TMMAC: A TDMA Based Multi-Channel MAC Protocol using a Single Radio Transceiver for Mobile Ad Hoc Networks

Jingbin Zhang †, Gang Zhou †, Chengdu Huang ‡, Ting Yan †, Sang H. Son †, John A. Stankovic †

† Department of Computer Science, University of Virginia

‡ Department of Computer Science, University of Illinois

{jz7q, gzhou, ty4k, son ,stankovic}@cs.virginia.edu chuang30@cs.uiuc.edu

Abstract

In this paper, we propose a medium access control (MAC) protocol called TMMAC for Mobile Ad Hoc Networks (MANETs). TMMAC requires only a single half-duplex radio transceiver on each node. In addition to explicit frequency negotiation which is adopted by the conventional multi-channel MAC protocols, TMMAC introduces lightweight explicit time negotiation. This two-dimension negotiation enables TMMAC to exploit the advantage of both frequency diversity and TDMA. Performance evaluation results show that TMMAC achieves up to 62% higher communication throughput over the state-of-the-art multi-channel MAC protocols for single-transceiver wireless devices. Moreover, through this two-dimension negotiation, TMMAC achieves aggressive power saving by allowing nodes that are not involved in communication to go into doze mode. In addition, broadcast is supported in an effective way in TMMAC. We also study the impact of different interference ranges on TMMAC. Based on that, we propose Differentiated Power TMMAC to improve the performance of TMMAC when the interference range is much larger than the communication range. Differentiated Power TMMAC maintains the significant improvement of TMMAC over the state-of-the-art multi-channel MAC protocols when the interference range is 1.75 times the communication range.

Categories and Subject Descriptors: C.2.2 [Computer-Communication Networks]: Network Protocols

General Terms: Algorithm

Keywords: Mobile Ad Hoc Networks, Medium Access Control, Multi-Channel, TDMA.

1 Introduction

Media access control is an essential part of the wireless communication stack and it has obtained intensive research attention. Among them, multi-channel MAC design emerges as a hot topic, when frequency diversity is explored to reduce radio congestion and provide higher communication throughput.

This paper focuses on how to incorporate both the advantages of frequency diversity and TDMA into the MAC design without heavy overhead, when each node in the network is only equipped with a single half-duplex radio transceiver. Typically, such hardware can not transmit and receive at the same time, but it can dynamically switch its frequency. Many of the previous multi-channel MAC designs [31] [22] [3] require multiple radio transceivers. Multiple radios not only result in higher product prices, but also consume more power from energy-constrained devices. Moreover, most current IEEE 802.11 devices are equipped with a single half-duplex transceiver. Therefore, a MAC protocol based on a single half-duplex transceiver is more adaptable to IEEE 802.11 [1] than the MAC protocols requiring multiple transceivers.

In this single transceiver context, conventional multi-channel MAC designs adopt explicit frequency negotiation [24] [12] [17] [14] [29], through certain kinds of control messages. This one-dimension negotiation enables these MAC protocols to adopt the advantage of frequency diversity and achieve better performance than IEEE 802.11.

In this paper, we propose a TDMA based multi-channel MAC protocol called TMMAC. In addition to conventional frequency negotiation, TMMAC introduces lightweight explicit time negotiation. In TMMAC, time is divided into fixed periods, which consists of a negotiation window followed by a communication window. The communication window is slotted and the duration of each time slot is the time needed for a single data packet transmission or reception. During the negotiation window, each node decides not only which channels to use, but also which time slots to use for data communication. Then each node adopts the negotiated frequency for each time slot to transmit or receive data packets. From the TDMA's point of view, TMMAC is a traffic-adaptive and energy-efficient TDMA scheduling algorithm.

This two-dimension negotiation enables TMMAC to take full advantages of both frequency diversity and TDMA. It has the following advantages over the conventional one-dimension frequency negotiation based MAC protocols:

- In the typical frequency negotiation based MAC protocols, a node first negotiates which channel to use with its destination. After negotiation, a node stays in the negotiated frequency for a specified period of time for data communication. During that time, a node can not communicate with other nodes, which use different frequencies even when this node is idle, due to the lack of time negotiation. In this case, the bandwidth may not be fully utilized. However in TMMAC, through explicit frequency negotiation plus explicit time negotiation, nodes can use different frequencies to communicate with different neighbors, within different time slots. This property allows TMMAC to achieve more efficient bandwidth usage and improves the throughput.

- TMMAC avoids contention based data communication in the communication window. In the conventional frequency negotiation based MAC protocols, normally RTS/CTS handshaking is used to avoid collisions, if two data transmissions using the same frequency are within two hops. However in TMMAC, collisions are avoided by putting the two data transmissions into two different time slots. Our experiment shows that TMMAC makes a better choice, especially when the system load is low.

- In TMMAC, a node can go into doze mode in a time slot whenever it is not scheduled to transmit or receive a packet. This property allows TMMAC to achieve aggressive power saving.

- TMMAC supports broadcast very efficiently. In TMMAC, a node asks all its neighbors to switch to the same frequency only at the time slots when it transmits the broadcast packets. In this case, the time needed for all its neighbors to use the same frequency becomes minimal and each broadcast message only needs to be sent once.

The rest of this paper is organized as follows. We review related work in Section 2. Then we present the details of the TMMAC protocol design in Section 3. Section 4 presents some discussion and optimization of TMMAC. Section 5 contains a complete evaluation of TMMAC. In Section 6, we study the impact of different interference ranges on TMMAC and propose Differentiated Power TMMAC to address this issue. We give the conclusions and future work in Section 7.

2 Related Work

A large number of multi-channel MAC protocols and TDMA scheduling algorithms have been proposed in the literature and put into use. Many multi-channel protocols are based on special hardware assumptions. [26] [28] assume the use of frequency hopping spread spectrum (FHSS) wireless cards, and in [11] the busy-tone ability is required for the radio hardware. In [18] [31] [19] [22] [3], either multiple radio transceivers or a single sophisticated transceiver is required to be capable of listening to multiple frequencies at the same time. For example, in Dynamic Channel Assignment (DCA) [31], two radio transceivers are used. One listens to the control channel and the other listens to the data channel simultaneously. In TMMAC, we do not have such special hardware requirements. TMMAC only requires a single radio transceiver that can work on different frequencies at different times.

Multi-channel MAC protocols that are closely related to TMMAC are the ones that extend IEEE 802.11 Distributed Coordination Function (DCF) protocol [1] and use certain kinds of control messages for frequency negotiation. Typical protocols in this group are [24] [12] [17] [14] [29]. Among these protocols, MMAC [24] is the most related one to TMMAC. MMAC assumes time synchronization in the network and the time is divided into fixed-length beacon intervals. Each beacon interval consists of a fixed-length ATIM (Ad Hoc Traffic Indication Messages) window, followed by a communication window. During the ATIM window, every node listens to the same default channel and uses ATIM, ATIM-ACK (ATIM-Acknowledgement) and ATIM-RES (ATIM-Reservation) packets to negotiate which channel to use for data transmission or reception. After the ATIM window ends, nodes that have successfully negotiated channels with their destinations send out data packets using IEEE 802.11 DCF [1], i.e. exchanging RTS/CTS before sending out data packets, for channel access and congestion avoidance. Nodes that do not achieve successful negotiations can go into doze mode to save power. From simulation studies, MMAC successfully exploits multiple channels to achieve higher throughput than IEEE 802.11.

Besides the explicit frequency negotiation, pseudo random number generators are also used in [4] to help frequency allocation and switches. Nodes have different random numbers at different times, and communication is allowed when neighboring nodes have overlapping numbers.

There are also many TDMA scheduling algorithms [10][9][27][21][5][23] proposed for ad hoc networks in the literature. These TDMA scheduling algorithms are mainly designed for sharing a single channel in the network and providing collision-free channel access scheduling in that single channel. In these protocols, frequency diversity is not exploited to improve communication throughput.

3 Design of TMMAC

In this section, we present the TMMAC protocol. Like IEEE 802.11 Power Saving Mechanism (PSM) [1] and MMAC [24], TMMAC divides the time into fixed-length beacon intervals and uses a small ATIM window at the beginning of each beacon interval for negotiations. During the ATIM window, four types of messages are used for negotiations, which are ATIM, ATIM-ACK, ATIM-RES and ATIM-BRD (ATIM-Broadcast) packets. The ATIM, ATIM-ACK and ATIM-RES packets are used for unicast negotiations and the ATIM-BRD packets are used for broadcast negotiations. During the negotiation, each node decides which frequency to use for each time slot for data communication. Then at each time slot, each node adopts the negotiated frequency to transmit or receive data packets.

Before proceeding to the detailed description of TMMAC, we first list all the assumptions which we use for TMMAC as follows:

- Each node only has a single half-duplex transceiver. The transceiver can not send a message and receive a message simultaneously. Also, the transceiver can not access multiple channels at the same time.

- Every half-duplex transceiver has N available channels. All the N available channels have the same bandwidth. The channels are well separated and the communications in different channels do not noticeably interfere with each other.

