the algorithm that is executed by each node in order to calculate the index of the branch to which it is affiliated. The index of the branch will allow the node to which group it belongs. We use the hierarchical addresses of ZigBee based on the Cskip formula as explained in the standard [22].

For channel allocation, each node has a 3-hop neighborhood bitmap which enables it to choose dynamically its own channel. The node with the highest priority in the 3-hop neighborhood chooses its channel first. Priorities are assigned according to the network addresses. The node with the smallest network address has the highest priority. A node proceeds to its channel allocation as soon as it becomes the node with the smallest address among its 3 hop-neighborhood that is not yet assigned a channel. A bitmap that represents all the nodes is also used to announce which nodes have completed the channel allocation process in order for each node to know if it is its turn to choose a channel. When a node chooses its channel, it broadcasts it in the beacon frame.

Each node then builds a 16-bit bitmap for nodes that belong to $Group$ 1. It also builds 16-bit bitmap for nodes that belong to group $Group$ 2. Where each bit in the bitmaps indicates the occupancy of the equivalent channel in 1-hop neighborhood. These bitmaps are then included to the beacon which allows nodes to build a 2-hop channel allocation bitmaps. All four bitmaps are then included in the beacon to build a 3-hop channel allocation view for each group of nodes.

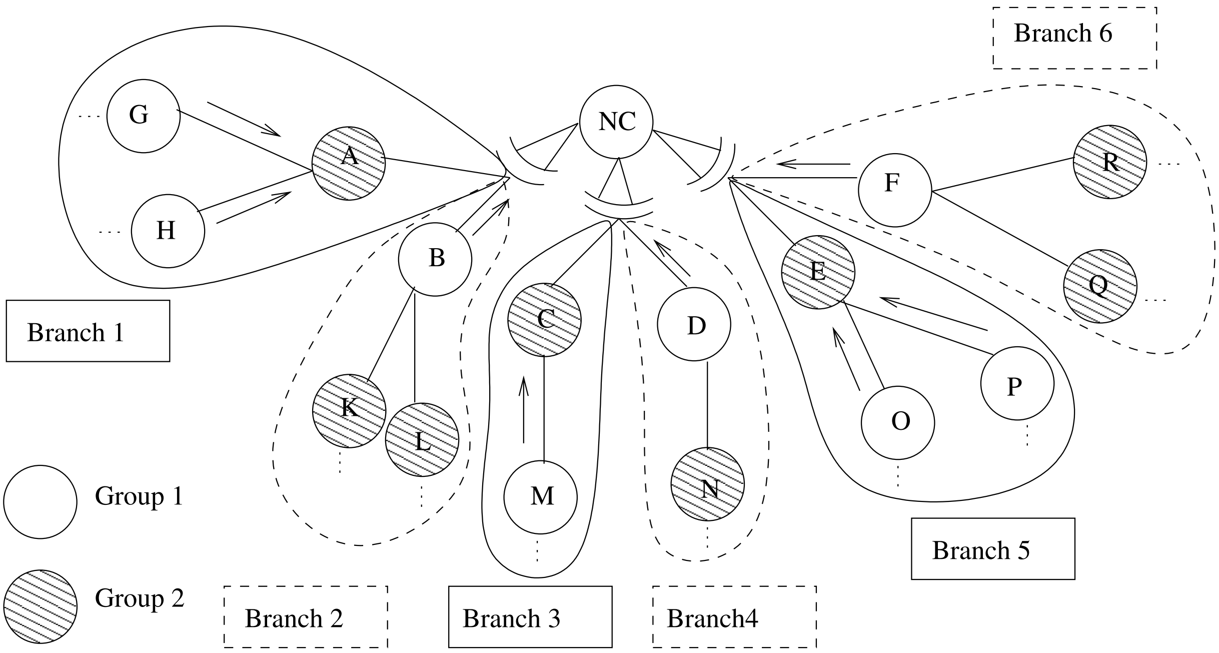

Figure 3. Network segmentation. Each interface divides its descendants into two branches. In the first branch, even depth nodes belongs to $Group\ 1$, odd depth nodes belongs to $Group\ 2$. In the second branch, odd depth nodes belongs to $Group\ 1$, even depth nodes belongs to $Group\ 2$.

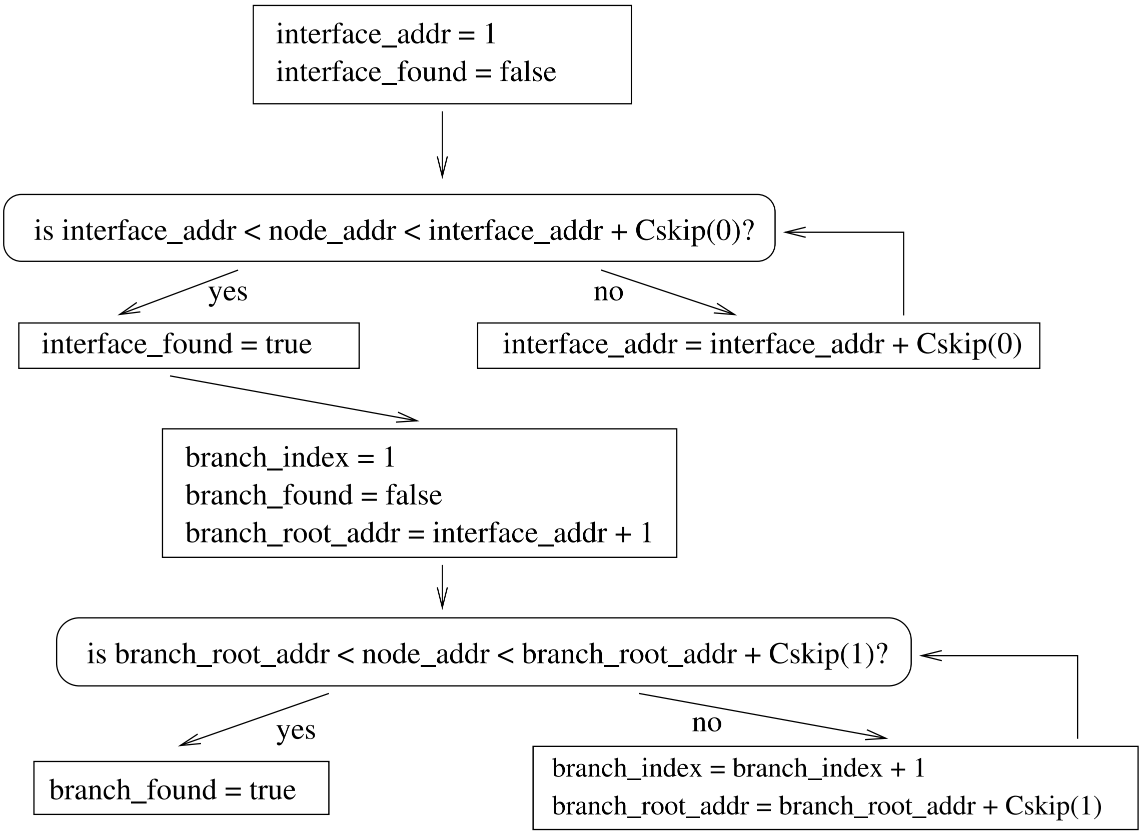
interface_addr = 1
interface_found = false
is interface_addr < node_addr < interface_addr + Cskip(0)?
yes
no
interface_found = true
interface_addr = interface_addr + Cskip(0)
branch_index = 1
branch_found = false
branch_root_addr = interface_addr + 1
is branch_root_addr < node_addr < branch_root_addr + Cskip(1)?
yes
no
branch_found = true
branch_index = branch_index + 1
branch_root_addr = branch_root_addr + Cskip(1)

