A hybrid multi-channel MAC protocol for wireless ad hoc networks

Duc Ngoc Minh Dang · Choong Seon Hong · Sungwon Lee

Published online: 22 August 2014
© Springer Science+Business Media New York 2014

Abstract In a regular wireless ad hoc network, the Medium Access Control (MAC) protocol coordinates channel access among nodes, and the throughput of the network is limited by the bandwidth of a single channel. The multi-channel MAC protocols can exploit multiple channels to achieve high network throughput by enabling more concurrent transmissions. In this paper, we propose a hybrid and adaptive protocol, called H-MMAC, which utilizes multi-channel resources more efficiently than other multi-channel MAC protocols. The main idea is to adopt the IEEE 802.11 Power Saving Mechanism and to allow nodes to transmit data packets while other nodes try to negotiate the data channel during the Ad hoc Traffic Indication Message window based on the network traffic load. The analytical and simulation results show that the proposed H-MMAC protocol improves the network performance significantly in terms of the aggregate throughput, average delay, fairness and energy efficiency.

Keywords Multi-channel · MAC protocol · Ad hoc networks

1 Introduction

The popular IEEE 802.11 standard [1] provides a Medium Access Control (MAC) protocol called the Distributed Coordination Function (DCF). To avoid the hidden terminal problem, IEEE 802.11 DCF uses the channel reservation technique by exchanging “Request to Send” (RTS) and “Clear to Send” (CTS) packets before any data packets are sent. In addition to the RTS–CTS–DATA–ACK handshake, IEEE 802.11 DCF employs a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) technique, where each node always senses the channel before any transmission. If the channel is busy, the node waits until the channel is idle and then goes through a random back-off time before retrying. The back-off procedures reduce the probability of collisions when nodes try to send RTS packets and ensure fairness among contending transmissions.

There are three non-overlapping channels in IEEE 802.11b and g, and twelve non-overlapping channels in IEEE 802.11a. Although IEEE 802.11 provides multiple channels for wireless communications at the physical layer, the MAC protocol is designed only for a single channel. If multiple channels are exploited, many concurrent transmissions can take place without interference. Thus, we can achieve a higher network throughput than using a single channel. However, it is difficult to design a MAC protocol that can exploit multiple channels with a single half-duplex transceiver. Although the transceiver can switch the channel radio dynamically, it cannot sense all channels simultaneously, and it may lose the channel reservation packets from its neighbors on other channels. This leads to a new type of hidden terminal problem in a multi-channel environment, which we refer to as the multi-channel hidden terminal problem [11].
The rest of this paper is organized as follows. Section 2 reviews related work on Multi-channel MAC protocols in ad hoc networks. And the IEEE 802.11 Power Saving Mechanism (PSM) is briefly reviewed in Sect. 3. Our proposed protocol is described in detail in Sect. 4. Then, the throughput analysis based on the Markov chain model is given in Sect. 5. We compare the proposed H-MMAC protocol with other protocols through extensive simulations and discuss the simulation results in Sect. 6. Finally, we conclude this paper in Sect. 7.

2 Related work

The major challenges of multi-channel transmissions are negotiating a data channel while also decreasing the collisions and the control packet overhead. Recently, many multi-channel MAC protocols have been proposed, and they can be classified into the following categories [2]: Dedicated Control Channel, Common Hopping, Split Phase, and Parallel Rendezvous.

In the Dedicated Control Channel approach [3–10], each node has two radios. One radio is tuned to the channel dedicated to the control packets and another can switch to any other channels for the data transmissions. When a sender has data for a receiver, it transmits an RTS packet (Fig. 1a). The receiver responds with a CTS packet on the control channel to confirm the data channel that was suggested by the sender. Since all nodes listen on the control channel all the time, each node can keep track of the status of the other nodes and data channels. One representative of this approach is Dynamic Channel Allocation (DCA) [3]. The performance of DCA is affected by the ratio of $T_s/T_c$, where $T_s$ and $T_c$ are the transmission duration of the data packets and the control packets, respectively. In [4], a Transmission Opportunity (TXOP) strategy is proposed to solve the control channel saturation problem. By bringing a power control scheme to Multi-channel MAC, Dynamic Channel Assignment with Power Control (DCA-PC) [5] and Symmetric Power Control (Symmetric TPC) [7] are improvements of DCA. In DCA-PC, all control packets RTS/CTS/RES are transmitted using the maximum power on the control channel, while the DATA/ACK packets are transmitted with the minimum required power on the data channel. The Signal to Noise Ratio-based multi-channel MAC protocol (SNRMP) [8] is based on a criterion proposed for the receiver to choose the “best” data channel in terms of the lowest interference power. Multi-channel MAC with Hopping Reservation (MMAC-HR) [9], another improvement of DCA, resolves the multi-channel exposed terminal problem. This approach does not require time synchronization but requires two radios, thereby increasing the cost of the device as well as increasing the power consumption. By using a single transceiver, Asynchronous Multi-channel MAC (AM-MAC) [10] targets the low-cost and low-power objectives, while guaranteeing no collisions.