- Every half-duplex transceiver can switch its channel dynamically. In our implementation, the time needed for switching the channel is set to $224us$ according to IEEE 802.11 [1].

- We assume all the nodes are synchronized. In this paper, we do not go deep into how the nodes are synchronized. Either GPS [13] or solutions like IEEE 802.11 timing synchronization function (TSF) [1] can be applied. The protocol tolerates a certain level of time drift among different nodes by expanding the length of each time slot used to send or receive a message.

- We define the interference range as follows. Suppose node B is at the edge of node A's communication range as shown in Figure 1, and node A is sending a packet to node B, the interference range is the maximum distance from node D to node B when node D's transmission cause node B to fail to receive the packet from node A. Here we only consider the interference from a single node when we define the interference range. If we consider the interference from multiple nodes, the actual interference range is larger than the one defined above [33]. For ease of description, we first assume that the interference range is similar or slightly larger than the communication range. This is a common assumption used by many existing TDMA scheduling algorithm, such as NAMA [5] and TRAMA [23] (NAMA and TRAMA assume that the communication range is the same as the interference range). However, in reality, this assumption may not be true [33]. We discuss the impact of different interference ranges on TMMAC in Section 6 and propose a solution called Differentiated Power TMMAC in Section 5.4 to address this issue.

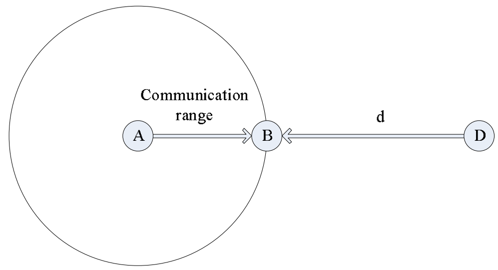


Figure 1: Interference range.

3.1 Main Data Structures

To present the TMMAC in detail, we first describe three main data structures used in TMMAC. They are Channel Usage Bitmap (CUB), Channel Allocation Bitmap (CAB), and Task Queue. The following subsections describe these three data structures, respectively.

3.1.1 Channel Usage Bitmap

The Channel Usage Bitmap is the main data structure that needs to be maintained at each node in TMMAC. Each node maintains a list of CUBs, each of which represents the usage information of one channel. The list of all the CUBs in each node is called the Channel Usage Bitmap List (CUB List). If the radio transceiver has N available channels, the CUB List in each node is comprised of N CUBs. We use CUB$_i$ to denote the CUB for channel i. Figure 2 shows an example of a CUB List.

	Slot 1	Slot 2	Slot 3	Slot 4	...	Slot M
CUB$_1$	1	1	0	0	...	0
...	...	...	...	...	...	...
CUB$_N$	1	0	0	0	...	1

Figure 2: An example of a CUB List.

As shown in Figure 2, CUB$_i$ in the CUB List represents the usage information of channel i at different time slots for the current beacon interval. If the beacon interval has M time slots, each CUB contains M elements in it. The value of M is determined by the length of the communication window and the length of the time slot, which is discussed in more detail in Section 3.2.3 and Section 3.3. We use C_iS_j to denote the j^{th} element in CUB$_i$. Each element C_iS_j is comprised of a single bit, in which 1 means that channel i at time slot j is already allocated by its neighbors or itself and 0 means that channel i at time slot j is still available for allocation.

The rules to change the values of the CUB List are described as follows:

1. The value of each element in the CUB List is reset to 0 when the node is powered up or it is at the start of each beacon interval.

2. For unicast, if the sender and receiver nodes agree on using channel i at time slot j for data transmission, the elements

at time slot j of all the CUBs in both the sender and receiver are set to 1. For broadcast, if the sender decides to send a broadcast message at time slot j using channel i, the elements at time slot j of all the CUBs in the sender are set to 1.

3. If a node nearby overhears an ATIM-ACK, or ATIM-RES packet, which contains the Channel Allocation Bitmap List (CAB List), C_iS_j of its CUB List is set to 1 if C_iS_j of the CAB List contained in the packet is 1. If a node overhears an ATIM-BRD packet, the elements at time slot j of all its CUBs are set to 1 if C_iS_j of the CAB List contained in the packet is 1.

3.1.2 Channel Allocation Bitmap

The Channel Allocation Bitmap (CAB) has the same data structure as CUB. The list of all the CABs is called the Channel Allocation Bitmap List (CAB List). We use CAB$_i$ to represent the allocation information of channel i of a single negotiation. C_iS_j is also used to denote the j^{th} element in CAB$_i$.

The CAB List contains the result of a single negotiation during the ATIM window and the CAB List is transmitted along with ATIM-ACK, ATIM-RES or ATIM-BRD packets, indicating which channels and which time slots are allocated by this negotiation. In the CAB List, if the value of C_iS_j is 1, it means that channel i at time slot j at this beacon interval is negotiated to be allocated for this transmission and 0 means that this channel at this time slot is not allocated.

3.1.3 Task Queue

A task queue maintains the scheduling information of a node for the current beacon interval. If the beacon interval has M time slots, a task queue contains M items in it. Each item in a task queue contains three fields: State, Address and Channel as shown in Figure 3. The State field has to be one of these three values: SEND, RECEIVE or DOZE.

Item 1	State: SEND	Address: Node TO	Channel: 1
Item 2	State: DOZE	Address: NULL	Channel: NULL
…	…	…	…
Item M	State: RECEIVE	Address: Node FROM	Channel:2

Figure 3: An example of a task queue.

If a node has negotiated to send a packet at time slot i, the State field of the i^{th} item is set to SEND. Then the Address field of the i^{th} item contains the receiver's address and the Channel field includes the channel number to be used for this transmission. If a node is negotiated to receive a packet at time slot i, the State field of the i^{th} item is set to RECEIVE. Then the Address field contains the sender's address and the Channel field includes the channel number to be used for this transmission. If a node neither sends a packet nor receives a packet at time slot i, the State field is set to DOZE. This node can go into doze mode at time slot i to save power.

3.2 Overall Architecture of TMMAC

Figure 4 shows the overall architecture of TMMAC. As shown in the figure, the time in TMMAC is divided into fixed-length beacon intervals and each beacon interval is comprised of a small ATIM window and a communication window. The communication window is further divided into time slots. Each time slot should be long enough to accommodate a data packet transmission.

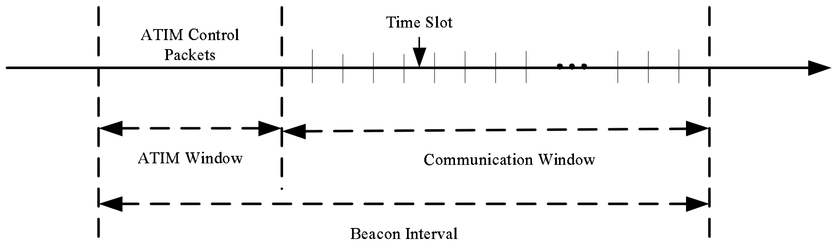


Figure 4: Overall architecture of TMMAC.

During the ATIM window, all the nodes listen to the same default channel. Four types of messages are used for negotiation: ATIM, ATIM-ACK, ATIM-RES and ATIM-BRD. They are called ATIM control packets. In TMMAC, the communication during the ATIM window is contention based and uses the same scheme as the one used in IEEE 802.11 DCF. During the negotiation, the sender and the receiver decide which channels and which time slots to use for data communication.

It is desirable to keep the size of the ATIM window as small as possible to leave more bandwidth for the communication window and to save more power. However, it also needs to be large enough to accommodate enough successful negotiations to make efficient use of the communication window. Section 5.4 discusses the impact of different ATIM window sizes on the performance of TMMAC in more detail.

3.2.1 Unicast Negotiation

For unicast packets, the ATIM, ATIM-ACK and ATIM-RES packets are used for negotiation. If node S wants to send some packets to node R, node S first sends an ATIM packet to node R containing its CUB List and the number of packets it wants to send. After receiving the ATIM packet, node R decides which channels and which time slots to use based on the information of its CUB List and the CUB List from the sender S. The selection procedures used by node R are described as follows:

First, node R does an OR operation on node S's CUB List and its own CUB List to generate a temporally combined CUB List. If the value of C_iS_j in the combined CUB List is 1, it means that channel i at time slot j can not be allocated for this transmission. If the value of C_iS_j is 0, it means that channel i at time slot j can be allocated for this transmission. If there are multiple available channels at one time slot, at most one channel can be chosen for data transmission, because a node has a single half-duplex radio transceiver in TMMAC.

Second, following the first rule, node R randomly select channels and time slots, among available ones from the combined CUB List for this transmission. If node R can not allocate enough time slots as the packet number specified in the ATIM packet, node R allocates as many time slots as possible. In this case, some packets in node S can not be transmitted in this beacon interval and it needs to wait for the following beacon intervals.

After node R decides which channels and which time slots to use for this transmission, node R updates its CUB List and task queue, and generates the corresponding CAB List. If C_iS_j is selected for transmission, C_iS_j in the CAB List is set to 1. Otherwise it is set to 0.