Figure 4. Algorithm for calculating the index of the branch to which a node is affiliated.

16 channels are not sufficient in order to allow all nodes to communicate at the same time without interference when the network is dense. Hence, the channel allocation process should exploit spatial reuse or allow interfering nodes to use same channels using CSMA/CA. In our channel allocation process, we will try to decrease the interference as much as possible. A node first tries to find a free channel in its 3-hop neighborhood of its group, it it does not find one, it tries to find a free channel in its 2-hop neighborhood of its group. In case there are no available channels in its 2-hop neighborhood, the node tries to find a free channel in its 1-hop neighborhood of its group. Finally, if it does not find a free channel, it randomly takes a channel among the least used channels in its 1-hop neighborhood that are part of its group. Indeed, because each node signals its channel in the beacon it broadcasts, each node can manage the number of neighboring nodes using the same channel. Note that whenever multiple channels are available, nodes randomly choose one of the available channels. Figure 5 depicts the steps executed by each node before choosing a channel.

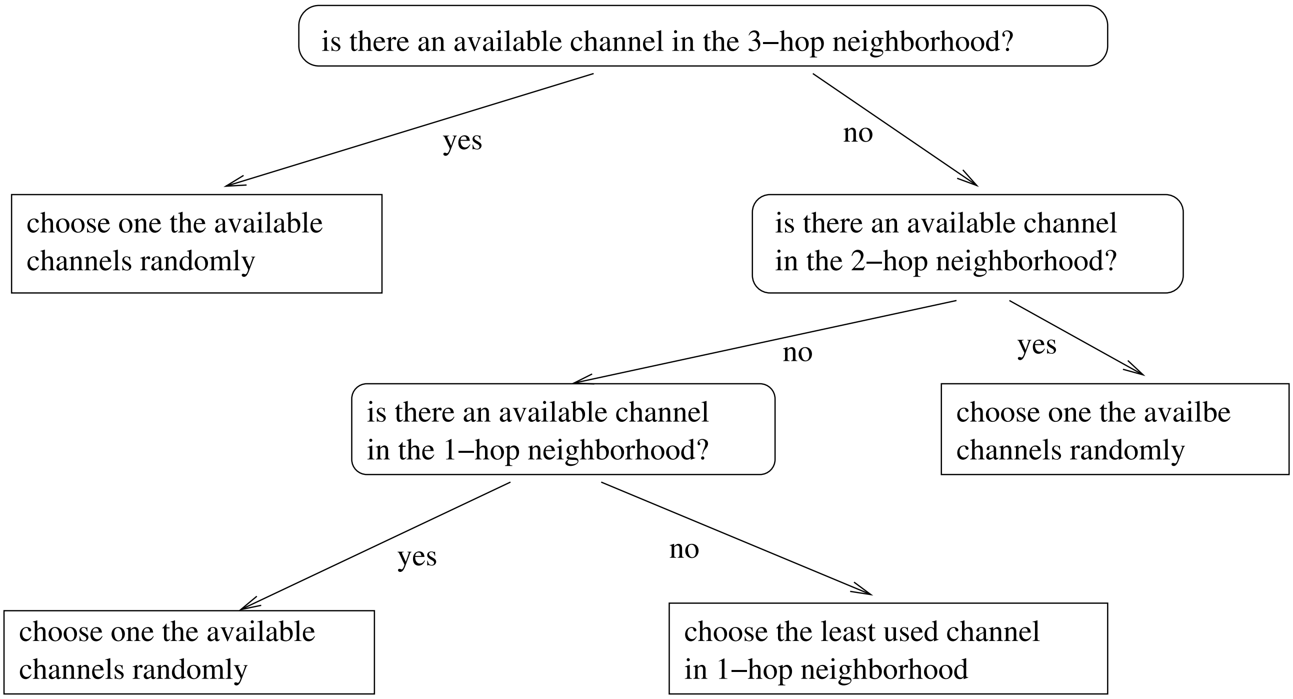
is there an available channel in the 3−hop neighborhood?
yes
no
choose one the available channels randomly
is there an available channel in the 2−hop neighborhood?
no
yes
is there an available channel in the 1−hop neighborhood?
choose one the availbe channels randomly
yes
no
choose one the available channels randomly
choose the least used channel in 1−hop neighborhood

Figure 5. Algorithm for choosing the a channel.

