

# A Multi-channel MAC Protocol for Ad Hoc Wireless Networks

Jungmin So

Dept. of Computer Science, and  
Coordinated Science Laboratory  
University of Illinois at Urbana-Champaign  
Email: jso1@uiuc.edu

Nitin H. Vaidya

Dept. of Electrical and Computer Eng., and  
Coordinated Science Laboratory  
University of Illinois at Urbana-Champaign  
Email: nhv@uiuc.edu

## Technical Report

January 2003

**Abstract**— This paper proposes a medium access control (MAC) protocol for ad hoc wireless networks that can utilize multiple channels dynamically to improve performance. IEEE 802.11 standard provides multiple channels for use, but its MAC protocol is designed only for a single channel. We can achieve improved throughput with multiple channels because multiple transmissions can take place simultaneously without interfering each other.

We modify IEEE 802.11 DCF protocol to enable hosts to utilize multiple channels by switching channels dynamically. Our scheme requires only one transceiver for each host. The main idea is to use ATIM windows as in power saving mechanism (PSM) of IEEE 802.11 DCF, so that hosts can negotiate and select channels during the window using ATIM messages. Our scheme improves throughput of the network significantly, especially when the network is highly congested. We have simulated our protocol to verify its performance compared to IEEE 802.11 and another multi-channel MAC protocol. The results show that our protocol performs better than both. The improvement over the existing multi-channel MAC protocol is obtained even though our protocol uses simpler hardware.

### I. INTRODUCTION

IEEE 802.11 standard for wireless LAN [1] has a medium access control (MAC) protocol designed for sharing a single channel between hosts. Due to the broadcast nature of wireless transmission, when two hosts are communicating, all other hosts within the range of the two hosts must defer their communication in order to avoid collision. This results in a significant throughput degradation as the number of active hosts increases.

A lot of work has been done to improve the throughput of wireless networks, and various approaches have been proposed. One approach is to control the transmission power [2]. The basic idea of *power control* is to have

the sender transmit with power just enough to reach the target node, so that the space blocked by that particular transmission is minimized. Another approach is to use directional antennas instead of omnidirectional antennas [3], [4], [5]. Directional antennas are able to transmit signal in one direction, so that the hosts located in other directions can communicate concurrently without interfering each other.

These two approaches improve the throughput by increasing *spatial reuse*, but still they use only a single channel. The approach in this paper, is to achieve improved performance using multiple channels [6], [7], [8], [9], [10]. Data transmitted in different channels do not interfere with each other, thus can take place in the same region simultaneously. So the throughput can increase significantly, proportional to number of channels in ideal case. Although not studied in this paper, having multiple channels can provide further benefits in addition to increased throughput, such as a more simple way to support QoS.

|                    | 802.11b           | 802.11a            |
|--------------------|-------------------|--------------------|
| Physical Layer     | DSSS <sup>1</sup> | OFDM <sup>2</sup>  |
| Maximum Speed      | 11Mbps            | 54Mbps             |
| Frequency Band     | 2.4GHz            | 5GHz               |
| Number of Channels | 3                 | 8 / 4 <sup>3</sup> |

TABLE I  
FEATURES OF IEEE 802.11 STANDARDS

<sup>1</sup>Direct Sequence Spread Spectrum

<sup>2</sup>Orthogonal Frequency Division Multiplexing

<sup>3</sup>8 for indoor use, 4 for outdoor use

IEEE 802.11 standard already has multiple channels available for use. Table I summarizes the features of IEEE 802.11 standards. IEEE 802.11b physical layer (PHY) has 14 channels, 5MHz apart in frequency [1]. But to be totally non-overlapping and thus feasible for use in the same region, the frequency spacing must be at least 30MHz. So channels 1, 6 and 11 are typically used for communication in current implementations, and thus we have 3 channels available for use. IEEE 802.11a provides 12 channels, 8 in the low part of the band for indoor use and 4 in the upper part for outdoor use [11].

There is enough motivation for concurrent use of multiple channels, but the current MAC protocol is designed for sharing a single channel. The main reason for this is that each IEEE 802.11 host is equipped with one half-duplex transceiver, so it can only transmit or listen on one channel at a time. So when a host is listening on a particular channel, it cannot hear communication taking place on a different channel. Due to this, as observed in [9], if a single-channel MAC protocol (such as IEEE 802.11 DCF) is applied in a multi-channel environment wherein each node may dynamically switch channels, performance degradation may occur (unless additional precautions are taken to manage dynamic channel selection).

In this paper, we propose a MAC protocol which enables hosts to dynamically negotiate channels so that multiple communication can take place in the same region simultaneously, each in different channel. The main idea is to divide time into fixed-time intervals using beacons, and have a small window at the start of each interval to indicate traffics and negotiate channels for use in that interval. A similar approach is used in IEEE 802.11 power saving mechanism (PSM) [1], which is explained in section III.

The rest of the paper is organized as follows. Section II reviews the related work in this topic. Section III provides some background information. Section IV presents our proposed protocol in detail. Section V describes the simulation model we use, and also discusses the results of our simulation. Section VI discusses some issues in our protocol and ways to improve it. Finally, section VII concludes and presents our plans for future work.

## II. RELATED WORK

There are many related papers that study the benefit of using multiple channels. Dual Busy Tone Multiple Access [12] divides a common channel into two sub-channels, one data channel and one control channel. Busy tones are transmitted on a separate control channel

to avoid hidden terminals, while data is transmitted on the data channel. This scheme uses only one data channel and is not intended for increasing throughput using multiple channels.

Hop Reservation Multiple Access [13] is a multi-channel protocol for networks using slow frequency hopping spread spectrum (FHSS). The hosts hop from one channel to another according to a predefined hopping pattern. When two hosts agree to exchange data by RTS/CTS handshake, they stay in a frequency hop for communication. Other hosts continue on hopping, and more than one communication can take place on different frequency hops. This scheme can be done using only one transceiver for each host, but it only applies to slow frequency hopping networks, and cannot be used in systems using other mechanisms such as direct sequence spread spectrum (DSSS).

Nasipuri et al. [7] proposes a multi-channel CSMA protocol with “soft” channel reservation. If there are  $N$  channels, the protocol assumes that each host can listen to all  $N$  channels concurrently. A host wanting to transmit a packet searches for an idle channel and transmits on that idle channel. Among the idle channels, one that was used for last successful transmission is preferred. In [6] the protocol is extended to select the best channel based on signal power observed at the sender side.

