A Power Saving MAC Protocol for Wireless Networks

Eun-Sun Jung[1] and Nitin H. Vaidya[2]

[1] Dept. of Computer Science
Texas A&M University, College Station
Email: esjung@cs.tamu.edu

[2] Dept. of Electrical and Computer Engineering,
and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Email: nhv@crhc.uiuc.edu

Technical Report

July 2002

***Abstract*—This paper presents a protocol which improves on the power saving mechanism in the IEEE 802.11 Distributed Coordination Function (DCF). In the power saving mechanism (PSM) for DCF, all nodes are synchronized by beacons. In each beacon interval, there is a fixed time interval called the *ATIM window* where every node has to be awake. During the ATIM window, a source node informs a destination node about a pending packet by transmitting an ATIM frame. When the destination node receives an ATIM frame, it replies with an ATIM-ACK. Both the source and destination nodes stay awake for the remaining beacon interval. The source can transmit data after the ATIM window finishes. A node that does not have traffic to send or receive can enter the doze state after its ATIM window finishes.**

During the ATIM window in PSM, no data transmission is allowed. Thus, the available bandwidth in PSM is reduced according to the ATIM window size. Also, energy is consumed in transmitting and receiving ATIM and ATIM-ACK frames. This paper proposes a protocol that removes the overhead of the ATIM window and uses the bandwidth for data transmission. Simulation results show that removing the ATIM window gives better aggregate throughput and energy saving.

I. INTRODUCTION

BATTERY power is one of the critical resources in wireless networks. Due to limited battery power, various energy efficient protocols have been proposed to reduce energy consumption. Various approaches are proposed for different protocol layers, including work on battery management [1], [4], [13], power control [5], [11], [12], and energy-efficient protocols using directional antennas [16], [21].

Since the wireless network interface consumes a significant amount of energy, a large body of research has focused on reducing energy consumption. A power saving mode is often used to reduce energy consumption by putting the wireless network interface into a doze state.

In this paper, we propose a new power saving MAC protocol using the power saving mode. We consider two states for the wireless network interface. Specifically, a wireless network interface can be in either the *awake* or *doze* states. In the awake state, there are three different modes, *transmit*, *receive*, and *idle*, and each consumes a different amount of energy. In the doze state, the wireless network interface consumes much less energy as compared to the awake state. However, there exists transition delay and additional energy consumption when a node changes its state from doze to awake (or vice versa). For

This research is supported in part by National Science Foundation grants 99-76626 and 01-96410.

instance, [9] and [6], [14] report 250 μs and 800 μs for the transition time, respectively.

The IEEE 802.11 standard [19] specifies two medium access control protocols – PCF (Point Coordination Function) for a centralized protocol and DCF (Distributed Coordination Function) for a fully distributed protocol. Both protocols support a power saving mechanism (hereafter referred as PSM) which requires nodes in the network to be synchronized by periodic beacon transmissions. In this paper we only focus on PSM in DCF.

Fig. 1 illustrates the PSM in DCF. As the figure indicates, time is divided into beacon intervals in PSM. At the beginning of each beacon interval, there exists a specific time interval, called the ATIM window (Ad-hoc Traffic Indication Message window), where every node is awake. When a node has a packet to transmit, it first transmits an ATIM frame to the destination node during the ATIM window. When the destination node receives the ATIM frame, it replies with an ATIM-ACK. After the ATIM and ATIM-ACK handshake, both the source and the destination will stay awake for the remaining beacon interval to perform the data transmission. A node that has not transmitted or received an ATIM frame during the ATIM window may enter the doze state after finishing its ATIM window.