These operations run only during the setup phase or if changes in the topology occur. The setup phase ends when all nodes have chosen frequencies. Then, data exchange phase can start. $[T_1; T_2]$ is divided into intervals, each interval is divided into two time slots. During the first time slot, $Group$ 1 are senders and $Group$ 2 are receivers. During the following time slot nodes exchange roles, receivers become senders and senders become receivers. Each node knows its own depth. Thus, it is able to alternate between sending and receiving states. Nodes receive data frames from lower depth neighbors and transmit frames to one of their higher depth neighbors using slotted CSMA/CA algorithm.

4. Performance Evaluation

In this section, we evaluate the overall performance of HMC-MAC and we compare it to other methods. The metrics used for the evaluation is the aggregate throughput (which is the number of received packets per seconds at the sink), packet delivery ratio, number of frame repetitions and number of dropped packets. We used NS-2 simulator. We implemented the beacon propagation process where nodes send beacons in a TDMA fashion as explained in part 3. Each node includes its 1-hop and 2-hop neighbors in the beacon frame enabling a 3-hop neighborhood discovery. The network dimensions are $100 \times 100\ m^2$ with a communication range of $20\ m$. Our topologies correspond to hierarchical topologies of 50 nodes with a maximum depth of 7 (Lm = 7), and 3 children by coordinator (Cm = 3), there are no end-devices in the network. We used the ZigBee hierarchical routing protocol in all simulations [22]. We also assumed that number of available channels is 16. We fixed the packet queue size to 200. Each point in the graph is the mean over 50 iterations. These parameters are the same for all the evaluated methods.

In this paper, we do not study the effect of the overhead of the synchronization period nor the build-up phase of the network. Moreover, all the presented results are obtained from a 20 s observation window during $[T_1; T_2]$ activity period. During this window, we have 80 alternations between group intervals (each interval being $125\ ms$ long). This evaluation is done after the network has reached a stable state and all nodes have joined in. In this paper we do not study the impact of the duration of group intervals on the network performance.

In this version of HMC-MAC, we considered only 3 interfaces for the sink and two children for each interface. There is no rational for the number of interfaces, 3 is just a case study. It is clear that with more interfaces we can achieve better throughput. We limited the number of children per interface to 2 in order to avoid competition between these nodes and thus avoid collisions on the sink on each interface. The simulation is carried out by varying the number of packets generated by each node, the topologies and the type of traffic production. Also note that we do not do any packet or frame aggregation in the network.

4.1. Traffic Production

We defined two types of traffic generation profiles: periodic generation and burst generation. In the periodic traffic generation, nodes generate data in a periodic manner similar to a CBR (Constant Bit Rate) traffic. In the burst traffic generation, nodes generate packets by doubling the rate of periodic generation in the first second and then refrain from generating traffic in the following second. This type of traffic generation emulates sudden data bursts. With both traffic generation profiles we reach the same number of generated packets. But, in the burst generation, nodes compete much more to access to the medium for 50% of the time compared to the periodic generation and spend the other 50% to forward packets accumulated in the packet queue. This results in higher offered load during the first 50% compared with the periodic traffic. Indeed, the same number of packets is generated in 50% of the time. Figure 6 shows how traffic is generated in both modes.

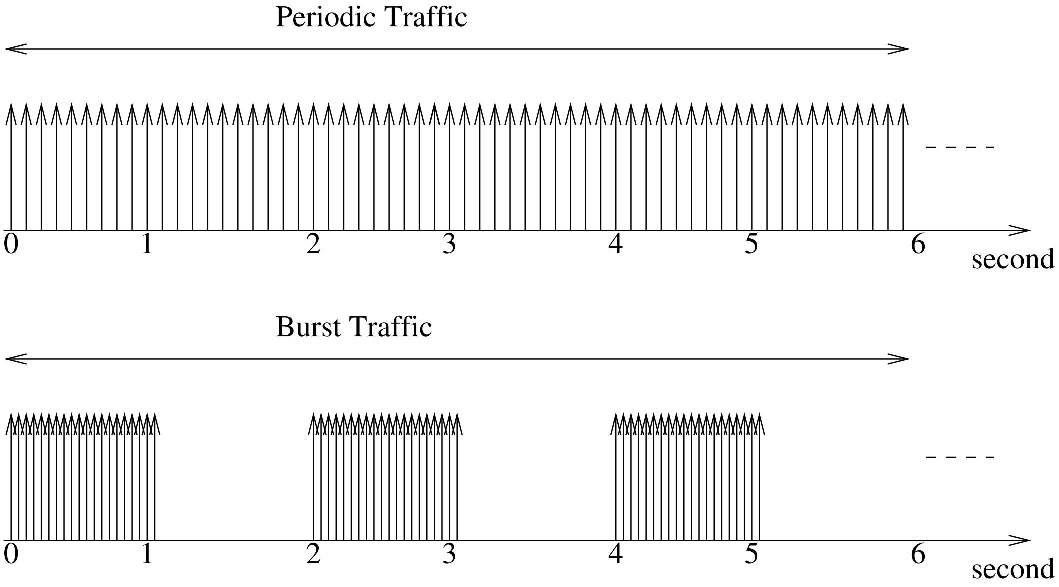

Figure 6. Traffic generation in periodic and burst modes.

4.2. Comparison with other Methods

Most protocols in the literature use 2-hop information to allocate channels and some of them use random allocation. Others divide the network into clusters. Accordingly, we chose to compare our protocol with four different allocation methods: HMC without segmentation method, 2-hop method, random method and cluster method. These are variants of HMC-MAC but inspired by protocols from the literature.

In the HMC without segmentation method, a node applies the same channel allocation scheme as HMC-MAC but without taking into account the even or odd depths nor the even or odd branches. This method is evaluated in order to show this importance of network segmentation. In the 2-hop method, a node chooses an available channel in its 2-hop neighborhood. In case all channels are used, it randomly chooses a channel. This method is used to evaluate the effect of doing an allocation based on 3-hop information and not only 2-hop information. In the random method, a node randomly chooses a channel from the list of authorized channels without checking the availability of this channel. This method is simply a default method used for comparison sake only. In the cluster method, the network is divided into 3 clusters. Each interface with its descendants represents a cluster. We allocate a different channel for each cluster. All nodes belonging to the same cluster use CSMA/CA to compete for channel access on the same channel. This method is the equivalent of 3 sub-networks using CSMA/CA without FDMA each sub-network using an orthogonal channel.