Wu et al. [9] propose a protocol that assigns channels dynamically, in an on-demand style. In this protocol, called *Dynamic Channel Assignment* (DCA), they maintain one dedicated channel for control messages, and other channels for data. Each host has two transceivers, so that it can listen on the control channel and the data channel simultaneously. RTS/CTS packets are exchanged on the control channel, and data packets are transmitted on the data channel. In RTS packet, the sender includes suggested data channel information according to the channel condition around itself. The receiver, on receiving RTS, decides which channel to communicate and includes the selected channel information in CTS packet. Then DATA and ACK packets are exchanged on the agreed data channel. This protocol does not need synchronization and can utilize multiple channels with little control message overhead. But it does not perform well in an environment where all channels have the same bandwidth. When the number of channels is small, one channel dedicated for control messages can be costly. In case of IEEE 802.11b, only 3 channels are available, so having one control channel can be interpreted as using 33% the total bandwidth as the control overhead. On

the other hand, if the number of channels is large, the control channel can become a bottleneck and prevent data channels from being fully utilized. [9] mentions that if every control packet has a length of  $L_c$  and data packet length is  $L_d$ , maximum number of channels should be no more than  $L_d/3L_c$ .

Jain et al. [8] propose a protocol that uses similar scheme as [9] in having one control channel and  $N$  data channels, but selects the best channel according to the channel condition at the receiver side. By intelligently selecting the data channel, it achieves throughput improvements, but still has the same disadvantages as DCA.

Compared to the above works, our protocol operates with one transceiver per host, and thus does not require any hardware change to current IEEE 802.11 device. Also, it does not require a separate control channel but considers all channels as identical. Instead, our scheme requires synchronization among all the hosts. IEEE 802.11 MAC protocol has a synchronization scheme that works for wireless LANs. Also, [14] discusses synchronization mechanisms for multi-hop networks. At the start of each interval we require all hosts to listen to a common channel in order to exchange traffic indication messages. During this interval hosts do not exchange data packets. So this duration of time will be an overhead in our scheme. But as we will see in later sections, it achieves better throughput and flexibility than maintaining a separate control channel, in an environment where the number of available channels ranges from 3 to 10, and all channels have the same fixed bandwidth.

### III. PRELIMINARIES

In this section, we present some background information on IEEE 802.11 DCF and power saving mechanism.

#### A. IEEE 802.11 Distributed Coordination Function

In IEEE 802.11 DCF, a node reserves the channel for data transmission by exchanging RTS/CTS messages with the target node. When a node wants to send packets to another node, it first sends an RTS (Ready to Send) packet to the destination. The receiver, on receiving RTS, replies by sending CTS (Clear to Send) packet to the sender. RTS and CTS packets include the expected duration of time for which the channel will be in use. Other hosts that overhear these packets must defer their transmission for the duration specified in the packets. For this reason, each host maintains a variable called *Network Allocation Vector* (NAV) that records the duration of time it must defer its transmission. This whole process is called *Virtual Carrier Sensing*, which allows

the area around the sender and receiver to be reserved for communication, thus avoiding the hidden terminal problem. Fig. 1 illustrates the operation of IEEE 802.11 DCF.

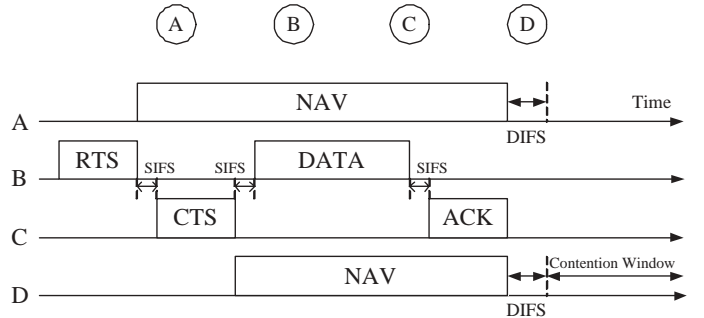


Fig. 1. Operation of IEEE 802.11 DCF. When B is transmitting a packet to C, A overhears the RTS packet and sets its NAV until the end of ACK, and D overhears the CTS packet and sets its NAV until the end of ACK. After the transmission completes, the stations wait for DIFS and then contend for the channel.

If a node has a packet to send but observes the channel busy, it performs a *random backoff* by choosing a backoff counter no greater than an interval called *contention window*. Each host maintains a variable  $cw$ , the contention window size, which is reset to a value  $CW_{min}$  when the node is initiated. Also, after each successful transmission,  $cw$  is reset to  $CW_{min}$ . After choosing a counter value, the node will wait until the channel becomes idle, and start decrementing the counter. The counter is decremented by 1 after each “slot” time, as long as the channel is idle. If the channel is busy, the node will freeze the counter until the channel is free again. When the backoff counter reaches 0, the node will try to reserve the channel by sending RTS to the target node. Since two nodes can pick the same backoff counter, the RTS packet may be lost because of collision. Since the probability of collision is higher as the number of nodes increase, a node will interpret the absence of CTS as a sign of congestion. In this case, the node will double its contention window to lower the probability of congestion.

Before transmitting a packet, a node has to wait for a small duration of time even if the channel is idle. This is called *interframe spacing*. Four different intervals enable each packet to have different priority when contending for the channel. SIFS, PIFS, DIFS, and EIFS are four interframe spacings, in an increasing order. A node waits for DIFS before transmitting RTS, but waits for SIFS before sending CTS or ACK, which is shorter. So an ACK packet will win the channel when contending with

RTS or DATA packets.

### B. IEEE 802.11 Power Saving Mechanism

A node can save energy by going into *doze* mode. In doze mode, a node consumes much less energy compared to normal mode, but cannot send or receive packets. So it is desirable for a node to enter the doze mode only when there is no need for exchanging data. In IEEE 802.11 power saving mechanism (PSM), this power management is done based on *Ad hoc Traffic Indication Messages* (ATIM). Time is divided into beacon intervals, and every node in the network is synchronized by periodic beacon transmissions. So every node will start and finish each beacon interval almost at the same time.

At the start of each beacon interval, there exists an interval called ATIM window, where every node should be in awake state and be able to exchange messages. If a node A has buffered packets destined for B, it sends an ATIM packet to B during this interval. If B receives this message, it will reply back by sending ATIM-ACK to A, and both A and B will stay awake for that entire beacon interval. If a node has not sent or received any ATIM packets during the ATIM window, it enters doze mode and stays until the next beacon time. This process is illustrated in Fig. 2.

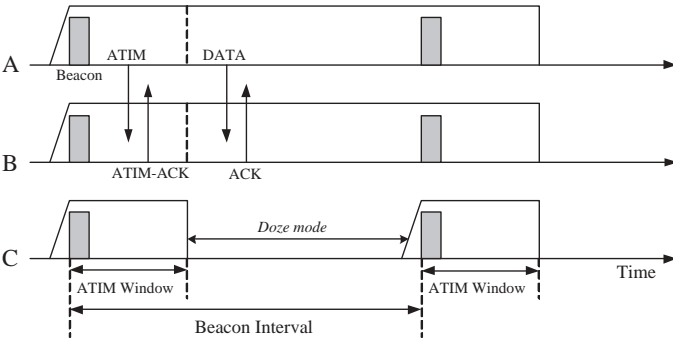


Fig. 2. Operation of IEEE 802.11 PSM. Nodes are synchronized by beacons. During the ATIM window, A sends an ATIM packet to B, notifying that it has packets for B. So A and B stays awake for the whole beacon interval. Since C does not send or receive any ATIM packets, it enters doze mode after the ATIM window until the end of beacon interval.