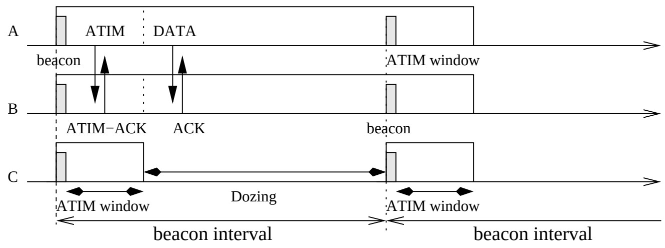


Fig. 1. Power saving mechanism (PSM) for DCF: Node A announces a buffered packet for B using an ATIM frame. Node B replies by sending an ATIM-ACK, and both A and B stay awake during the entire beacon interval. The actual data transmission from A to B is completed during the beacon interval. Since C does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

As shown in [8], [22], the performance of PSM is significantly affected by the size of the ATIM window. The optimal ATIM window size is closely related to the network load, the size of beacon interval, etc. As mentioned earlier, during the ATIM window all nodes are awake and only ATIM and ATIM-ACK frames can be transmitted. Real data transmission can only occur after the ATIM window. Overhead in energy consumption is incurred for transmitting or receiving additional ATIM and ATIM-ACK frames, and there is overhead in time due to the ATIM window. For example, if the beacon interval is 100 ms and the ATIM window size is 20 ms, 20% of bandwidth is wasted. In this paper, we propose a new power saving scheme which removes the ATIM window and uses the bandwidth for data transmission.

The rest of the paper is organized as follows. Section II reviews the related work. Section III presents our proposed protocols. Section IV describes our simulation model and discusses the simulation results. Section V concludes the paper.

II. RELATED WORK

Simulation results for the power saving mechanisms of two wireless LAN standards, IEEE 802.11 and HIPERLAN, are presented in [22]. It shows the sizes of a beacon interval and an ATIM window in IEEE 802.11 have a significant impact on throughput and energy consumption.

As shown in [22], a fixed ATIM window size in IEEE 802.11 cannot perform well in all situations. The optimal ATIM window size depends on various factors. A mechanism to choose the ATIM window size dynamically is proposed in [8]. As observed in [8], the power saving mechanism in IEEE 802.11 does not provide much energy savings because nodes have to stay awake for a whole beacon interval even if they have few packets to transmit. The protocol in [8] allows nodes to power off their network interface during the beacon interval whenever they finish announced packet transmissions, thus improving energy savings.

The synchronization of beacon intervals when using DCF can be difficult in multi-hop wireless networks. Some solutions are proposed in [20].

In PAMAS [15], each node uses two separate channels, one for control and the other for data packet transmissions. By using the control channel, a node determines when and for how long to power off the wireless network interface.

Similar to PAMAS, S-MAC [24] allows nodes to sleep during neighbors' transmissions. Nodes enter the doze state after hearing an RTS or CTS destined for a neighbor. S-MAC is designed for wireless sensor networks. To reduces contention latency, long messages are fragmented into many smaller fragments, then transmitted in bursts.

Span [3] utilizes the IEEE 802.11 power saving mechanism. Span elects coordinators, which periodically rotate their roles. The coordinators stay awake and forward traffic for active connections. Non-coordinators follow the power saving mechanism of IEEE 802.11 DCF. Nodes buffer the packets for dozing destinations and announce these packets during the ATIM window as in DCF. 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 enter the doze state if they do not have traffic to send or receive.

The protocol proposed in [25] also uses the power saving mechanism of IEEE 802.11. However, unlike IEEE 802.11, the protocol in [25] uses information from the network layer to reduce packet delivery latency. When a node receives routing packets, such as route request, route reply, etc., the node will stay awake for a predefined time duration, which is much longer than a beacon interval. Nodes involved in packet forwarding will be awake for a longer time, so that the end-to-end latency is reduced.

GAF [23] uses location information, provided by GPS, to form "virtual grids". All nodes in the same grid are equivalent in terms of traffic forwarding. GAF guarantees that one node in each grid stays awake in order to forward traffic.

Bluetooth [2] is designed for a low-cost and low-power wireless network. Bluetooth devices are usually organized into so called piconets, which consist of one master and up to 7 slave devices. Bluetooth provides three different low power modes (*sniff*, *hold* and *park*) to reduce energy consumption. Energy efficiency in Bluetooth is studied in [7], [26].

III. PROPOSED POWER SAVING MECHANISM

We now present the proposed power saving mechanism, referred to as NPSM (New PSM) hearafter. Since NPSM is similar to the IEEE 802.11 MAC protocol, we first describe how IEEE 802.11 works.