Then node R replies back with an ATIM-ACK message, which contains the generated CAB List. The nodes in the vicinity of node R can update their CUB Lists by overhearing the ATIM-ACK message based on the rules described in Section 3.1.1. The neighbors of node R can determine the current channel usage information of their neighbors. After receiving the ATIM-ACK packet from node R, node S updates its CUB List and task queue based on the CAB List from node R and sends out an ATIM-RES packet containing the same CAB List to node R. By overhearing the ATIM-RES packet, the nodes near node S update their CUB Lists based on the rules described in Section 3.1.1 to obtain the current channel usage information of their neighbors.

3.2.2 Broadcast Negotiation

For broadcast packets, the ATIM-BRD packets are used for negotiation. If node S has some packets to broadcast during this beacon interval, node S first selects the channels and the time slots it wants to use based on its own CUB List. The selection procedures for ATIM-BRD are described as follows:

First, node S selects the time slots, at which all the channels are not used by any of its neighbors yet. If node S can not allocate enough time slots for the broadcast messages, node S allocates as many time slots as possible.

Second, after node S allocates the time slots for the broadcast packets, node S can select any channel for each chosen time slot for the broadcast messages. Only one channel can be selected for each chosen time slot.

After node S decides which channels and which time slots to use for the broadcast packets, node S updates its CUB List and task queue, and generates the corresponding CAB List. If C_iS_j is selected, C_iS_j in the CAB List is set to 1. Otherwise, it is set to 0.

Then node S broadcasts the ATIM-BRD packet with the CAB List in it. By overhearing the ATIM-BRD packet, the nodes in the vicinity of node S learn at which time slots they will receive the broadcast packets and which channels to use, and update their CUB Lists based on the rules described in Section 3.1.1. The values of the corresponding items in the task queues are also updated.

3.2.3 Data Packet Transmission

After the ATIM window, nodes can start to send packets, receive packets or go into doze mode based on its task queue. At the i^{th} time slot, if the i^{th} item of the task queue is set to SEND, the node first switches its channel to the specified channel when it enters the i^{th} time slot and then transmits the packet which is destined to the specified receiver. To avoid head of line blocking problem in a single FIFO queue, in which a packet needs to be sent is blocked by the packet at the top of the queue, TMMAC maintains per-neighbor FIFO queues. The node which is negotiated to receive the packet at the i^{th} time slot also first switches its channel to the specified channel and then waits for the incoming packet. If the incoming packet is unicast, the receiver sends back an acknowledgement after it receives it. If the incoming packet is broadcast, the receiver does not send back an acknowledgement. If the sender does not receive the acknowledgement after it sends out a unicast packet, that packet is retransmitted at the next available time slot. If the i^{th} item of the task queue is set to DOZE, the node changes to the doze mode to save power.

The length of each time slot used to send a data packet should at least accommodate the time needed to switch the channel, to send the data packet, and to send the acknowledgement. In order to compensate for possible time drift due to some failures in time synchronization, each time slot needs to include some extra time. The length of the extra time should be two times the maximum possible time drift to totally avoid the overlapping of the communication at different time slots.

3.3 Overhead Analysis

The size of the CUB and CAB lists is very critical in TMMAC, because they are transmitted along with ATIM control packets. If they are big, the costs of the ATIM control packets may be high.

Both the CUB and CAB Lists are comprised of $M \times N$ bits, where M is the number of time slots in each beacon interval and N is the number of channels used. There are two ways to control the sizes of ATIM control packets. The first one is to control the length of each beacon interval. In this way, we can control the number of time slots in each beacon interval. For example, in our simulations, we use a beacon interval of $100ms$ and a communication window of $80ms$. By setting the data packet size to 512 bytes and the bit rate to 2Mbps which are shown in Section 5.1 with more detail, the length of each time slot is $2.96ms$, including the extra time used to compensate for possible time drift. In that setting, each beacon interval is comprised of 27 time slots and 27 bits are enough to represent one CUB or one CAB. The second one is to control the number of CUBs or CABs transmitted along with the ATIM control packets. When the number of available channels is limited, it does not hurt to include the complete CUB List or CAB List into the ATIM control packets. For example, the IEEE 802.11b only contains 3 well separated channels. In this case a CUB List or a CAB List only contains 51 bits and the sizes of the ATIM control packets used in TMMAC are similar to the sizes of RTS and CTS packets in IEEE 802.11 DCF. However, if the number of the available channels is large, for example, 50 channels, it is not desirable to include the whole list. One possible approximation is to only include several CUBs which have the least utilizations up to now in the ATIM packet. Then the receiver allocates the channels whose CUBs are contained in the ATIM packet and generates the corresponding CABs for the ATIM-ACK packet.

4 Discussion and Optimization

There are several issues in TMMAC that may degrade the performance of TMMAC. In this subsection, we discuss these issues and provide approaches to improving the performance of TMMAC.

4.1 CUB Inconsistency Problem

In Section 3, we discussed how TMMAC exploits both frequency diversity and TDMA in an efficient way. However, the correctness of TMMAC depends on reliable broadcast, which is hard to achieve cheaply. With contention based communication in the ATIM window, a problem called the CUB inconsistency problem may occur, which results from the Hidden Terminal

Problem [6]. Figure 5 shows an example of how such a problem happens.

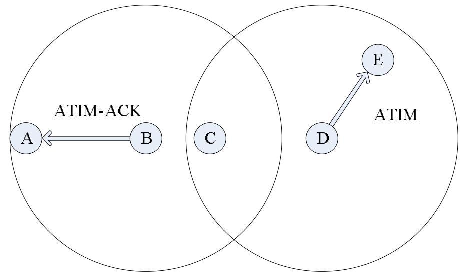


Figure 5: An example of CUB inconsistency problem.

In Figure 5, node A first sends an ATIM packet to node B. Node B replies back with an ATIM-ACK packet containing the CAB List. However, node D may not notice the transmission from node B to Node A because it is outside the communication range of node B and it may send out an ATIM packet to node E at that time. In this case, node C that is in the vicinity of node B may fail to receive the ATIM-ACK packet due to the interference from node D. Thus, node C's CUB List can not be updated, and the information contained in node C's CUB List becomes obsolete and inconsistent with the current channel usage information in its vicinity. With inconsistent CUB Lists in the network, some channels at some slots may be allocated multiple times by the nodes nearby and these packets are prone to be lost due to collisions.

As we can see from the example, the CUB inconsistency problem is actually a typical Hidden Terminal Problem [6]. In contention-based communication, it is costly to avoid the Hidden Terminal Problem when transmitting a broadcast message [25][8] and it is expensive to provide reliable broadcast [7][20]. Another way to solve this problem is to avoid contention-based communication during the ATIM window and use TDMA to send the ATIM control packets as the schedule packets sent in TRAMA [23]. However, it is not cost effective especially when multiple channels are used, because in each beacon interval we need to leave at least one time slot to each node for transmitting the schedule packet no matter whether this node has packets to send or not. And during this time slot, all the nodes within its two-hop neighborhood can not transmit packets and need to switch to the same channel. The overhead increases as the density of the nodes increases. By using contention-based communication for the ATIM control packets, only the nodes which have packets to send consume the precious bandwidth when all the nodes switch to the same frequency. In our implementation, rather than trying to avoid that problem, we allow the existence of the CUB inconsistency problem and try to minimize its negative impact, that is, TMMAC is not a totally collision-free MAC. However, the collision rate remains very low.

With the existence of the CUB inconsistency problem, the selection procedures used to decide which channels and which time slots to use for data communication become critical. One simple way to do it is to use sequential selection procedures, in which we select the channels and time slots sequentially or based on certain rules. Another way is to use random selection procedures, in which we select the channels and time slots randomly from the available ones. The randomization used in the selection procedures works as follows: first, choose the time slots from the available ones randomly; once a time slot is decided, the channel is selected randomly. Through random selection, the packet loss ratio due to collisions is reduced.