## IV. PROPOSED MULTI-CHANNEL MAC (MMAC) PROTOCOL

In this section, we present our proposed scheme. Before describing the protocol in detail, we first summarize our assumptions.

- $N$  channels are available for use, and all channels have the same bandwidth. None of the channels

overlap, so the packets transmitted on different channels do not interfere with each other. Hosts have prior knowledge on how many channels are available.

- Each host is equipped with a single half-duplex transceiver. So a host can either transmit or listen at a time, but cannot do both simultaneously. Also, a host can listen to or transmit on only one channel at a time. So when listening to one channel, it cannot sense carrier on other channels. Unlike our scheme, many other multi-channel MAC protocols require each host to have multiple transceivers [3], [8], [9].
- The transceiver is capable of switching its channel dynamically. The time elapsed for switching the channel is less than  $1\mu s$  [15], [16], which is a negligible overhead.
- Each host periodically sends out beacons to synchronize time in a distributed manner as in IEEE 802.11 power saving mechanism. When transmitting a beacon, the host includes a timestamp of its local timer. If a node receives a beacon from another node, it cancels its beacon and adjust its timer according to the timestamp included in the beacon. [14] argues that this scheme may not be enough for multi-hop networks, and suggests several solutions to synchronize time in multi-hop networks. Solutions to synchronize time are outside the scope of this paper, and this paper assumes synchronization that is achieved through beacons.

Now we describe our proposed scheme in detail. From now on, our protocol will be referred as *Multi-channel MAC* (MMAC).

### A. Preferable Channel List (PCL)

Each node maintains a data structure called *Preferable Channel List* (PCL), that indicates which channel is preferable to use for the node. PCL records the usage of channels inside the transmission range of the node. Based on this information, the channels are categorized into three states.

- *High preference* (HIGH): This channel has been already selected by the node itself for use in the current beacon interval. If a channel is in this state, this channel must be selected. At most one channel can be in this state at each node.
- *Medium preference* (MID): This channel has not been taken yet for use in the transmission range of the host. If there is no HIGH state channels, a channel in this state will be preferred.

- *Low Preference (LOW)*: This channel is already taken at least once for use in the node's transmission range. To balance the channel load as much as possible, there is a counter for each channel in the PCL to record how many source-destination pairs have planned to use the channel for that interval. If all channels are in LOW state, a node selects the channel with the least count.

The channel states are changed in the following way.

- All the channels in PCL are reset to MID state when the node is powered up, and at the start of each beacon interval.
- If the source and destination nodes agree upon a channel, they both record the channel to be in HIGH state.
- If a node overhears ATIM-ACK or ATIM-RES packet (explained in the next section), it changes state of the channel specified in the packet to be LOW, if it was previously in MID state. When the state of a channel changes from MID to LOW, the associated counter is set to 1. If the channel was previously in HIGH state, it stays in HIGH state. If the channel was already in LOW state, the counter for the channel is incremented by 1.

### B. Channel Negotiation during ATIM Window

In MMAC, periodically transmitted beacons divide time into beacon intervals and ATIM windows are used as in IEEE 802.11 PSM. The nodes that have packets to transmit negotiate channels with the destination nodes during this window. In the ATIM window, every node must not only stay in awake state, but also must listen to the *default channel*. The default channel is one of multiple channels available, which is selected for exchanging ATIM packets. Note that this channel is used for sending data outside the ATIM window, similar to other channels.

If a node S has buffered packets destined for D, it will notify D by sending an ATIM packet. S includes its preferable channel list (PCL) in the ATIM packet. D, upon receiving ATIM packet, selects one channel based on the sender's PCL and its own PCL. As explained in the next section, the receiver's PCL has higher priority in selecting the channel. After D selects a channel, it includes the channel information in ATIM-ACK packet and sends it to S. When S receives ATIM-ACK packet, it sees if it can also select the channel specified in ATIM-ACK. S can select the specified channel only except when S has already selected another channel (according to rules for selecting the channel, explained in the subsequent section). If S selects the channel specified

in ATIM-ACK, S sends ATIM-RES packet to the D, with S's selected channel specified in the packet. The ATIM-RES (ATIM-Reservation) is a new type of packet used in our scheme, which is not in IEEE 802.11 PSM. The ATIM-RES packet notifies the nodes in the vicinity of S which channel S is going to use, so that the neighboring nodes can use this information to update their PCL. Similarly, ATIM-ACK packet notifies the nodes in the vicinity of D. After the ATIM window, S and D will switch to the selected channel and start communicating by exchanging RTS/CTS.

If S cannot select the same channel as D, because it has already selected another channel, it cannot send packets to D during the beacon interval, and has to wait for the subsequent beacon interval to negotiate channels again. Even though S finishes transmitting all the scheduled packets on the selected channel during the beacon interval, it has to buffer all the packets destined for D until the next beacon interval. Since this can be a waste of bandwidth, we may want to let S send packets to D by switching its channel to the same channel as D, in the beacon interval. This issue is discussed in more detail in section VI.

When multiple nodes start sending ATIM packets at the beginning of a beacon interval, ATIM packets will collide with each other. To avoid such collisions, Each node waits for a random backoff interval before transmitting ATIM packet. The backoff interval is chosen in the range of 0 and  $CW_{min}$ .

Note that the receiver can always select a channel for use. Even if all the channels are selected for use in the receiver's transmission range, the receiver can select one of the channels. This is possible because the sender and receiver still exchange RTS/CTS before sending DATA packet, after the ATIM window. If two source-destination pairs that are closely placed choose the same channel, they will have to contend with each other just as in original IEEE 802.11.

Power saving is not the main goal of our protocol, but a node may save power by going into *doze* mode, if it has not transmitted or received ATIM packets during the ATIM window. The possibility of integration with IEEE 802.11 PSM is one of the advantage of our protocol. However, in our simulations, nodes do not go into doze mode.

### C. Rules for Selecting the Channel

When a node receives an ATIM packet, it selects a channel and notifies the sender by including the channel information in the ATIM-ACK packet. The receiver

tries to select the “best” channel based on information included in the sender’s PCL (preferable channel list) and its own PCL. By the best channel we mean the channel with the least scheduled traffic, as elaborated below. This selection algorithm will balance the channel load as much as possible, so that bandwidth waste caused by contention and backoff is minimized. For this reason, we count the number of source-destination pairs that have selected the channel by overhearing ATIM-ACK or ATIM-RES packets, and select the one with the lowest count. This scheme assumes that every source-destination pair will deliver the same amount of traffic in a beacon interval, which may not be true. A better approach may be to count the number of packets scheduled to be transmitted on the channel in the beacon interval. To do this, the source may need to include the number of pending packets in the ATIM packet. We take the former approach in this paper, and discuss the latter approach in section VI.