A. *IEEE 802.11 MAC Protocol*

The DCF in IEEE 802.11 uses an exchange of RTS (Ready to Send) and CTS (Clear to Send) packets between the sender and receiver prior to transmission of a data packet [1].

When a node S wants to transmit a packet to a node D it chooses a "backoff" counter uniformly distributed in the interval $[0, cw]$, where cw is size of the so-called *contention window*. cw at node S is reset to a value CW_{min} at the beginning of time, and also after each successful transmission of a data packet by S. Now, if the transmission medium is not idle, S waits until it becomes idle. Then, while the medium is idle, the backoff counter is decremented by 1 after each "slot" time [2]. Eventually, when the backoff counter reaches 0, S transmits an RTS packet for the intended destination D. When D receives the RTS, if D can communicate with S at the present time, D replies by sending a CTS to node S. Node S, on receipt of the CTS, sends DATA to D. After receiving DATA successfully, D sends an acknowledgment (ACK) to S. Now, it is possible that two nodes may choose their backoff counters such that they both transmit their RTS packets simultaneously, causing a collision between the RTS packets. In this case, node S will not receive a CTS. Absence of the CTS is taken as an indication of congestion, and node S doubles its contention window size cw, picks a new backoff counter uniformly distributed over $[0, cw]$, and repeats the above procedure.

IEEE 802.11 DCF also incorporates physical and "virtual" carrier sensing. The details are omitted here, however, note that the proposed protocol borrows the physical and virtual carrier sensing mechanism from 802.11 DCF.

B. *Time Synchronization*

The proposed NPSM uses periodic beacon transmissions to synchronize nodes in the network, as in IEEE 802.11 PSM. In PSM for IEEE 802.11 DCF, time is synchronized in a distributed manner. Each node maintains a local timer and transmits a beacon, which contains a timestamp of local timer. When a node receives a beacon from a neighbor and its beacon frame has not been transmitted, it cancels its beacon transmission. The node will update its local timer, if the timestamp of the received beacon frame is more recent than the value of its own timer. Reference [20] proposes other solutions to achieve synchronization in multi-hop networks, which could also be used with NPSM.

C. *Removing the ATIM window*

NPSM removes the ATIM window from IEEE 802.11 PSM in order to reduce control overhead. As mentioned earlier, time is divided into *beacon intervals* in NPSM. At the start of a beacon interval, every node enters an awake state for a specified duration called *DATA window*.

---

[1] For small data packet the RTS-CTS exchange may be omitted.

[2] Before counting down the backoff counter, a node waits for a duration of *DIFS* time.

The DATA window can be considered analogous to the ATIM window in PSM since every node is awake during the DATA window. However, the difference is that we do not transmit ATIM or ATIM-ACK frames during the DATA window. Instead, nodes transmit data packets during the DATA window without any ATIM or ATIM-ACK transmission. The purpose of the ATIM window in PSM is to announce pending packets to destination nodes. NPSM has a different way to achieve the same function. The basic idea is whenever a node transmits a packet to a destination it includes the number of pending packets in the packet. We describe the details next.

D. Announcing pending packets

In IEEE 802.11 PSM, the purpose of the ATIM window is to announce the existence of pending packets. To achieve the same goal, in NPSM, each node maintains counters to indicate the number of pending packets to transmit or receive.

The following counters are maintained by each node X:

- $T(i)$: the number of packets pending at this node (i.e., node X) for node i.
- $R(i)$: the number of packets destined for node X known (to X) to be pending at node i.
- R_{total}: sum of $R(i)$ over all neighbors i of node X. R_{total} is the total number of packets destined for node X known to be pending at all its neighbors.
- $Up(i)$: the number of packets that the neighbor node i needs to transmit or receive. Node X learns $Up(i)$ by overhearing packet transmission from node i, as described below.

The above counters are included in DATA, RTS, CTS, and ACK packets, as listed in Table I, and explained below.

TABLE I

THE COUNTER INCLUDED IN EACH PACKET

Packet type	Counters included in the packet
DATA	$T(destination)$ and R_{total}
RTS	$T(destination)$ + R_{total}
CTS, ACK	$T(source)$ + R_{total}

- *DATA:* When node i transmits a DATA packet to node j it includes $T(j)$ and R_{total}[3]. When node j receives the DATA packet from node i it updates $R(i)$. (Update to $R(i)$ also changes R_{total} at node j.)
- *RTS:* An RTS from node i to node j includes $T(j)$ + R_{total}.
- *CTS* and *ACK:* A CTS and an ACK from node j to node i includes $T(i)$ + R_{total}.