However, each node has only one radio in the Split Phase approach [11–20]. Time is divided into an alternating sequence of a control interval or contention interval and a data transmission interval, as shown in Fig. 1(b). During the control interval, all nodes have to switch to the control channel and try to reserve data channels to be used in the data transmission interval. In the next interval, both the sender and receiver tune to the agreed data channel for the data transmissions. MMAC [11] is an example of this approach. E-MMAC [12] applies a power control algorithm to MMAC in order to exploit multiple channels and improve frequency reuse. One of the disadvantages of both the MMAC and E-MMAC protocols is the waste of the data channel’s bandwidth in the control interval when all nodes are on the control channel while the other data channels are free. Therefore, the duration of the control interval affects the network performance. Traffic Aware Multi-channel MAC (TAMMAC) [13] and TDMA-based Multi-channel MAC (TMMC) [14] involve a mechanism

Fig. 1 Four approaches for the multi-channel MAC protocol [2]. a Dedicated control channel approach, b split phase approach, c common hopping approach, d parallel rendezvous: McMPC

© Springer
to adjust the control interval dynamically according to the traffic on the network. However, TMMAC achieves more
power savings by allowing nodes that are not involved in the data transmissions to go into doze mode. ECU-
MAC [15] tries to utilize the channel resources during the Ad hoc Traffic Indication Message (ATIM) window by
allowing nodes which have the first successful negotiation for each channel start the transmission immediately.
Multi-channel MAC with Channel Distribution (CD-MAC) [16] is another scheme that attempts to utilize the channel
during the control interval. The control interval is divided
into two windows. The first window is used for sender-
receiver pairs to negotiate which channel can be used to
compete for the final data channel. Then, the sender-
receiver pairs can compete to obtain a channel to transfer
packets in the second window. This scheme can prevent the
collisions during the control interval, unlike MMAC. In our
previous work [17], the proposed protocol can exploit
multiple channels more efficiently, but cannot adapt to the
network traffic load. The Multi-channel MAC protocol
[18], which provides QoS for ad hoc network, has two
regions in the control interval. In the first region, the real-
time traffic users contend with higher priority than the data
traffic users, and at the second region, all users contend
together with the same priority. Novel Multi-channel MAC
(Novel) [19] combines the advantages of using a common
control channel and a common control period. It employs
two transceivers and employs both the Split Phase and
Dedicated Control Channel approaches. Both transceivers
can be used to negotiate the data channel like MMAC in
the first phase, and then they behave like DCA in the
second phase.

In the Common Hopping approach [21, 22], the nodes
also have one radio. Nodes that do not exchange data hop
through all channels synchronously. As shown in Fig. 1(c),
the hopping pattern cycles through channels 1, 2 and 3.
Now, all nodes are on channel 1, and if a sender wants to
send data packets to a receiver, it sends an RTS packet to
the receiver on the current common channel. If the receiver
receives the RTS properly, it replies with a CTS packet on
the same channel. Both the sender and receiver then stop
hopping and remain on channel 1 during the data trans-
mission, whereas the other idle nodes continue hopping to
channel 2. After the data transmission, the sender and
receiver rejoin the Common Hopping sequence with the
other idle nodes. This approach requires all nodes to hop
frequently and requires tight time synchronization.
According to IEEE 802.11, the required time to switch
channel is 224 μs, so the channel switching time cannot be
negligible. Channel Hopping Multiple Access (CHMA) [21]
is an example of this approach. Pipelining Multi-channel
MAC (x-Mc) [22] is similar to the pipeline technique. The
transmission task is divided into many subtasks. All of the
subtasks are transmitted on different channels sequentially.
All nodes that have data packets to transmit should contend
to transmit the first subtask on the first channel. If the first
subtask is transmitted successfully, all of the other subtasks
also succeed in being transmitted.

To make multiple agreements at the same time and
resolve the congestion on the common channel, Parallel
Rendezvous protocols [23–25] are proposed. Both Slotted
Seeded Channel Hopping (SSCH) [23] and Multi-channel
MAC (McMAC) [24] protocols require only one radio. Each
node hops independently among channels. A sender needs to
synchronize with a receiver in order to transmit a packet. As
a result, they suffer from the missing receiver problem [6].
Figure 1(d) shows the operation of McMAC. Each node
picks a seed to generate a different pseudo random hopping
sequence. An idle node follows its “home” hopping
sequence. Each node puts its seed in every packet that it
sends, so its neighbors eventually learn its hopping
sequence. When a sender has data packet to send, it flips a
coin and transmits with some probability p during each time
slot. If it decides to transmit, it tunes to the current channel
of the receiver and sends an RTS packet. If the receiver replies
with aCTS packet, both the sender and receiver stop hop-
ing to exchange data packets. The data transmission may
take place over some time slots. After the data transmission
is complete, the sender and receiver return to their original
hopping sequences. Another protocol based on the fre-
cquency hopping and parallel rendezvous approaches is Fast
and Slow Hopping MAC protocol [25]. Each node has two
radios. One radio follows a fast hopping while the second
radio follows a slow hopping.