Figure 6 shows the packet loss ratios due to collisions of IEEE 802.11 DCF, TMMAC-RANDOM and TMMAC-SEQUENTIAL. The settings of the simulations are as follows: 200 nodes are randomly deployed in a $1000m \times 1000m$ area and 120 constant-bit

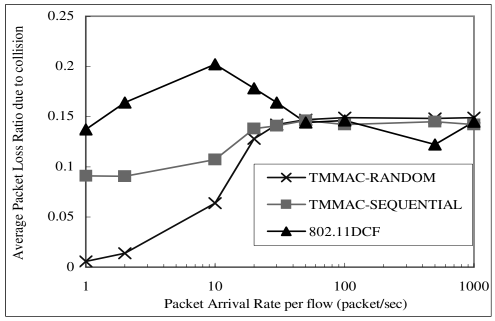


Figure 6: Packet loss ratio due to collisions with different packet arrival rates per flow.

rate (CBR) flows are randomly picked; each node has 3 non-overlapping channels; the interference range is the same as the communication range, which is $250m$. As we can see from the figure, when the network load is low, TMMAC-RANDOM has the least packet loss ratio due to collisions while IEEE 802.11 DCF has the largest. It illustrates the following two points. First, when there are many free time slots and channels, the random selection procedures can help to distribute the traffic into different time slots and channels to avoid some collisions. That is why TMMAC-RANDOM has the least packet loss ratio due to collisions. Second, the packet losses caused by the CUB inconsistency problem is less severe than that caused by contention-based communication adopted by IEEE 802.11 DCF. That is why TMMAC-SEQUENTIAL has smaller packet loss ratio than IEEE 802.11 DCF. However, when the network load is high, the packet loss ratio of TMMAC-RANDOM becomes similar to those of TMMAC-SEQUENTIAL and IEEE 802.11 DCF. It indicates that when all the time slots and channels are used, there is no space left for the random selection to avoid collisions if the CUB inconsistency problem happens. In the following simulations, the random selection procedures are used for TMMAC.

4.2 Time Synchronization

TMMAC requires time synchronization like many MAC protocols, such as IEEE 802.11 PSM [1], MMAC [24] and many TDMA scheduling algorithms [10][9][27][21][5][23]. Time synchronization can be achieved either through GPS [13] or solutions like IEEE 802.11 timing synchronization function (TSF) [1]. For example, MMAC suggests using a beaconing mechanism similar to IEEE 802.11 TSF. In that scheme, each node sets a random backoff for a beacon message at the start of each beacon interval. If a node receives a beacon message before it sends its own one, it drops its own beacon message.

To send separate packets for time synchronization is not cost effective. A better approach is to embed the timestamp into the ATIM control packets. At the beginning of each beacon interval, a random backoff for a beacon message is set only when a node does not have any packet to send. The node which has packets to send directly sets a backoff for the ATIM packets or the ATIM-BRD packets. The backoff for beacon messages is longer than that for the ATIM packets or the ATIM-BRD packets. When an ATIM packet or ATIM-BRD packet is transmitted, the MAC layer timestamps the packet. A node which sets a backoff for a beacon message can drop the beacon message if it receives an ATIM packet or an ATIM-BRD packet or a beacon message before it sends the beacon message out. This approach can reduce the number of beacon messages and leave more time for ATIM control packets. The disadvantage is that it makes the MAC layer a little complicated.

5 Performance Evaluation

In this section, we evaluate the performance of the TMMAC protocol by simulation. We compare our scheme with IEEE 802.11 DCF which uses a single channel, and MMAC [24] which is a typical multi-channel MAC protocol using a single radio transceiver (MMAC was explained in Section 2). The following four metrics are used to evaluate the performance of TMMAC:

- Aggregate throughput. Aggregate throughput is the total throughput of all the nodes in the network. It shows how well TMMAC is able to utilize the existing bandwidth.
- Average transmission time. Average transmission time is the average MAC delay for each successful data packet transmission. It is computed by dividing the total MAC delay of the whole network by the total number of successful packet transmissions.
- Per packet energy. Per packet energy is the value of total energy consumed by the whole network divided by the total number of packets successfully transmitted. It illustrates the price we pay for each successful data packet transmission.
- Packet loss ratio due to collisions. As shown in Section 4.4, TMMAC has the CUB inconsistency problem during the negotiation, which causes some packet losses because some channels at certain time slots may be allocated multiple times by the nodes nearby. Here we only show the packet loss ratio due to collisions to better understand how often the collisions happen in TMMAC compared to IEEE 802.11 DCF and MMAC. We do not include the packet losses due to buffer overflow into the computation. Also we do not consider the retransmissions when counting the number of lost packets to better understand the frequency of collisions.

5.1 Simulation Settings

We have implemented MMAC and TMMAC in GloMoSim [32], a scalable discrete event simulator developed by UCLA. Unless otherwise specified, the simulation settings of all of our simulations are as follows: the bit rate is 2Mbps for each channel; 3 channels are used in our simulations to conform to the IEEE 802.11b which has 3 orthogonal channels[1]; the beacon interval is set to $100ms$; the length of the ATIM windows is $20ms$; the communication range of each node is $250m$; the interference range is $313m$, which is 1.25 times the communication range to make the simulations more realistic; the data packet size is 512 bytes; constant-bit rate (CBR) traffic is used in the application layer; GF [16] routing protocol is used in the routing layer; all the points in our performance figures are computed from 20 trials and each trial lasts for 50 seconds. Two application scenarios are used to show the performance of TMMAC, MMAC and 802.11 DCF as shown in Figure 7.

By adding the time needed to switch the channel, send the data packet and the acknowledgement, the duration of each time slot is $2.82ms$. In order to compensate for possible time drift when the time synchronization is not perfect in real ad hoc networks, we increase the time slot to $2.96ms$. With such setting, each beacon interval is comprised of 27 available time slots for each channel and both the CUB and CAB lists contain 51 bits.

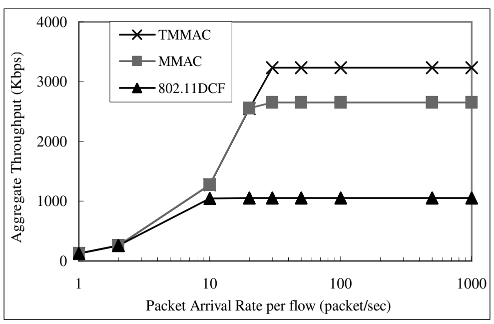
[1]IEEE 802.11a [2] provides 12 orthogonal channels. 4 channels are left for outdoor use and the remaining 8 are for indoor use.

Application Scenario 1	A wireless LAN with the size of $150m \times 150m$. Every node in the network is within one hop. 64 nodes are randomly deployed in the area, in which half of the nodes are chosen as sources and the other half are chosen as destinations. A node is either the source for a CBR flow or the destination for a CBR flow.
Application Scenario 2	A multi-hop network with the size of $1000m \times 1000m$. 200 nodes are randomly deployed into this area. Then 120 nodes are randomly chosen as the sources and 120 nodes are randomly chosen as the destinations. In this set of simulations,nodes can be sources or destinations for multiple CBR flows simultaneously.

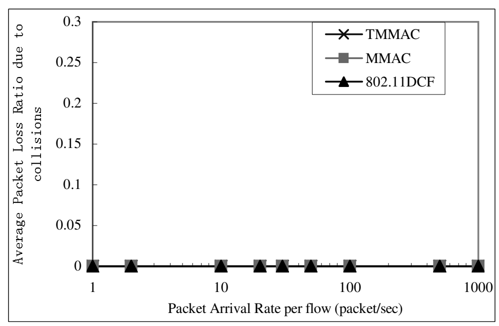
Figure 7: Two application scenarios used for simulations.

5.2 Simulation Results in Application Scenario 1

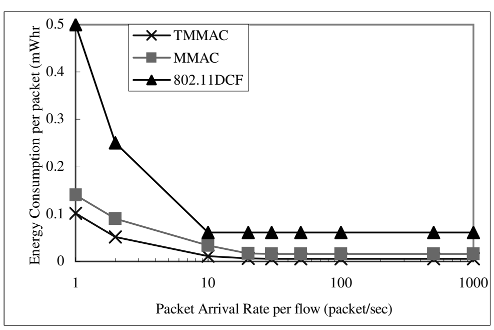
In this section, We compare and analyze the performance of TMMAC, MMAC and IEEE 802.11 DCF in Application Scenario 1. We vary the packet arrival rates of the CBR flows to show their performance at different network loads.



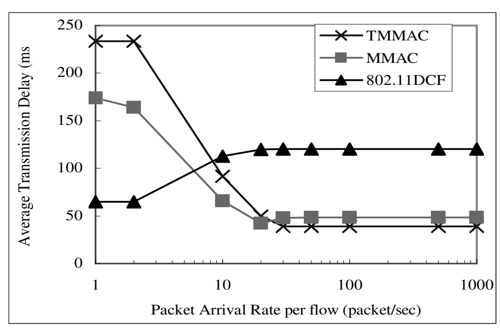
(a) Aggregate throughput



(b) Packet loss ratio due to collisions



(c) Energy consumption per packet



(d) Average transmission delay

Figure 8: Performance evaluation with different system loads in Application Scenario 1.

