MAC Protocol using Asynchronous Multi-channels in Ad Hoc Networks

Chunghwan Son, Neung-Hyung Lee, and Saewoong Bahk
School of Electrical Engineering and Computer Science, and INMC,
Seoul National University, Seoul, Korea
Email: {chson, nhlee, sbahk}@netlab.snu.ac.kr

Abstract—In this paper, we consider a simple but efficient medium access control (MAC) protocol that can be used in an ad hoc network with asynchronous multi-channel environments. The approach in our proposal is to use time offset between channels where each channel has its own frame starting time. We assume that the network runs in power saving mode (PSM) of the IEEE 802.11 standard protocol which channel structure consists of beacon frame, ad hoc traffic indication message (ATIM) window, and data window. By asynchronously coordinating beacon frame start times between channels, nodes that have no tasks involved in the current channel can switch to some other channels for more consecutive data transmission trials. Through extensive simulations, we confirm that our multi-channel access scheme results in significant throughput enhancement compared to the conventional access scheme that uses multi-channels in a synchronized manner. Combining our proposal with a power saving scheduling algorithm, we can achieve further throughput improvement. Our analysis and simulation results are used to prove the advantage of our proposals.

I. INTRODUCTION

An ad hoc network consists of multiple nodes that communicate with each other without a fixed infrastructure. If the network has a fixed infrastructure, it is much more reliable and easier to manage. For example, a wireless local area network (WLAN) has an access point (AP) that performs several functions for network management.

The IEEE 802.11b has 13 or 14 channels and three of them can be used simultaneously without interfering with each other. In a WLAN, it is hard to use multiple channels at the same time although nodes including the AP have multiple transceivers. This is because the AP always takes part in communication as a sender or a receiver. In an ad hoc network, however, multiple transmissions over multiple channels can be possible because each node can operate in a distributive and independent manner. In [2], multi-channel MAC (MMAC) in an ad hoc network was proposed and a single transceiver is considered to handle the hidden terminal problem in multi-channel environments. It works well in light load, while it performs badly in heavy load because of the channel matching problem.

In this paper, we propose a MAC protocol in a multi-channel ad hoc network. Our proposal uses one transceiver to make its implementation cost effective. The objective in our design is to reduce access delay and to enhance throughput performance by using asynchronous multiple channels. Asynchronous channels are created by using different time offsets between channels.

The IEEE 802.11 standard specifies how a node lowers power consumption by using power saving mode (PSM). Assuming an independent basic service set (IBSS), the IEEE 802.11 divides a beacon interval into two windows: ad hoc traffic indication map (ATIM) window and DATA window. Each node that has a packet for transmission sends an ATIM packet to its destination node during the ATIM window. If the ATIM packet and corresponding ACK are successfully exchanged, the source can try the transmission of the data packet during the DATA window. Nodes that did not participate in exchanging ATIM packets will enter the doze mode during the following DATA window and wake up at the next beacon interval.

In the IEEE 802.11 standard, nodes that won the contention during the ATIM window should have another contention for data transfer during the DATA window. As the double contention is inefficient, in this paper, we adopt the scheduling based structure proposed in [7] that runs a scheduling algorithm right after the ATIM window, thereby getting rid of the contention during the DATA window.

To evaluate our proposal, we perform numerical analysis and simulation analysis. Our analysis results confirm that our proposals perform better than MMAC especially in heavy load. The rest of this paper is organized as follows. In Section II, we propose our scheme and combine it with a scheduling algorithm. Mathematical analysis is given in Section III. The simulation results are described in Section IV, followed by the conclusion in Section V.

II. PROPOSED SCHEME

A. Multi-channel access Algorithm

In the IEEE 802.11b network, there are 13 or 14 channels. When communication is in need, the AP chooses one of these channels after considering each channel condition. However, there are at least three channels that are not overlapping one another. Therefore we are motivated to design a MAC protocol that uses three channels together in an efficient manner. In this section, we propose a simple scheme which is suitable for a multi-channel ad hoc network. For simplicity, we use three non-overlapping channels in explanation.

There have been researches on multi channel ad hoc networks [1]-[4]. In [1] and [3], multiple transceivers for each node are assumed which is not practical. In [1], a control channel is introduced to make channel management easier but
it creates some implementation complexity. In this paper, we assume one transceiver per node and no control channel.

The main idea in our proposal is using different time offsets between channels and adopting PSM of the 802.11 standard. The beacon interval is divided into three subintervals, and the second channel starts its beacon interval after the first subinterval of the first channel elapses. Fig. 1 shows the channel structure. Nodes that have completed their tasks in channel 1 can move into channel 2 or 3 for more transmission trials by joining the contention in the ATIM window of the corresponding channel. So, our scheme uses the whole bandwidth of the channels that are not selected. As our scheme experiences the ATIM window three times more often compared to the conventional scheme, it is acceptable to have a shorter ATIM window and a longer DATA window for throughput enhancement.