Here we describe the channel selection algorithm in detail. Suppose that node A has packets for B and thus sends an ATIM packet to B during the ATIM window, with A’s PCL included in the packet. On receiving the ATIM request from A, B decides which channel to use during the beacon interval, based on its PCL and A’s PCL. The selection procedure used by B is described as follows.

- 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 MID state at both A and B, it is selected. If there are multiple of them, one is selected arbitrarily.
- 4) Else if there is a channel which is in 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 LOW state, add the counters of sender’s PCL and receiver’s PCL, and the channel with the least count is selected. Tie is broken arbitrarily.

After selecting the channel, B sends an ATIM-ACK packet to A, specifying the channel it has chosen. When A receives the ATIM-ACK packet, A will see if it can also select the channel specified in the ATIM-ACK packet. If it can, it will send an ATIM-RES packet to B, with A’s selected channel specified in the packet. If A cannot select the channel which B has chosen, it does not send an ATIM-RES packet to B.

The process of channel negotiation and data exchange

in MMAC is illustrated in Fig. 3. During the ATIM window, A sends ATIM to B and B replies with ATIM-ACK indicating to use channel 1. This ATIM-ACK is overheard by C, so channel 1 will be in LOW state in C’s preferable channel list. When D sends ATIM to C, C selects channel 2. After the ATIM window, the two communications (between A and B, and C and D) can take place simultaneously.

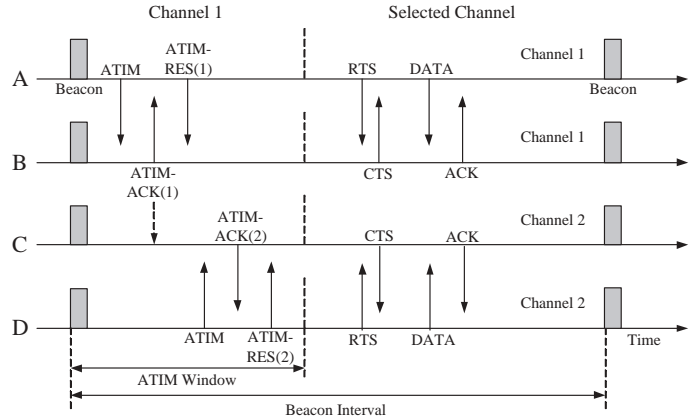


Fig. 3. Process of channel negotiation and data exchange in MMAC.

## V. PERFORMANCE EVALUATION

In this section we evaluate the performance of our protocol by simulation. We have compared our scheme with IEEE 802.11, as well as the Dynamic Channel Assignment protocol (DCA), proposed in [9] (DCA was explained in section II). Recall that the DCA protocol [9] uses a separate channel for exchanging control messages and uses other channels for data. This approach is also taken by [8], [10]. We have used two metrics to evaluate the performance of our protocol.

- 1) *Aggregate throughput over all flows in the network:*

The main goal of our protocol is to increase total throughput in the network using multiple channels. Thus, this metric will directly show how our protocol achieves the goal. Ideally, a multi-channel MAC will improve the total throughput by a factor of  $n$  over single-channel MAC given that  $n$  channels are available. But this is not true in practice because of the overhead required for negotiating channels and avoiding the hidden terminal problem.

- 2) *Average packet delay over all flows in the network:* Average packet delay is the duration between the time when the MAC layer of the sender receives



a packet to send, and the time the packet reaches the destination. So the delay is a sum of queueing delay, backoff delay, and transmission delay. The queue size at each node is 50 packets. We ignore the lost packets, and only measure delay of the packets that are correctly received by the receiver. Since MMAC uses ATIM windows, and data cannot be transmitted in these windows, packets will have to wait, which can result in increased packet delay.

### A. Simulation Model

For simulations, we have used ns-2 [17] with CMU wireless extension [18]. Simulations are performed in two network scenarios, wireless LAN and multi-hop networks. The bit rate for each channel is 2Mbps. The transmission range of each node is approximately 250m, and beacon interval is set to 100ms. Each source node generates and transmits constant-bit rate (CBR) traffic. Each simulation was performed for a duration of 40 seconds, and each data point in the result graphs is an average of 30 runs.

Unless otherwise specified, we assume 3 channels. Also we assume packet size is 512 bytes, and ATIM windows are 20ms long. To study the impact of these factors on the throughput and packet delay, we also performed simulations varying these parameters. The parameters we vary are number of nodes in the network, packet arrival rate of CBR traffic, packet size, ATIM window size, and number of channels.

1) *Wireless LAN*: In the simulated wireless LAN, all nodes are within each other's transmission range. So every source node can reach its destination in a single hop. The number of nodes we used are 6, 30, and 64. For each scenario, half of the nodes are sources, and other half are destinations. So a source has at most one destination. The impact of a source having multiple destinations or a destination having multiple sources is not studied in this scenario, but it is studied in the multi-hop network scenario.

First, we examine the throughput and packet delay varying the network load. We use the packet arrival rate of CBR flows to vary the network load. After that, we study the impact of different factors on the throughput. Packet size, ATIM window size, and number of channels are the factors we consider.

2) *Multi-hop network*: For a multi-hop network, 100 nodes are randomly placed in a 500m x 500m area. 40 nodes are randomly chosen to be sources, and 40

nodes are chosen to be destinations, such that all source-destination pairs are within a single hop. But a node may be the source for multiple destinations, and a node may be the destination for multiple sources. In a multi-hop network, we capture the situation where different traffic loads are present in different regions, which is not captured in wireless LAN scenario.

### B. Simulation Results

Simulation results are presented in this section. Note that in the graphs, the curves labelled as "802.11" refer to original IEEE 802.11 single channel MAC, the curves labelled as "DCA" indicate the DCA protocol from [9], and the curves labelled as "MMAC" indicate our proposed scheme.

First we present results from simulations performed for a wireless LAN. Fig. 4 shows the aggregate throughput of different protocols as the network load increases. The network sizes are 6, 30, and 64 nodes in Fig. 4(a), (b) and (c) respectively. When the network load is low, i.e. the network is not saturated, all protocols perform similarly. As the network load draws near saturation, MMAC performs significantly better than IEEE 802.11, and also does better than DCA. Since there are 3 channels, DCA uses 1 channel for control packets and other 2 channels for data. By using this separate control channel, DCA achieves almost 100% throughput improvement over IEEE 802.11. But as the number of channel increases, the throughput improvement of DCA for the added channel becomes less, because of bottleneck on control channel, as we will see later. MMAC uses all 3 channels for data exchange, but cannot achieve 200% improvement over IEEE 802.11, because of its overhead for channel negotiation. The overheads in MMAC are periodic beacon transmissions and ATIM packets. As the graphs show, MMAC performs 20%-30% better than DCA. The throughput improvement of MMAC over DCA may not be dramatic, but it is important that MMAC achieves this throughput using only a single transceiver for each host. Thus, the improvement is achieved using simpler hardware.