Whenever any neighbor node k overhears an RTS, CTS, DATA, or ACK packet from node i to node j, node k updates $Up(i)$ to $T(j)$ + R_{total} included in the packet. $Up(i)$ indicates the minimum data transmission that the node i will perform while staying awake. $Up(i)$ is reset to zero at the beginning of each beacon interval for all i. Node k knows that node i will stay awake as long as the counter, $Up(i)$ is greater than zero. A node uses $Up(i)$ to decide if it can enter the doze state after the DATA window finishes, as described next.

Note that instead of including the number of pending packets for a destination, $T(destination)$, an alternative is to include the list of destinations to whom the source has pending packets (and the amount of data pending to them). We have not evaluated this alternative yet. The approach evaluated here may introduce a short term unfairness since nodes that have not received any packet may enter the doze state. However, this is also the case in IEEE 802.11 PSM – if a node has not received an ATIM or ATIM-ACK, it will enter the doze state. In this paper, we do not include the list of destinations for all the pending packets. This is an issue for future work.

E. Transition to doze state

When the current DATA window expires, a node decides whether it should extend the DATA window (and stay awake longer) or go to the doze state. The node decides to extend the DATA window if any of the following conditions are satisfied.

- As explained earlier, a node maintains an estimate of the number of packets it needs to receive from its neighbors. If the node has not finished receiving all the packets (i.e., R_{total} is greater than zero) it will stay awake longer.
- In NPSM, a node can infer the neighbors' state (doze or awake) from overheard information. For example, node k can assume that node i will be awake for packet transmission if $Up(i)$ maintained by k is greater than zero. When node k has a packet to transmit to a neighbor i, and $Up(i)$ is greater than zero, k will remain awake and try to transmit packets to the destination.

---

[3] R_{total} in DATA packet is necessary only when the RTS-CTS handshake is omitted for small DATA packet.

In our simulation of NPSM, the beacon interval and the *initial* DATA window are set to 100 ms and 20 ms, respectively. The DATA window size is increased in increments of 5 ms. When the increased DATA window expires, the same process happens as when the initial DATA window expired. This process is repeated until the next beacon interval is started. Thus, if the network is highly loaded, it is possible the node does not enter the doze state at all.

Fig. 2 illustrates how NPSM works. Suppose node A has packets to send to node B. Node A transmits packets during the DATA window without using any ATIM frames. Since node A has not finished all packet transmissions after the original DATA window, both A and B will stay awake for 5 ms longer. Since node A includes the number of packets to transmit to node B within its first DATA packet, B knows it has not received all the packets from A. When the increased DATA window expires, and if all packet transmissions have finished, both A and B can go to the doze state, thus saving energy. Since node C does not have any packet to transmit or receive it enters the doze state when the initial DATA window expires.

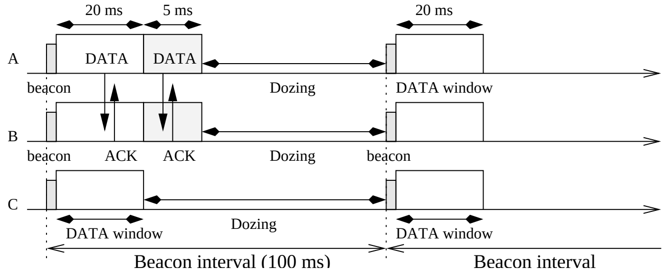


Fig. 2. NPSM does not have the ATIM window in order to reduce the overhead of transmitting extra control packets (ATIM and ATIM-ACK). This gives more bandwidth for data transmissions as compared to PSM.

IV. PERFORMANCE EVALUATION

We have simulated the proposed NPSM and the PSM in IEEE 802.11, as well as the IEEE 802.11 MAC without using power saving mode. Three metrics are used to evaluate the MAC protocols.