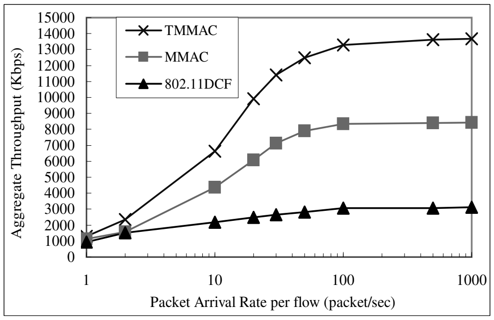
Figure 8 shows the performance of TMMAC, MMAC and IEEE 802.11 DCF with different packet arrival rates in Application Scenario 1. As shown in Figure 8(a), when the network load is near saturation, the aggregate throughput of MMAC is 2.5 times that of IEEE 802.11 DCF, because MMAC successfully exploits 3 channels. However, TMMAC achieves 22% more aggregate throughput than MMAC. That is because TMMAC avoids contention based communication during the communication window. Contention based communication adopted by MMAC and IEEE 802.11 DCF wastes some time in backoff and sending RTS and CTS. Figure 8(b) shows that in a wireless LAN, the packet loss ratios due to collisions of TMMAC, MMAC and 802.11 DCF are almost 0 in spite of the increase of the packet arrival rate per flow. In a single hop wireless LAN, the CUB inconsistency problem described in Section 4.1 is not likely to happen, because everyone can hear from each other. Figure 8(c) shows their per packet energy consumptions. As we can see from the figure, both TMMAC and MMAC consume much less energy compared to IEEE 802.11 DCF, because the nodes in TMMAC and MMAC can switch into power saving mode when they are not scheduled. However, TMMAC achieves more aggressive power saving because it can decide whether it goes into doze mode at each time slot. When the packet arrival rate is low, per packet energy consumption in TMMAC is 65% of that in MMAC. That is because the majority of the energy in both TMMAC and MMAC is consumed during the ATIM window when nodes are negotiating with each other. As the packet arrival rate increases, the energy saving during the communication window in TMMAC becomes more obvious. Per packet energy consumption in TMMAC is only 30% of that in MMAC when the packet arrival rate is above 10 packets/sec. Figure 8(d) compares their average transmission delays. When the packet arrival rate is low, TMMAC has the largest average transmission delay while IEEE 802.11 DCF has the smallest. It is caused by the following two reasons. First, in TMMAC and MMAC, nodes are not allowed to transmit any packets until they successfully negotiate with their destinations. That is why both TMMAC and MMAC have higher transmission delay than IEEE 802.11 DCF. Second, random selection procedures adopted by TMMAC distribute the traffic into different time slots, which means that the packets are not sent as early as possible when it enters the communication window. That is why TMMAC has higher transmission delay than MMAC. When the network is overloaded, by avoiding contention based communication during the communication window, TMMAC achieves the smallest transmission delay compared to MMAC and IEEE 802.11 DCF.

5.3 Simulation Results in Application Scenario 2

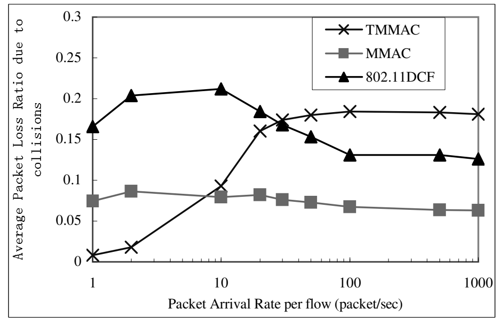
In this section, We show the performance evaluation of TMMAC, MMAC and IEEE 802.11 DCF in Application Scenario 2. We also vary the packet arrival rates of the CBR flows to show their performance at different network loads.

Figure 9 compares the performance of TMMAC, MMAC and IEEE 802.11 DCF with different packet arrival rates in Application Scenario 2. As shown in Figure 9(a), the aggregate throughput of TMMAC is 1.62 times that of MMAC, and is 4.39 times that of IEEE 802.11 DCF. The improvement in Application Scenario 2 is much larger than that in Application Scenario 1. This is due to the following two reasons. First, in a multi-hop network, there are more channel contentions for RTS and CTS packets in both MMAC and 802.11 DCF. With more contentions, the retransmissions of the control packets and backoff happen more frequently. Second, in TMMAC all the available bandwidth in different channels can be fully utilized once it is scheduled, but in MMAC and 802.11 DCF, nodes need to wait whenever the channel is busy.

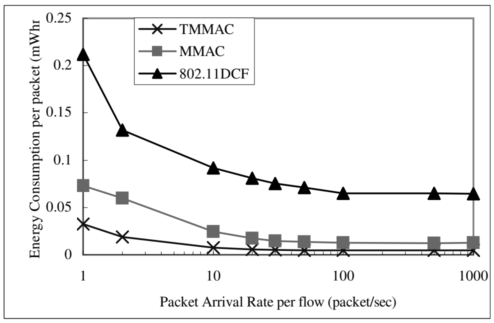
Figure 9(b) shows the packet loss ratios due to collisions of the three MAC protocols. Interestingly, TMMAC shows much lower packet loss ratio when the network load is low, because TMMAC is able to randomly distribute the small traffic load into different time slots and channels. However, when the traffic load becomes higher, TMMAC suffers a little higher loss ratio, which is caused by the following two reasons. First, when all the available channels and time slots are fully allocated, the chance to use the same channels and time slots by multiple transmissions nearby is high if the CUB inconsistency problem



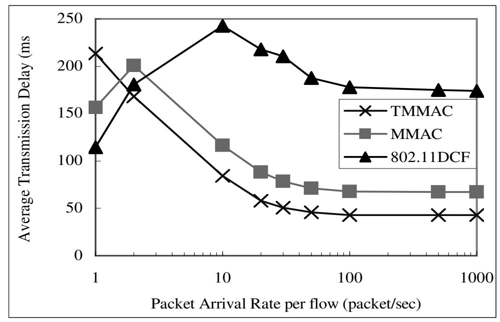
(a) Aggregate throughput



(b) Packet loss ratio due to collisions



(c) Energy consumption per packet



(d) Average transmission delay

Figure 9: Performance evaluation with different system loads in Application Scenario 1.

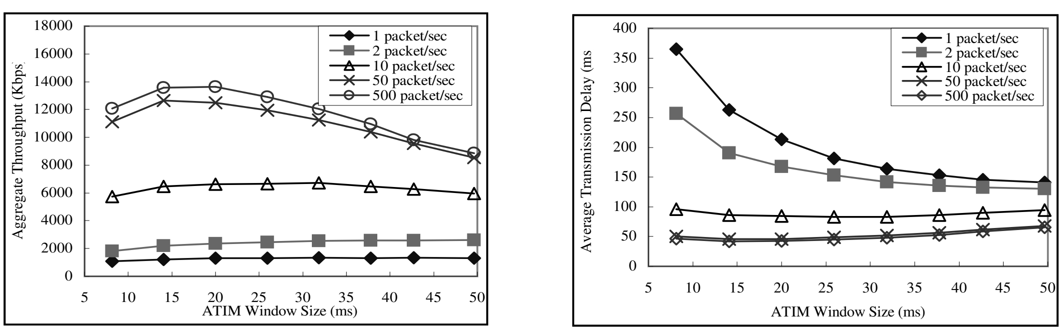
happens. Second, the interference range in these simulations is 1.25 times the communication range. Even if there is no CUB inconsistency problem, some packets may be lost in TMMAC due to the interference. However, the impact of larger interference range on IEEE 802.11 DCF and MMAC is smaller, because the use of carrier sense and RTS/CTS packets can help to avoid some collisions during data packet transmissions.

Figure 9(c) shows that although TMMAC suffers higher packet loss ratio due to collisions when the network is overloaded, TMMAC consumes much lower energy for each successful packet transmission in a multi-hop network. Per packet energy consumption in TMMAC is only 37% of that in MMAC when the network is overloaded. That is because the power saving mode in TMMAC is more aggressive and it achieves much higher aggregate throughput.

Figure 9(d) illustrates the average transmission delays of different MAC protocols. Both TMMAC and MMAC have higher transmission delays than IEEE 802.11 DCF when the packet arrival rate is 1 packet/second. When there are too many flows in the network, some nodes need to wait several beacon intervals to complete their negotiation. For example, in this set of simulations, we have 120 random flows in the network. Suppose each flow has an average of three hops. We can see that the majority of the 200 nodes are involved in transmitting packets, but only about 40 to 50 nodes can successfully negotiate with their destinations in a $20ms$ ATIM window. This indicates that when there are many flows and each flow only transmits a small number of packets per beacon interval, we may need to increase the size of the ATIM window to reduce the transmission delay. Section 5.4 shows the simulation results with different ATIM window sizes. Another interesting observation is that both MMAC and IEEE 802.11 DCF have peak points , while TMMAC does not. We conclude the following two reasons for the peak point in IEEE 802.11 DCF. First, when the packet arrival rate increases, the increase of the contention results in the increment of the transmission delay at the first stage. Second, as the packet arrival rate keeps increasing, the performance improvement brought by the good links in the four corners of the simulation field starts to reduce the impact of the increased contention. The data communication in the four corners of the simulation field has higher success ratio than that at the center. It is obvious, because there is no node outside the boarders of the simulation field and the number of nodes which can interfere with the data communication in the corners is limited. When the good links have more packets to transmit, it starves some bad links at the center. Meanwhile, it achieves higher throughput, which reduces the average transmission delay. This is also the reason why the average packet loss ratio of IEEE 802.11 DCF shown in Figure 9(b) starts to decrease after the packet arrival rate exceeds 10 packets/second. MMAC has the similar reasons for the peak point except that MMAC has one more reason for the decrease of the transmission delay after the peak point. When the packet arrival rate increases, more packets can be transmitted by a single negotiation in MMAC, which reduces the average transmission delay. Overall, when the packet arrival rate per flow is above 2 packets/second, the transmission delay of TMMAC starts to outperform those of MMAC and IEEE 802.11 DCF. TMMAC achieves 35%-38% lower transmission delay than MMAC when the packet arrival rate is above 20 packets/second.

