A Multi-channel MAC Protocol with TDMA Initialization for Rapid Home Sequence Propagation

Xiaojuan Jiang\textsuperscript{†}, Masaki Bandai\textsuperscript{†}, and Takashi Watanabe\textsuperscript{†}

\textsuperscript{†} Graduate School of Information Shizuoka University
\textsuperscript{‡} Faculty of Science and Technology Sophia University

E-mail: \textsuperscript{†} gs10302@a.inf.shizuoka.ac.jp, watanabe@a.inf.shizuoka.ac.jp, \textsuperscript{‡} bandai@sophia.ac.jp

\textbf{Abstract} To improve wireless network throughput, the multi-channel media access (MAC) protocols have been proposed. They are classified into two categories, i.e., centralized protocols and decentralized protocols. In this paper, we address the problems of them, i.e., bottleneck problem in the control channel and long propagation time, and propose a multi-channel MAC protocol called TIMMAC which combines the centralized protocol and the decentralized protocol. We use several simulations to evaluate our protocol. The results show the proposed protocol has better performance in throughput and end-to-end delay than McMAC protocol.

\textbf{Keyword} Ad hoc network, multi-channel, common hopping, multiple rendezvous

1. INTRODUCTION

Over the last two decades, wireless network technologies have been developed rapidly. As a typical wireless network, Ad hoc network attracts researchers’ attention. Ad hoc network is being applied in many fields, such as transportation, military, entertainment. However, there are still challenges in ad hoc network, such as throughput improvement, interference suppression and security enhancement. Multi-channel can improve the channel utilization and reduce the overhead brought by data transmissions. It allows communicating pairs to transmit simultaneously, hence multi-channel as a way of throughput improvement is being studied by many researchers [6] [7] [8] [9] [10].

The main problem of multi-channel is how to distribute multiple channels to communicating pairs effectively. Depend on the strategy of distribution [5] divides multi-channel MAC protocols in to four types, i.e., dedicated control channel protocols, common hopping protocols, split phase protocols and multiple rendezvous protocols (Table 1). In the first three types, all the nodes exchange the channel reservation message which is necessary before sending data in a specific channel. In other words there is only one transmission pair reserve data channel at a specific time. While in the decentralized protocols nodes make agreement in distinct channels. So the reservation messages can be sent synchronously in different channels. While in the multiple rendezvous protocols nodes make agreement in distinct channels. So the reservation messages can be sent synchronously in different channels. Because of this feature we focus on the decentralized protocols.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Interface</td>
<td>&gt;&gt;2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Control channel</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Rendezvous</td>
<td>Single</td>
<td>Single</td>
<td>Single</td>
</tr>
</tbody>
</table>

Table 1 Categories of multi-channel protocols

The rest part of this paper is organized as following. First we introduce the related works. In the related works we present the centralized phase protocol and the decentralized phase protocol in detail and list the main problem of each protocol. In part 3, we propose a multi-channel protocol named TIMMAC. We give the detail work of TIMMAC and explain how it deals with the migration of nodes. In part 4, we show the evaluation...
of our protocol. In the last part, we give the summary and future works.

2. RELATED WORKS

We classify multi-channel protocols into two categories. One is the centralized multi-channel MAC protocol; the other one is the decentralized multi-channel MAC protocol. The centralized multi-channel MAC protocol includes dedicated control channel protocols, common hopping protocols, split phase protocols. We take common hopping protocol for example to explain how centralized protocols work. In common hopping all the nodes jump from channel to channel together at the same pace. The transmission pair stops jumping and starts transmission in current channel while the other nodes continue to jump. When transmission ends, the transmission pair back to the jumping pattern again. In [1], authors suggested the time nodes dwell on each channel should be long enough to receive a collision-avoidance control packet from neighbor. Before sending data the sender sends RTS to receiver and then all nodes hop to next channel. If the sender receives the reply message from the receiver both of them stay on this channel and start transmitting data. [2] divides time into frame which is consisted of slots, and the slot is divided into sub-slot more: ACK, RTS, CTS. Each slot starts from ACK sub-slot. In this scheme, if more than one transmission starts in the same time slot, the collision will occur; therefore all of these potential transmissions can not continue (Fig. 1). [5] calls it “bottleneck problem in control channel” (Fig. 2).

The Fig.3 shows the scheme of decentralized protocol. In decentralized protocols each node holds a specific “home sequence”. The nodes jump following own “home sequence”. The sender tunes to the channel of the receiver when he has data to send, and then both of them stay in the current channel until complete the transmission. After the transmission, nodes return to own home sequence. MoMAC [3] and SSCH [4] are examples of this scheme. In [3], when a node has data to send, it deviates from home sequence (HM) and tunes to receiver’s channel with probability $P_{deviate}$. This decreases the probability of collision. In [4], authors suggest each node holds several HSs, which is uniquely determined. When a node wants to start a transmission, instead of tune to receivers channel it waits any receiver coming to his channel. If the sender wants to transmit with this receiver frequently it changes HS to the one which resemble to that of receivers. The decentralize protocols ease the bottleneck problem in the control channel but it brings a new problem, i.e., how does the sender gets the jumping information of the receiver. Both [3] and [4] assume the nodes got the hopping information at the first time they met each other. However, it takes long time to make all nodes know the HSs. We call it propagation time. To solve this problem mentioned above, we propose a multi-channel protocol, TDMA Initialized Multi-channel MAC protocol (TIMMAC).