1) *Aggregate throughput over all flows in the network*
   Since the throughput may be degraded by using a power saving protocol, we measure the aggregate throughput.
2) *Total data delivered per unit of energy consumption (or, Kbits delivered per joule)*
   This is calculated as the total data delivered by all the flows divided by the total amount of energy consumption over all the nodes.
3) *Average packet delay over all flows in the network*
   Packet delay is calculated as the difference between the time when MAC layer receives a packet from the upper layer and the time when a node receives an ACK from the receiver for the packet. We calculate average delay over all flows in the network.

A. Simulation Model

We used ns-2 [18] for our simulations. Each simulation was performed for a duration of 20 seconds. The channel bit rate is 11 Mbps and the transmission range is 250 m. Different network sizes are simulated, as described later. In each scenario, a source node generates and transmits constant-bit rate traffic. The packet size of each flow is fixed at 512 bytes unless otherwise specified. (We also performed some simulations varying packet size.)

For the energy model, we assume the wireless network interface consumes 1.65 W, 1.4 W, 1.15 W, and 0.045 W in the transmit, receive, and idle modes and the doze state, respectively [10], [17].

We use 800 μs as the doze-to-awake transition time [6], [14], which is a more conservative than the 250 μs estimate in [9]. During this transition time, a node will consume twice as much power as the idle mode (i.e., 2.3 W).

The initial energy for each node is 1000 joules so nodes do not run out of energy during the simulations. All the simulation results are averages over 30 runs.

The beacon interval is set to 100 ms for both PSM and NPSM. The ATIM window size for PSM and the *initial* DATA window for NPSM are set to 20 ms (recall that the DATA window is changed dynamically). Simulations were performed in both a wireless LAN and a multi-hop network as described below.

1) Wireless LAN Scenario: Simulated network sizes are 20, 40, and 60 nodes for a wireless LAN. By a wireless LAN, we mean all nodes are within each other's transmission range. In each scenario for 20 and 40 node networks, half the nodes are source nodes and the other half are destination nodes. Thus, there are 10 flows in a 20 node network.

For a 60 node network, we simulate a scenario where one source transmits packets to multiple destinations. Twenty sources transmit packets to 40 other destination nodes – each source node transmits two flows.

We varied the total network load to observe the effect of network load on aggregate throughput and energy consumption. Simulated network loads are 10%, 20%, 30%, 40%, and 50%, measured as a fraction of the channel bit rate of 11 Mbps. For example, at a network load of 10%, the total bit rate of all traffic sources is $0.1 \times 11 = 1.1$ Mbps. Each traffic source has the same bit rate. Thus, with a total load of 10%, and 10 traffic sources, each traffic source has a rate of 0.11 Mbps.

2) Multi-hop Network Scenario: For a multi-hop network, 50 nodes are randomly placed in a 1000 x 1000 m^2 area. Ten source and ten destination nodes are randomly chosen. Note that a source or destination node can also be an intermediate node that forwards traffic for other nodes. The average route length of the flows is 4 hops with a range of 2 to 6 hops. Each traffic source generates a data rate of 10, 20, 30, 40, or 50 Kbps.

B. Simulation Results

We now present our simulation results. We show the simulation results for the wireless LAN case first, followed by the simulation results for a multi-hop network. The graphs in this paper show three curves labeled as 802.11, PSM and NPSM. The curve labeled as 802.11 corresponds to IEEE 802.11 DCF *without* using the power saving mode. The curve labeled as PSM indicates IEEE 802.11 DCF with PSM. The curve labeled as NPSM is for the scheme proposed in this paper.