Another challenge is the limit in power capacity of bat-
tery-powered wireless nodes. IEEE 802.11 PSM is used to
conserve energy for ad hoc networks by allowing nodes to
enter doze mode when there is no need for data exchange.
Since MMAC is based on IEEE 802.11 PSM, one of its advan-
tages is its power saving capability. DCA employs two
radios, so that while one radio is on the data channel for the
data transmissions, another radio can negotiate on the con-
trol channel. We propose the H-MMAC protocol by com-
paring the advantages of both DCA and MMAC protocols.
The H-MMAC protocol is an improvement of the MMAC
protocol in that the H-MMAC protocol allows some nodes to
transmit data packets during the ATIM window in high
network load. Each node maintains the data structures to
keep track of the status of its neighbors and to balance the
network load on each data channel. Comparing to MMAC,
H-MMAC has some advantages:

- Nodes choose the transmission mode adaptively (either
  Normal or Extended transmission mode) according to
  the network condition in order to achieve the high
  network performance.
In high network load, all data channels are utilized for the data transmissions during the ATIM window. The throughput improvement of H-MMAC compared to MMAC is

\[
\text{Improvement} = \left(\frac{N_{ch} - 1}{N_{ch}}\right) \frac{\text{ATIM\_window\_size}}{\text{Data\_window\_size}}
\]

(1)

Since some nodes might be on the data channels during the ATIM window, H-MMAC has fewer number of nodes contending the default channel to exchange ATIM packets than MMAC. Therefore, H-MMAC has higher success probability of the ATIM negotiations on the default channel than MMAC.

3 IEEE 802.11 PSM

In IEEE 802.11 PSM, an Ad hoc Traffic Indication Message (ATIM) is used for power management. Figure 2 illustrates the operation of IEEE 802.11 PSM. Time is divided into beacon intervals, and all nodes are synchronized by periodic beacon transmissions. IEEE 802.11 PSM uses a short interval called the ATIM window at the start of each beacon interval. All nodes have to awake for the ATIM window. The sender and receiver exchange ATIM-Request/ATIM-Acknowledgement in the ATIM window. After the ATIM window, both the sender and receiver exchange DATA/ACK packets, while the nodes that do not have packets to send or receive go into doze mode in order to save energy. In doze mode, a node consumes much less energy compared to idle mode, but it cannot send or receive packets.

Our proposed H-MMAC protocol adopts the IEEE 802.11 PSM. During the ATIM window, nodes contend to exchange ATIM packets to negotiate the data channel. After that, nodes switch to the agreed data channel for the data transmissions. The details of the H-MMAC protocol are described in the following sections.

4 Proposed H-MMAC protocol

First, we summarize our assumptions as follows:

- There are \(N_{ch}\) non-overlapping channels that can be used. A beacon interval is divided into two sub-intervals: the ATIM window and the data window. One channel is defined as a default channel (CH1) just in the ATIM window. The default channel is used to transfer data packets like other channels outside the ATIM window.
- Nodes have prior knowledge of how many channels are available in the network.
- Each node has a single half-duplex transceiver such that it can either transmit or listen but cannot do both simultaneously. The transceiver is capable of switching the channel dynamically.
- All nodes are time-synchronized and operate IEEE 802.11 DCF mechanism. The clock synchronization can be achieved by using Global Positioning System (GPS) or IEEE 802.11 Timing Synchronization Mechanism (TFS) [1]. In addition, several clock synchronization protocols have been proposed in [26–28]. The synchronization overhead is small and the maximum clock offset can be achieved as 15 \(\mu s\) [28].

Next, we define two transmission modes (Tx mode):

- Normal Transmission (N-Tx): the transmission performed within the data window.
- Extended Transmission (E-Tx): the transmission performed within the data window and the next ATIM window. It can be extended to the next data window when the other nodes do not use this channel. The E-Tx is longer than the N-Tx.

Figure 3 shows an example of the transmission modes. Since the ATIM window on the channel 1 is used for nodes to exchange ATIM packets, only Normal transmission mode can be used on this channel. The Extended transmission mode is from the beginning of the current data window to the end of the next ATIM window or the end of the next beacon. Normally, the Extended transmission finishes at the end of the next ATIM window (Extended Transmission on channel 2 in Fig. 3). However, the Extended transmission is extended to one more data window only if there is no node using this channel in order to keep the fairness among nodes in the network (Extended transmission on channel 3 in Fig. 3). Node has to sense the channel idle for certain time to make sure that the channel
is not used by other nodes in this data window. If the node exchanged the control packets successfully, the contention window is set to minimum value $CW_{\text{min}}$. So, the Extended transmission nodes have to sense the channel and if the channel is idle at least $T_{\text{channel-switching}} + DIFS + CW_{\text{min}} + Slot\_time$, they can continue their transmissions.

There are four node’s types during the ATIM window: Normal, Ongoing, Limited and Unknown. These types help a node to keep track of the status of its neighbor nodes.

- Normal: the node which is on the default channel during the current ATIM can have up-to-date information about its neighbor nodes.
- Ongoing: the node which is exchanging data during the current ATIM window because of its E-Tx mode.
- Limited: the node that lost its neighbors status because it was Ongoing in the last beacon.
- Unknown: if a node does not know about its neighbor, then the neighbor node is Unknown node.

A node itself is in either Normal or Ongoing type. Normal means that the node is on the default channel during the ATIM window, whereas Ongoing indicates that the node is exchanging data on the data channel. In the point of view of a node, a neighbor node of its can be in one of the four types above. In the current ATIM window, if the neighbor is in Ongoing type, it will be in Limited type in the next ATIM window and in Normal type in the next next ATIM window. If a node itself is in Ongoing type in the current ATIM window, it does not overhear the ATIM packets on the default channel, so it considers all Normal neighbor nodes as Unknown nodes.