4.3. Aggregate Throughput

Figures 7 and 8 present the results in terms of aggregate throughput. For HMC-MAC protocol, it is clear that when the number of generated packets increases the aggregate throughput increases. All other protocols reach saturation (Figure 7) or rise slowly (Figure 8) when the number of generated packets becomes greater than 8 pkts/sec/node due to the high number of collisions and frame repetitions. For the cluster method, the number of packets collected by the NC does not exceed 100 packets/sec and reaches saturation when the number of generated packets exceeds 4 pkts/sec/node.

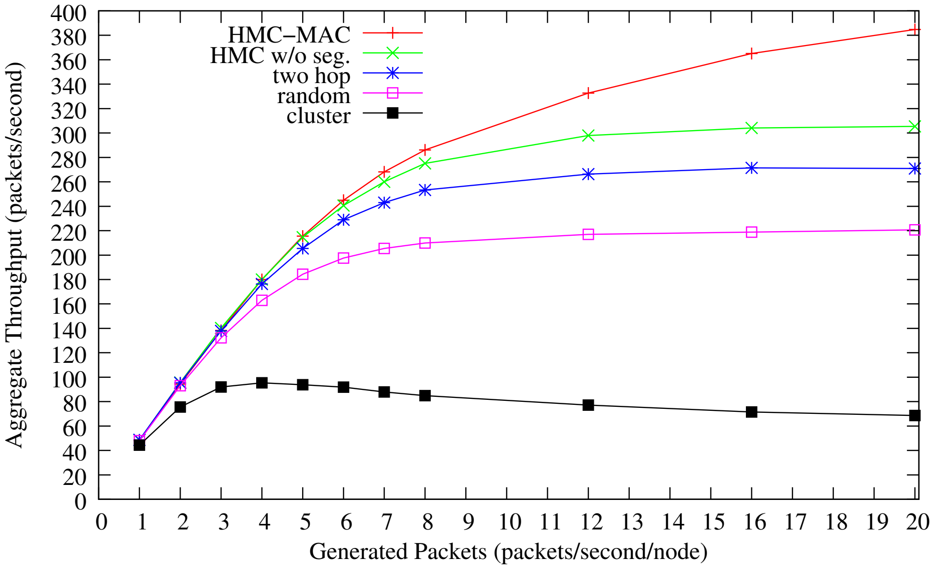

Figure 7. Aggregate throughput for periodic traffic.

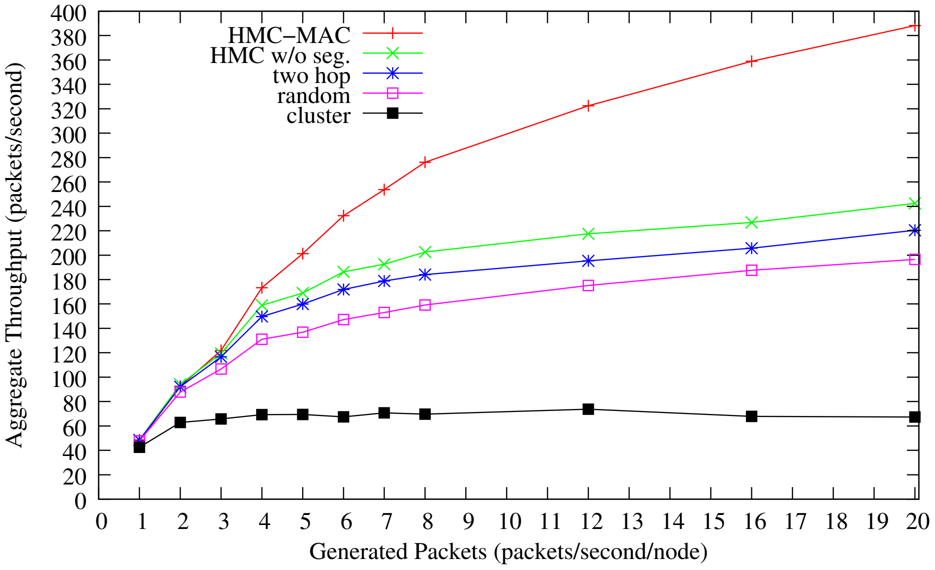

Figure 8. Aggregate throughput for burst traffic.

The performances of our proposition are close to those obtained with HMC without segmentation, for low traffic rates (under 6 pkts/sec/node for periodic traffic scenario and under 3 pkts/sec/node for burst traffic scenario). On the other hand, for higher traffic rates HMC-MAC copes much better than the other protocols. This is due to the fact that HMC-MAC divides the network into two groups. It always takes into consideration the channel usage in the groups before choosing a channel. This means, with HMC-MAC we have 16 channels for each group, whereas in HMC without segmentation and the other methods we have 16 channels for all the nodes. In addition, segmenting the network and defining the receivers during each interval helps HMC-MAC to avoid sending frames for nodes that are not in reception mode on the right channel. This leads to higher packet delivery rate with HMC-MAC.

By comparing the periodic and the burst traffic generations, we note that HMC-MAC outperforms other protocols even more in burst traffic generation. For example, with 12 pkts/sec/node, when we use a periodic traffic generation, HMC-MAC increases the received packets at the sink by around 76.8%, 34.7%, 20% and 10.4% compared to cluster method, random method, 2-hop method and HMC without segmentation method, respectively. When we use a burst traffic generation, the number of packets received by the sink is increased by almost 77%, 45.7%, 39.4% and 32.5% respectively. This is due to the same reasons stated earlier and shows that HMC-MAC is able to better manage the access to the medium under high burst traffic than other protocols that suffer more from packet loss due to lack of rendez-vous between senders and receivers.

