Abstract—The medium access control (MAC) protocol, which coordinates channel access between wireless stations, significantly affects network throughput of wireless ad hoc networks. A MAC protocol that is based on a multi-channel model can achieve high throughput by enabling more simultaneous transmission pairs in the network. In this paper, we investigate different designs of multi-channel MAC protocols and classify them according to different negotiation strategies for channel selection. We then propose a novel design by combining the advantages of two different categories. The simulation results show that our proposed protocol can significantly outperform two representative protocols.

Keywords – Wireless Ad Hoc Networks, IEEE 802.11, Medium Access Control, Multi-Channel

I. INTRODUCTION

Wireless Local Area Networks (WLANs) have become increasingly popular in recent years. WLANs can be configured as two network types: infrastructure and ad hoc. The rapid and economy deployment in response to communication needs makes the wireless ad hoc network attractive in the applications such as emergency rescue and battlefield communication.

IEEE 802.11 is an international WLAN standard which covers the specification for the Medium Access Control (MAC) sublayer and the Physical (PHY) layer. One commonly used function to coordinate channel access for wireless stations (STAs) in the IEEE 802.11 MAC protocol is called Distributed Coordination Function (DCF).

If all transmissions among STAs are taken place on a single channel, signal interference will limit network throughput. In the IEEE 802.11b and the IEEE 802.11a systems, there are totally three and twelve non-overlapping channels available, respectively. Using multiple channels in the MAC protocol can increase network throughput by enabling multiple STAs in the same interference zone to transmit data simultaneously [15][17].

When the STAs are located on different channels and are unaware of the channel activities of their neighbors, multi-channel MAC protocols suffer from some problems such as multi-channel hidden terminal, deafness, and broadcast problems [16]. An STA is called multi-channel hidden terminal if it causes interference on one of its neighbors by attempting transmission after switching to the same channel this neighbor is currently used. A deafness problem occurs when an STA continuously attempts to contact with another STA which is located on a different channel. The sending of broadcast messages to the STAs which are located on different channels becomes costly, since multiple unicast messages should be sent instead.

In this paper, we investigate the design of multi-channel MAC protocols by comparing and classifying the existing work. More importantly, we propose a novel protocol to enhance network throughput by tackling the weaknesses of existing protocols.

The remainder of this paper is organized as follows. We give a survey on the related work in Section II. The proposed novel protocol is presented in Section III. We evaluate the performance in Section IV, and draw conclusions in Section V.

II. RELATED WORK

In a multi-channel MAC protocol, the most important work is on the negotiations between the sender and the receiver of a transmission pair to decide an appropriate channel for data exchanges. In this work, we need to decide when and where to do negotiations, and how to agree with a common channel. According to various decision strategies, we classify various multi-channel MAC protocols into different categories.

A. Channel Negotiation Strategy

Based on the place to do negotiations, there are two general approaches: dedicated control channel and common control period (called split phase in [1]). In the first approach, one or more control channels (All protocols of this category except for MCMAC [7] use only one control channel) are dedicated to exchange control packets for negotiations. The remaining available channels are called data channels. Any transmission pair can do negotiations anytime on the dedicated control channel and then switch to the commonly selected data channel for data exchanges. Examples of this category include AMNP [2], DCA [3], DPC [4], RBCS [5], MCDA [6], and MCMAC [7].

In the second approach, there is no dedicated control channel but one data channel (called common channel) will temporarily act as a control channel. All STAs will synchronously alternate between control and data periods. During a control period, all STAs have to switch to the common channel and do negotiations with the destined STAs.
of pending data packets. A transmission pair after successful negotiations has to wait for the ending of the current control period and then will switch to the commonly selected data channel for data exchanges. Note that the common channel can be selected as a data channel for data exchanges. Examples of this category include MMAC [8] and MAP [9].

We compare the minimal number of transceivers that is necessarily equipped with each STA for both approaches. In the first approach, two transceivers are needed if one control channel is used. One is for keeping in touch with other STAs on the dedicated control channel. The other is for dynamically switching to the selected data channel for data exchanges. The protocols [2][10] using only one transceiver would suffer from the problems we mentioned in the introduction. In the second approach, only one transceiver is needed and this transceiver has to stay in control and data periods alternately.