4.1 Neighbor information list and preferable channel list

Each node maintains its data structures, which are called the Neighbor Information List (NIL) and the Preferable Channel List (PCL). The NIL stores the information about the neighbor nodes such as: channel, type and transmission mode, while the PCL stores the information about the channels such as: the state and counter. The counter of a channel shows how many node pairs already reserved that channel.

4.1.1 Neighbor information list—NIL

Each list entry keeps the information about the neighbor nodes including channel (CHL), type and Tx mode. The channel field shows which channel the neighbor node is going to switch to in the data window. An idle node has channel 0. There are four types of neighbor nodes: Normal, Ongoing, Limited, and Unknown, and two Tx modes: Normal (N-Tx) and Extended (E-Tx). Table 1 shows the NIL of node A at the end of the third ATIM window.

Figure 4 illustrates the type transition of the neighbor node that performed an E-Tx transmission. If the neighbor node uses E-Tx mode from beacon 1, its type is changed to Ongoing, Limited, and Normal in the ATIM window of beacons 2, 3, and 4, respectively in the NIL. Let us consider node A in Fig. 5. In the first beacon, node A is a Normal node and performs an E-Tx transmission. Node A is an Ongoing node which is exchanging data in the second ATIM window and node A misses the control packets from its neighbor nodes. The worst case is that node A misses the control packets from the nodes that are going to perform E-Tx mode, for example, nodes C and I. As a result of this, node A becomes a Limited node in the third ATIM window and it overhears the control packets to update its NIL. It is certain that nodes C and I are on the default channel in the fourth ATIM window, which means that node A can do a handshake with node C to transmit data. In other words, node A is a Normal node in the fourth ATIM window.

In the second ATIM window, node A does not hear the control packets exchanged by the nodes that are on the default channel. These neighbor nodes are updated as the Unknown type. However, node A is aware of the Ongoing nodes that are performing the data transmissions such as nodes E and F. In the third ATIM window, these Unknown nodes are still Unknown nodes until node A overhears the control packets from them and updates its NIL. In the fourth ATIM window, the remaining Unknown nodes are changed automatically to Normal nodes.
A node needs to know the current status of its neighbor nodes in order to exchange data. It is therefore very important for a node to keep its NIL updated. A node updates its type itself (Normal or Ongoing) and then updates its NIL before each beacon as shown in Table 2. For example, node A is a Normal node. If there is an Unknown node with N-Tx mode in its NIL, it updates the records of that node to Channel 0, Normal type, and N-Tx mode. Node A continues updating for all records of its NIL. During the ATIM window, whenever it overhears ATIM packets from a neighbor node, the channel and the Tx mode are updated to the corresponding transmission mode for that neighbor node's record in the NIL.

4.1.2 Preferable channel list—PCL

Each channel in the PCL is in either the Selected or Not_Selected state. Selected state means that the channel has already been chosen by a node in the current beacon; otherwise, it is a Not_Selected channel. There is at most one channel in Selected state at each node for each beacon. The counter of a channel shows how many node pairs have already reserved that channel, and helps to balance the channel load as much as possible. The data channel which has a Not_Selected state and zero counter is an idle channel. Table 3 shows node A’s PCL after it has finished exchanging ATIM packets with node B in the fourth ATIM window (Fig. 5).

The PCL is updated whenever the node overhears ATIM-ACK/ATIM-RES packets or when the node selects a channel to use in the data window.

- All of the channels are reset to the Not_Selected state at the start of each beacon interval.
- If the sender and receiver nodes reserve a channel to exchange data, this channel is changed to the Selected state in both the sender’s and receiver’s PCLs.
- When a node knows that a neighbor node of its plans to use a channel through the ATIM-ACK/ATIM-RES packets, it increases the counter of that channel by one.

4.2 Extended transmission

Figure 6 shows the performance comparisons of the different transmission modes. Normal mode is similar to MMAC, where the nodes are not allowed to exchange data during the ATIM window. Extended mode means that the nodes always perform E-Tx transmission mode, regardless of the network load. In Hybrid mode, the node decides which transmission mode is used based on the network traffic load: Normal mode in low network load and Extended mode in high network load. Figure 6(a, b) shows the network performance versus the packet arrival rate in the network of 12 channels and 40 nodes. And Fig. 6(c, d) shows the network performance versus the number of nodes in the network of 12 channels and packet arrival rate of 80 packets/s per flow. The other simulation parameters

<table>
<thead>
<tr>
<th>Table 2 NIL’s update procedure: update from the top to the bottom</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node A’s type</strong></td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
</tr>
</tbody>
</table>

Any Normal or Ongoing type
X Non-zero channel
Ongoing node that is not Ongoing type

 Springer
Table 3 Node A’s PCL.

<table>
<thead>
<tr>
<th>Channel</th>
<th>State</th>
<th>Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Not_Selected</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Selected</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Not_Selected</td>
<td>0</td>
</tr>
</tbody>
</table>

are given in Table 5 in Sect. 6. In both scenarios, the performance of the Hybrid mode is higher than the Extended and Normal mode cases. Let us consider Fig. 5 again. In the first beacon, nodes A, B, E, and F perform in E-Tx mode. They cannot transmit in the data window of the second beacon, although some channels are idle. They use a small ATIM window more, but they waste a large data window. As a result, the performance suffers a long delay and a low throughput. When the network load is low (few nodes or low packet arrival rate), nodes should not use Extended transmission mode. Therefore, we propose the Hybrid mode, which adapts to the network traffic load as follows.