We can also notice that HMC without segmentation outperforms the 2-hop method and that 2-hop method is more efficient than the random method. This is due to the fact that 2-hop does not take into consideration the channel usage in its 3-hop neighborhood and suffer form the type of collisions explained in 3.2. It is also important to note that we are using a slotted CSMA/CA algorithm in all the evaluated methods. Slotted CSMA/CA is a variant of non-persistent CSMA/CA, and this is why we reach a saturation point and not a decrease point.

4.4. Packet Delivery Ratio

Figures 9 and 10 show the results in terms of delivery ratios at the sink node. Similarly, HMC-MAC achieves better delivery ratios than other protocols during high data rate and burst traffic.

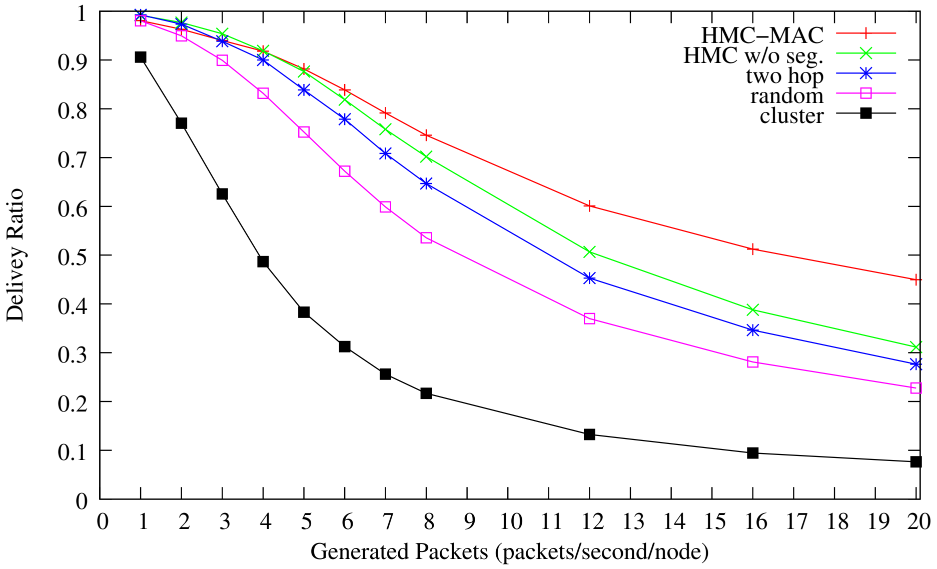

Figure 9. Delivery ratio at the sink for periodic traffic.

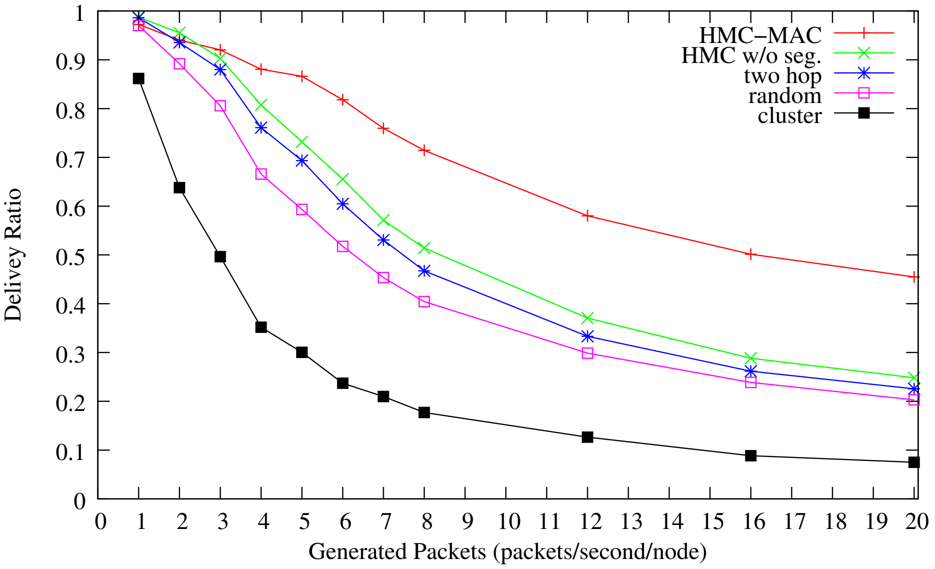

Figure 10. Delivery ratio at the sink for burst traffic.

4.5. Queue Overflow

Figures 11 and 12 show the results in terms of packet loss due to the queue overflow. We observe high rate of queue overflow in HMC-MAC protocol because of the limited capacity of the packet queues especially on the nodes situated near the sink. Indeed, HMC-MAC is able to route packets with much less loss on the medium than the other protocols. This leads to accumulation of packets in the nodes that are situated near the sink and that have a high number of descendants. These nodes are thus unable to transmit these accumulated traffic and suffer from high rates of queue overflow. This phenomena is also known as the funneling effect.

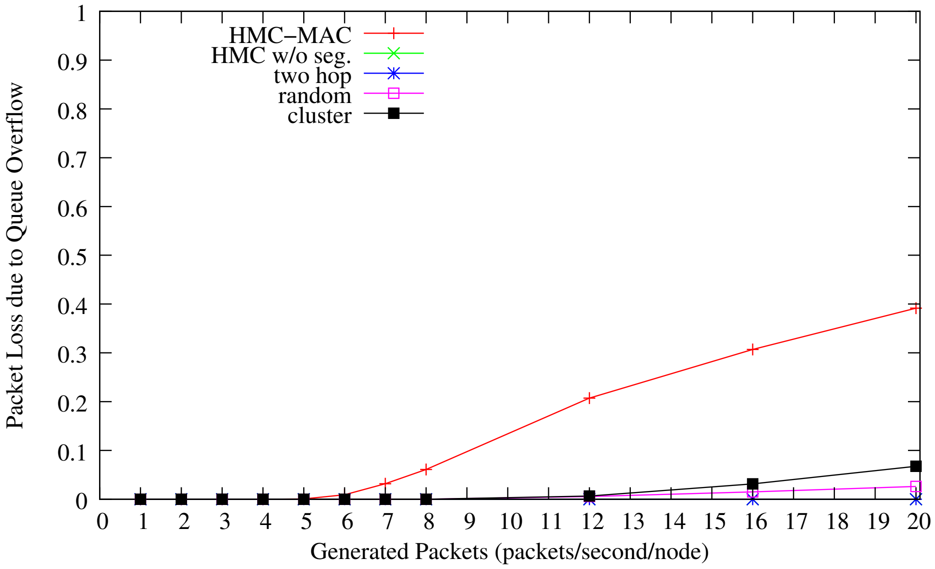

Figure 11. Packet loss due to queue overflow for periodic traffic.