B. Multi-channel Access Algorithm with Proposed Scheduling

In the DATA window, only the tasks that have successfully exchanged ATIM messages in the ATIM window are able to transmit according to the scheduling map of the AP. The scheduling gets rid of the contention in the data window, thereby achieving high channel utilization. The nodes that have completed their tasks in the DATA window are able to join the contention in the next coming ATIM window by other channel. Therefore, our goal of scheduling is to maximize the number of nodes that can switch to the following channels before their ATIM windows start. This goal is similar to that of the shortest task first algorithm in [8].

Let \( M \) be the number of tasks that have successfully exchanged ATIM messages in an ATIM window. Denote \( S_i \) and \( R_i \) as the \( i \)th successful ATIM sender and receiver, respectively. The size of the \( i \)th task is defined as

\[
T_i = \sum_{k=0}^{M} [U(S_i = S_k) + U(S_i = R_k)] + U(R_i = S_k) + U(R_i = R_k),
\]

where \( U(x) = \begin{cases} 1, & \text{if } x \text{ is true;} \\ 0, & \text{otherwise.} \end{cases} \) (3)

Define the first neighbor channel (FNC) as the channel with the first coming ATIM window, and the second neighbor channel (SNC) as the channel with the second coming ATIM window. For instance, channel 3 is the FNC of channel 2 and the SNC of channel 1. Let us define \( N_{th1} \) as the number of tasks that can be transmitted in the DATA window before the beacon frame of the FNC starts, and \( N_{th2} \) as the number of tasks that can be finished before the beacon frame of the SNC starts. \( N_{th1} \) and \( N_{th2} \) are dependent on the ATIM window size and the sizes of DATA packets for transmission.

Our considered scheduling policy consists of two steps. First, it considers all the task sizes and chooses one that satisfies

\[
Q = \arg\min_{i} T_i.
\]

In this scheme, the node with a smaller number of tasks is scheduled first, and the node that have completed their tasks before the beacon frame of the FNC switch to the FNC for more trials. The performance can be improved by using the two step scheduling algorithm shown in Fig. 2. In the first scheduling, we pick \( N_{th1} \) tasks, and assuming the nodes involved in \( N_{th1} \) tasks have completed their tasks, the scheduling algorithm updates the task size information. The same scheduling is performed once again for the nodes left in the current channel with updated task size information. The objective of each scheduling is to maximize the number of nodes that can be switched to the neighbor channel(s) for fast new trials.

III. ANALYSIS

For simple analysis, we assume that each node has packets for transmission at any time. When a node is successful in medium access during the ATIM window, it transmits an ATIM
packet to the destination. If the destination node stays on the same channel, it responds with the ACK packet to the sender. If it is in another channel, the source can not get the ACK packet from the destination. To handle this problem in an efficient manner, we assume each node has two queues and operates them in the following way. When a node fails to get the ACK in the ATIM window, it stores the packet in the second queue and tries to send a next packet from the first queue. When this node participates in the ATIM contention in the neighbor channel, the head-of-line packet in the second queue will be tried first.

In our analysis, we first find out the expected number of tasks, \( M \), that are successful in ATIM message exchange. Once we obtain \( M \), we can calculate the system throughput clearly. Therefore we focus on the analysis of contention in the ATIM window by defining variables as follows.

- \( \tau \): the probability that a station transmits an ATIM packet at each time slot
- \( p \): collision probability of ATIM packets
- \( M \): the number of tasks successful in ATIM and ACK exchange during an ATIM window
- \( w \): minimum contention window size \( CW_{\text{min}} \)
- \( m \): the value such that maximum contention window \( CW_{\text{max}} = 2^m w \)
- \( E[n]\): the expected number of nodes contending for a given channel for an ATIM window
- \( N \): the total number of nodes

Our analysis adopts and manipulates the results given in [6]

\[
\tau = \frac{2}{1 + w + pw \sum_{i=0}^{m-1} (2p)^i}, \quad (5)
\]

where

\[
p = 1 - \tau (1 - \tau)^{E[n]-1} \frac{E[n]}{N}. \quad (6)
\]

The second term in the collision probability represents that only a station transmits a packet during an ATIM window and the corresponding receiver is on the same channel. Then

\[
E[M] = \frac{ATIM\_size (1-p)}{W_{\text{backoff}} \times \text{SLOT\_time} + \text{TRANS\_time}}. \quad (7)
\]

Here, \( \text{TRANS\_time} \) is the total time for exchanging an ATIM packet and the ACK packet, and \( W_{\text{backoff}} \) represents the average of minimum backoff numbers of nodes that contend for the medium. \( W_{\text{backoff}} \) can be calculated by

\[
W_{\text{backoff}} = \frac{1-p - p(2p)^m w}{1 - 2p} \frac{1}{2M} \text{ or } \approx \frac{1-p - p w}{1 - 2p} \frac{1}{2M} \quad (8)
\]

\[
N - E[n] = \alpha(E[M] - E[N_{\text{th1}}]) + \beta(E[M] - E[N_{\text{th2}}]) \quad 0 < \alpha, \beta \leq 2 \quad (9)
\]