The network traffic load depends on the number of nodes want to exchange data and the number of data packets that they are going to exchange. A sender observes the network traffic load and uses the following three criteria to select the transmission mode:

- First, the node uses the collision factor to identify the network density. The more nodes contend the default channel, the more collisions occur. When a collision occurs, the retry counter is increased. We use the E-Tx Retry Threshold as the threshold of the Extended transmission selection. If the retry counter is larger than E-Tx Retry Threshold, the sender will determine whether or not to use E-Tx mode based on how many packets in its buffer.
- Second, let $T_{DATA}$ be the time required for transmitting one data packet. Since the sender does not know which channel will be selected by the receiver, the sender finds the minimum counter $Counter_{min}$ in its PCL. The total number of sender–receiver pairs that are going to use this channel is equal to or greater than $Counter_{min} + 1$. With the fairness assumption, the sender decides to use E-Tx mode if the number of data packets in its buffer is larger than $Data\_window\_size / \left( (Counter_{min} + 1) T_{DATA} \right)$.
- Finally, we define the E-Tx Node Threshold to limit the number of E-Tx transmissions on each data channel, except the default channel. After the two above conditions are satisfied, the sender checks how many nodes selected the E-Tx transmissions on each channel based on its NIL. If there is a channel which has the number of E-Tx transmissions less than the E-Tx Node Threshold, the sender selects the E-Tx mode and sends ATIM packet.

Fig. 6 Performance of different transmission modes of the H-MMAC protocol. 
- Aggregated throughput 40 nodes, 12 channels, 
- Average delay 40 nodes, 12 channels, 
- Aggregate throughput 80 packets/s, 12 channels, 
- Average delay 80 packets, 12 channels.
to the receiver. After selecting the channel, the receiver also checks how many nodes selected E-Tx mode on the selected channel. And the receiver has a final decision about the transmission mode and sends ATIM-ACK packet with the selected channel and the transmission mode to the sender.

4.3 Control frames

For our proposed protocol, both the sender and receiver have to exchange ATIM/ATIM-ACK/ATIM-RES packets on the default channel in order to select an appropriate data channel and transmission mode for their data transmissions. We modify the ATIM frame format and define two ATIM-ACK/ATIM-RES frames as shown in Fig. 7. Assume that we have a maximum of 12 channels (IEEE 802.11a). We use 1 bit to represent the state of each channel, Selected or Not_Selected, and 3 bits for the counter of each channel. The Tx mode field specifies which transmission mode is used.

4.4 Operation of the H-MMAC protocol

In the ATIM window, there are four types of neighbor nodes: Ongoing, Limited, Normal, and Unknown. A node must be in either a Normal or Limited type in order to be a receiver.

1. When a node has data to send, it checks the receiver’s type in its NIL. If the receiver’s type is Ongoing or Unknown, it has to wait for the next beacon to try again.
2. The sender checks the receiver’s channel in its NIL and its channel in its PCL. If the Selected channels of the sender and receiver are different, the sender also has to wait for the next beacon.
3. Based on the traffic load, the sender suggests which transmission mode is used.

4. The sender attaches its PCL and transmission mode to the ATIM packet and sends it to the receiver.
5. Upon receiving the ATIM packet, the receiver selects the best channel from its PCL and the sender’s PCL by using Algorithm 1. It also has to make the final decision about the transmission mode. Then the receiver sends an ATIM-ACK packet indicating the selected channel and the transmission mode to the sender.
6. The sender sends an ATIM-RES packet to confirm the data channel and the transmission mode selected by the receiver.
7. After the ATIM window, the sender and receiver switch to the agreed channel to exchange data.
8. Both the sender and receiver go to doze mode after finishing their data transmissions, and wake up again at the beginning of the next ATIM window.

Algorithm 1 Algorithm to select the "best" channel

if There is a Selected channel in the destination’s PCL then
  This channel is selected.
else if There is a Selected channel in the source’s PCL then
  This channel is selected.
else
  The channel with the least counter value is selected.

end if

Figure 5 shows an example of the operation of the H-MMAC protocol. (X−Y) and (X′−Y) denote the sender X and the receiver Y using N-Tx and E-Tx mode, respectively. In the first beacon, nodes A, B, E, and F use E-Tx mode. In the second ATIM window, nodes A, B, E, and F are Ongoing nodes, therefore they lose the ATIM packets from their neighbor nodes. In the third ATIM window, nodes A, B, E, and F become Limited nodes, which have limited information about their neighbor nodes. Node A only knows about nodes B, E, and F, and it does not have any information about the other nodes. Node D is an Unknown node in node E’s NIL. After node E overhears the ATIM packet from node D (nodes D and G exchange ATIM packets), it knows that node D is now on the default channel and updates the information about node D in its NIL. If node E has data packets for node D, it can exchange ATIM packets with node D. Then nodes E, D, and G switch to channel 2 for the data transmissions. Since node F knows about node B’s status, node F can do handshake with node B to exchange data in the third beacon. Node A lost the control packets of the other nodes from the second ATIM.
All $b_{i,k}$ are expressed in terms of $b_{0,0}$ which is determined as follows

$$1 = \sum_{i=0}^{m} \sum_{k=0}^{W_i-1} b_{i,k} = \sum_{i=0}^{m} b_{0,0} \frac{W_i - k}{W_i} = \sum_{i=0}^{m} \frac{b_{0,0} W_i + 1}{2}$$