5.4 Impact of ATIM Window Size

So far in our simulations, the ATIM window size is fixed to be $20ms$. Overall, an ATIM window size of $20ms$ works well as shown in Section 5.2 and Section 5.3. However, a fixed ATIM window size does not always achieve the best performance under all network situations in TMMAC. Like IEEE 802.11 PSM, the size of the ATIM window plays an important role in the performance of TMMAC. As shown in [30], there is no fixed size of the ATIM window that performs well under all situations In IEEE 802.11 PSM. In this section, we study the impact of different ATIM window sizes on TMMAC.



(a) Aggregate throughput

(b) Average Transmission Delay

Figure 10: Impact of different ATIM window sizes.

Figure 10 shows the performance of TMMAC with different ATIM window sizes in Application Scenario 2. The five curves show results for five different packet arrival rates: 1 packet/second, 2 packet/second, 10 packet/second, 50 packet/second, and 500 packet/second. We first analyze the performance of TMMAC when the packet arrival rate is low. As shown in Figure 10(a), when the packet arrival rate is 1 packet/second or 2 packet/second, the size of the ATIM window does not have a significant impact on the aggregate throughput. However, the average transmission time changes greatly if the size of the ATIM window varies as shown in Figure 10(b). When the packet arrival rate is low, by using a larger ATIM window, more nodes can successfully negotiate with their destinations within one beacon interval. In this case, nodes need to wait less beacon intervals to transmit their data packets, which reduces the transmission delay.

Now we analyze the performance of TMMAC when the packet arrival rate is high. When the packet arrival rate is 50 packet/second or 500 packet/second, the average transmission delay does not change much if the size of the ATIM window changes, because a single negotiation can schedule multiple data packets. However, the aggregate throughput is sensitive to the size of the ATIM window. When the ATIM window is too small, the aggregate throughput decreases, because the number of successful negotiations is not large enough to take full use of the communication window. When the ATIM window is too large, the aggregate throughput also drops, because there is less time left for data communication.

Recall that in Application Scenario 2, there are 120 CBR flows. If the number of the CBR flows is reduced significantly, a smaller ATIM window is desirable. When there are only several flows, a smaller ATIM window is able to accomodate all the negotiations and to leave more bandwidth for the communication window. Further, using a smaller ATIM window can save more power. The optimal ATIM window size not only depends on the packet arrival rate per flow, but also depends on the number of flows in the network. As in IEEE 802.11 PSM, there is no optimal ATIM window size that works well under all situations in TMMAC. The final solution is to change the ATIM window size dynamically based on the network situation. Jung et al. [15] propose a scheme to change the ATIM window size dynamically in IEEE 802.11 PSM for wireless LANs. However, changing the ATIM window size dynamically in a multi-hop network is more complicated, due to the packet losses caused by radio interference. Moreover, TMMAC introduces channel diversity and TDMA in the communication window, which further complicates the problem. How to dynamically change the ATIM window size in TMMAC is left for future work.

6 Differentiated Power TMMAC

So far we have explored the performance of TMMAC when the interference range is a little larger (1.25 times) than the communication range. However in reality, the interference range may be much larger than the communication range [33]. In this section we first explore the impact of different interference ranges on TMMAC, and also on MMAC and IEEE 802.11 DCF. Then we propose a solution called Differentiated Power TMMAC to improve the performance of TMMAC when the interference range is much larger than the communication range. Unless otherwise specified, the simulation settings used in this section are the same as those specified in Section 5.1.

6.1 Impact of Interference Range

When the interference range is larger than the communication range, even if the CUB inconsistency problem does not happen, TMMAC is inclined to suffer packet losses due to collisions. As shown in Figure 11, because the interference range is larger than the communication range, node D which is outside the communication range of node B can interfere with node B and cause B's failure to successfully receive the packet from node A.

If the interference range is only a little larger than the communication range, the impact is not severe as shown in Section 5.3 . However, as the interference range increases, more transmissions will be interfered with. This problem happens because TMMAC only tries to avoid multiple allocations using the same channel and the same time slot within the communication range. It does not take into consideration the allocations of the channels and time slots which are outside the communication

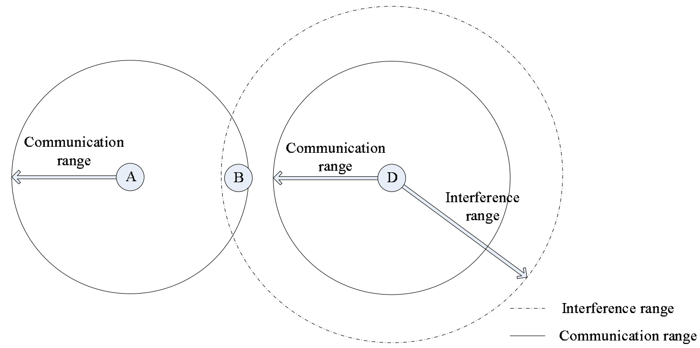
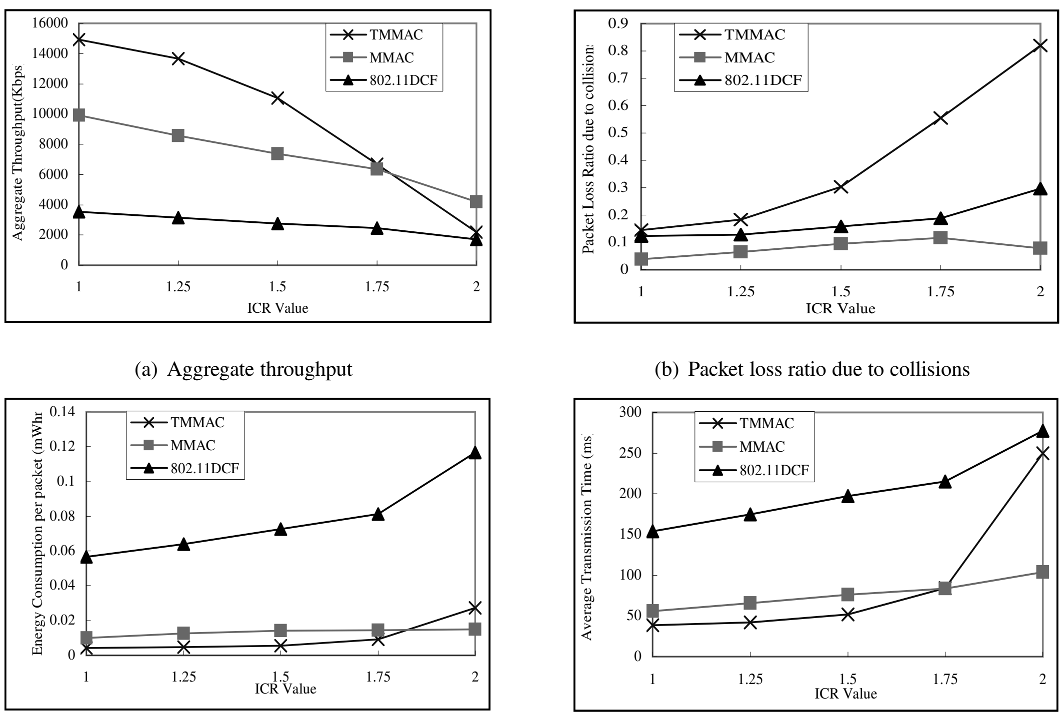


Figure 11: Impacts of interference range.

range, but inside the interference range. The increase of the interference range also impacts the performance of MMAC and IEEE 802.11 DCF.



(a) Aggregate throughput

(b) Packet loss ratio due to collisions

(c) Energy consumption per packet

(d) Average transmission delay

Figure 12: Performance evaluation with different ICR values.

Figure 12 shows the performance evaluation of TMMAC, MMAC and IEEE 802.11 DCF with different ICR values (Interference range to Communication range Ratio) in Application Scenario 2. In this set of simulations, the packet arrival rate of each flow is fixed to be 500 packets per second when the network is overloaded. We first analyze the performance of MMAC and IEEE 802.11 DCF. As shown in Figure 12, the performance of both MMAC and IEEE 802.11 DCF does not drop as much as TMMAC does when the ICR value increases. There are two reasons for that. First, using carrier sense before sending out packets enables them to avoid collisions at certain level. Second, as mentioned earlier, the good links in the four corners are less affected by the increase of the interference range, because they are protected by the boarder of the terrain in the simulations.

