A Group-Based Multi-Channel MAC Protocol for Wireless Ad Hoc Networks

Yung-Da Cheng [a, b] and Jang-Ping Sheu [b, c]
[a] Department of Electronic Engineering, Technology and Science Institute of Northern Taiwan, Taipei, Taiwan
[b] Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan
[c] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
E-mail: ydzheng@axp1.csie.ncu.edu.tw, sheujp@cs.nthu.edu.tw

Abstract

When we exploit multiple channels in MAC protocol, we can achieve a higher network throughput than using one single channel due to that multiple transmissions can take place simultaneously. In this paper, we proposed a novel group-based multi-channel MAC protocol which cannot only utilize multiple channels to transmit data packets but allow using multiple channels to propagate control packets. The protocol we presented is simple and suitable for wireless ad hoc networks with multiple available channels. The simulation results show that our protocol has the superior performances in network throughput to previous work.

Key words: MAC protocol, performance analysis, wireless ad hoc networks.

1. Introduction

IEEE 802.11 DCF standard is the most widely accepted medium access control (MAC) protocol in wireless ad hoc networks. There are lots of off-the-shelf network devices could be chosen for constructing a wireless ad hoc network. As the IEEE 802.11 wireless ad hoc networks become more and more popular, how to increase the network throughput becomes one of the urgent issues. Since the IEEE 802.11 DCF is a single channel protocol, its throughput is limited by the bandwidth. Nodes are inhibited from transmitting if there is a node transmitting in the same hop. In other words, only one communication can be established within a hop at any time. This may potentially reduce the network performance. To eliminate this problem, researchers have proposed protocols to exploit multi-channel capability. By using a multi-channel protocol, different nodes can transmit simultaneously on different channels. The concurrent transmissions will increase throughput of network.

As reviewed in Section 2, several multi-channel MAC protocols are proposed to improve the network throughput. The first category of them, named Dedicated Control Channel, are protocols that use a single dedicated control channel to exchange control information and use the rest channels for data transmission [8, 9, 10]. The second category of protocols, named Channel Hopping, let all network nodes hop together among channels and stop after agreement of transmission [1, 2, 3, 7, 11]. The third category of protocols, named Split Interval, divides channel time into fixed-time intervals using beacons, and divides each beacon interval into contention interval and data interval [5, 6]. In the contention interval, nodes in the network exchange only control message on a single control channel and make several agreements of communication. In the data interval, each communication pairs is assigned on one of the multiple channels for transmitting data.

In this paper, we proposed a multi-channel MAC protocol which splits the channel time into intervals just like the third category of protocols mentioned above. In the contention interval, the proposed protocol divides nodes into several groups, and lets nodes within each group negotiate in a distinct channel to make agreements of transmission. In the data interval, each pair with agreements is assigned on one of multiple channels for transmitting data. The main idea is to divide nodes into several groups in the contention interval. Thus, they can exchange control information in several channels at the same time rather than in a single channel as MMAC [6] does. The analysis and simulation results show that our protocol

978-1-4244-1890-9/07/$25.00 ©2007 IEEE

has the superior performances in network throughput to previous work.

The rest of this paper is organized as follows. In Section 2, we briefly review the related work. In Section 3, we present the proposed protocol in detail. In Section 4, we develop an analytical model to analyze the performance of our protocol. Simulation results are given in Section 5. Conclusion is made in Section 6.

2. Related Works

The multi-channel MAC protocols can be roughly classified into three categories: Dedicated Control Channel [8, 9, 10], Channel Hopping [1, 2, 3, 7, 11], and Split Interval [5, 6]. In the Dedicated Control Channel category, there is only one control channel to exchange control information at any time. Each node is equipped with two transceivers, a control transceiver and a data transceiver. The control transceiver operates on the control channel to exchange control packets with other nodes in the communication range and to obtain rights to access data channels. The data transceiver is able to dynamically switch to one of the data channels to transmit data packets and acknowledgements. One of the well-known protocols is the Dynamic Channel Allocation (DCA) protocol [8]. In DCA, when a sender intends to communicate with a receiver, the sender sends a RTS packet to the intended receiver through its control transceiver which operating on the control channel. After receiving the RTS packet, the receiver selects a data channel for subsequent communication and sends a CTS packet back to the sender with the selected channel number attached. After receiving the CTS packet, the sender responds a channel-reservation packet to the receiver. Then the sender and the receiver switch their data transceivers to the selected channel to communicate with each other.