(5)

$$b_{0,0} = \frac{2(1-p)(1-2p)}{(1-2p)(1+W_0) + p W_0 (1-(2p)^m)}$$

(6)

As a packet is transmitted when the back-off counter is zero, regardless of the back-off stage, the probability $\tau$ that node transmits in a time slot is

$$\tau = \sum_{i=0}^{m} b_{0,0} \frac{b_{0,0}}{1-p} \frac{2(1-p)(1-2p)}{(1-2p)(1+W_0) + p W_0 (1-(2p)^m)}$$

(7)

The conditional collision probability $p$ is the probability that more than one node transmits in a time slot and can be expressed as

$$p = 1 - (1-\tau)^n-1$$

(8)

Consequently, based on the Eqs. (7) and (8), the probability $\tau$ of transmitting in a randomly chosen time slot is solved using the numerical techniques. The Eqs. (7) and (8) are applied to calculate the probability $\tau^a$ and $\tau^d$ for two contention periods in both the ATIM window and the data window, respectively.

Now, we consider the contention model of the ATIM window as shown in Fig. 9. In the MMAC and H-MMAC protocols, when the sender makes the channel reservation with its receiver successfully, it does not continue contending the default channel. For simplicity, we assume that there are $n^d$ nodes which always try to contend the default channel to send the ATIM/ATIM-ACK/ATIM-RES packets. We rewrite Eqs. (7) and (8) to calculate the probability of transmission $\tau^a$ and the collision probability $p^a$ for the ATIM window as

$$\tau^a = \frac{2(1-p^a)(1-2p^a)}{(1-2p^a)(1+W_0) + p^a W_0 (1-(2p^a)^m)}$$

$$p^a = 1 - (1-\tau^a)^n-1$$

(9)

We adopt the contention model in [31]. The probability $p^d$, $p^b$, $p^a$, $p^c$, and $p^r$ in each time slot during the ATIM window are given as

$$\begin{align*}
   p^d &= (1-\tau^d)^n^d \\
   p^b &= 1 - p^d = 1 - (1-\tau^d)^n^d \\
   p^c &= n^c (1-\tau^c)^n^c-1 \\
   p^r &= p^b - p^c = 1 - (1-\tau^c)^n^c - n^c (1-\tau^c)^n^c-1
\end{align*}$$

(10)

The duration for a collision transmission $T_{col}$ and the duration for a successful transmission $T_{suc}$ are

$$\begin{align*}
   T_{col} &= T_{atim} + T_{df} \\
   T_{suc} &= T_{atim} + T_{atim_\text{-ack}} + T_{atim_\text{-res}} + 2 \cdot T_{df} + T_{dfs}
\end{align*}$$

(11)

Let $Z^a$ be the interval between the two continuous idle time slots. The mean value of $Z^a$ is expressed as

$$E[Z^a] = T_{idle} + \frac{P_{col}}{P_{idle} + P_{col}} T_{col}$$

(12)

Let $X^a$ be the interval from the channel access contention to the time when the ATIM/ATIM-ACK/ATIM-RES packets are transmitted successfully. The mean value of $X^a$ is given by

$$E[X^a] = \frac{1}{P_{suc}} T_{idle} + \frac{P_{col}}{P_{suc}} T_{col} + \frac{P_{suc}}{P_{suc}} T_{suc}$$

(13)

The average duration of a successful ATIM handshake is determined by Eq. (13). Let $T_{atim_\text{-window}}$ and $T_{data_\text{-window}}$ be the ATIM interval and the data interval, respectively. Let $n_{suc}$ be the number of successful ATIM handshakes during the ATIM window.

$$n_{suc} = \left\lfloor \frac{T_{atim_\text{-window}}}{E[X^a]} \right\rfloor$$

(14)

Since we try to balance the load for all data channels, the number of nodes on each data channel $n^d$ is

$$n^d = \frac{n_{suc}}{N_{ch}} = \frac{1}{N_{ch}} \left( \frac{T_{atim_\text{-window}}}{E[X^a]} \right)$$

(15)

The average duration for a successful data transmission is

$$E[X^d] = \frac{1}{P_{suc}} T_{idle} + \frac{P_{col}}{P_{suc}} T_{col} + \frac{P_{suc}}{P_{suc}} T_{suc}$$

(16)

where
\[
\tau^d = \frac{2(1 - p^d)(1 - 2p^d)}{(1 - 2p^d)(1 + W_0) + p^d W_0 (1 - (2p^d)^{\kappa})}
\]
\[
p^d = 1 - (1 - \tau^d)^{\kappa}
\]
\[
p^d_{\text{idle}} = (1 - \tau^d)^{\kappa}
\]
\[
p^d_{\text{busy}} = 1 - p^d_{\text{idle}} = (1 - (1 - \tau^d)^{\kappa})
\]
\[
p^d_{\text{sec}} = n^d \tau^d (1 - \tau^d)^{\kappa - 1}
\]
\[
p^d_{\text{coll}} = p^d_{\text{busy}} - p^d_{\text{sec}} = 1 - (1 - \tau^d)^{\kappa} - n^d \tau^d (1 - \tau^d)^{\kappa - 1}
\]
\[
T^d_{\text{coll}} = T_{\text{ctrl}} + T_{\text{difs}}
\]
\[
T^d_{\text{sec}} = T_{\text{ctrl}} + T_{\text{difs}} + T_{\text{data}} + T_{\text{ack}} + 3 \cdot T_{\text{difs}} + T_{\text{difs}}
\]