In this derivation, we assumed that nodes have exponentially distributed random backoff numbers. The approximate form in eq. (8) is the result of the summation of each backoff step. The left term in eq. (9) represents the number of nodes that can not join the contention during the ATIM window of the current channel. This means that those nodes could not switch from the other two neighboring channels to the current channel because of their unfinished tasks. Here, \( \alpha \) and \( \beta \) are the variables that represent the ratio of the number of nodes with some jobs to the number of tasks. For example, if \( \alpha \) and \( \beta \) are 2, each node has one task only. This means the number of nodes that can be released to the neighbor channel after completing the given tasks equals twice the number of tasks. If they are 1, each node is involved in two unfinished tasks in the other channels. So two nodes can be released after finishing two tasks. From these, the throughput can be given by

\[
\text{Throughput} = \frac{\text{DATA\_size} \cdot M \cdot 8}{BEACON\_int} \cdot 3, \quad (10)
\]

where \( BEACON\_int \) represents the length of the beacon interval.

IV. SIMULATION RESULTS

To evaluate our proposed scheme, we performed extensive simulations. We compare the throughput of our scheme with that of MMAC that does not have the channel offset. It is assumed that each node has identical capability. The number of nodes varies from 20 to 40. The physical transmission rate is 2Mbps for data frames and 1Mbps for control frames such as ATIM and ACK frames. The size of data frame is fixed at 1500 bytes. The incoming traffic of each node has a Poisson distribution. Other parameters are given in Table I.

<table>
<thead>
<tr>
<th>TABLE I WIRELESS LAN PHYSICAL PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>------------------------------------</td>
</tr>
<tr>
<td>Slot time</td>
</tr>
<tr>
<td>SIFS</td>
</tr>
<tr>
<td>DIFS</td>
</tr>
<tr>
<td>(CW_{\text{min}})</td>
</tr>
<tr>
<td>(CW_{\text{max}})</td>
</tr>
<tr>
<td>Preamble duration</td>
</tr>
<tr>
<td>PLCP header duration</td>
</tr>
<tr>
<td>Beacon interval</td>
</tr>
</tbody>
</table>

Figs. 3 through 8 compare the throughput performances of our scheme and MMAC. For simple and clear comparison, we assume that three channels have the same quality.

Fig. 3 shows that our proposals outperform MMAC. When the network load is light, i.e., the packet inter-arrival time is long, our schemes and MMAC show the same network throughput. As the network load gets heavier, our scheme shows 2.5 times higher throughput compared to MMAC. This is because if the network load is light, the channel matching problem does not likely happen. However, if the network is heavy, the waste of bandwidth in the ATIM window due to the channel matching increases, resulting in throughput decrease. When the scheduling algorithm in [7] is with our proposal represented as 'proposal with scheduling' in the figure, we observe the throughput increase by about 5% in case of the heavy load.
Fig. 3. Throughput performance according to the traffic load ($N = 20$ and $ATIM = 0.01s$).

Fig. 4. Throughput performance according to the traffic load ($N = 20$ and $ATIM = 0.02s$).

Fig. 5. Throughput performance according to the traffic load ($N = 30$ and $ATIM = 0.01s$).

Fig. 6. Throughput performance according to the traffic load ($N = 30$ and $ATIM = 0.02s$).

Fig. 7. Throughput performance according to the traffic load ($N = 40$ and $ATIM = 0.01s$).

Table II: Throughput Comparison (ATIM Size of 0.01s)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Analysis(Mbps)</th>
<th>Simulation(Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2.35</td>
<td>2.35</td>
</tr>
<tr>
<td>30</td>
<td>2.11</td>
<td>2.28</td>
</tr>
<tr>
<td>40</td>
<td>1.88</td>
<td>2.05</td>
</tr>
</tbody>
</table>

We compare the simulation results with the analysis results in Table II. For analysis, each node is assumed to have two queues. According to the traffic load and pattern, $\alpha$ and $\beta$ vary from 0 to 2. The analysis results are not much different from the simulation results when $\alpha$ and $\beta$ are one. For $N = 30$ and 40, the gaps between the analysis and simulation results are less than 9%, respectively.

V. CONCLUSION

In this paper, we proposed an enhanced protocol of the IEEE 802.11 standard that is suitable for multi channel ad

...
hoc networks. We analyzed our protocol mathematically and performed simulations extensively to investigate how well our proposal works. Introducing time offsets between channels, a node can switch to another channels for more transmission trials. After switching, it sends a channel request message during the corresponding channel’s ATIM window. Additionally we enhanced our proposal by adopting a scheduling algorithm that serves a task of the node with less tasks first to maximize the total sleeping time. As more nodes can be released sooner from their jobs, they can switch to another channel and try another transmission, thereby enhancing the total throughput.

The advantage of our schemes is that they just use one transceiver and solve the channel matching problem simply. This paper considered the case of three channels for simplicity. Our simulation results showed that our schemes approximately double the throughput performance of MMAC when the network traffic is high, and there was no performance degradation in the light load.

REFERENCES

Fig. 10. Throughput performance according to the beacon interval ($N = 30$ and $ATIM = 0.02s$).