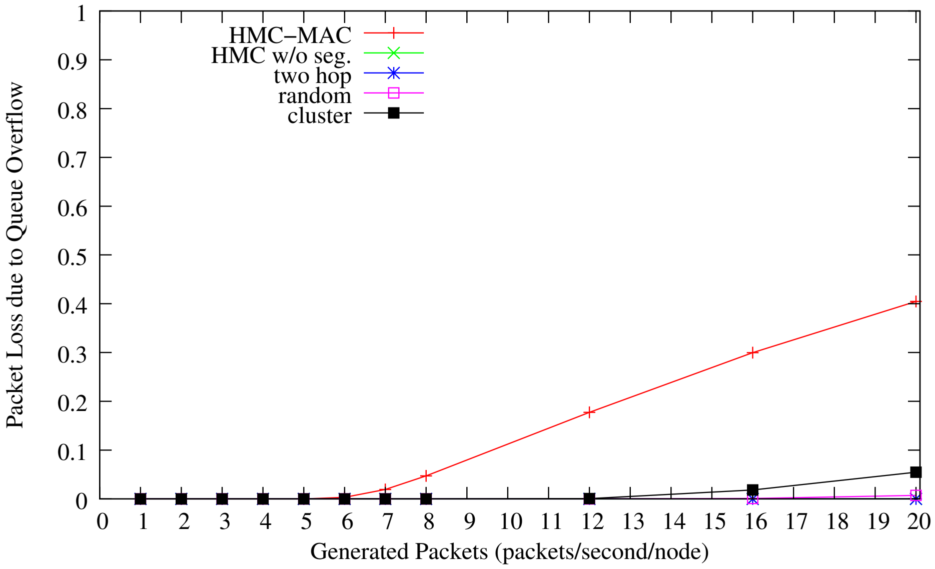

Figure 12. Packet loss due to queue overflow for burst traffic.

4.6. Number of Repetitions

Figures 13 and 14 show the results in terms of number of repetitions (including collisions and packet loss). Results show that when the number of generated packets increases the number of collisions increases as well.

These results show, as stated earlier, high repetition rate for the protocols that do not apply a rendez-vous mechanism. Indeed, when a node sends a packet to a node which is trying to send a packet to another node on a different channel, this packet will be lost because the receiver is currently busy on a different channel. HMC-MAC protocol does not suffer from that problem, and thus, reduces the number of frame repetitions compared to the other methods. Consequently, HMC-MAC avoids wasting energy by reducing the number of repetitions.

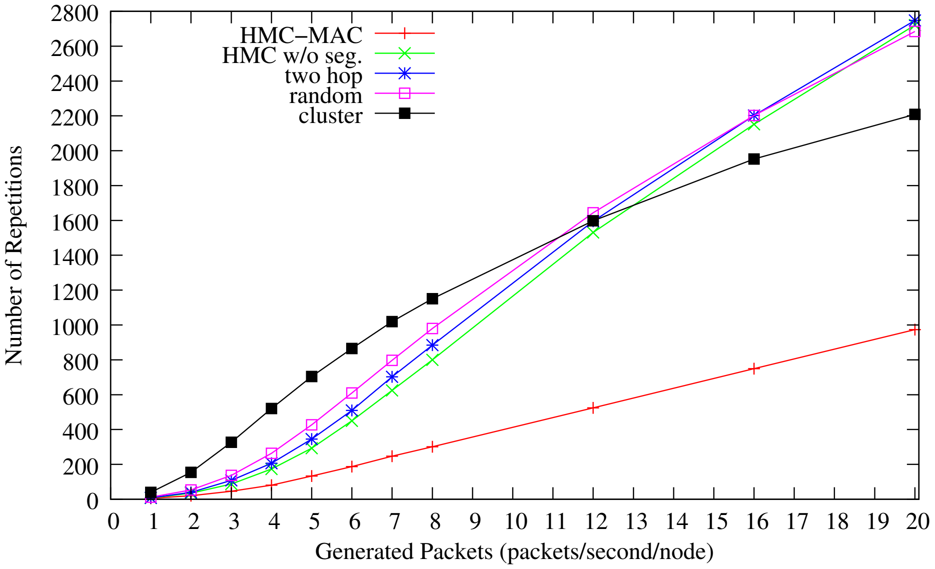

Figure 13. Number of repetitions for periodic traffic.