5.1 IEEE 802 analysis

In IEEE 802.11, only one channel is used for the data transmissions. There are \(n\) nodes which contend the channel for the data transmissions. We can calculate the average duration \(E[X^n]\) of a successful data transmission by using Eq. (16) where \(n^d = n\). Since there is a data packet with the length \(L_{\text{data}}\) sent in each \(E[X^n]\), the saturated throughput of the IEEE 802.11 is given by

\[
S_{\text{IEEE802}} = \frac{L_{\text{data}}}{E[X^n]} \tag{17}
\]

5.2 MMAC analysis

There are \(N_{ch}\) available channels and \(n\) nodes in the network. During the ATIM window, all nodes have to contend the default channel for the ATIM packets transmission. As we consider the saturated throughput, each node always has data packets to send. That means there are \(n\) nodes contending the default channel during the ATIM window, \(n^d = n\). We can calculate the value of \(\tau^d, E[X^n]\) and \(E[X^n]\) through Eqs. (9), (13) and (16).

The average number of data packets \(N_{\text{data}}^{\text{MMAC}}\) sent on \(N_{ch}\) channels during each data window \(T_{\text{data-window}}\) is

\[
N_{\text{data}}^{\text{MMAC}} = N_{ch} \cdot \frac{T_{\text{data-window}}}{E[X^n]} \tag{18}
\]

It is also the average number of data packets sent in each beacon. We can determine the saturated throughput by

\[
S_{\text{MMAC}} = \frac{N_{\text{data}}^{\text{MMAC}}}{T_{\text{beacon}}} \tag{19}
\]

5.3 H-MMAC analysis

Different from MMAC, H-MMAC allows nodes to extend their data transmissions to the next ATIM window. In the next ATIM window, there are some nodes on \(N_{ch} - 1\) data channels which are exchanging data packets. There are \(n\) nodes in the network, \(n^d\) available nodes on the default channel during the ATIM window and \(n^d\) available nodes on each data channel during the data window. We have

\[
n^d = n - n^d \cdot (N_{ch} - 1) \tag{20}
\]

From Eqs. (9), (15) and (20), we can calculate \(n^d\) and the \(E[X^n]\) is calculated by Eq. (16).

Since we analyze the saturated throughput, after the channel reservation is made, nodes perform the extended transmission with the duration of the beacon length for \(N_{ch} - 1\) channels, or perform the normal transmission on the default channel. Therefore, the average number of data packets sent during each beacon can be described as
Fig. 11 The impact of the number of channels on the performance of different multi-channel MAC protocols. a Aggregate throughput 80 nodes, 100 packets/s, b dropped packets 80 nodes, 100 packets/s,

\[ N_{data}^{H-MMAC} = \frac{T_{data\_window}}{E[X^d]} + (N_{ch} - 1) \cdot \frac{T_{beacon}}{E[X^d]} \]  \hspace{1cm} (21)

And the saturated throughput of the H-MMAC protocol is

\[ S_{data}^{H-MMAC} = N_{data}^{H-MMAC} \cdot \frac{L_{data}}{T_{beacon}} \]  \hspace{1cm} (22)

Nodes in H-MMAC can choose the transmission mode according to network traffic load. If the network load is low, nodes choose the normal transmission mode which is the same as nodes in MMAC. When the network load is high, nodes will choose the extended transmission mode to increase the performance. In order to utilize all channels with the extended transmission mode, there are at least two nodes on each channel for data transmission, \( n^d = 2 \). And the number of nodes \( n^e \) on the default channel during the ATIM window is larger than \( n^d \cdot N_{ch} \). Using the Eq. (20), we have

\[
\begin{align*}
  n^e &= n - n^d \cdot (N_{ch} - 1) \\
  n^e &\geq n^d \cdot N_{ch} \\
  n^e &\geq 2 \\
  \Rightarrow &\geq n^d \cdot (2N_{ch} - 1) 
\end{align*}
\]  \hspace{1cm} (23)

Theoretically, we can infer that when the number of nodes is less than \( n^d \cdot (2N_{ch} - 1) \), nodes use the normal transmission mode and then the performance of H-MMAC is the same as MMAC. Otherwise, when the number of nodes satisfies Eq. (23), nodes use the extended transmission mode and H-MMAC has higher performance than MMAC.

Figure 10 shows the throughput comparison of the IEEE 802.11 MAC with single channel and the multi-channel MAC MMAC and H-MMAC protocols. The multi-channel MAC protocols have higher throughput than the IEEE 802.11 MAC. Moreover, the H-MMAC protocol can utilize the multiple channels efficiently, and has higher throughput than the MMAC protocol.

6 Performance evaluations

In this section, we describe the simulations of IEEE 802.11 [1], DCA [3], MMAC [11], \( \pi \)-MC [22] and our proposed H-MMAC protocol on our developed packet-level simulation tool in Matlab.