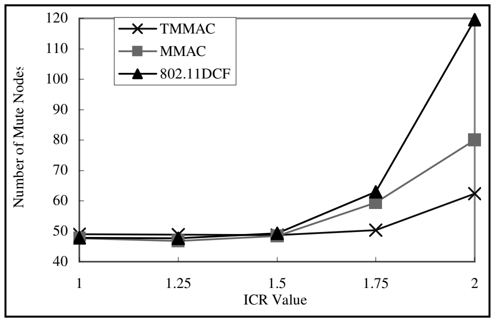


Figure 13: Number of Mute Nodes with different ICR values.

The performance achieved from the good links helps to maintain the overall performance. However, it raises a severe problem which can not be revealed directly from the aggregate throughput.

Before proceeding, we define another metric, called the number of mute nodes, to show their performance when the ICR value increases. A node is called a mute node if it fails to successfully send out a single packet during the whole simulation. Figure 13 shows the number of mute nodes with different ICR values. When the ICR value exceeds 1.5, the number of mute nodes in MMAC and IEEE 802.11 DCF increases very fast as shown in Figure 13. When the ICR value equals to 2, 60% of the nodes in the network can not send out a single packet in IEEE 802.11 DCF. However, the aggregate throughput of both MMAC and IEEE 802.11 DCF does not drop as fast as the number of mute nodes increases. This indicates that although the aggregate throughput does not drop dramatically, the network has failed to function properly in MMAC and IEEE 802.11 DCF. When the ICR value is large, only the nodes in the four corners can keep on transmitting packets successfully, while the nodes at the center can not send out a single packet.

Now we analyze the performance of TMMAC when the ICR value increases. As shown in Figure 12, the performance of TMMAC first degrades slowly as the ICR value increases. The performance of TMMAC when the ICR value equals 1.25 is comparable to that of TMMAC when the ICR value equals 1. However, the performance of TMMAC degrades very fast when the ICR value exceeds 1.5, because the area in which nodes are involved in interfering with the transmission at the center increases exponentially as the interference range increases. When a large number of nodes which are outside the communication range of the node at the center are involved in interfering the transmission at the center, the probability that none of these nodes transmits a packet when the node at the center is receiving a packet is low, especially when the network is overloaded. When the ICR value equals 2, TMMAC only achieves 27% higher aggregate throughput than 802.11 DCF and it suffers as high as 82% packet loss ratio due to collisions. This imposes a great challenge to TMMAC and also other TDMA scheduling algorithms which assume that the interference range is the same as the communication range [5][23][33].

6.2 Design of Differentiated Power TMMAC

As shown in Section 6.1, the performance of TMMAC decreases significantly, if the interference range is much larger than the communication range. To reduce the performance degradation caused by large interference range, one approach is to detect the on-line interference relations, which can be done by the RID protocol [33]. However it is costly to use, especially when the nodes are mobile. In this paper, we propose a solution called Differentiated Power TMMAC to address this problem. In Differentiated Power TMMAC, ATIM control packets and data packets are transmitted by using different sending powers. We use a higher sending power for ATIM control packets and a lower sending power for data packets. We use the communication range H_c and interference range H_i to denote the communication range and interference range when we use a higher sending power to transmit ATIM control packets, respectively. Also we use the communication range L_c and interference range L_i to denote the communication range and interference range when we use a lower sending power to send out data packets. Although a higher sending power for ATIM control packets consumes more energy, ATIM control packets are short compared to data packets. Further, one negotiation may schedule multiple data packets. Thus, the high energy consumption for ATIM control packets can be compensated for.

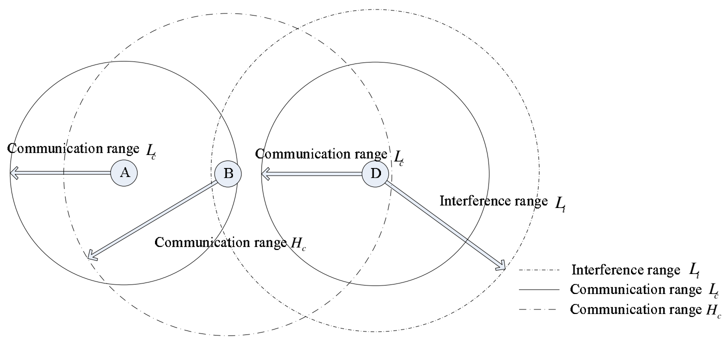
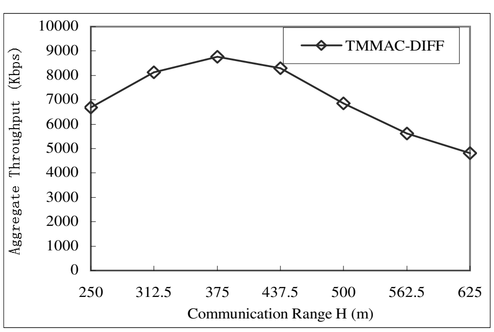
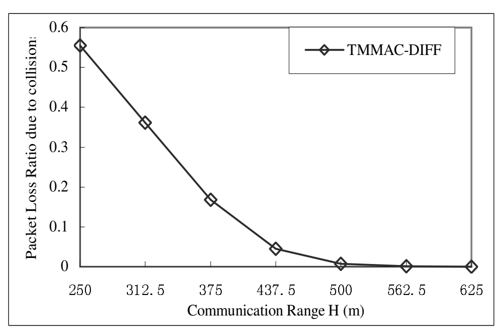


Figure 14: An example of using Differentiated Power TMMAC.

Figure 14 shows an example of using Differentiated Power TMMAC. When node B is sending out its ATIM control packets, it uses a higher sending power whose communication range covers node D. In this case, node D which is outside the communication range L_c of node B, but inside the interference range L_i is able to hear the ATIM control packets from node B. If the CUB inconsistency problem does not happen, node D is able to overhear the ATIM control packets from node B, and it will not choose the same channels and time slots for data transmissions. Thus, the collisions can be avoided. The main question in Differentiated Power TMMAC is how large the communication range H_c should be.



(a) Aggregate throughput



(b) Packet loss ratio due to collisions

Figure 15: Impact of different communication range H_c.

Figure 15 shows the impact of different communication ranges H_c on Differentiated Power TMMAC in Application Scenario 2. TMMAC-DIFF in the figure represents Differentiated Power TMMAC. In this set of simulations, we set the ICR value to 1.75, which means that the interference range L_i is $437m$. The packet arrival rate of each flow is fixed to be 500 packets per second. As shown in Figure 15(a) and Figure 15(b), the aggregate throughput of TMMAC-DIFF first increases as the communication range H_c increases, because the packet loss ratio due to collisions decreases a lot. However, as the communication range H_c keeps increasing, the aggregate throughput of TMMAC-DIFF drops, although the packet loss ratio due to collisions still keeps decreasing. The reason is that when the communication range H_c is beyond the interference range L_i , TMMAC-DIFF becomes conservative, because the allocations of channels and time slots which are outside the interference range L_i are also taken into consideration. From Figure 15, we also can see that the highest aggregate throughput is achieved when the communication range H_c is about 15% smaller than the interference range L_i. From the definition of the interference range, a node which is within the interference range of another node is not necessarily to be interfered by that node. We use an example shown in Figure 16 to illustrate it.

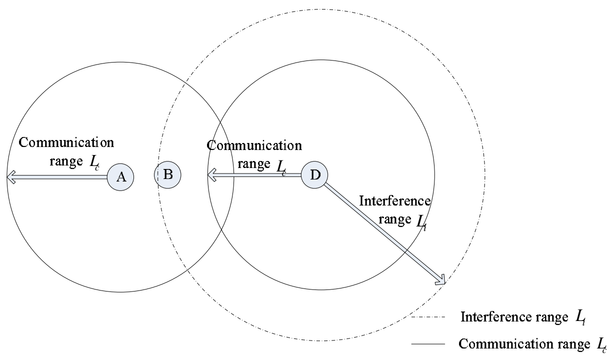


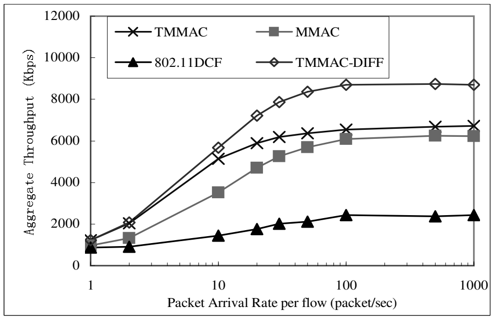
Figure 16: An example of interference.

In Figure 16, node B is within the interference range of node D. However, the transmission of node D may not cause node B to fail to receive the packet from node A, because node B is quite close to node A and the link between them is strong. Using a little shorter communication range H_c, TMMAC-DIFF allows more concurrent transmissions and achieves higher aggregate throughput, even with higher packet loss ratio due to collisions.