Next, we compare the maximal number of data channels that can be fully utilized together (called data capacity in the paper) for both approaches. Let \( T_C \) and \( T_D \) respectively denote the average time to complete negotiations and the average time to complete data exchanges for a transmission pair. Suppose that \( T_D < T_C \). Then, in the first approach, there are at most \( T_D/T_C \) transmission pairs that can finish negotiations on the control channel during one \( T_D \) on the data channel. As discussed in [3], the data capacity is \( T_D/T_C \). In the second approach, let the length of a control period be \( T_C \). Then, the total number of transmission pairs that can finish negotiations during one control period is \( T_D/T_C \). All these transmission pairs will simultaneously enter the data period and will be distributed to at most \( T_D/T_C \) data channels. Hence, the data capacity is \( T_D/T_C \).

The advantages and disadvantages of these two approaches are summarized below.

- **Dedicated control channel**
  - **Advantages**: The dedicated channel can serve as a broadcast channel. Channel negotiations can be done anytime.
  - **Disadvantages**: The dedicated channel on which only control packets are transmitted may have low channel utilization. The data capacity is dependent on the bandwidth of the dedicated channel, which cannot be dynamically adjusted.

- **Common control period**
  - **Advantages**: The common channel can be used to transfer data packets during data periods, which increases channel utilization. The hardware cost is low because only one transceiver is needed. The data capacity is dependent on the length of a control period, which can be dynamically adjusted.
  - **Disadvantages**: Time synchronizations among STAs are needed. It takes high cost to transmit any broadcast message during a data period. Channel negotiations cannot be done anytime, and hence this may cause a long delay on the delivery of any pending data packet.

Actually, there are other approaches to do negotiations. HRMA [11] and SSCH [12] use rendezvous time slots during frequency hopping to do negotiations. Each STA uses multiple unicasts in HMCP [13] and a broadcast channel in PCAM [14] to inform neighboring STAs its channel setting. Any neighbor that wishes to exchange data with an STA can switch to the same channel of this STA.

### B. Channel Selection Strategy

In the following, we discuss how to agree with a common channel during negotiations. We consider this issue from two aspects: selection criterion and decision maker.

We classify various channel selection strategies into two categories: global and local schedules. Every STA can know the channel activities of all other STAs in the global schedule, but can know the channel activities of neighboring STAs in the local schedule.

The MAP [9] protocol has a global schedule and a common control period is used to do negotiations. Every STA will collect these negotiation data and will have a global view about how many transmission pairs will take place and how long each transmission pair will spend. According to this information, every STA can use the same shortest-job-first scheduling algorithm to schedule these transmission pairs onto appropriate channels. Though the optimal schedule can be found, this method can not be applied to multi-hop network environments.

In the local schedule, there are three criteria that can be used: *idle state* (e.g., AMNP [2], DCA [3], DPC [4], RBCS [5], MMAC [7], and MAP [9]), *traffic load* (e.g., MMAC [8] and HMCP [13]), and *random assignment* (e.g., MCDA [6]). Using the idle state, the channel that will become idle at the earliest is selected. Using the traffic load, the channel that has the lightest traffic load is selected. Using the random assignment, the channel is randomly selected. Using the first two criteria will incur the overhead on information collection, while using the last criterion may have poor performance.

Next, we discuss who makes the final decision on the channel selection during negotiations. The decision maker may be a sender, a receiver, or a hybrid sender and receiver.

In the sender-based scheme, always a sender decides a data channel according to its own selection criterion to exchange data with a receiver. The receiver may either conditionally or unconditionally accept the decision. If the receiver unconditionally accepts, this kind of protocols needs another channel contention on the selected data channel to prevent collision at the receiver side. If the receiver conditionally accepts, it can reject the decision if its own selection criterion is violated. This may involve several rounds of negotiations between the sender and receiver. Each round of negotiations involves a two-way handshake. Examples of this scheme contain DPC [4] and MCDA [6].

In the receiver-based scheme, a sender tells its corresponding receiver the state (idle state or traffic load) of each data channel it knows. Then the receiver compares this received information with its own one and then decides a data channel to be used. If the neighbors at the sender side need to be informed with the decision result, these negotiations will involve a three-way handshake. The delivery of channel states becomes an overhead. Examples of this scheme contain DCA [3], MMAC [8], and MCMAC [7].
In the hybrid-based scheme, a sender tells its corresponding receiver both the state of each data channel and a recommendation of one data channel. If the receiver accepts the recommendation, this scheme works as a sender-based one. Otherwise, this scheme works as a receiver-based one. An example of this scheme is AMNP [15].