Fig. 5 shows the average packet delay of the protocols as the network load increases. The difference between IEEE 802.11 and other protocols in delay is due to the fact that with only one channel, a packet has to wait longer to occupy the channel when the network load is high. When comparing DCA and MMAC, MMAC shows higher delay in the network scenario with 6 nodes. Then the delay of two protocols becomes similar with 30 nodes, and MMAC outperforms DCA in 64-

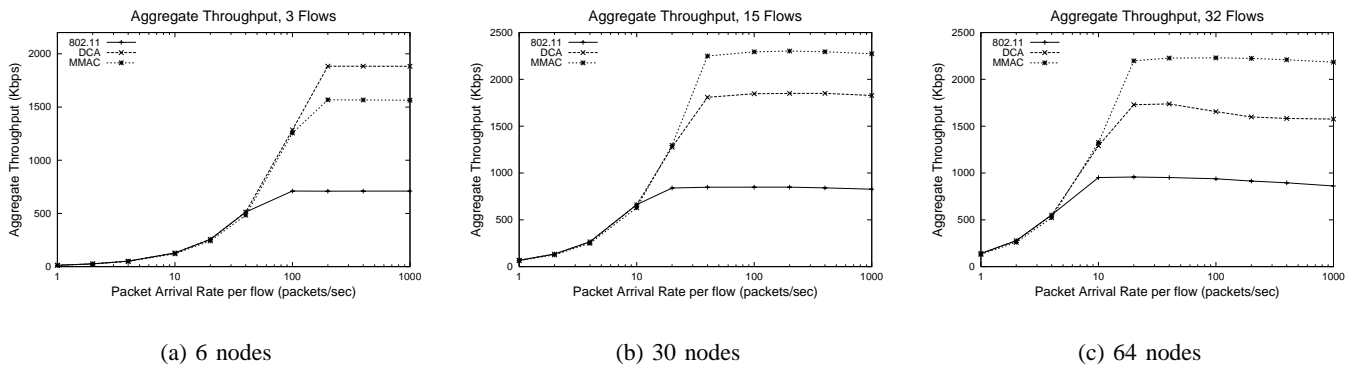


Fig. 4. Aggregate Throughput vs. Packet Arrival Rate in a wireless LAN.

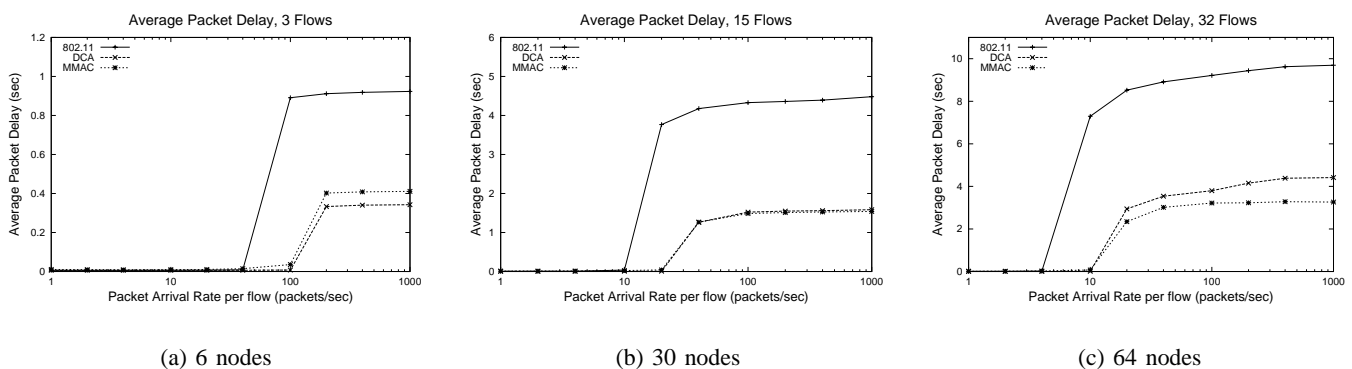


Fig. 5. Average Packet Delay vs. Packet Arrival Rate in a wireless LAN.

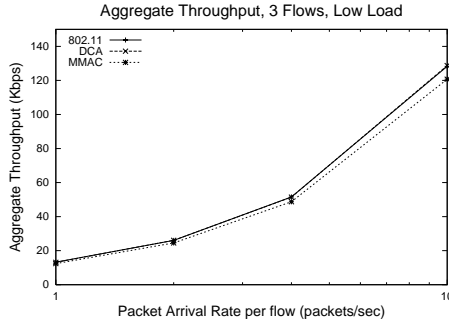
node scenario. In 6-node scenario, MMAC shows higher delay even though it achieves higher throughput. This is because only ATIM packets are allowed to be exchanged in ATIM windows. For 20ms in every 100ms, packets have to wait for channel negotiation, whether or not the channel is idle. But when the number of nodes becomes large, DCA suffers from high contention at the control channel which results in high packet delay. MMAC does not have this problem, because it does not maintain a separate channel for control messages. Since Fig. 5 shows the delay for a wide range of network load, the difference in delay when the network load is low is not shown clearly. Fig. 6 shows the aggregate throughput and average packet delay of the protocols only when the network load is low. We can see from Fig. 6(b) that MMAC has significantly larger packet delay than other protocols, even though all the protocols achieve almost the same amount of throughput.

each flow to be 100 packets/sec, so the network is already being saturated. In general, aggregate throughput is higher when packet size is large, mainly because of less control overhead. With larger packets, larger amount of data is transmitted for one RTS/CTS exchange, and thus contending over the channel occurs less frequently. we can observe from Fig. 7 that for DCA, throughput is close to IEEE 802.11 when the packet size is smaller than 256 bytes, but shows sharp increase when the packet size is greater than 256 bytes. Recall that DCA uses a common channel for all nodes to exchange control messages. When packet size is too small, control channel becomes a bottleneck so that available data channels cannot be utilized to its maximum. This indicates that as the number of channels increase, the packet size should also increase in DCA to fully utilize the data channels. MMAC does not have this bottleneck problem, since there is

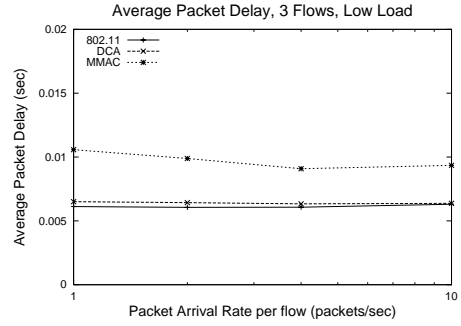
In Fig. 7, we simulated the protocols using different packet sizes. We have fixed the packet arrival rate of

Now we look at results from multi-hop network. As stated in the previous section, in our multi-hop network



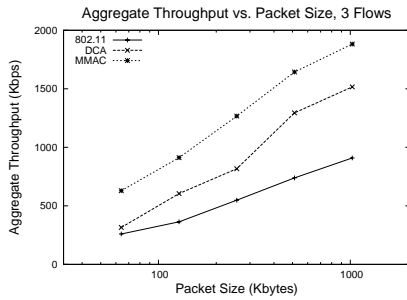


(a) Aggregate Throughput

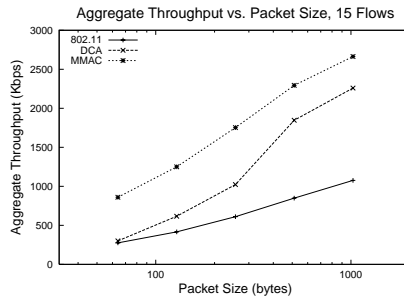


(b) Average Packet Delay

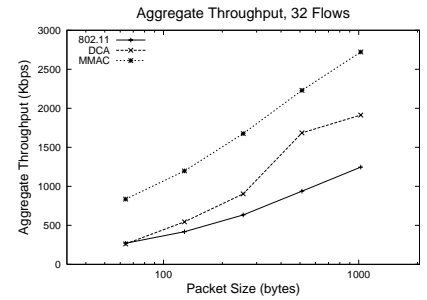
Fig. 6. Aggregate Throughput and Average Packet Delay when the network load is low, in a wireless LAN.



(a) 6 nodes



(b) 30 nodes



(c) 64 nodes

Fig. 7. Aggregate Throughput vs. Packet Size in a wireless LAN.

simulations, a node can be a source for multiple destinations, or it can be a destination for multiple sources. Fig. 8 shows the aggregate throughput of different protocols as the network load increases. 3 and 4 channels are used in Fig. 8(a) and 8(b) respectively. In 8(a), MMAC performs better than DCA, but the difference is smaller than in wireless LAN case. This is due to the following reasons. First, in a multi-hop network, all 3 channels may not be fully utilized in the entire area. In the region where the network can benefit from having the third channel, MMAC does better than DCA. But in the region where only 2 channels are needed, DCA does better than MMAC because it can utilize 2 data channels without ATIM window overhead. Second, in MMAC, if a node has flows to two different destinations, each destination may choose a different channel, and one flow may have to wait for an entire beacon interval to negotiate the channel again. Also, if a node is a destination for two flows from other sources, these two flows must be transmitted on the same channel, diminishing the

benefit of having multiple channels. As the network load becomes very high, throughput of DCA drops faster than MMAC. This is because a single control channel is shared by every node in DCA. When the network load is very high, the collision rate of control packets increases, degrading the throughput. We call this *control channel saturation*.

The impact of control channel saturation is also shown in Fig. 8(b). MMAC gains significant benefit from not having a dedicated control channel. As mentioned in section II, the maximum number of channels that can be fully utilized using DCA is  $L_d/3L_c$ , given that  $L_d$  is the data packet size and  $L_c$  is the control packet size. But even when the number of channels is less than  $L_d/3L_c$ , the throughput suffers from contentions among the control packets if the number of channels is close to this value. This control channel bottleneck can be removed if we use larger packets. The results using large packets are shown in Fig. 9(a) and 9(b). Here we used 1024 bytes as the packet size. As the results

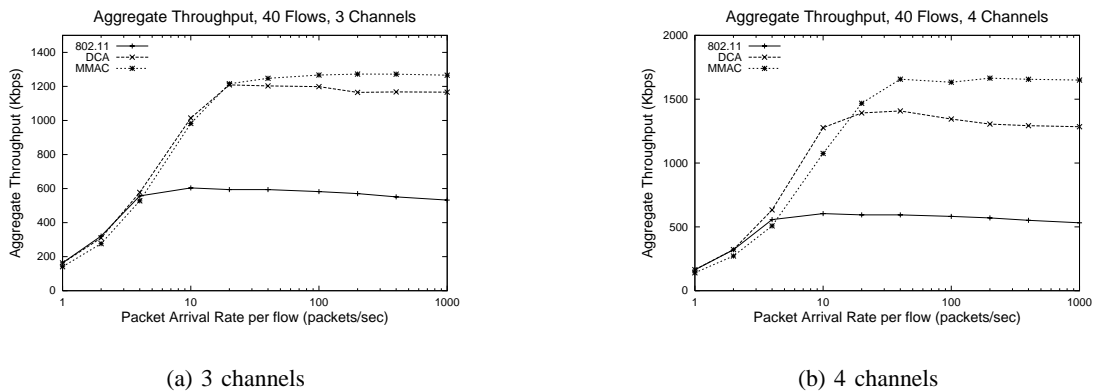


Fig. 8. Aggregate Throughput vs. Packet Arrival Rate in a multi-hop network. Packet size is 512 bytes.

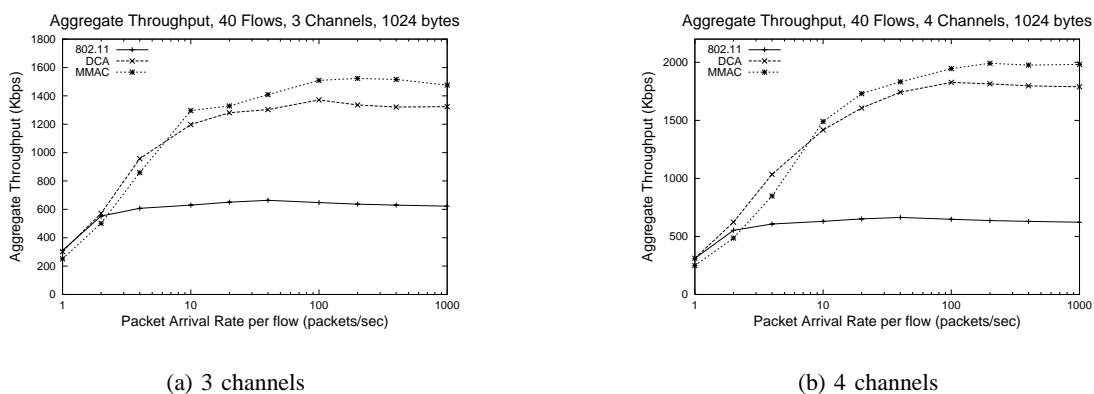


Fig. 9. Aggregate Throughput vs. Packet Arrival Rate in a multi-hop network. Packet size is 1024 bytes.

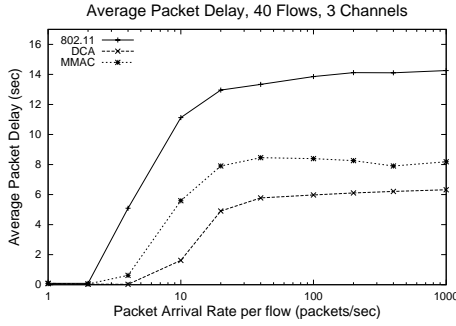
show, MMAC only does slightly better than DCA, both with 3 and 4 channels. However, MMAC achieves this improvement with simpler hardware than DCA.

Fig. 10(a) and 10(b) shows the average packet delay of the protocols as the network load increases. Packet size is again 512 bytes in these graphs. With 3 channels, MMAC shows higher delay than DCA, even though MMAC achieves higher throughput. This is due to the same reasons explained in wireless LAN scenario. However, when 4 channels are available, MMAC shows lower delay than DCA. We can see that the average packet delay of DCA is almost the same with 3 and 4 channels. This is because DCA does not benefit from having one more channel because of control channel saturation, as mentioned above. But MMAC benefits from having the fourth channel and the average delay becomes lower. ATIM window overhead in MMAC does not increase with the number of channels, as long as the ATIM window is long enough to exchange all the ATIM

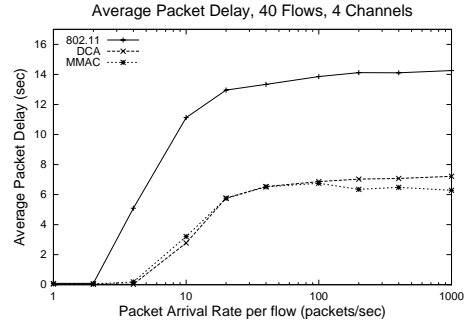
messages necessary.

We have fixed the ATIM window size to  $20ms$  so far in this paper. But this may be undesirable due to the following reason. When there are small number of flows in the network, using 20% of each beacon interval for exchanging ATIM messages is wasteful. Much of the time the channel will be left as idle, because data packets are not allowed to be transmitted in this interval. On the other hand, if there are very large number of flows in the network, a longer ATIM window would be needed to exchange all the ATIM messages between nodes to negotiate channels. Thus the ATIM window size affects the throughput of MMAC protocol<sup>4</sup>. To study this impact, aggregate throughput is measured using different ATIM window sizes. Fig. 11 shows the result. For this network scenario, an ATIM window

<sup>4</sup>A similar observation about impacts of ATIM window on IEEE 802.11 power saving mechanism is reported in [19].



(a) 3 channels



(b) 4 channels

Fig. 10. Average Packet Delay vs. Packet Arrival Rate in a multi-hop network.

size of around  $15\text{-}20\text{ms}$  is shown to be the best for throughput. When the ATIM window size is less than  $15\text{ms}$ , not all nodes can exchange ATIM messages and negotiate channels during this interval. Nodes that have not successfully exchanged ATIM messages stay in the default channel. So the multiple channels cannot be fully utilized, resulting in degraded throughput. If the ATIM window is longer than  $20\text{ms}$ , the throughput decreases because while the overhead increases, no more ATIM messages are exchanged after  $20\text{ms}$  and thus cannot benefit from having a longer ATIM window. The optimal ATIM window size depends mainly on the number of flows in the beacon interval, because an ATIM packet exchange is required for each flow. Since the number of flows is often dynamically changed, it is desirable to also make the ATIM window size change dynamically. Changing ATIM window size dynamically to achieve maximum throughput is left as a future work.

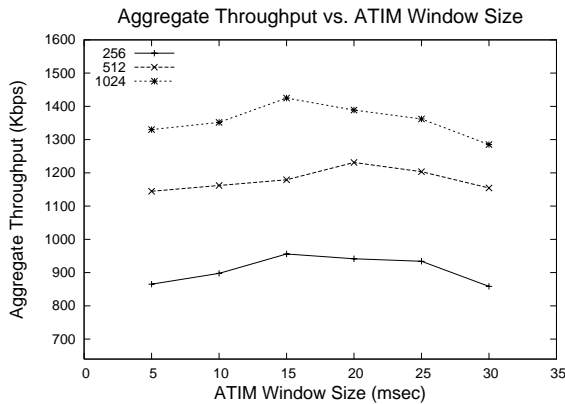


Fig. 11. Aggregate Throughput vs. ATIM Window Size in a multi-hop network.

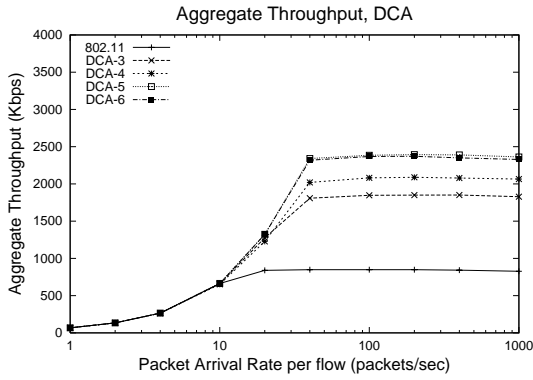
Finally, we measured the throughput of different protocols varying number of channels. This simulation was done for a wireless LAN with 30 nodes. We used 512 bytes for packet size, and the number of channels vary from 3 to 6. The results are shown in Fig. 12. In the graphs, “MMAC-3” indicates MMAC protocol with 3 channels, and “DCA-4” refers to DCA protocol with 4 channels. The throughput of IEEE 802.11 is also shown in the graphs. Because of control channel saturation, DCA does not benefit from having additional channels when the number of channels becomes larger. MMAC does better than DCA with the same number of channels, when the network load is high.

DCA and MMAC have their own ways to avoid the hidden terminal problem and access multiple channels dynamically. DCA uses one separate channel to exchange control packets, whereas MMAC uses ATIM windows to negotiate channels. In DCA, the bandwidth of control channel has a major effect on the performance. In MMAC, the ATIM window size takes the role. As the results show, MMAC can achieve better throughput than DCA, with simpler hardware.

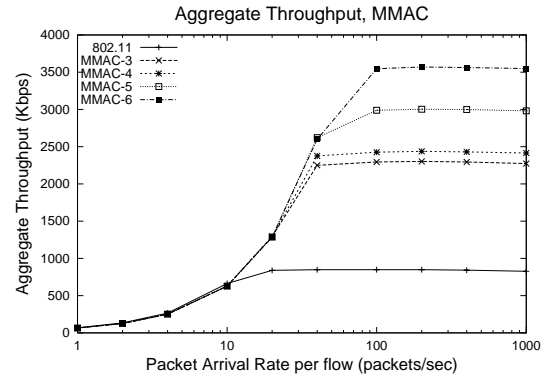
## VI. DISCUSSION

In this section, we discuss some issues to be considered regarding our scheme and possible ways to improve it.

When a node is sending packets to two different destinations, these two destination nodes may select a different channel. For example, suppose that we have nodes A, B and C in the network, as in Fig. 13. Node A has packets destined for B, and also packets destined for C. During channel negotiation, node B selects channel 1 and node C selects channel 2. If A selects channel



(a) DCA



(b) MMAC

Fig. 12. Aggregate Throughput vs. Packet Arrival Rate in a wireless LAN (30 nodes).

1, it can only transmit packets destined for B, and all the packets destined for C must wait until next beacon interval to negotiate the channel again. This behavior of MMAC protocol raises several issues. First, to avoid *head of line blocking* problem, the packets that cannot be transmitted because of channel mismatch must be kept in a separate buffer, and be restored to the front of the buffer at the end of the beacon interval. This complicates the queue management. Also, it is possible that the same channels are selected by each node in the subsequent beacon intervals, starving the flow from A to C. In our scheme, if node A has to send ATIM packets to B and C, A chooses randomly which one to send the packet to first. This randomness will prevent complete starvation, although there can be short-term unfairness among the flows. Instead of randomly choosing among the destinations, A can send an ATIM packet first to the destination which is the target node of the first packet in its queue. This modification will improve the fairness of the protocol.