In the Channel Hopping category, each node is equipped with one transceiver to send and receive packets. Time synchronization is needed between nodes in this approach. All network nodes have a common hopping sequence and hop together among channels. When all nodes hop into a channel, they start to negotiate for transmission. Once an agreement for transmission is made, the communication pair stops to hop and begin to transmit data. The rest nodes continue to hop to next channel in the hopping sequence to negotiate for the channel access right. One example of this category of protocol is Hop-Reservation Multiple Access (HRMA) protocol [11].

In the last, Split Interval, category of protocols, channel time is divided into fixed-time intervals using beacons, and each beacon interval is further divided into contention interval and data interval. In the contention interval, nodes in the network exchange only control message on a single control channel and make several agreements of communication. In the data interval, each communication pairs is assigned on one of the multiple channels for transmitting data. Time synchronization is also employed in this approach. The MMAC protocol proposed in [6] is the most well-known example which divides channel time into fixed-time intervals using beacons, and has a small window, called ATIM window at the start of each beacon interval to negotiate channels to be used during the rest of the beacon interval. MMAC uses one transceiver and adopts time synchronization. This MMAC protocol employs the beaconing mechanism in IEEE 802.11 power saving mode (PSM) to divide channel time into beacon intervals. In the ATIM window, every node listens to the default channel. Nodes that have packets to transmit negotiate channels with the destination nodes during ATIM window.

3. The Group-Based Multi-Channel MAC (GMAC) Protocol

To focus on the Split Interval category of multi-channel MAC protocols as described in last section, it is found that nodes are able to send data packets to each other on multiple non-overlap channels simultaneously in data interval, but nodes use only one single channel to exchange control information with each other during contention interval. Such kind of mechanism wastes most of channels bandwidth during contention intervals. Therefore, in this paper, we propose a novel group-based multi-channel MAC protocol for wireless ad hoc networks that not only makes multiple data packet transmissions possible but admits multiple control packet transmissions. Thus, more communication pairs can be formed in the contention interval and the network throughput can be increased.

Our protocol, named as GMAC hereafter, has the following assumptions. As illustrated in Fig. 3.1, there are k channels available and all channels have the same bandwidth. Channels are divided into m groups, where $m \leqq k$. Each node is equipped with one transceiver which can randomly switch among channels. Nodes are synchronized, such that all nodes begin and end their beacon intervals at the same time. Channel time is divided into an alternating sequence of beacon intervals. The beaconing mechanism is same as which of IEEE 802.11 power saving mode. A beacon interval is composed of a contention interval and a data interval. During a contention interval, nodes that have packets to transmit negotiate channels with the destination nodes. If some of nodes make agreements, they are admitted to transmit message during data interval.

In GMAC protocol, we divided channels and nodes into groups for improving throughput. Since there are *k* available channels and *m* groups, each group owns $\lfloor k/m \rfloor$ or $\lceil k/m \rceil$ channels for communication. Without loss of generality, we supposed that *k* can be exactly divided by *m* without remainder. Let $g = k/m$. It means that a group owns *g* channels. Assume there are *n* nodes in the whole networks. Thus, each group owns n/m member nodes. Assume each node has a unique ID. Any node is able to decide which group it will go into by calculating the remainder of the node ID divided by *m*. So each node knows which group itself belongs to and which groups the other nodes belong to if it has the node IDs. Since there are *g* channels in each group, we let one of them to be the contention channel on which the member nodes can negotiate during the contention interval. And we named that channel to be *group contention channel*, as indicated in Fig. 3.1. The *group contention channel* is the first channel of the group, so every node knows all of the *group contention channels*.

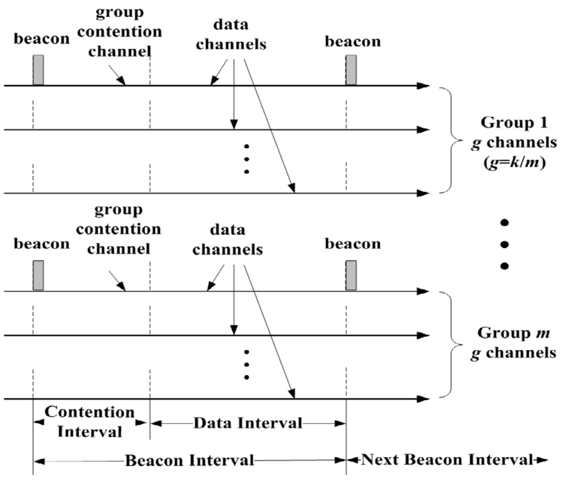

Figure 3.1: Group channel and time interval.