III. PROPOSED PROTOCOL

Our proposed protocol combines the advantages of using the dedicated control channel and the common control period. We use two transceivers at each STA. One is called control transceiver that permanently operates on the dedicated control channel, and the other is called data transceiver that can switch among data channels.

We involve power management into our protocol design. In the ad hoc mode for IEEE 802.11 WLAN, each STA can turn into sleep mode for power saving, and will wake up periodically at each ATIM window. During an ATIM window, an intended sender will send a message to the receiver, and then the receiver will keep awake. In MMAC [8], the ATIM window is acted as the common control period during which channel negotiations are performed. Indeed, the tasks to wake up and negotiate with other STAs can be combined together.

Compared with MMAC, we use two transceivers to do negotiations during an ATIM window on two channels. Consequently, we can double the data capacity if an equal sized ATIM window is used in both schemes. Moreover, we can achieve the same data capacity with MMAC using a half of an ATIM window and leave more time for data transmissions. During each non-ATIM period, our proposed protocol behaves like the protocol which is based on the dedicated control channel but with some enhancements.

First, we enable the control channel to be a data channel by allowing the control transceiver to transmit data packets too. In the extreme case, an STA can be concurrently involved in two separated data transmissions. One is on the control channel and the other is on the data channel. This would facilitate the data exchanges in a bidirectional connection.

Second, we consider the data retransmission issue that is seldom discussed in the literature. In DCA [3], a data retransmission has to be re-negotiated on the control channel. This would burden the overhead of the control channel, particularly when channel errors are high. We propose to perform data retransmissions on the same selected channel without extra negotiations.

We do not limit any type of selection criterion and decision maker in the description of our proposed protocol, but do limit it in the performance evaluation. The basic operations of the proposed protocol are described below.

Step 1 (if enabling power management): Each STA should wake up when entering an ATIM window and tune the data transceiver into a default data channel. An STA can do negotiations with other STAs using both transceivers simultaneously. Any data transmission occurs after the ending of the current ATIM window.

(a) If the control channel is selected for data exchanges, the control transceiver is used. If a data channel is selected, the data transceiver is used. Both transceivers can operate simultaneously if necessary.

(b) If the data transceiver at either the sender or the receiver has associated with one particular data channel, this data channel has to be selected in the future negotiations during the remaining time of the current ATIM window. That is, an STA can only agree to use one data channel for its data transceiver during an ATIM window.

Step 2: If any transmission error occurs during data exchanges, an STA performs a data retransmission process on the same channel. The retransmission for the same data packet can be repeated until a maximal retry limit is reached.

Step 3: An STA can initiate negotiations on the control channel during non-ATIM periods for any new data packet.

(a) We can not change the associated channel with a data transceiver when there is any pending transmission that will be served by the transceiver. Hence, if the currently used data channel at the sender is different from that at the receiver. We use the control transceiver to serve data exchanges on the control channel instead.

(b) When a data transceiver switches to a different channel, it has to hold a time period whose length is equal to the time to transmit a data packet of maximal size. This holding is for avoiding possible collision after switching to the new channel.

We evaluate performance using the traffic load as the selection criterion and the sender as the decision maker. An STA maintains each counter for each available channel (including control and data channels). This counter records the number of times that the corresponding channel has been selected by neighboring STAs for data exchanges. This information can be collected by overhearing all the negotiations. A sender will select the channel with the lowest count when doing channel selection and then will inform the receiver the selected channel. The sender has to contend the channel access on the selected channel before data transmissions. Moreover, the sender can send out all pending data packets destined to the same receiver.

A scenario example is shown in Figure 1, where only two channels (one control channel and one data channel) are available. The embedded number (1 in the figure) in a control packet indicates that the selected channel is the data channel. During the ATIM window, STA A informs STAs B and C of pending data packets at the same time. Then these data transmissions take place on the data channel after the ATIM window by following the sequence RTS-CTS-DATA-ACK. Meanwhile, STA B has pending data packets for STA C and hence negotiates with STA C on the control channel. Then STA B switches to the data channel and contends the channel access.
IV. PERFORMANCE EVALUATION

We write simulation programs using the CSIM [18]. We mainly compare our protocol (abbreviated as Novel) with DCA and MMAC.