6.1 Simulation model

Each node can have a random location, and can be a source or a destination. The destination of each source is one of the neighbor nodes that are in the source’s transmission
range. Each source node generates and transmits constant-bit-rate (CBR) traffic. A node which is in transmit, receive, idle or doze state consumes the power of 1.65, 1.4, 1.15 and 0.045 W, respectively. The other simulation parameters in our simulations are listed in Table 5. Each simulation was performed for 10 s, and the simulation results are the average of 30 runs. We use the Jain’s fairness index [32] as a performance metric.

In the simulation, we use the following metrics to evaluate the performance of different protocols.

\[
\text{Throughput} = \frac{\text{Packet Size} \times \text{No. Successful Packets}}{\text{Total SimTime}}
\]

\[
\text{Packet Delivery Ratio} = \frac{\text{Packet Received by Receiver}}{\text{Packet Generated by Sender}}
\]

\[
\text{Average Delay} = \frac{\text{Total Packet Delay}}{\text{No. Successful Packets}}
\]

\[
\text{Fairness Index} = \left( \frac{\sum \text{Throughput}}{\text{Number Node} \times \sum \text{Throughput}^2} \right)^2
\]

\[
\text{Energy Efficiency} = \frac{\text{Total Energy Consumption}}{\text{No. Successful Packets}}
\]

6.2 Simulation results

The main purpose of the multi-channel MAC protocols is to exploit multiple channels to improve the network performance with high throughput, low delay and fairness. Figure 11 shows the impact of the number of channels on the performance of different multi-channel MAC protocols. We simulated with 80 nodes where packets arrived at each node at the rate of 100 packets/second. When the number of channels increases, there are more chances to transmit data packets. The collision probability decreases and a data packet does not need to wait for a long time until transmitted. As a result, the aggregate throughput, packet delivery ratio and fairness index increase (Fig. 11a, c, e), and the packet loss, average delay and the energy consumption per data packets decrease (Fig. 11b, d, f). However, when the number of channels increases, the aggregate throughput of DCA does not always increase, as shown in Fig. 11(a). As we mentioned above, when the number of data channels is greater than \( T_d/T_c \), the data channel resource cannot be fully utilized because of the control...
protocols, it is affected by the channel switching time and the back-off duration. In the proposed H-MMAC protocol, nodes can transmit during the ATIM window while other nodes try to negotiate the channel. Therefore, the throughput of H-MMAC is higher than that of the other protocols, and about 22% higher than that of MMAC. The multi-channel MAC protocols enable more data packet transmissions than the IEEE 802.11 MAC protocol. That is why the average delays of the multi-channel MAC protocols are lower than that of IEEE 802.11 MAC. Our proposed H-MMAC protocol has a lower delay than the DCA, MMAC and π-Mc protocols as shown in Fig. 14(d). The IEEE 802.11 MAC has a starvation issue that results in low fairness when the network traffic is high. Similar to the IEEE 802.11 MAC, nodes have to contend the control channel to reserve the data channel for one data transmission in the π-Mc and DCA protocols. They also suffer the starvation problem. In the MMAC and H-MMAC protocols, after two nodes made the channel reservation successfully, they do not contend the default channel and thus the remaining nodes have chances to make the channel reservation. During the ATIM handshake, nodes choose the “best” channel based on their NILs and PCLs. Through the channel selection algorithm, nodes have more chance to transmit data packets. As a result, H-MMAC has a higher fairness index as shown in Fig. 14(e). As the packet arrival rate increases, the collision probability increases. It leads to the increasing dropped packets (Fig. 14b) and the decreasing packet delivery ratio (Fig. 14c). Like MMAC, H-MMAC also implements a power saving mechanism. All of the nodes that do not have data to exchange or cannot contend the data channel in the ATIM window enter doze mode with a very low power consumption of 0.045 W. In the other protocols, all of the idle nodes have to stay awake and consume an idle power of 1.15 W. With less energy consumption and higher throughput, the energy efficiency of the H-MMAC protocol is better than that of the other protocols as shown in Fig. 14(f).

7 Conclusions and future work

In this paper, we propose a hybrid and adaptive multi-channel MAC protocol, called H-MMAC, which utilizes almost all of the channel resources to improve the network performance. By using the PSM and allowing nodes to transmit data during the ATIM window based on the network traffic load, the H-MMAC protocol achieves higher performance than the other multi-channel MAC protocols. The analytical and simulation results show that H-MMAC’s performance is increased significantly when the number of channels is large. Moreover, the performance of H-MMAC is not significantly affected by a long ATIM window compared to MMAC.

Beside the selected channel information, the nodes can also exchange their transmission power information on the default channel during the ATIM window. Therefore, we can apply a power control algorithm for the H-MMAC protocol to improve the spatial reuse of the wireless channels.

Acknowledgments This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2013-H0301-13-4006) supervised by the NIPA (National IT Industry Promotion Agency).

References


Springer
Sungwon Lee received the Ph.D. degree from Kyung Hee University, Korea. He is a professor of the Computer Engineering Department at Kyung Hee University, Korea. Dr. Lee was a senior engineer of Telecommunications and Networks Division at Samsung Electronics Inc. from 1999 to 2008. He is an editor of the Journal of Korean Institute of Information Scientists and Engineers: Computing Practices and Letters.