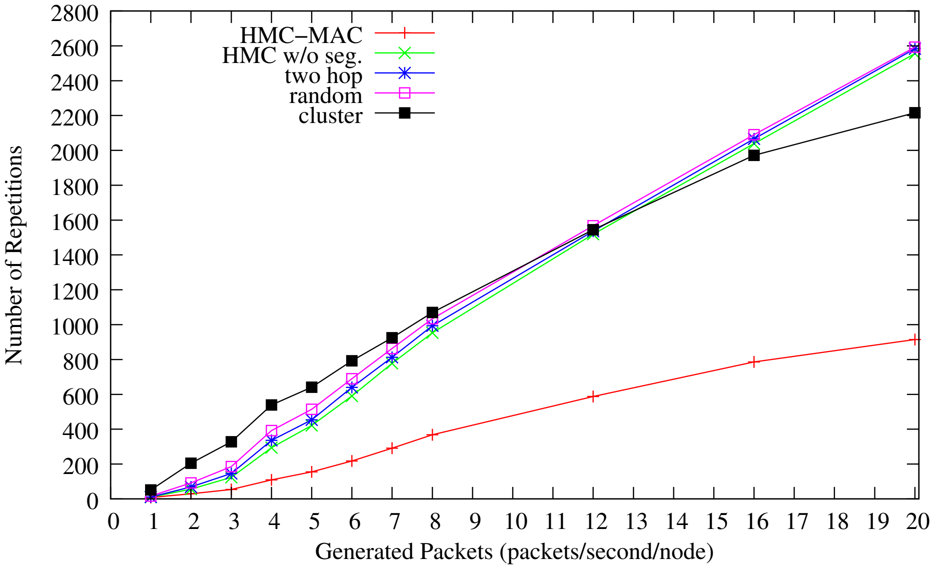

Figure 14. Number of repetitions for burst traffic.

4.7. Number of Dropped Packets

Figures 15 and 16 present the results in terms of number of dropped packets. According to the CSMA/CA algorithm of the IEEE 802.15.4 standard [21], the frame is repeated four times and is dropped if it is not acknowledged or not sent after the fourth repetition.

For higher traffic rates, we obtain a higher dropped packet rates as we have more packet loss and more repetitions. For periodic traffic generation, we note that HMC-MAC reduces the number of dropped packets compared to other protocols. For example, for 16 pkts/sec/node, when we use a periodic traffic generation, HMC-MAC reduces the number of dropped packets by around 77%, 68%, 64% and 60% of that of cluster method, random method, 2-hop method and HMC without segmentation method, respectively. Results are substantially similar in case of burst traffic.

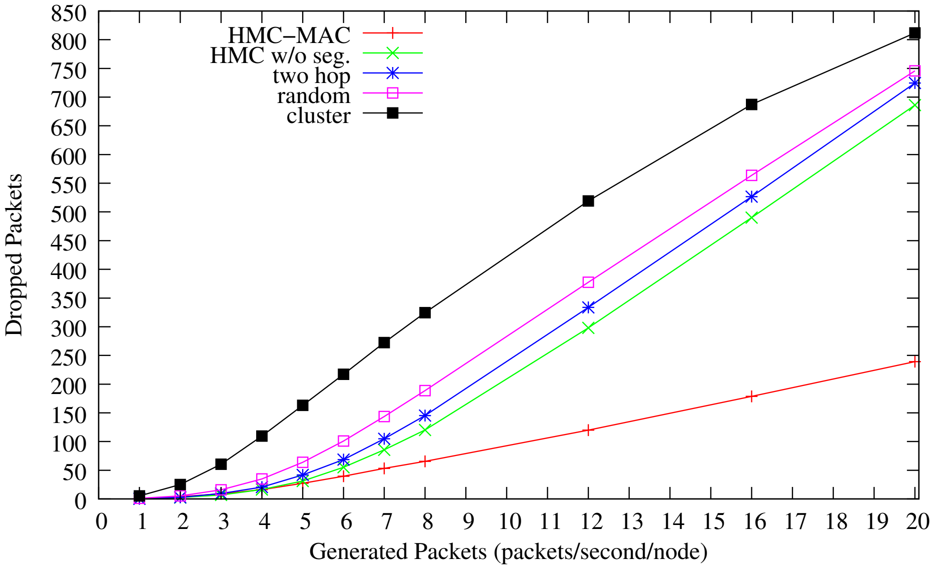

Figure 15. Number of dropped packets for periodic traffic.

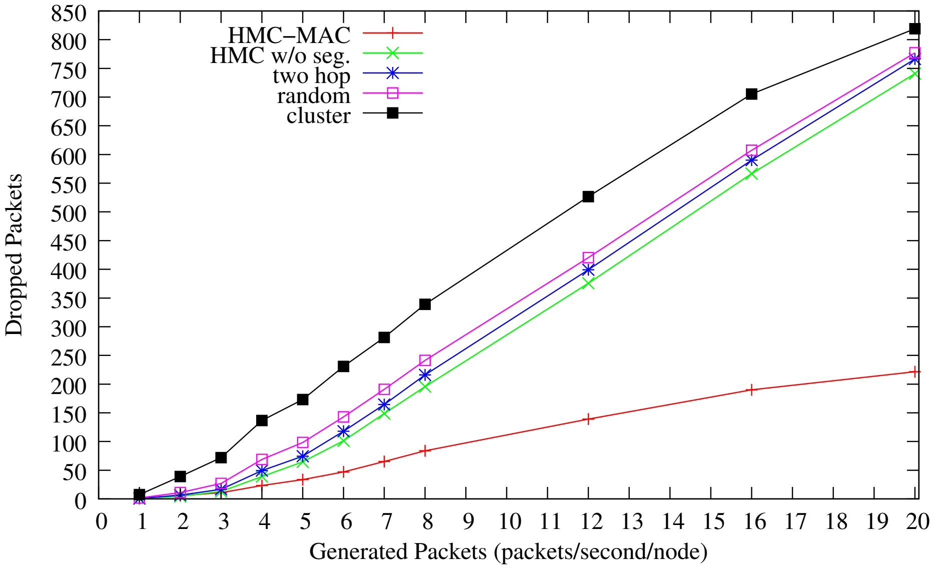

Figure 16. Number of dropped packets for burst traffic.

5. Conclusion and Perspectives

In this paper, we evaluated under high data rates the performance of HMC-MAC, a hybrid multi-channel MAC protocol for WSNs. It is based on 3-hop neighborhood channel allocation to avoid interference when data frame acknowledgements are used. We evaluated by simulation the efficiency of HMC-MAC and compared it to other MAC protocols. We considered the packet delivery rate, the number of collisions, frame repetitions, and dropped packets ratio as the main metrics. Results showed how HMC-MAC considerably reduces the number of collisions and frame repetitions compared to the other methods, and this in two different traffic scenarios and different random network topologies. HMC-MAC copes much better than the other protocols when we increase the offered load, this is even more obvious when we use a burst traffic generation.