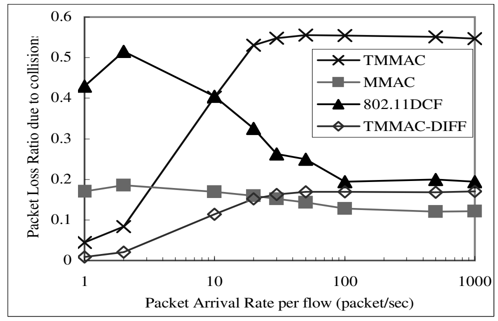
6.3 Performance Evaluation of Differentiated Power TMMAC

In this section, We show the performance evaluation of TMMAC-DIFF, TMMAC, MMAC and IEEE 802.11 DCF in Application Scenario 2. In this set of simulations, the ICR value is set to 1.75 and the communication range H_c used in TMMAC-DIFF is $380m$, which is $57m$ shorter than the interference range L_i. We also vary the packet arrival rates of the CBR flows to show their performance at different network loads.

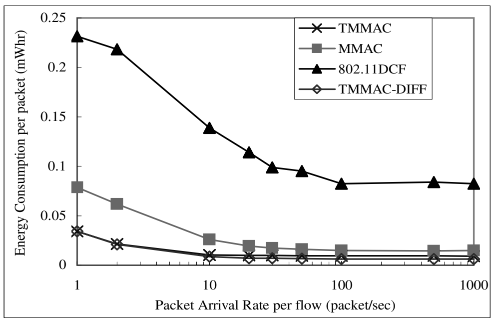
Figure 17 shows the performance of different MAC protocols with different packet arrival rates in Application Scenario 2. As we can see from Figure 17(a) and 17(b), TMMAC-DIFF shows 26% higher aggregate throughput than the original TMMAC and shows 40% higher aggregate throughput than MMAC. Also the packet loss ratio due to collisions drops from 55% to 16%, which is similar to MMAC and 802.11 DCF. From Figure 17(c) we can see that although TMMAC-DIFF uses a much higher sending power for ATIM control packets, the energy consumption per packet remains low, which is 42% of MMAC. That is because the sizes of ATIM control packets are small compared to those of data packets and one single negotiation can schedule



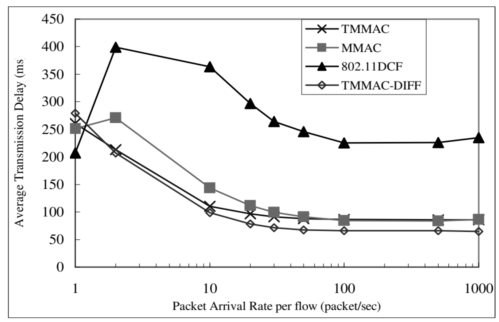
(a) Aggregate throughput



(b) Packet loss ratio due to collisions



(c) Energy consumption per packet



(d) Average transmission delay

Figure 17: Performance evaluation with different system loads in Application Scenario 2.

multiple transmissions. Figure 17(d) shows that TMMAC-DIFF also has smaller transmission delay compared to other MAC protocols. When the network is overloaded, the transmission delay in TMMAC-DIFF is 75% of that in MMAC.

7 Conclusions and Future Work

In this paper, we present the TMMAC protocol, which is a TDMA based multi-channel MAC protocol using a single radio transceiver. TMMAC requires time synchronization in the network and divides the time into fixed beacon intervals. Nodes that have packets to transmit negotiate which channels and which time slots to use for data communication with their destinations during the ATIM window. This two-dimension negotiation enables TMMAC to exploit the advantages of both frequency diversity and TDMA in an efficient way. From our performance evaluation, TMMAC achieves up to 62% higher communication throughput over the state-of-the-art multi-channel MAC protocols using a single radio transceiver. Further, TMMAC is able to support broadcast in an effective way and is highly power-efficient.

In the paper, we also explore the impact of different interference ranges on TMMAC. Performance evaluation results show that the performance of TMMAC decreases as the interference range increases. We propose Differentiated Power TMMAC to improve the performance of TMMAC when the interference range is much larger than the communication range. Differentiated Power TMMAC improves the communication throughput of TMMAC by 26% when the interference range is 1.75 times the communication range.

For future work, we would investigate how to dynamically change the size of the ATIM window to improve the performance of TMMAC in terms of throughput, energy consumption and transmission delay. We also look forward to the implementation and evaluation of TMMAC in a real large scale wireless network.

References

[1] IEEE 802.11 Working Group, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,” 1997.

[2] IEEE 802.11a Working Group, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 1: High-speed Physical Layer in the 5 GHz band,” 1999.

[3] A. Adya, P. Bahl, J. Padhye, A.Wolman, and L. Zhou, “A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks,” in IEEE International Conference on Broadband Networks (Broadnets), 2004.

[4] P. Bahl, R. Chandra, J. Dunagan, “SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks,” in Proceedings of MobiCom, 2004.

[5] L. Bao and J.J. Garcia-Luna-Aceves, “A New Approach to Channel Access Scheduling for Ad Hoc Networks,” in Proceedings of MobiCom, 2001.

[6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A Media Access Protocol for Wireless LANs,” In Proceedings of ACM SIGCOMM, 1994.

[7] J. Chang and N.F. Maxemchuk, “Reliable Broadcast Protocols,” in ACM Transactions on Computer Systems (TOCS), vol. 2, pp. 251-273, 1984.

[8] C-Y. Chiu, E.H-K. Wu, G-H. Chen, “A Reliable and Efficient MAC Layer Broadcast (Multicast) Protocol for Mobile Ad Hoc Networks,” in Global Telecommunications Conference, 2004.

[9] I. Chlamtac and A. Farago, “Making Transmission Schedules Immune to Topology Changes in Multi-hop Packet Radio Networks,” in IEEE/ACM Transactions on Networking, 1994.

[10] I. Chlamtac and S. Kutten, “A Spatial-Reuse TDMA/FDMA for Mobile Multi-hop Radio Networks,” in Proceedings of InfoCom, 1985.

[11] J. Deng and Z. Haas, “Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks,” in IEEE ICUPC, October 1998.

[12] F. Fitzek, D. Angelini, G. Mazzini and M. Zorzi, “Design and Performance of an Enhanced IEEE 802.11 MAC Protocol for Multihop Coverage Extension,” in IEEE Wireless Communications, December 2003.

[13] I.A. Getting, “The Global Positioning System,” IEEE Spectrum 30, December 1993.

[14] N. Jain and S. R. Das, "A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks," in IEEE IC3N, October 2001.

[15] E.S. Jung and N.H. Vaidya, "An Energy Efficient MAC Protocol for Wireless LANs," in Proceedings of InfoCom, June 2002.

[16] B. Karp, "Geographic Routing for Wireless Networks," Ph.D. theses, Harvard University, 2000.

[17] J. Li, Z. J. Haas, M. Sheng, and Y. Chen, "Performance Evaluation of Modified IEEE 802.11 MAC for Multi-Channel Multi-Hop Ad Hoc Network," in IEEE AINA, 2003.

[18] A. Nasipuri, J. Zhuang, and S. R. Das, "A Multichannel CSMA MAC Protocol for Multihop Wireless Networks," in IEEE WCNC, September 1999.

[19] A. Nasipuri and S. R. Das, "Multichannel CSMA with Signal Powerbased Channel Selection for Multihop Wireless Networks," in IEEE Vehicular Technology Conference, September 2000.

[20] E. Pagani and G.P. Rossi, "Reliable Broadcast in Mobile Multihop Packet Networks," in the Proceedings of ACM/IEEE MobiCom, 1997.

[21] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," in Wireless Networks, 1999.

[22] A. Raniwala and T. Chiueh, "Architecture and Algorithm for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network," in IEEE InfoCom 2005, March 2005.

[23] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," in Proceedings of SenSys, 2003.

[24] J. So and N. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver," in Proceedings of MobiHoc, 2004.

[25] K. Tang and M. Gerla, "MAC Reliable Broadcast in Ad Hoc Networks," in the Proceedings of MILCOM, 2001.

[26] Z. Tang and J.J. Garcia-Luna-Aceves, "Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks," in IEEE InfoCom, March 1999.

[27] Z. Tang and J.J. Garcia-Luna-Aceves, "A Protocol for Topology-Dependent Transmission Scheduling," in Proceedings of WCNC, 1999.

[28] A. Tyamaloukas and J. J. Garcia-Luna-Aceves, "Channel-Hopping Multiple Access," in IEEE ICC, 2000.

[29] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks," in IEEE InfoCom 2001, April 2001.

[30] H. Woesner, J.P. Ebert, M. Schlager, and A. Wolisz, "Power-Saving Mechanisms in Emerging Standards for Wireless LANs: The MAC Level Perspective," IEEE Personal Communications, Jun 1998.

[31] S. Wu, C. Lin, Y. Tseng, J. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks," in Proceedings of ISPAN, 2000.

[32] X. Zeng, R. Bagrodia, M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks," in The 12th Workshop on Parallel and Distributed Simulations, 1998.

[33] G. Zhou, T. He, J.A. Stankovic, T. Abdelzaher, "RID: Radio Interference Detection in Wireless Sensor Networks," in Proceedings of InfoCom, 2005.