In the contention interval, each node should switch its transceiver to its *group contention channel* to contend channels with other member nodes. If a node makes agreement of transmission with another node, one of the *g* channels is selected for transmission in the data interval according to the channel selection rules which are the same as MMAC and will be briefed below. If a node wants to communicate with a destination node in another group, it will flip a coin with probability p_j to switch its transceiver to the *group contention channel* of that group. If it makes an agreement of transmission, it will be assigned a channel inside that destination group for data communication. The reason to flip a coin to switch transceiver is to increase the probability for source node and destination node to meet with each other at a same channel. We can tune the p_j to increase the meeting probability as calculated in following section..

Similar to [6], each node maintains a data structure called the preferable channel list (PCL) that indicates which channel is preferable for this node. PCL is used to record states of channel that inside the transmission range of this node. In our proposed protocol, each node records the states of channels inside its group. The channel states are categorized into the HIGH, MID, and LOW states just the same as MMAC [6]. The channel states are also changed in the same way as that of MMAC.

During contention interval, each node listens on its *group contention channel*. If a node *A* has a packet to send to another node *B*, node *A* will check first if node *B* is in another group. If it is, node *A* will flip a coin with probability p_j to switch its transceiver to node *B*'s *group contention channel*. Then, node *A* begins to transmit a Ch-Req packet including its PCL to node *B*. Upon receiving the Ch-Req packet, nodes *B* will select the most preferable channel according to the channel selection rules and send back a Ch-Ack packet including the information of selected channel to node *A*. After receiving the Ch-Ack packet, node *A* will transmit a Ch-Rsv packet including the selected channels information to inform nodes inside transmission range to update their PCL. If Ch-Rsv is successfully received by node *B*, the communication pair of nodes *A* and *B* is formed. All nodes overhear the Ch-Ack or the Ch-Rsv should update the state of the selected channels in their PCL. Both nodes *A* and *B* must update state of the selected channels to be HIGH.

Here we brief the channel selection rules of MMAC. Suppose that node *A* has packet for *B* and sends Ch-Req packet to *B* with *A*'s PCL attached. Node *B* examines *A*'s PCL and its PCL and selects channel according to following procedure.

1. If there is a HIGH state channel in *B*'s PCL, this channel is selected.
2. Else if there is a HIGH state channel in *A*'s PCL, this channel is selected.
3. Else if there is a channel which is in the MID state at both *A* and *B*, it is selected. If there are multiple channels in this state, one is selected arbitrarily.

4. Else if there is a channel which is in the MID state at only one side, *A* or *B*, it is selected. If there are multiple of them, one is selected arbitrarily.
5. If all of the channels are in the LOW state, add the counters of the sender's PCL and the receiver's PCL. The channel with the least count is selected. Ties are broken arbitrarily.

4. Analytical Model and Numerical Results

In this section, we present an analytical model to analyze the performance of the proposed protocol. According to the model, we derive numerical results under a certain operation condition. We made the following simplification for our protocol. Although the proposed protocol is able to operate in multi-hop environment, we suppose it is operating just in a single hop environment for analysis simplicity. Besides, all nodes of the network are in a saturation condition. That means the transmitting buffer of each node is never empty at any time.

A. The modal

As illustrated in Fig. 3.1, there are *k* channels and *n* nodes in the network. Channels and nodes are divided into *m* groups. The process to calculate the throughput of network can be divided into three steps:

1. Calculate the number of agreements made, i.e. communication pairs formed, during contention interval.
2. Calculate the number of communication pairs distributed into each channel according to the channel selection rules.
3. Calculate the number of packets transmitted on each channel during data interval. Thus, the throughput of each group can be derived and so does the network throughput.

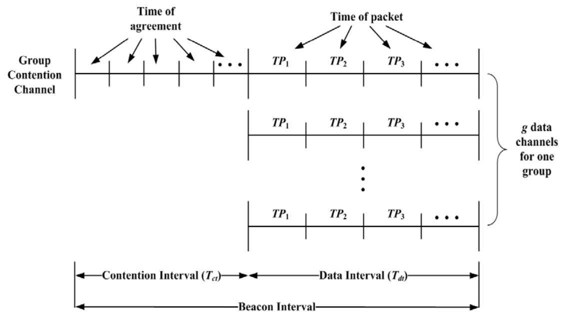

Figure 4.1: The time of agreement and the time of packet (TP_i)

Step 1: Calculate the number of agreements made

If the group number is just one, it is straightforward to calculate the number of agreements. For the assumption of saturation condition, every node in a group has at least one packet to transmit at any time. The traffic load during the contention interval is fixed and depends only on the number of nodes in a group. Let T_{ct} denote the time length of contention interval, T_a denote the average time duration of an agreement made, and N_a denote the number of agreements made. Based on the Bianchi model proposed in [4], T_a can be derived from following equation:

$$T_a = T_s + \sigma \frac{(1 - P_{tr})}{P_{tr} P_s} + T_c\left(\frac{1}{P_s} - 1\right) \tag{1}$$

where,

T_s is the average time when the channel is sensed busy because of a successful transmission.

T_c is the average time when the channel is sensed busy by each station during a collision.

σ is the duration of an empty slot time.

P_{tr} is the probability that there is at least one transmission in the considered slot time.

P_s is the probability that a transmission occurring on the channel is successful.

Equation (1) expresses the average amount of time spent on the channel in order to observe the successful transmission of a packet payload. In other words, it is the average time duration for making an agreement. Since T_a can be derived as (1) shown, we can further calculate the N_a as $N_a = \lfloor T_{ct}/T_a \rfloor$.

If the group number is more than one, nodes are evenly divided into groups according to their node IDs. Nodes intend to transmit packet may find their destination node are in another group. In other words, source node may miss to meet destination node during the contention interval. If the group number is *m*, the miss-meeting probability is $(m - 1)/m$. GMAC uses a flip coin mechanism to decrease the miss-meeting probability. If a node wants to communicate with a destination node in another group, it will flip a coin with probability p_j to switch its transceiver to the *group contention channel* of the destination node. This mechanism does increase the meeting probability as shown in following computation. Following notations are defined at first:

p_{sg}: the probability that destination node is in the same group as source node;

p_j: the probability that source node switch its transceiver to destination node's contention channel;

p_{st}: the probability that a node stays in its own group;

p_m: the probability that source node meet destination node in the same contention channel.

The probability p_{sg} is equal to $1/m$ if nodes are evenly distributed among m groups. The probability p_j is a factor we can tune to increase the p_m. The probability p_{st} is calculated as $p_{st} = p_{sg} + (1- p_{sg})(1- p_j)$. It means a node will stay in its own group in two cases. First, when it and its destination node are in the same group (p_{sg}), second, if its destination is in another group and it flips a coin and decides not to jump to another group (i.e. $(1- p_{sg})(1- p_j)$). To calculate p_m we suppose source node is *S* and destination node is *D*. Notice that each node, including node *D*, may jump to another group. Because node *D* has its own destination node, node *D* may jump to another group if its destination node is in another group. Therefore, node *S* has three chances to meet node *D*. First, node *D* is in the same group as node *S* and *D* stays in its original group. This probability is $p_{m1} = p_{sg} \times p_{st}$. Second, node *D* in another group, node *S* jumps to that group and node *D* stays in that group. This probability is $p_{m2} = (1\text{-}p_{sg}) \times p_j \times p_{st}$. Third, node *D* in another group, node *S* stays in its group and node *D* jumps into the group of node *S*. This probability is $p_{m3} = (1\text{-}p_{sg})(1\text{-}p_j)(1\text{-}p_{st})/(m\text{-}1)$. Finally, the probability p_m that node *S* meets node *D* can be obtained by $p_m = p_{m1} + p_{m2} + p_{m3}$. The whole equation is listed below.

$$\begin{aligned} p_m = {} & p_{sg} \times p_{st} + (1\text{-}p_{sg}) \times p_j \times p_{st} \\ & + (1\text{-}p_{sg})(1\text{-}p_j)(1\text{-}p_{st})/(m\text{-}1) \end{aligned} \tag{2}$$

Using (2) to calculate the meeting probability, we have the maximum meeting probability occurs when p_j is equal to 0.5. The meeting probability p_m is equal to 0.625, 0.5, 0.4375 and 0.375 for number of groups equal to 2, 3, 4 and 6, respectively.

Obviously, increasing the group number will lower the meeting probability and thus, decrease the number of agreements during contention interval. But, in the other hand, increasing the group number may divide nodes into more groups and lower the contention in each group and utilize more channels for contention. This may increase the number of agreements. To compute the number of agreements for multiple groups, we still use (1) to calculate the average time duration of an agreement. The only difference is the P_s probability in (1) should be multiplied with the probability p_m.

Step 2: Calculate the number of communication pairs distributed on each channel

Here, we discuss if the communication pairs formed in a contention interval can be evenly distributed into each data channel. Assume that there are q nodes to contend channels, g channels in a group, and r communication pairs formed in a contention interval. According to the channel selection rules, the first communication pair formed will be assigned to channel 1 definitely. The second communication pair formed has a probability p to be assigned to channel 1 and a probability 1 - p to be assigned to channel 2. This is because if the source node or destination node of second communication pair is just a same node as the source node or destination node of the first communication pair, the second communication pair will be assigned by channel selection algorithm to the same data channel as the first communication pair was. For example, if the first communication pair is node *A* and node *B*, and the second communication pair is node *B* and node *C*, the two communication pairs will be assigned to data channel 1 together. The probability p can be derived as follows.

$$p = \frac{2}{q}\frac{(q\text{-}2)}{q} + \frac{(q\text{-}2)}{q}\frac{2}{q} + \frac{2}{q}\frac{1}{q}$$

After the second pair assigned, the expected nodes number assigned to channel 1 can be derived as:

$$2 + \frac{2}{q} * \frac{(q\text{-}2)}{q} * 1 + \frac{(q\text{-}2)}{q} * \frac{2}{q} * 1 + \frac{2}{q} * \frac{1}{q} * 0$$

The expected nodes number assigned to channel 2 can be derived as:

$$\frac{(q\text{-}2)}{q} * \frac{(q\text{-}3)}{q} * 2$$

And, the expected nodes number assigned to channel 3 through channel g are all 0.

Applying the same method, we can assign channel for the third pair through the r'th pair. To describe this in a more general form, we define the following notations.

$P_{i,j}$: The probability that the i'th communication pair is assigned to channel j.

Ps_i: The probability that the i'th communication pair is successfully assigned to any channel.

$N_{i,j}$: The expected number of nodes assigned to channel j after the i'th communication was assigned.

Nu_i: The expected number of nodes unassigned to any channel after the i'th communication was assigned.

C_j: The total expected number of communication pairs assigned to channel j in data interval.

Ps_i can be computed as following equation:

$$Ps_i = \sum_{\mathrm{j}} P_{i,j}$$

Nu_i can be computed as following equation:

$$Nu_i = q - \sum_{j} N_{i,j}$$

$P_{i,j}$ can be computed as following equations:

$$P_{1,1} = 1 \tag{3-1}$$

$$P_{i,j} = 0 \quad \text{for } i < j \tag{3-2}$$

$$P_{i,j} = \frac{N_{i-1,j}}{q}\frac{Nu_{i-1}}{q} + \frac{Nu_{i-1}}{q}\frac{N_{i-1,j}}{q} + \frac{N_{i-1,j}}{q}\frac{N_{i-1,j}-1}{q} \quad \text{for } i \in (2,g),\ j \in (1,i\text{-}1) \tag{3-3}$$

$$P_{i,j} = \frac{Nu_{i-1}}{q}\frac{Nu_{i-1}-1}{q} \quad \text{for } i \in (2,g),\ j = i \tag{3-4}$$

$$P_{i,j} = \frac{N_{i-1,j}}{q}\frac{Nu_{i-1}}{q} + \frac{Nu_{i-1}}{q}\frac{N_{i-1,j}}{q} + \frac{N_{i-1,j}}{q}\frac{N_{i-1,j}-1}{q} + \lambda_{i,j} \quad \text{for } i \in (g+1,r)\ \ j \in (1,g) \tag{3-5}$$

$$\lambda_{i,j} = \begin{cases} \dfrac{Nu_{i-1}}{q}\dfrac{Nu_{i-1}-1}{q} & \text{if } C_j = \text{Min}(C_1 \ldots C_g) \\ = 0 & \text{else} \end{cases} \tag{3-6}$$

Equation (3-1) expresses that the first communication pair will be assigned to channel 1 definitely. Equation (3-2) expresses that the *i*'th pair is impossible to be assigned to channel *j* if *j* is greater than *i*. The only channels for second pair to be assigned are channel 1 or channel 2. Equation (3-3) expresses the probability that the *i*'th pair to be assigned to channel that has been previously assigned communication pairs when there are still empty channels. Equation (3-4) expresses the probability that the *i*'th pair to be assigned to an empty channel *j*. Equation (3-5) expresses the probability that the *i*'th pair to be assigned to channel that has been previously assigned communication pairs when there is no empty channel. If the nodes of the *i*'th pair are the same as any nodes of the already assigned pair, the *i*'th pair should be assigned to the same channel as the already assigned pair. This is expressed in the first 3 terms of (3-5). If the nodes of the *i*'th pair are not the same as any nodes of the already assigned pairs, the *i*'th pair will be assigned to a channel that was assigned least pairs. This is expressed in the fourth term, $\lambda_{i,j}$, of (3-5).

$N_{i,j}$ can be computed as following equations:

$$N_{1,1} = 2 \tag{4-1}$$

$$N_{i,j} = 0 \qquad \text{for } i < j \tag{4-2}$$

$$N_{i,j} = N_{i-1,j} + \left(\frac{1}{Ps_{i-1}}\right) * \left(\frac{N_{i-1,j}}{q}\frac{Nu_{i-1}}{q} * 1 + \frac{Nu_{i-1}}{q}\frac{N_{i-1,j}}{q} * 1 + \frac{N_{i-1,j}}{q}\frac{N_{i-1,j}-1}{q} * 0 \right) \quad \text{for } i \in (2,g),\ j \in (1,i\text{-}1) \tag{4-3}$$

$$N_{i,j} = \frac{1}{Ps_{i-1}}\frac{Nu_{i-1}}{q}\frac{Nu_{i-1}-1}{q} * 2 \quad \text{for } i \in (2,g),\ j = i \tag{4-4}$$

$$N_{i,j} = N_{i-1,j} + \left(\frac{1}{Ps_{i-1}}\right) * \left(\frac{N_{i-1,j}}{q}\frac{Nu_{i-1}}{q} * 1 + \frac{Nu_{i-1}}{q}\frac{N_{i-1,j}}{q} * 1 + \frac{N_{i-1,j}}{q}\frac{N_{i-1,j}-1}{q} * 0 + \lambda_{i,j} * 2 \right) \quad \text{for } i \in (g+1,r),\ j \in (1,g) \tag{4-5}$$

Equation (4-1) expresses that the first communication pair will be assigned to channel 1 definitely. So, the expected number of nodes assigned to channel 1 after the first communication pair assigned is 2. Equation (4-2) expresses that the *i*'th pair is impossible to be assigned to channel *j* if *j* is greater than *i*. Equation (4-3) expresses the expected nodes number of channel *j* after *i*'th pair assigned when there are still empty channels. The first term of the equation is the original expected nodes number after the previous pair assigned. The rest terms of this equation express the expected nodes number to be assigned to channel *j* for the *i*'th pair assignment. Equation (4-4) expresses the expected nodes number to be assigned to an empty channel *j*. Equation (4-5) expresses the expected nodes number of channel *j* after *i*'th pair assigned when there is no empty channel.

For given *q, r,* and *g,* we can compute all $P_{i,j}$ and $N_{i,j}$ for $i \in (1,r)$ and $j \in (1,g)$. The expected value of the number of communication pairs assigned to each channel can be further derived as follow equation:

$$C_j = \sum_{i=1}^{r} P_{i,j}/Ps_i \qquad j \in (1,g) \ ,$$

Step 3: Calculate the number of packets transmitted on each channel during data interval

As illustrated in Fig. 4.1, TP_i denotes the time duration of transmitting the *i*'th packet in data interval. Let T_{dt} denote the time length of data interval. TP_i can be computed from (1). Since we have already known the number of communication pairs assigned to each data channel, i.e. C_j, we can compute the number of packets transmitted for a given T_{dt} for each channel. Thus, throughput of a group can be computed and throughput of the whole network can be derived.

B. Numerical results

The values of parameters used to obtain the time of agreement T_a are the same as which in [4]. The channel rate is set to be 1 Mbps. Thus, the transmission time of one bit is just 1μs. Payload of a packet is fixed to be 8184 bits. Propagation delay is 1μs. Empty slot time σ is 50μs. SIFS is 28μs. DIFS is 128μs. RTS is 288μs. CTS is 240μs. ACK is 240μs.

Ch-Req packet length is the RTS packet length plus the data length of PCL data structure. For comparing with MMAC [6] on the same base, we set T_{ct} to 20 ms and T_{dt} to 80 ms. We compute the throughput of several network scenarios. Set k to be 3, 6, and 12, set group number to be 1, 2, 3, and 6, and set node number to be 50, 100, and 200. The analysis results are illustrated in section 5 for comparing with simulation results.

5. Simulations

We use Glomosim, a discrete event simulator developed by UCLA, to implement our simulations. The propagation range of each node is 250 meters. All nodes can communicate with each other in one hop. Each node generates and transmits constant-bit rate (CBR) traffic. Each node randomly selects a node as its destination node. The traffic load is heavy enough to let each node in a saturation state. Channel bit rate is set to 1 Mbps. Packet size is fixed as 8284 bits. Contention interval length is set to 20 ms. Data interval length is set to 80 ms. Each simulation was performed for duration of 100 seconds.

In Fig 5.1, channel number k is set to be 3. We simulate and analyze 50, 100, and 200 nodes to be divided into 1 group and 3 groups, respectively. Network throughput are measured and computed. If the group number is set to 1 in GMAC, it is just the same as MMAC. Notice that when group number is set to 1 and node number is set to 50, the throughput is about 1 Mbps which is compatible with the throughput value simulated by MMAC [6]. The throughputs of 3 groups are better than the throughputs of 1 group (MMAC), no matter the node numbers are 50, 100, or 200. The analysis results are very near to the simulation results, especially when the node number is large enough.

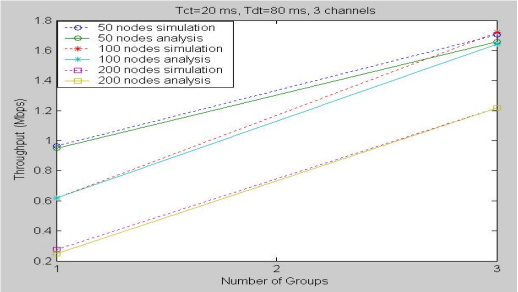

Figure 5.1: The throughput vs. number of groups for 3 channels

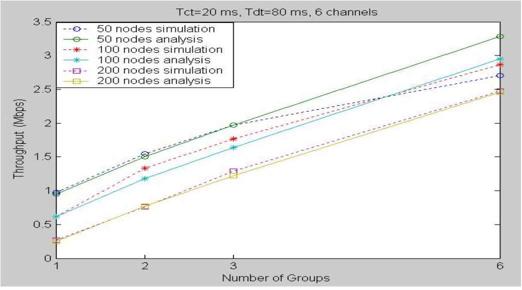

Figure 5.2: The throughput vs. number of groups for 6 channels

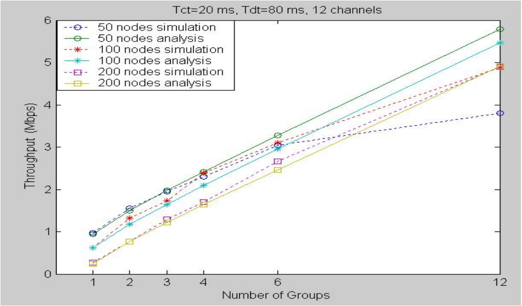

Figure 5.3: The throughput vs. number of groups for 12 channels

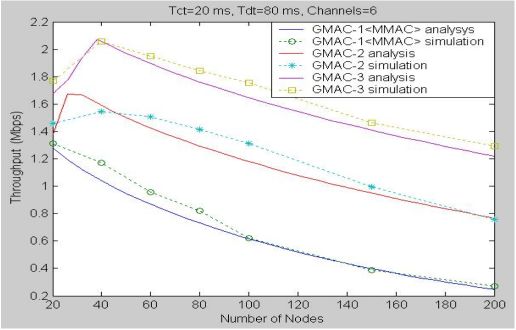

Figure 5.4: The throughput vs. number of nodes for 6 channels

In Fig. 5.2, channel number k is set to be 6. We simulate and analyze 50, 100, and 200 nodes to be divided into 1 group, 2 groups, 3 groups and 6 groups respectively. Network throughput are measured and computed. In Fig. 5.3, channel number k is set to be 12. We simulate and analyze 50, 100, and 200 nodes to be divided into 1 group, 2 groups, 3 groups, 4 groups, 6groups and 12 groups respectively. GMAC performs better than MMAC does, no matter the node numbers are 50, 100, or 200.

In Fig. 5.4, we check the impact of node number to the network throughput. The channel number k is set to 6. GMAC-1 denotes the setting of 1 group which behaves the same as MMAC. GMAC-2 and GMAC-3 denote the setting of 2 groups and 3 groups, respectively. The simulation results are very close to the prediction of analysis. GMAC-3 performs better than GMAC-2 and GMAC-2 performs better than GMAC-1 (MMAC).

6. Conclusion

In this paper, we have presented a GMAC protocol that utilizes multiple channels to improve throughput in wireless networks by dividing nodes into groups. Because of dividing nodes into groups, nodes can negotiate channels with their destination nodes in multiple channels rather than in a single channel as MMAC does. Analysis and simulation results show that GMAC successfully exploits multiple channels to improve total network throughput over MMAC. The performance of GMAC and MMAC depends on the network situation, but as the results show, GMAC performs better or at least comparable to MMAC. Since GMAC utilizes multiple channel during contention interval rather than single channel so that GMAC can raise the probability of transmission control packets in contention interval and increase network throughput. In addition, we proposed an analytical model to compute the network throughput. The simulation results prove the correction of the analytical model.

7. References

[1] Asimakis Tzamaloukas and J.J. Garcia-Luna-Aceves, “A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks,” in Proc. of IEEE INFOCOM, vol.1, pp. 189 – 198, Anchorage, AK, USA, April 2001.

[2] Asimakis Tzamaloukas and J.J. Garcia-Luna-Aceves, “Channel-Hopping Multiple Access,” in Proc. of IEEE International Conference on Communications, Vol. 1, pp. 415 – 419, New Orleans, LA, USA, June 2000.

[3] Asimakis Tzamaloukas and J. J. Garcia-Luna-Aceeves, “Channel Hopping Multiple Access with Packet Trains for Ad Hoc Networks”, in Proc. of IEEE Mobile Multimedia Communications, Tokyo, Japan, November 2000.

[4] Giuseppe Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function”, IEEE journal on Selected Areas in Communications, Vol. 18, pp. 535 – 547, March 2000.

[5] Jenhui Chen, Shiann-Tsong and Chin-An Yang, “A New Multichannel Access Protocol for IEEE 802.11 Ad Hoc Wireless LANs”, in Proc. of Personal, Indoor and Mobile Radio Communications (PIMRC), Vol. 3, pp. 2291 – 2296, Beijing, China, September 2003.

[6] Jungmin So and Nitin Vaidya, “Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver,” in Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 222 – 233, Roppongi Hills, Tokyo, Japan, May, 2004.

[7] Paramvir Bahl, Ranveer Chandra and John Dunagan, “SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks”, in Proc. of International Conference on Mobile Computing and Networking (MobiCom), pp. 216 – 230, Philadelphia, PA, USA, September, 2004.

[8] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng and Jang-Ping Sheu, “A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks,” in Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN), pp. 232-237, Dallas, TX, USA, Dec. 2000.

[9] S.-L. Wu, Yu-Chee Tseng, Chih-Yu Lin and Jang-Ping Sheu, “A Multi-Channel MAC Protocol with Power Control for Multi-hop Mobile Ad Hoc Networks,” The Computer Journal, Vol. 45, No. 1, pp. 101-110, 2002.

[10] Wing-Chung Hung, K.L.Eddie Law, A. Leon-Garcia, “A Dynamic Multi-Channel MAC for Ad Hoc LAN,” in Proc. of 21st Biennial Symposium on Communications, pp. 31-35, Kingston, Canada, June 2002.

[11] Zhenyu Tang and J.J Garcia-Luna-Aceves, “Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks,” in Proc. of IEEE INFOCOM, New York, NY, USA, pp. 194 – 201, March, 1999.