A. Simulation Model

For simplicity, we consider a fully connected topology where all STAs are within the radio range of each other. Also, we disable power management in our scheme for a fair comparison with DCA. The arrival of data packets at a sender is followed by the Poisson process. We assume that each data packet is of the same size of 1024 bytes. The network parameters are followed by the IEEE 802.11 system. We assume that each channel has the same bandwidth of 2Mbps. The default channel number is three. The wireless channel condition is modeled as a two-state Markov process with good and bad states.

In the simulation, we use the following metrics to measure the performance.

\[
\text{Throughput} = \frac{\text{Packet\_Length} \times \text{No\_Successful\_Packets}}{\text{Total\_SimTime}}
\]

\[
\text{Average\_Delay} = \frac{\text{Total\_Packet\_Delay}}{\text{No\_Successful\_Packets}}
\]

\[
\text{Drop\_Rate(\%)} = \frac{\text{No\_Drop\_Packets}}{\text{No\_Total\_Generated\_Packets}}
\]

\[
\text{Utilization} = \frac{\text{Packet\_Length} \times \text{No\_Successful\_Packets}}{\text{Total\_SimTime} \times \text{No\_Channels}}
\]

B. Simulation Results

Figure 2 shows throughput comparisons as the number of sending STAs (i.e., sender nodes) increases. The network load becomes heavy when the number of STAs is large. When the network load is low, all protocols perform similarly. As network load grows to near saturation, Novel performs significantly better than DCA as well as MMAC. This is because the bandwidth of the control channel is fully utilized by the proposed protocol. Novel which uses the counter-based channel selection can simplify the channel decision process and can balance channel load on the network.

DCA has to compute the release time of each data channel and agree with a free channel before data transmission. Moreover, only one data packet can be transmitted after negotiations. This results in complex channel negotiations and ineffective reservations when data transmissions fail on the selected data channel.

MMAC has to pay the penalty of the ATIM window during which data packets can not be delivered. We have selected the best ATIM window size for MMAC in the simulation, but its throughput is still lower than Novel.

Figure 3 shows the average packet delay as the network load increases. As can be seen, DCA and Novel have the similar delay. MMAC has a high delay, since STAs have to wait for the ending of the current ATIM window before transmitting data packets. Moreover, not every transmission pair can finish channel negotiations during an ATIM window. These pairs have to wait for the next ATIM window, and this will further postpone packet deliveries.

Figure 4 compares the drop rates of different protocols. Novel has the lowest drop rate among all. DCA suffers from a
high drop rate because of serious contentions (per-packet negotiations and re-negotiations) on the control channel.

Figure 5 presents throughput comparisons when varying the number of available channels. The number of sending STAs is 80. When the available channel number is small, Novel performs better than DCA. This is because Novel has one more channel (control channel) to be a data channel than DCA. However, DCA gets large improvement as the channel number increases, and may even slightly outperform Novel. This is because the occupation of data deliveries on the control channel becomes significant in Novel. We found that Novel can perform similarly with DCA in the case of the high channel number if we disable the control channel to be a data channel. In MMAC, the maximal number of transmission pairs that can successfully finish negotiations during an ATIM window is limited. Hence the throughput can not be increased more when we add more channels.

![Figure 5. Throughput vs. Number of channels.](image)

Figure 6 compares utilizations when varying the number of available channels. Again, the number of sending STAs is 80. As can be seen, Novel has better channel utilization than other protocols. The utilizations of Novel and DCA become increasing and then decreasing as the number of channels increases. Novel and DCA can fully utilize all the channels when there are 3 and 5 channels, respectively. Adding more channels to these two protocols will have no benefits, since the maximal number of transmission pairs that can successfully finish negotiations on the control channel is bounded. As discussed before, the performance of MMAC is mainly dependent on the size of an ATIM window but not the number of channels.

![Figure 6. Utilization vs. Number of channels.](image)

Figure 6. Utilization vs. Number of channels.

References:


