An Energy Efficient MAC Protocol for Wireless LANs

Eun-Sun Jung and Nitin H. Vaidya
Texas A&M University
{esjung,vaidya}@cs.tamu.edu

Technical Report TR01-017

Department of Computer Science
Texas A&M University

July 31, 2001

Abstract— **This paper presents an optimization of the power saving mechanism in the Distributed Coordination Function (DCF) in IEEE 802.11 standard. In the 802.11 power saving mode specified for DCF, time is divided into so-called *beacon intervals*. At the start of each beacon interval, each node in the power saving mode periodically wakes up for a duration called the *ATIM Window*. The nodes are required to be synchronized to ensure that all nodes wake up at the same time. During the ATIM window, the nodes exchange control packets to determine whether they need to stay awake for the rest of the beacon interval. The size of the ATIM window has a significant impact on energy saving and throughput achieved by the nodes. This paper proposes an adaptive mechanism to dynamically choose a suitable ATIM window size. We also allow the nodes to stay awake for only a fraction of the beacon interval following the ATIM window. On the other hand, 802.11 DCF mode requires the nodes to stay awake either for the entire beacon interval following the ATIM window or none at all. Simulation results show that the proposed approach outperforms the IEEE 802.11 power saving mechanism in terms of throughput and the amount of energy consumed.**

Keywords— **energy efficiency, power saving, MAC, wireless, dynamic ATIM window.**

I. INTRODUCTION

WIRELESS hosts are often powered by batteries. The batteries can provide a finite amount of energy, therefore, to increase the battery lifetime, it is important to design techniques to reduce energy consumption by the wireless hosts. In the last few years, there has been significant activity with the goal of conserving energy. Past research has investigated energy conserving mechanisms at various layers of the protocol stack, including work on routing [1] [2] [3] [4], medium access control (MAC) [5] [6] [7] [8] [9], and transport protocols [10] [11] [12]. This paper focuses on an energy efficient MAC protocol for wireless LANs, which attempts to reduce energy consumption by putting the wireless interfaces in a "doze" state when deemed reasonable.

A wireless interface can be in *awake* state, *doze* state or *off* state [13] – we may say that a node is in a certain state, when its wireless interface is in that state. In the *off* state, the wireless interface consumes no power. In the *doze state* also a node cannot transmit or receive, and consumes very little power. In the *awake state*, a node may be in one of three different modes, namely, transmit, receive, and idle modes, and consumes somewhat different power in each mode (although all three modes in awake state consume significantly more power than the doze state). In this paper, we only consider the awake and doze states. Thus, the proposed MAC protocol may place a wireless interface in either the awake or the doze state at any given time. Transition from the doze state to the awake state requires small duration of time (for instance, [14] reports $250\mu s$ for this transition). Also, transition from the doze state to the awake state results in additional energy consumption during the transition [15].

As noted above, except in the off state, the wireless network interface consumes energy. For instance, Lucent IEEE 802.11 WaveLAN card [16] [17] consumes 1.65 W, 1.4 W and 1.15 W in the transmit, receive and idle modes, respectively, in the awake state. In the doze state, WaveLAN consumes 0.045 W [16]. Clearly, a significant amount of energy is consumed even in the idle mode. This

This research is supported in part by National Science Foundation.

occurs due to the CSMA/CA mechanism in IEEE 802.11 which requires each awake node to continually listen to the channel.

IEEE 802.11 consists of two components: PCF and DCF. PCF (Point Coordination Function) is a centralized medium access control protocol, whereas DCF (Distributed Coordination Function) is a fully distributed protocol [13]. IEEE 802.11 specifies a power saving mechanism for both PCF and DCF. This paper focuses on the power saving mechanism proposed for DCF (Distributed Coordination Function).

In DCF, time is divided into the so-called *beacon intervals* by means of a distributed protocol for beacon transmission. One purpose of the beacons is to synchronize the different nodes [13][18] – so far as our power saving scheme is concerned, any other mechanism, such as GPS [19], may also be used to synchronize the nodes if available. The beaconing protocol has a shortcoming (particularly when used in conjunction with the power saving mode): when two groups of nodes that are initially partitioned from each other come together it may not be possible for them to synchronize their beacon intervals. However, the beaconing protocol can work in networks that are connected and remain connected. In this paper, we do not consider the problem of synchronizing the nodes, and assume that a mechanism exists for this purpose. For the rest of our discussion, we will assume that time is divided into *beacon intervals* that begin and end approximately at the same time at all nodes. A similar approach is also taken in [18].

The *power saving* mechanism specified for DCF requires that, at the start of each beacon interval, each node must stay awake for a fixed time interval, called ATIM window (ATIM stands for Ad-hoc Traffic Indication Message). Thus, during an ATIM window, all nodes are in awake state. The ATIM window is utilized to announce any packets pending transmission to nodes in doze state, as described later. An earlier investigation [20] shows that any fixed size of the ATIM window cannot perform well in all situations, when throughput and energy consumption are considered. This paper presents an adaptive mechanism to dynamically adjust the size of ATIM window. We refer to our protocol as Dynamic Power Saving Mechanism (DPSM). Note that DPSM allows different nodes to use different ATIM window sizes.

The rest of the paper is organized as follows. Section 2 presents an overview of the DCF power saving mechanism in IEEE 802.11, and Section 3 reviews the related work. Proposed approach, DPSM (Dynamic Power Saving Mechanism), is presented in Section 4. Section 5 describes our simulation model and discusses the simulation results. Section 6 concludes the paper.

II. POWER SAVING MECHANISM FOR DCF IN IEEE 802.11

Figure 1 illustrates the power saving mechanism (hereafter referred as PSM) for DCF. As noted before, time is divided into beacon intervals [13]. At the start of each beacon interval, each node stays awake for an ATIM window (Ad-hoc Traffic Indication Message window) interval. We describe the power saving mechanism using Figure 1.

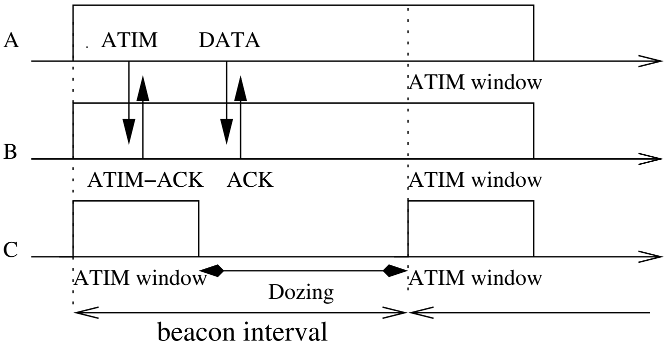


Fig. 1. Power save mode for DCF: Node A announces a buffered packet for B using an ATIM frame. Node B replies by sending an ATIM-ACK, and both A and B stay awake during the entire beacon interval. The actual data transmission from A to B is completed during the beacon interval. Since C does not have any packet to send or receive, it dozes after the ATIM window.

In PSM, when any node has a packet destined for another node, this packet is announced during a subsequent ATIM window. For instance, in Figure 1, node A announces a packet destined for node B by transmitting an "ATIM frame" during the ATIM window. The transmission of an ATIM frame is performed using the CSMA/CA (collision avoidance) mechanism specified in 802.11. When a node has sent an ATIM frame to another node, such as node A in our example, the node remains awake for the entire beacon interval. A node that receives an ATIM frame replies by sending an ATIM-ACK. Such a node remains awake for the entire beacon interval, after transmitting the ATIM-ACK.

In our example, node B sends ATIM-ACK to node A and remains awake for the rest of the beacon interval. Transmission of one or more data packets from node A to B can now take place during the beacon interval, after the end of the ATIM window. A node that has no outstanding packets to be transmitted can go into the doze state at the end of the ATIM window, if it does not receive an ATIM frame during the ATIM window. In the example in Figure 1, node C dozes after the ATIM window, thus saving energy. All dozing nodes again wake up at the start of the next beacon interval.

In PSM specified in 802.11, all nodes use the same (fixed) ATIM window size, as well as identical beacon intervals [13]. Since the ATIM window size critically affects throughput and energy consumption, a fixed ATIM window does not perform well in all situations, as shown in [20]. If the ATIM window is chosen to be too small, there may not be enough time available to announce buffered packets (by transmitting ATIM frames), potentially degrading throughput. If the ATIM window is too large, there would be less time for the actual data transmission, since data is transmitted after the end of the ATIM window, again degrading throughput at high loads. Large ATIM windows can also result in higher energy consumption since all nodes remain awake during the ATIM window. At low load, in particular, large ATIM windows are unnecessary. Thus, a static ATIM window size cannot always perform well. This paper proposes a dynamic mechanism for choosing an ATIM window size.

III. RELATED WORK

[20] presents simulation results for the power saving mechanisms of two wireless LANs standards, IEEE 802.11 and HIPERLAN. It shows that the different sizes of beacon interval and ATIM window in IEEE 802.11 have significant impact on throughput and energy consumption.

IEEE 802.11 uses a handshake consisting of RTS (Request-To-Send) and CTS (Clear-To-Send) preceeding transmission of a data packet. [21] proposes a power-conserving algorithm that requires a node to enter the doze state if it overhears RTS/CTS for data transmission between some other nodes (the RTS and CTS packets specify duration of the impending data transfer, which can be used to determine the doze period). However, this approach is not always suitable due to the time and energy costs associated with a doze-to-active transition – these costs can make it expensive to transition between awake and doze states on a per-packet basis. Our scheme, on the other hand, makes transitions between awake and doze states at most once per beacon interval.

SPAN [18], a power saving technique, elects a group of "coordinators" which are changed periodically. The coordinators stay awake and forward traffic for active connections. Non-coordinators follow the power saving mechanism in IEEE 802.11. Nodes buffer the packets for dozing destinations and announce these packets during ATIM window. SPAN introduces a new advertised traffic window following an ATIM window. During this advertised traffic window, the announced packets and the packets for the coordinators can be transmitted. After this window, only the packets for the coordinators can be transmitted, and non-coordinators can go to doze state if they do not have traffic to send or receive.

[22] proposes a communication protocol in wireless microsensor networks, where each node usually has limited energy capacity and limited wireless channel bandwidth. A cluster-based routing protocol, LEACH (Low-Energy Adaptive Clustering Hierarchy), is proposed to minimize global energy consumption by distributing the load to all the nodes.

In PAMAS [5], each node uses two separate channels for control packet and data packet transmissions. Using the control channel, a node determines when to power off and the power-off duration. This scheme has the disadvantage of requiring two channels for communication.

IV. PROPOSED DPSM SCHEME

We propose a *dynamic power saving mechanism* (DPSM). We now describe the features of DPSM scheme.

- In the proposed DPSM scheme, each node independently chooses an ATIM window size based on observed network conditions. This might potentially result in each node using a different ATIM window size.
- In PSM specified in 802.11, when a node transmits or receives an ATIM frame during an ATIM window, it must stay awake during the entire beacon interval. This approach allows a single ATIM frame from, say node A to node B, to be followed up by multiple data packets during the remaining beacon interval – that is, a single ATIM frame and ATIM-ACK exchange can be used to deliver several packets in the same beacon interval. While this approach has its advantages, at low loads, this approach results in a much higher energy consumption than necessary. In the proposed DPSM scheme, we allow a node to enter the doze state after completing any transmissions that are explicitly announced in the ATIM window. For instance, if node A has only one packet pending, say, for node B, and no other node has a packet pending for node B, then both nodes A and B would enter the doze state after completing transmission of the single packet (when using DPSM) – on the other hand, with PSM in 802.11, both A and B will remain awake for the entire beacon interval. As noted above, 802.11 approach has the benefit of being able to transmit multiple packets to a destination following a single ATIM frame. We incorporate another mechanism described in the next item to gain the benefit of the IEEE 802.11 approach.

As noted previously, there is a finite delay associated with the doze-to-awake transition, in addition to a higher energy consumption. Therefore, in our scheme, a node will not enter the doze state after completing packet transmissions if the remaining duration in the current beacon interval is “too small” – specifically, in our simulations, a node will not enter the doze state if the remaining duration is less than 1600μs – the delay for both doze-to-awake transition and awake-to-doze state transition are considered as 800μs each [15][23].

- When a node, say node A, successfully transmits an ATIM frame to another node, say node B, node A will not transmit another ATIM frame to the same destination in the same beacon interval. Instead, node A includes in each packet sent to B the number of packets still pending for node B. This information allows node B to determine when it has received all the packets pending at node A at the time of data transmission. If node A could not deliver all pending packets that were previously announced to node B, and the current beacon interval expires, nodes A and B both stay up in the next beacon interval, with B anticipating the remaining packets from node A, without node A having to send an ATIM frame to node B. Node A delivers the remaining packets to B, and then may enter the doze state if it has no other announced pending packets to transmit or receive. Similarly, node B may enter the doze state after receiving the previously announced packets, if it is not aware of any other pending packets.
- DPSM specifies rules for dynamically increasing and decreasing the ATIM window size. This may result in different nodes using different ATIM window sizes. We specify a finite set of ATIM window sizes that may be used by each node, with the smallest ATIM window size being denoted as *ATIMmin*. Each allowed window is called a $level$.
- As noted previously, transmission of each ATIM frame is performed using the CSMA/CA mechanism in 802.11. In particular, a node wanting to transmit an ATIM frame picks a backoff interval in the range $[0, cw]$ where cw denotes the *contention window*. The initial cw value is called $cwmin$. The backoff interval is decremented by 1 after each “clock tick” if the channel is sensed as idle [13]. An ATIM frame is transmitted when the backoff interval reaches 0. When the ATIM frame is received by the destination node, it will respond by sending an ATIM-ACK. However, the ATIM frame may collide with an ATIM frame transmitted by another node. In this case, the ATIM-ACK will not be sent. If an ATIM-ACK is not received in response to the transmitted ATIM frame, the node transmitting the ATIM frame doubles the value of cw, selects a new backoff interval, and repeats the process. Note that, while the backoff interval is being decremented, say, at node A, the ATIM window of node A might end. In this event, the node will attempt to send an ATIM frame for the corresponding destination again in the next beacon interval. The cw used in the next beacon interval is also doubled. This will decrease the probability of collision within the next ATIM window. Backoff interval range (i.e., cw) for a given destination will not be reset to $cwmin$ until the node receives ATIM-ACK from that destination – that is, until a packet is successfully announced to that destination.
- Packet marking.

As noted above, when a node transmits an ATIM frame, an ATIM-ACK may not be received in response. In such cases, the node will retransmit the ATIM frame. We set the retry limit for ATIM frame as 3 – that is, ATIM frame will be transmitted three times, before the retry limit is reached. If ATIM-ACK has not been received after three transmissions, the transmitted packet is “marked” and rebuffered for another try (also up to 3 times) in the next beacon interval. Thus, after three attempts in a beacon interval, the ATIM frame for a given destination is only transmitted again in the next beacon interval. In the current interval, the node is then free to transmit ATIM frame to another node. A rebuffered packet can stay in the buffer for at most 2 beacon intervals. If ATIM frame cannot still be successfully transmitted, the packet will be dropped. The “marking” performed here is useful to dynamically increase ATIM window size, as explained below.

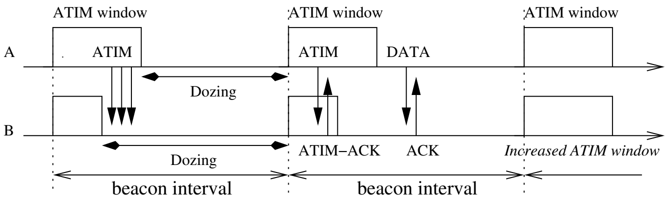


Fig. 2. DPSM: node A fails to announce a packet, because node B is already in doze state. A will $mark$ the packet and retry it during the next ATIM window. When B receives the marked packet it will increase its ATIM window.

Since in DPSM, nodes dynamically adjust their ATIM window size, it is possible for a node to have a different ATIM window size from the other nodes. For example, in Figure 2, node A has a larger ATIM window than node B. In the first ATIM window, let us assume that, due to contention with ATIM frames from other nodes, node A is not able to deliver its ATIM frame until after B’s ATIM window has ended. Eventually, node A does manage to send ATIM frame to B within three times. However, by then, node B is already in doze state, so B cannot reply with ATIM-ACK. In the next beacon interval, node A is able to deliver the ATIM frame within B’s current ATIM window. As described earlier, when node A cannot successfully transmit the ATIM frame in the first beacon interval, A will ”mark” and rebuffer the packet. In the second beacon interval in Figure 2, when node B receives the marked packet, as per our “ATIM window increase rules” described later, node B will increase its ATIM window size from the subsequent beacon interval.

- Each node piggybacks its own ATIM window size on all transmitted packets. Thus, each node may be aware of the ATIM window size used by some or all the other nodes. A node that has a small ATIM window may enter doze state earlier than a node that has a larger ATIM window. When transmitting the ATIM frame packets, each node gives preference to destination nodes whose ATIM window size is known to be small – when the ATIM window size for a destination node is not known, it is assumed to be equal to $ATIMmin$. Thus, the packets pending to be transmitted are sorted by the size of the ATIM window at their destinations. The ATIM frames are then transmitted in the order in which packets appear in the sorted order, with the provision that for any given destination, at most one ATIM frame is transmitted within a given beacon interval (recall that the packet announces the number of pending packets for the destination).

The mechanism for implementing the above scheme consists of several queues, one queue corresponding to each allowed level of the ATIM window, the smallest value of the ATIM window being $ATIMmin$. Now recall that a packet is be rebuffered on failure to transmit corresponding ATIM frame within a given beacon interval (either because the ATIM window ended, or because ATIM-ACK is not receieved despite three transmissions of the ATIM frame). In such cases, the packet is rebuffered in the queue corresponding to ATIM window size $ATIMmin$, to give a higher transmission priority to such packets.

Now, we describe the rules for dynamic adjustment of the ATIM window size. Initially, each node begins with ATIM window size equal to *ATIMmin* – in our simulations, *ATIMmin* is chosen to be 2 ms. A finite set of ATIM window sizes is specified; a node may use any ATIM window size from this set of values. We present the rules for increasing the ATIM window size, followed by the rule for decreasing the ATIM window size.

- There are four rules for increasing the ATIM window size, as listed below.

1. *Based on the number of pending packets that could not be announced during the ATIM window.*

After the ATIM window expires, a node checks its buffer to see if the current window was big enough to announce one ATIM frame to all destinations which have pending packets. If the number of pending packets that could not be announced is too large, it means that using the current ATIM window size is small to transmit an ATIM frame to all destinations that have pending packets. Thus, the node will increase its ATIM window by one level. In our simulation, if a node has more than 10 pending packets that could not be announced, it will increase its ATIM window size.

2. *Based on overheard information.*

Each node piggybacks its ATIM window size on all transmitted packets. If a node, say node A, overhears an ATIM window size that is at least two levels larger than that being used by node A, then node A increases its own ATIM window size by one level. We use this procedure, as opposed to simply copying the overheard ATIM window size when larger, to allow the ATIM window size reduction heuristic (described later) to work as desired.

3. *Receiving an ATIM frame after ATIM window.*

Suppose a node has a small ATIM window. For example, in Figure 3, the ATIM window size used by node B is smaller than that used by node C. In this case, because node B receives ATIM frame from node A, node B stays awake for the remaining beacon interval waiting for a data packet from node B. Now, after node B’s ATIM window has been completed, but node C’s window is not yet completed, node C sends an ATIM frame to node B. Thus, node B receives the ATIM frame after its own ATIM window has been concluded (node B is still awake expecting a data packet from node A). In such a situation, we allow node B to increase its ATIM window size by one level.

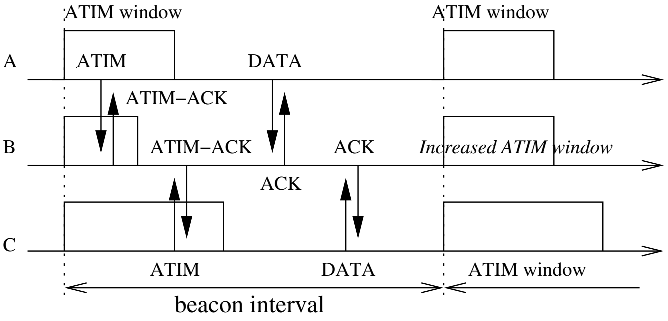


Fig. 3. Increasing ATIM window in DPSM. B receives ATIM frame from A and stays awake. After its ATIM window expires, B receives an ATIM frame from C. In this case, B will increase its ATIM window by one level from the next beacon interval.

4. *Receiving a marked packet.*

Earlier we described how a packet may be marked. When a node receives a marked packet, the node will increase its ATIM window size to the next higher level. Marked packet indicates failure to deliver the corresponding ATIM frame in a small number of attempts – this might indicate that the recipient node may be using too small ATIM window. For our earlier example in Figure 2, node A has a larger ATIM window than node B. When node A sends ATIM frame to B to announce a packet, B is already in doze state, so node

B cannot reply with ATIM-ACK.

If any of the above rules are satisfied, a node will increase its ATIM window size to the next higher value at the beginning of the next beacon interval. In our simulations, we allowed ATIM window values between 2 ms and 26 ms, in increments of 2 ms. Once the ATIM window size reaches 26 ms, it is not allowed to increase further. The proposed scheme may be modified to selectively turn off the power saving mode if a very large ATIM window size is required, indicating a high level of ATIM frame traffic – in such cases, instead of making the ATIM window larger, it might be beneficial to simply not use the power saving mode, and stay awake all the time. We have not evaluated this approach in the current paper, and impose an upper bound of 26 ms on the ATIM window size.

- Rule for decreasing ATIM window size: During an ATIM window, if a node has successfully announced one ATIM frame to all destinations that have pending packets and no window increasing rule defined above is satisfied, it means that the current ATIM window size was big enough. In such cases, the node will be decreasing its ATIM window size by one level (by 2 ms in our simulations). Recall that we also maintain a minimum ATIM window size so that when the ATIM window size reaches *ATIMmin*, it is not decreased any further. In our simulations, $ATIMmin$ is 2 ms.

V. PERFORMANCE EVALUATION

We simulated the proposed DPSM scheme, the PSM scheme in IEEE 802.11, and also 802.11 without any nodes in the power saving mode – we refer to the last one as WithOut Power Saving Mechanism (WOPSM). We simulated a LAN environment wherein all nodes are in each other's transmission range. We use two metrics to evaluate the proposed scheme.

1. Aggregate throughput over all flows in the network: One of our goals is to design an energy conserving MAC protocol that improves energy consumption without degrading throughput. Therefore, this metric is useful to measure if any throughput degradation occurs when using the proposed scheme.

2. Aggregate throughput per unit of energy consumption (or, throughput per joule): This metric measures the amount of data delivered per joule of energy. It is obtained by dividing the aggregate throughput over all flows by total energy consumption over all nodes in the network. Total energy consumption is the sum of each node's energy consumption during the simulation time. We use throughput per joule as a measurement of energy consumption – the greater the value of thruoghput per joule, the lower the energy consumption.

We do not use total energy consumption over a simulation duration as a metric, because the amount of useful "work" done (i.e., throughput achieved) by different schemes over a given amount of time may be different. For instance, some scheme may consume very little energy, but also achieve very little throughput. Thus, it is not fair to compare energy consumption of different schemes over a fixed time interval. Therefore, we use the throughput per joule metric.

A. Simulation Model

For our simulation, we used ns-2 with the CMU wireless extensions [24]. Each simulation was performed for a duration of 25 seconds in a wireless LAN environment, with the number of nodes chosen to be 8, 16, 32 or 64. In each case, half of the nodes are source nodes and the rest are destinations, for the simulated flows. For example, in 8 nodes senario, 4 nodes send packets to the other 4 nodes. Each flow transmits a constant-bit rate traffic, the rate of traffic being varied in different simulations. Channel bit rate is 2 Mbps and the packet size is fixed at 512 bytes.

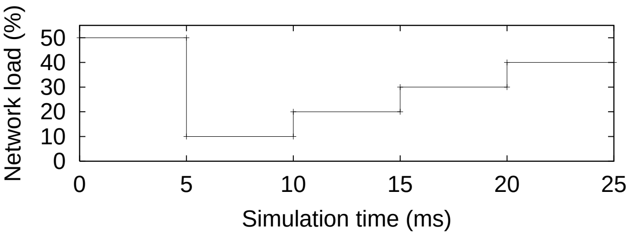


Fig. 4. In our dynamic network load senario, source nodes change their network load as the simulation time changes.

We varied the total network load to observe the effect of the network loads on throughput and energy consumption. Simulated network loads are 5%, 10%, 20%, 30%, 40%, and 50%, measured as a fraction of the channel bit rate of 2 Mbps. For instance, at network load of 10%, the total bit rate of all the traffic sources is $0.1 \times 2 = 0.2$ Mbps. Each traffic source has the same bit rate. Thus, with total load of 10%, and with, say 4 traffic sources, each traffic source has the rate of 0.05 Mbps.

We also simulated each scheme with dynamic network load. As Figure 4 shows, the source nodes start with a network load of 50% and change network load from 50% to 10% at 5 seconds. The nodes then change from 10% to 20%, from 20% to 30%, and from 30% to 40% at 5 second intervals.

For the enregy model, we use 1.65W, 1.4W, 1.15W, and

0.045W as values of power, consumed by the wireless network interface in transmit, receive, and idle modes and doze state, respectively. We use 800μs for a doze to awake transition time [15] [23], which is a much more conservative estimate than 250μs in [14]. During this transition time, a node will consume twice the power than idle mode.

The initial energy for each node was 1000 joules – no node runs out of its energy during our simulations. All the simulation results are averages over 30 runs. We use 100 ms for each beacon interval, which is the value specified for PSM in [13]. We simulated PSM, DPSM, and WOPSM. For PSM simulation, we varied the ATIM window size from 2ms to 50ms.

B. Simulation Results

Although the purpose of the power saving technique is to save as much energy as possible, we would like it to not degrade throughput. Thus, the goal is to use an ATIM window size that leads to a high energy saving without degrading throughput.

When a node is in doze state, it can save 1.105 joule of energy per 1 second of dozing time, since the power consumption difference between idle mode and doze state is 1.105W (1.15W - 0.045W). However, there is also energy loss due to the overhead of sending beacons, ATIM, and ATIM-ACK frames. Whenever a node transmits these packets, it will consume the power of transmit mode, and the neighbor nodes will consume the power of receive mode. Therefore, we have to make sure that the energy gain from the doze state is larger than the energy loss.

B.1 Fixed Network Load

This section presents simulation results in the case where the total network load is constant throughout the simulation. Later, we will present results for a case where the network load changes with time.

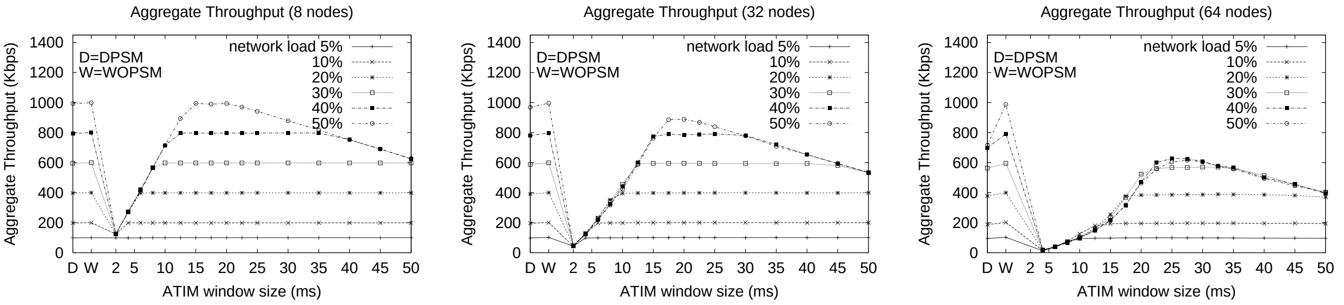
Figure 5 shows the aggregate throughput (aggregated over all flows) with fixed network load when using PSM, DPSM, and WOPSM schemes. In these figures, letter D on the horizontal axis corresponds to the proposed DPSM scheme, and W corresponds to the WOPSM scheme. The performance of the PSM scheme in 802.11 depends on the chosen ATIM value. The figure plots the throughput of PSM as a function of the ATIM value, the ATIM values being varied between 2 ms and 26 ms, as plotted on the horizontal axis. The results in Figure 5(a), (b) and (c) are for different number of nodes on the LAN.

As the figures show, ATIM window size affects throughput achieved with PSM quite significantly. On the other hand, proposed DPSM typically yields throughput comparable to WOPSM. Also, DPSM typically performs comparable or better than PSM, even if we use the optimal value of ATIM window size for PSM. Also, DPSM typically conserves more energy than PSM as we will discuss later in Figure 6.

At low load in Figure 5, throughput with PSM is less sensitive to the ATIM window size. However, as load is increased, the choice of ATIM window size can significantly affect the PSM throughput. If the ATIM window is too small, there is not enough time to announce all the pending packets, resulting in throughput degradation. If the ATIM window is too large, there is less time for actual data transmission, resulting in throughput degradation as well.

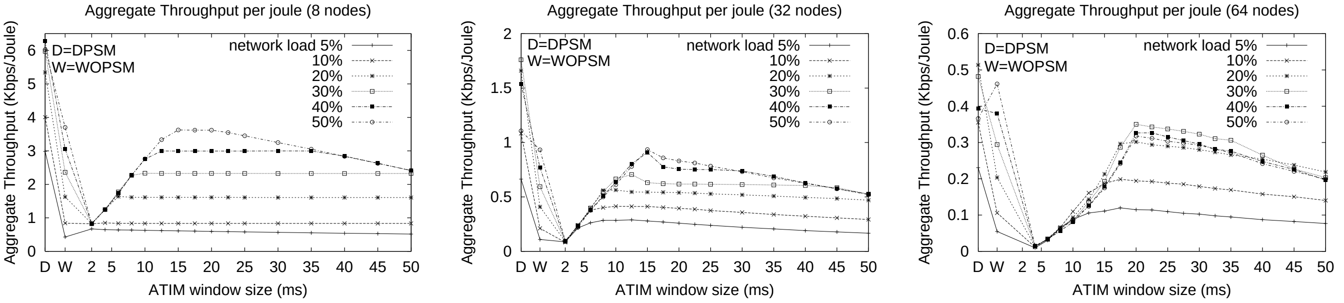
For example, in Figure 5 (a), 2 ms ATIM window is enough to acheive the desirable throughput for 5% of network load. However, with 50% of network load in Figure 5 (c), ATIM window of about 27.5 ms gives the best throughput. Moreover, although 27.5 ms of ATIM window size gives the best throughput here, throughput in PSM is significantly less compared to WOPSM. As the number of nodes increase or as the network load gets heavier, ATIM window size becomes a significant factor for both throughput and the energy consumption in PSM. Aggregate throughput is also degraded in DPSM with 50% of network load in 64 nodes in Figure 5 (c). This is because the highly loaded network needs more time for data transmission, but both PSM and DPSM use extra channel capacity for ATIM window. However, DPSM always performs comparable or better than PSM. As explained earlier, in DPSM, each packet includes the number of pending packets for the destination. Therefore, both source and destination nodes know how many pending packets at the source node. When the network load is highly loaded, there may be situation where the source node has not transmitted all the pending packets for the destination during the current beacon interval. In this situation, in DPSM, both source and destination nodes will stay awake during the next beacon interval without sending any further ATIM frame to the same destination. This allows DPSM performs better than PSM in Figure 5 (c).

Figure 6 shows the aggregate throughput per joule. DPSM always performs better than PSM. In particular, in Figure 6 (a), DPSM conserves 3 to 4 times more energy than PSM or WOPSM. For instance, in case of 10% network load, DPSM achieves 4 kbps/joule, while both PSM and WOPSM achieve 1 kbps/joule. Since with WOPSM, all nodes are always awake, the energy consumption is higher than PSM and DPSM. Although PSM allows a node to be in power saving mode, the node cannot enter doze state if it has at least one packet to transmit or receive. This is a disadvantage when the network load is low. In



(a) Aggregate Throughput (8 nodes) (b) Aggregate Throughput (32 nodes) (c) Aggregate Throughput (64 nodes)

Fig. 5. Aggregate Throughput in PSM, DPSM, and WOPSM (Fixed network load).



(a) Aggregate Throughput per joule (8 nodes) (b) Aggregate Throughput per joule (32 nodes) (c) Aggregate Throughput per joule (64 nodes)

Fig. 6. Aggregate Throughput per joule in PSM, DPSM, and WOPSM (Fixed network load).

DPSM, a node can enter doze state whenever it finishes the transmission. This allows a node to be in doze state longer than PSM, leading to reduced energy consumption as compared to PSM.

Energy gain from power saving mode becomes smaller when the network load gets higher or the number of nodes increases. For instance, throughput per joule in 32 node network (Figure 6 (b)) and 64 node network (Figure 6 (c)) are lesser than 8 node network (Figure 6 (a)). Note that the scale for the vertical axis in Figures 6(a), (b) and (c) is not identical. When the network load is high, there is less time for a node to be in doze state because of data transmission. Also, when the number of nodes in the network is large, there can be more collisions among the nodes. Therefore, a node will take more time to finish the data transmission, hence, it is more likely to stay in awake state. Lesser duration in doze state will yield lesser energy conservation.

In Figure 6 (c), PSM performs similar or worse than WOPSM. When the network is highly loaded, having power saving mode itself does not help the energy conservation – recall there are extra channel usage for ATIM window and energy loss for beacons, ATIM, and ATIM-ACK transmissions. However, DPSM always performs comparable or better than PSM due to the similar reasons explained in Figure 5 (c).

Now we discuss how a node using DPSM adapts ATIM window size. The first graph in Figure 7 shows the ATIM window changes at one of the source nodes during the simulation time using 16 node sinario with 10% constant network load. The second graph in Figure 7 shows the number of packets the node transmitted during a beacon interval beginning at the time shown on horizontal axis. The figure shows how a node using DPSM adjusts its ATIM window size during the simulation time and its impact on the number of packets the node transmitted. The node will increase its ATIM window when the increasing ATIM window rule is satisfied. The number of pending packets that could not be announced is the main factor for a node to increase its ATIM window. From Figure 7, whenever there is a peak for ATIM window increment, more packets are transmitted. When a node transmits an ATIM frame and does not get ATIM-ACK from the destination node, there can be two possible reasons. One is the network congestion during the ATIM window and the other is the des-

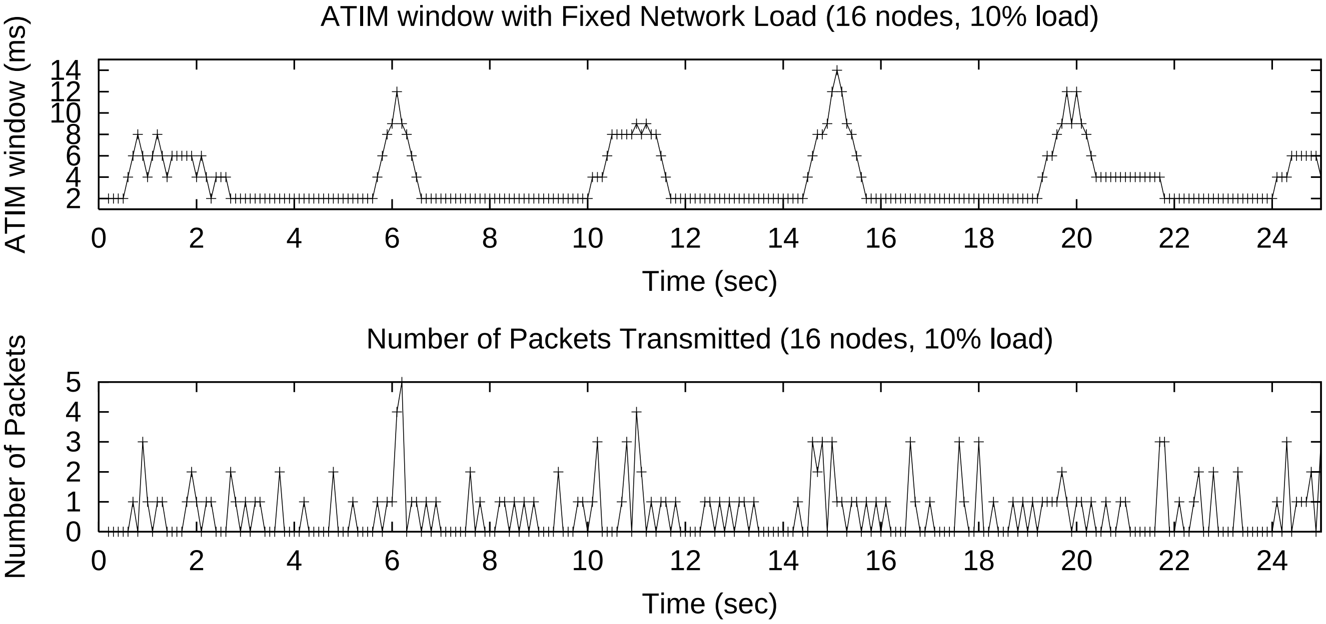


Fig. 7. DPSM allows a node to adjust its ATIM window according to the network load. Increasing ATIM window size gives the node more time to successfully transmit ATIM frames. More packets are transmitted after increasing ATIM window size.

tination in doze state. While the node cannot successfully transmit ATIM frames, the number of pending packets that could not be announced may grow. When the number of pending packets that could not be announced is too large, the node will decide to increase its ATIM window, assuming that the current ATIM window was not big enough. By increasing the ATIM window size, the node will have more time to transmit ATIM frames. Whenever the node receives ATIM-ACK, DPSM allows the node to transmit all the pending packets to the same destination without sending further ATIM frame. In Figure 7, the number of packets transmitted increases just after the ATIM window is increased. This indicates that DPSM fairly adjusts its ATIM window size according to the network load.

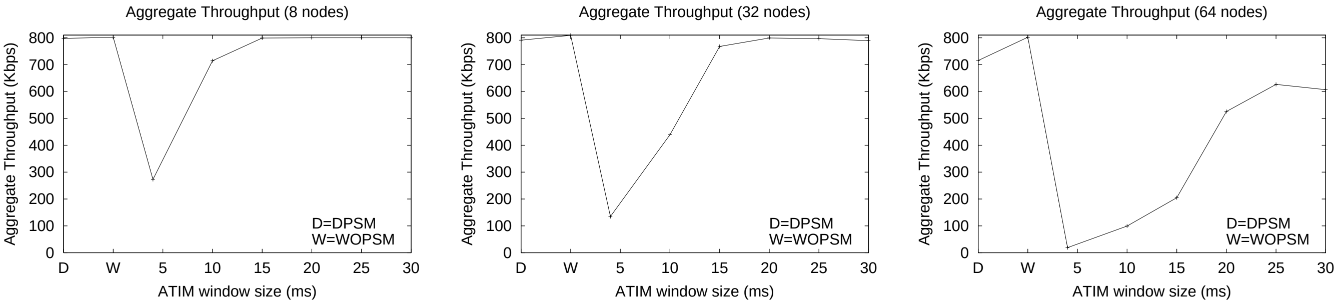
B.2 Dynamic Network Load

Section B.1 presented results for the case when aggregate load by the traffic flows was held constant. Now we present results for a case when the load is time-varying. The time-varying load used in these simulations is presented in Figure 4, as discussed previously. Figure 8 plots the aggregate throughput for the DPSM scheme (label D on horizontal axis), WOPSM scheme (label W on horizontal axis), and the PSM scheme (with different values of fixed ATIM window size for PSM). Figure 9 shows the corresponding throughput per joule. As discussed earlier, since ATIM window size significantly affects on both throughput and energy consumption, PSM does not perform well all the time. ATIM window size should be dynamically adjusted according to the network load. The simulation using dynamic network load shows that DPSM always performs well compared to PSM.

In Figure 8 (a), 15 ms of ATIM window in PSM gives comparable throughput as DPSM and WOPSM. However, the corresponding throughput per joule is significantly lower than DPSM in Figure 9 (a). When the number of nodes in the network increases, there may be more contention in order to transmit data. Thus, PSM requires a larger ATIM window size, as shown in Figure 8 (c). Also, PSM gives throughput per joule similar to WOPSM in Figure 9 (c) with the similar reason, explained in Figure 5 and Figure 6 – that is, extra channel capacity for ATIM window and extra energy overhead. DPSM performs better than PSM with dynamic network load in Figure 8 and Figure 9.

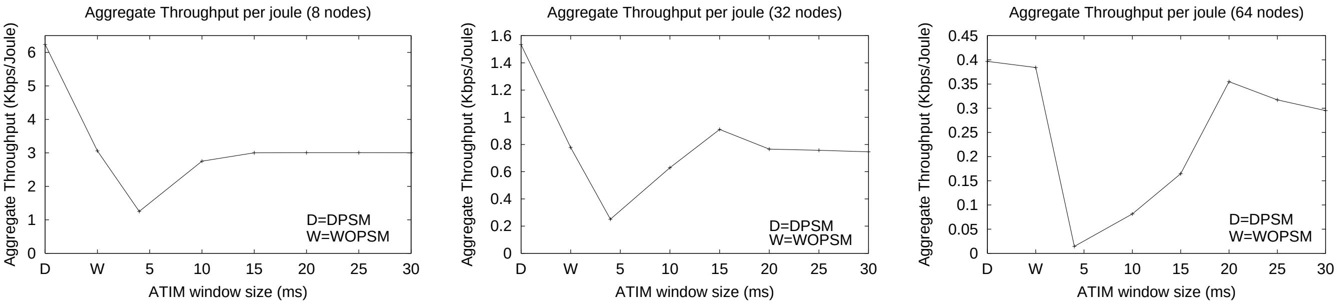
VI. CONCLUSIONS

In this paper, we presented an energy efficient MAC protocol, DPSM. The ATIM window size in PSM significantly affects the throughput and the amount of energy saving. As the network load gets heavier, the desirable ATIM window size becomes larger. Thus, a fixed ATIM window cannot perform very well all the time. In PSM, if the ATIM window is too small, the throughput degrades as the network load becomes heavier. If the ATIM window is too large, the energy gain from power saving mode becomes small,



(a) Aggregate Throughput (8 nodes) (b) Aggregate Throughput (32 nodes) (c) Aggregate Throughput (64 nodes)

Fig. 8. Aggregate Throughput in PSM, DPSM, and WOPSM (Dynamic network load).



(a) Aggregate Throughput per joule (8 nodes) (b) Aggregate Throughput per joule (32 nodes) (c) Aggregate Throughput per joule (64 nodes)

Fig. 9. Aggregate Throughput per joule in PSM, DPSM, and WOPSM (Dynamic network load).

since each node must stay awake during the ATIM window.

DPSM is based on PSM, but a node can dynamically adapt its ATIM window size according to observed network conditions. In DPSM, a node also can power off its wireless network interface whenever it finishes packet transmission for the announced packets. Simulation results show that the proposed scheme can improve energy consumption without degrading throughput.

REFERENCES

[1] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing," in *MOBICOM 2001*, jul 2001, http://www.isi.edu/scadds/publications.html.

[2] Jae-Hwan Chang and Leandros Tassiulas, "Energy conserving routing in wireless ad-hoc network," in *INFOCOM 2000*, mar 2000.

[3] Javier Gomez, Andrew T. Campbell, Mahmoud Naghshineh, and Chatschik Bisdikian, "PARO: Power-aware routing in wireless packet networks," in *Proc. 6th IEEE International Workshop on Mobile Multimedia Communications (MoMuC99)*, nov 1999.

[4] Laura M. Feeney, "An energy-consumtion model for performance analysis of routing protocols for mobile ad hoc networks," *Baltzer/ACM Journal of Mobile Networks and Applications (MONET) Special Issue on "QoS in Heterogeneous Wireless Networks"*, jun 2001.

[5] Suresh Singh and C. S. Raghavendra, "PAMAS - power aware multi-access protocol with signalling for ad hoc networks," in *MOBICOM '98*, oct 1998.

[6] A. Chockalingam and Michele Zorzi, "Energy efficiency of media access protocols for mobile data networks," *IEEE Transactions on Communications*, vol. 46, no. 11, nov 1998.

[7] Krishna M. Sivalingam, Mani B. Srivastava, and Prathima Agrawal, "Low power link and access protocols for wireless multimedia networks," in *VTC '97*, may 1997.

[8] Jeffrey P. Monks, Vaduvur Bharghavan, and Wen mei W. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *INFOCOM 2001*, apr 2001.

[9] Laura M. Feeney and Martin Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *INFOCOM 2001*, apr 2001.

[10] Sandeep Agrawal and Suresh Singh, "An experimental study of TCP's energy consumption over a wireless link," in *4th European Personal Mobile Communications Conference*, feb 2001.

[11] Michele Zorzi and Ramesh R. Rao, "Is TCP energy efficient?," in *Proc. IEEE MoMuC*, nov 1999.

[12] Robin Kravets and P. Krishnan, "Application-driven power management for mobile communication," *ACM/URSI/Baltzer Wireless Networks (WINET)*, 1999.

[13] The editors of IEEE 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, 1997.

[14] Ad Kamerman and Leo Monteban, “WaveLAN-II: A high-performance wireless LAN for the unilicensed band,” *Bell Labs Technical Journal*, vol. 2, no. 3, summer 1997, http://www.lucent.com/minds/techjournal/summer_97/pdf/paper08.pdf.

[15] Paul J. M. Havinga and Gerard J. M. Smit, “Energy-efficient TDMA medium access control protocol scheduling,” in *Proc. Asian International Mobile Computing Conference (AMOC 2000)*, nov 2000.

[16] *IEEE802.11 WaveLAN PC Card - User’s Guide, p.A-1.*

[17] Mark Stemm and Randy H. Katz, “Measuring and reducing energy consumption of network interfaces in hand-held devices,” *IEICE Transactions on Communications, special Issue on Mobile Computing*, vol. E80-B, no. 8, pp. 1125–31, 1997.

[18] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” in *MOBICOM 2001*, jul 2001.

[19] Wlodzimierz Lewandowski, Jacques Azoubib, and William J. Klepczynski, “GPS: primary tool for time transfer,” *Proc. of the IEEE*, vol. 87, no. 1, pp. 163–172, jan 1999.

[20] Hagen Woesner, Jean-Pierre Ebert, Morten Schlager, and Adam Wolisz, “Power-saving mechanisms in emerging standards for wireless LANs: The MAC level perspective,” *IEEE Personal Communications*, jun 1998.

[21] Juan Carlos Cano and Pietro Manzoni, “Evaluating the energy-consumption reduction in a MANET by dynamically switching-off network interfaces,” in *Proc. of the 6th IEEE Symposium on Computers and Communications*, jul 2001, http://www.disca.upv.es/pmanzoni/r/Papers/publications.html.

[22] Wendi R. Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *IEEE Proc. of the 33rd Hawaii International Conference on System Sciences*, jan 2000.

[23] Tajana Simunic, Haris Vikalo, Peter Glynn, and Giovanni De Micheli, “Energy efficient design of portable wireless systems,” in *International Symposium on Low Power Electronics and Design (ISLPED)*, jul 2000.

[24] The CMU Monarch Project., “The CMU monarch project’s wireless and mobility extensions to NS,” http://www.monarch.cs.cmu.edu/cmu-ns.html.