3. PROPOSED PROTOCOL DESCRIPTION

3.1 Basic concept

Not only to decrease the propagation time but to solve the bottleneck problem in the control channel at the same time we propose a TDMA Initialized Multi-channel MAC protocol (TIMMAC). TIMMAC combines the centralized phase and decentralized phase. We also call centralized phase TDMA phase, because in this phase the nodes access the media with the way of TDMA. To get the time synchronization we assume each node is equipped with GPS.
In our protocol the centralized phase is divided into small slots including a special slot (SP) (Fig. 4). Each node occupies a specific time slots to broadcast its home sequence. The SP slot is prepared for the new comer. If there is no new comer it would be idle. After the SP slot all the nodes come into decentralized phase. In the decentralized phase the nodes jumps away following “home sequence”. The Fig. 5 shows the MAC frame structure in centralized phase. The frame includes header, state, home sequence and trailer 4 parts. The state part records the owner of each slot in current time. The home sequence part indicates the jumping sequence of the sender. After the centralized phase, each node gets the home sequence of all other’s and then they come to decentralized phase. To avoid collision the node does not send data immediately. Instead, the node must sense the channel first, after a random time the node starts transmission.

3.2 Node immigration

In Ad hoc network, the nodes come in and leave out of network frequently. So we should consider the migration and immigration of nodes.

When a node wants to join the network, it waits for the SP slot (Fig.6 (a)) and then broadcasts its home sequence in SP slot. After receiving the message from new comer, all the nodes add a new slot as SP slot. That’s because the old SP slot has been used by the new comer. The Fig.6 (b) shows the slot state after the joining of new comer. Some times there is more than one node comes at the same time such as 2 nodes. Both of them use SP broadcasting home sequence. Therefore, the collision
occurs (Fig.7 (a)). This collision can be sensed by some other nodes. These nodes set the SP state as "collision" (Fig.7 (b)). As they broadcasting frames in centralized phase, the collision state of SP is also propagated and all nodes add more one idle slot as receiving the collision notice (Fig.7 (c)) in the next frame. The comers use each special slot with probability $p_i$. Fig.6 (d) shows the frame after successful occupation.

Fig.6 Migration of one node

Fig.7 Migration of 2 nodes

3.3 Node emigration

This section discusses the leaving of a node. If a node wants to leave in the next frame, it broadcasts the leaving information in the current frame. The node who received this information spreads it in its own slot. In Fig.8, node A and node B are hidden terminals to each other. The node A broadcasts the leaving information in the n-th frame. But node B can not receive it immediately. Every node including node C duplicates and broadcasts it, so Node B gets the leaving message of node A in time (Fig.9).

3.4 The solution of the hidden terminal problem in TDMA phase.

This section presents the solution of hidden terminal problem. Let’s see Fig.8 again. In this figure the node A and the node B are out of range each other, node A does not know node B, and vice versa. Since both of them use special slot to send their home sequences concurrently, the collision occurs in the overlap area. Node C who is in the overlap area senses this collision, then C announces this collision in its slot. After receiving the collision notice, A and B know each other's existence. Them both of them add more one special slot for each other. Though they can not hear each other they take into account of each other actually and access special slot with probability $p_a$ and $p_b$. After the reservation node A and node B put a "used flag" in each other’s slot.

4. FUNDAMENTAL EVALUATION
In this section shows the evaluation results. We use simulations to evaluate our protocol. The simulator is Qualnet. First we compare the propagation time in TIMMAC and traditional decentralized protocol, McMAC. The parameters are shown in Table 2 and topology is shown in Fig. 10. In Fig. 10, the distance intra transmission pair is 237 m, while the distance inter transmission pair is 127 m. The bidirectional transmissions occur between the node A and the node B, the node C and the node D concurrently. We calculate how many frames have passed until the node gets all other nodes’ home sequence. The Fig. 11 shows the result of node A. In TIMMAC after 4 frames the node A get all the home sequence, while in McMAC it needs 13 frames. The table 3 shows the result of other nodes.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of packets</td>
<td>10,000</td>
</tr>
<tr>
<td>Packet arrival interval</td>
<td>10 ms</td>
</tr>
<tr>
<td>Channel capacity</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Application protocol</td>
<td>CBR</td>
</tr>
</tbody>
</table>

Table 2 Parameters.

![Fig. 10 Simulation topology](image)

Then to evaluate the throughput and end-to-end delay we assume that there are 9 nodes scattered randomly in a 500*500 square. We compare the throughput (Fig. 12) and end-to-end delay (Fig. 13) of TIMMAC and McMAC in different channels. From Fig. 12 we can know the throughput increases as the number of channels increases in both TIMMAC and McMAC, while the throughput of TIMMAC is better than that of McMAC. Fig. 13 shows the TIMMAC has less end-to-end delay compare to McMAC.

![Fig. 12 Throughput](image)

![Fig. 13 End-to-end delay](image)

<table>
<thead>
<tr>
<th></th>
<th>TIMMAC</th>
<th>McMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node A</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>Node B</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Node C</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>Node D</td>
<td>3</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 3 Result of propagation time.
5. CONCLUSION

In this paper we give a brief explanation about the current multi-channel protocols. They are classified into two types: centralized protocols and decentralized protocols and we address the problems of them, i.e., bottleneck problem and long propagation problem of home sequences. Then, we proposed a multi-channel MAC protocols named TIMMAC which combines the characteristics of the centralized protocols and the decentralized protocols not only to decrease the propagation time but also to solve the bottleneck problem in the control channel. We have shown TIMMAC has less propagation time comparing with traditional decentralized protocol, McMAC, using simulations. The results enable high throughput and low end-to-end delay. In the future, we will extensively investigate TIMMAC with different parameters to analyze it.

REFERENCE