In addition to the problems stated above, this situation might also have impact on the throughput. Suppose that A had only a few packets for B. Then after sending all the scheduled packets, A becomes idle for the rest of the beacon interval. But A cannot send packets to C, even though A has received C's ATIM-ACK during ATIM window and knows which channel C will be listening on. To avoid waste of bandwidth, we can extend MMAC to allow nodes to switch channels inside the beacon interval. A node may switch channels according to the following rules.

- If node A finishes sending packets on its selected

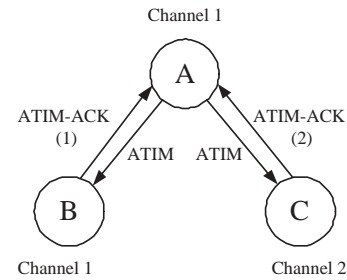


Fig. 13. An example network scenario. Assume there are other nodes in the vicinity of these three nodes, that affect the PCL of these three nodes. Node A has packets for B, and also packets for C. A exchanges ATIM messages with B first, and both select channel 1. After that, A sends an ATIM packet to C, and C selects channel 2. Since A will stay in channel 1 for the beacon interval, packets for C must be deferred until the next beacon interval.

channel, and does not know of any node that is planning to send packets to A, A may switch its channel.

- If A has received any ATIM packet during the ATIM window, A must stay on the selected channel for the entire beacon interval.
- After switching to another channel, node A must wait for one packet transmission interval before transmitting a packet, to gather information on condition of the new channel. This delay is required to avoid collision, because A does not have NAV (Network Allocation Vector) information for the new channel at the time it switches channels.

In our example, A can switch to channel 2 after sending all the scheduled packets to B, because it has not received any ATIM packets during the ATIM window. C

stays in channel 2 for the entire beacon interval, because it has received ATIM packet from A. So the communication between A and C can take place in channel 2 correctly. This extended scheme might increase the throughput, and the performance of this scheme will be studied in the future work.

Another issue is that in MMAC, a node counts the usage of a channel based on the ATIM-ACK or ATIM-RES packets it overhears, and selects the channel with the least count to balance the channel load. It means that the node is counting the number of *source-destination pairs*. The assumption here is that every flow has the same amount of traffic ready to be transmitted in the beacon interval, which may not be true. Different flows may have different number of packets pending to be sent. So it may be better to count the number of pending packets rather than number of source-destination pairs. This could be easily achieved. The source counts the number of pending packets for the destination and include the value in the ATIM packet. This number is echoed in the ATIM-ACK and ATIM-RES packet, so that the nodes in the vicinity of the source or destination can obtain the information. When selecting a channel, the node selects a channel with the least number of packets scheduled on the channel. This selection mechanism will achieve a better load balancing than the original scheme.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented a multi-channel MAC protocol that utilizes multiple channels to improve throughput in wireless networks. The proposed scheme requires only one transceiver for each host, while other multi-channel MAC protocols require multiple transceivers for each host [3], [8], [9]. The nodes in the network are synchronized by beacons, and the channels are negotiated in the ATIM window using ATIM packets. After the ATIM window, nodes switch to their selected channel and exchange messages on that channel for the rest of the beacon interval. Since multiple transmissions occur at the same time, MMAC improves the throughput of a wireless network significantly, even with small number of channels.

Simulation results show that MMAC performs significantly better than IEEE 802.11. Compared with DCA, MMAC performs better in terms of throughput, but often somewhat worse in terms of packet delay. It is important that MMAC achieves a comparable performance using simpler hardware than DCA. Also, since the ATIM windows are already specified in IEEE 802.11 PSM, if

PSM is used, MMAC can be easily integrated to achieve multi-channel accessibility without further overhead.

## REFERENCES

- [1] IEEE 802.11 Working Group, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," 1997.
- [2] J. Monks, V. Bharghavan, W. Hwu, "A Power Controlled Multiple Access Protocol for Wireless Packet Networks," in *IEEE INFOCOM*, April 2001.
- [3] A. Nasipuri, S. Ye, J. You and R. Hiromoto, "A MAC Protocol for Mobile Ad Hoc Networks using Directional Antennas," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, September 2000.
- [4] R. Roy Choudhury, X. Yang, R. Ramanathan and N. H. Vaidya, "Using Directional Antennas for Medium Access Control in Ad Hoc Networks," in *ACM MOBICOM*, Atlanta, GA, September 2002.
- [5] Y.-B. Ko, V. Shankarkumar and N. H. Vaidya, "Medium Access Protocols using Directional Antennas in Ad Hoc Networks," in *IEEE INFOCOM*, March 2000.
- [6] A. Nasipuri and S. R. Das, "Multichannel CSMA with Signal Power-based Channel Selection for Multihop Wireless Networks," in *IEEE Vehicular Technology Conference (VTC)*, September 2000.
- [7] A. Nasipuri, J. Zhuang and S. R. Das, "A Multichannel CSMA MAC Protocol for Multihop Wireless Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, September 1999.
- [8] N. Jain and S. Das, "A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks," in *Proceedings of the 9th Int. Conf. on Computer Communications and Networks (IC3N)*, October 2001.
- [9] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng and J.-P. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, September 2000.
- [10] W. Hung, K. Law and A. Leon-Garcia, "A Dynamic Multi-Channel MAC for Ad Hoc LAN," in *21st Biennial Symposium on Communications*, April 2002.
- [11] 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.
- [12] J. Deng and Z. Haas, "Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks," Florence, Italy, 1998.
- [13] Z. Tang and J. J. Garcia-Luna-Aceves, "Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks," in *IEEE INFOCOM*, 1999.
- [14] Y.-C. Tseng, C.-S. Hsu and T.-Y. Hsieh, "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks," in *IEEE INFOCOM*, June 2002.
- [15] S.-L. Wu, Y.-C. Tseng, C.-Y. Lin and J.-P. Sheu, "A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad Hoc Networks," *The Computer Journal*, vol. 45, pp. 101–110, 2002.
- [16] R. Garces and J. J. Garcia-Luna-Aceves, "Collision Avoidance and Resolution Multiple Access for Multichannel Wireless Networks," in *IEEE INFOCOM*, 2000.
- [17] VINT Group, "UCB/LBNL/VINT network simulator ns (version 2)." [Online]. Available: <http://www.isi.edu/nsnam/ns>

- [18] The CMU Monarch Project, "Wireless and Mobility Extension to ns."
- [19] H. Woesner, J. Ebert, M. Schlager and A. Wolisz, "Power-saving mechanisms in Emerging Standards for Wireless LANs: The MAC Level Perspective," *IEEE Personal Communications*, June 1998.