In the current version of HMC-MAC, the beacon propagation period is done in a pure TDMA fashion where each node has its own slot for broadcasting the beacon frame. In our future work, we plan on optimizing this period by reducing it using a mix of TDMA and FDMA based on the neighborhood discovery information. Also, we plan on optimizing the frequency of this period in order to limit the control traffic overhead. A trade-off should be found between the clock drift for maintaining synchronization between nodes, the dynamic aspect of the network in order to take into account new nodes joining the network and nodes leaving the network, and the control traffic overhead.

Managing node failure in the current version of the protocol is not yet considered. One of the enhancements for HMC-MAC would be to consider redundant links to maintain connectivity and network activity in case of a node failure.

In this paper, we did not optimize the routing protocol in order to achieve load balancing in the network. Hence, we observed a high rate of queue overflow because of the limited storage capacity of the packet queues especially on the nodes located near the sink. In our future works, we will investigate this point and try to achieve a better traffic load balancing to avoid queue overflow near the sink.

Author Contributions

All three authors have equally contributed to this paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. EkbataniFard, G.; Monsefi, R. A detailed review of multi-channel medium access control protocols for wireless sensor networks. *Int. J. Wirel. Inf. Netw.* **2012**, *19*, 1–21.
2. Incel, O.D. A survey on multi-channel communication in wireless sensor networks. *Comput. Netw.* **2011**, *55*, 3081–3099.
3. Diab, R.; Chalhoub, G.; Misson, M. Overview on Multi-Channel Communications in Wireless Sensor Networks. *Netw. Protoc. Algorithms Surv. Tutor.* **2013**, *5*, 1943–3581.
4. Jayasuriya, A.; Perreau, S.; Dadej, A.; Gordon, S. Hidden *vs.* Exposed Terminal Problem in Ad hoc Networks. In Proceedings of the Australian Telecommunication Networks and Applications Conference, Sydney, Australia, 8–10 December 2004.
5. Diab, R.; Chalhoub, G.; Misson, M. Hybrid Multi-Channel MAC Protocol for Wireless Sensor Networks: Interference Rate Evaluation. In Proceedings of the IEEE 78th Vehicular Technology Conference: VTC2013-Fall, Las Vegas, NV, USA, 2–5 September 2013.
6. Lohier, S.; Rachedi, A.; Salhi, I.; Livolant, E. Multichannel Access for Bandwidth Improvement in IEEE 802.15.4 Wireless Sensor Network. In Proceedings of the 2011 IFIP/IEEE Wireless Days, Niagra Falls, ON, Canada, 10–12 October 2012.
7. Priya, B.; Manohar, S.S. EE-MAC: Energy Efficient Hybrid MAC for WSN. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*, doi:10.1155/2013/526383.
8. Polastre, J.; Hill, J.; Culler, D. Versatile Low Power Media Access for Wireless Sensor Networks. In Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD, USA, 3–5 November 2004.
9. IEEE 802.15. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*; Standard 802.11; ANSI/IEEE: Piscataway, NJ, USA, 2005.
10. Salajegheh, M.; Soroush, H.; Kalis, A. HYMAC: Hybrid TDMA/FDMA Medium Access Control Protocol for Wireless Sensor Networks. In Proceedings of the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007), Athens, Greece, 3–7 September 2007.
11. Zhou, G.; Huang, C.; Yan, T.; He, T.; Stankovic, J.A. MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks. In Proceedings of the IEEE Infocom, Barcelona, Spain, 23–29 April 2006.
12. Incel, O.D.; Jansen, P.; Mullender, S. *MC-LMAC: A Multi-Channel MAC Protocol for Wireless Sensor Networks*; Technical Report TR-CTIT-08-61; Centre for Telematics and Information Technology, University of Twente: Enschede, The Netherlands, 2008.

13. Hoesel, L.; Havinga, P. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches. In Proceedings of the 1st International Workshop on Networked Sensing Systems (INSS 2004), Tokyo, Japan, 22–23 June 2004.
14. Borms, J.; Steenhaut, K.; Lemmens, B. Low-Overhead Dynamic Multi-channel MAC for Wireless Sensor Networks. In Proceedings of 7th European Conference on Wireless Sensor Networks (EWSN), Coimbra, Portugal, 17–19 February 2010.
15. Buettner, M.; Yee, G.; Anderson, E.; Han, R. X-MAC: A Short Preamble MAC Protocol For Duty-CycledWireless Sensor Networks. In Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), Boulder, CO, USA, 31 October–3 November 2006.
16. Wu, Y.; Stankovic, J.A.; He, T.; Lin, S. Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks. In Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM 2008), Phoenix, AZ, USA, 13–18 April 2008.
17. Kim, Y.; Shin, H.; Cha, H. Y-MAC: An Energy-efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks. In Proceedings of the International Conference on Information Processing in Sensor Networks, St. Louis, MO, USA, 22–24 April 2008.
18. Diab, R.; Chalhoub, G.; Misson, M. Evaluation of a hybrid multi-channel MAC protocol for periodic and burst traffic. In Proceedings of the IEEE 39th Local Computer Networks: LCN2014, Edmonton, AB, Canada, 8–11 September 2014.
19. Chalhoub, G.; Misson, M. Cluster-tree based energy efficient protocol for wireless sensor networks. In Proceedings of the IEEE International Conference on Networking, Sensing and Control, Chicago, IL, USA, 10–12 April 2010.
20. Diab, R.; Chalhoub, G.; Misson, M. Channel Allocation Evaluation for a Multi-channel MAC Protocol. In Proceedings of the IEEE International Symposium on Personal Indoor and Mobile Radio Communications, PIMRC, London, UK, 8–11 September 2013.
21. IEEE 802.15. *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*; Standard 802.15.4 R2006; ANSI/IEEE: Piscataway, NJ, USA, 2006.
22. Zigbee. *Zigbee Specification*; Standard Zigbee 053474r13; ZigBee Standards Organization: Davis, CA, USA, 2006.