B.1 Wireless LAN: varying the network load

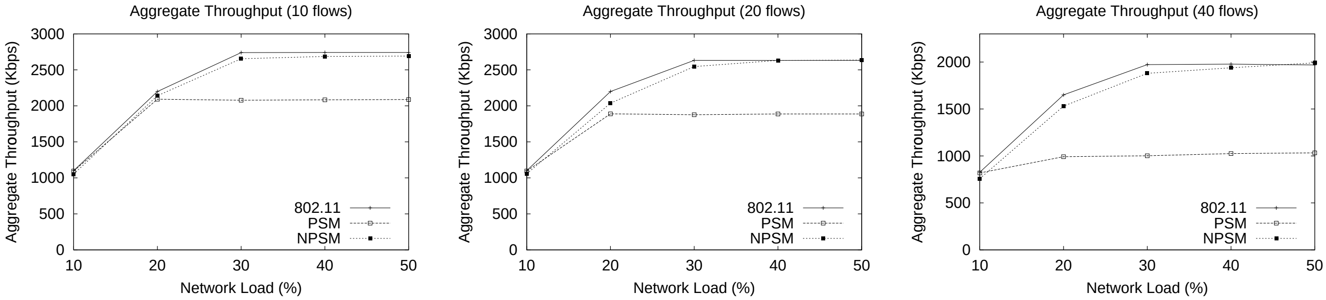
Fig. 3 shows the aggregate throughput (in Kbps) varying the network load using different schemes in a wireless LAN. When the network load is low, for instance at the network load of 10%, all schemes perform similarly for 20, 40, and 60 node networks in Fig. 3(a), (b), and (c), respectively. However, as the network load increases, the aggregate throughput of PSM degrades severely. This is mainly due to the overhead of the ATIM window. In PSM, 20% of the channel bandwidth is used for the ATIM window where only ATIM and ATIM-ACK transmissions are allowed. Therefore, there is less time for actual data transmission using PSM. The aggregate throughput of NPSM is also lower than IEEE 802.11 without PSM, but the degradation is not as significant as PSM. Since NPSM does not have the ATIM window, more bandwidth can be used for data transmissions. However, like PSM, NPSM also has extra beacon transmissions for synchronization. This is why the the aggregate throughput of NPSM is slightly lower than that of 802.11 in Fig. 3(a), (b), and (c). Note that the simulation results for the scenario with two flows per source node (Fig. 3(c)) are similar to those for one flow per source (Fig. 3(a) and (b)).

Fig. 4 shows the total data delivered per joule (Kbits/joule) for a wireless LAN with various network loads. NPSM performs the best among all schemes. Since we measure the total data delivered per joule, the poor aggregate throughput of PSM (see Fig. 3) results in lower total data delivered per joule. As observed in [8], the energy saving of PSM is poor when the network load is high. As mentioned earlier, PSM specifies that nodes have to be awake for the whole beacon interval even if they have few packets to transmit. This leads to less dozing time, resulting in less energy saving. Moreover, every node has to be awake during the ATIM window in PSM, transmitting extra ATIM and ATIM-ACK frames. This introduces extra energy consumption. In Fig. 4(a), PSM performs slightly better than 802.11. However, when the network load is high, PSM does not achieve energy savings as seen in Fig. 4(b) and (c) and also degrades the aggregate throughput as seen in Fig. 3(b) and (c). In NPSM, nodes do not use the ATIM window, and nodes can go to the doze state during a beacon interval if they do not have any traffic. Therefore, NPSM performs much better than PSM or 802.11 in Fig. 4(a), (b), and (c). Note that the simulation results for the scenario with two flows per source node (Fig. 4(c)) are similar to those of one flow per source (Fig. 4(a) and (b)). In Fig. 4(c), the data delivered per joule for PSM is slightly worse than IEEE 802.11 due to its poor throughput (see Fig. 3(c)).

Fig. 5(a), (b), and (c) show the average packet delay over all flows in a 20, 40, and 60 node network, respectively. IEEE 802.11 performs the best among all schemes because it does not use power saving mode. The delay for PSM is longer than that of 802.11, but shorter than that of NPSM. This is because with moderate and high loads, nodes are awake most of time (no dozing) in PSM. (Recall that in PSM, a node has to be awake for the whole beacon interval even if it has a few packets to transmit or receive.) NPSM gives the longest delay due to longer dozing time (yielding more energy saving, as seen in Fig. 4). As the network load increases, the average packet delay also increases in NPSM. When a destination node is in the doze state, packets at a source node have to stay in a buffer. This increases the packet delay in NPSM. Note that the average packet delay does not include the delay for lost packets. The simulation results for the scenario with two flows per source node (Fig. 5(c)) are similar to those of one flow per source scenarios (Fig. 5(a) and (b)).

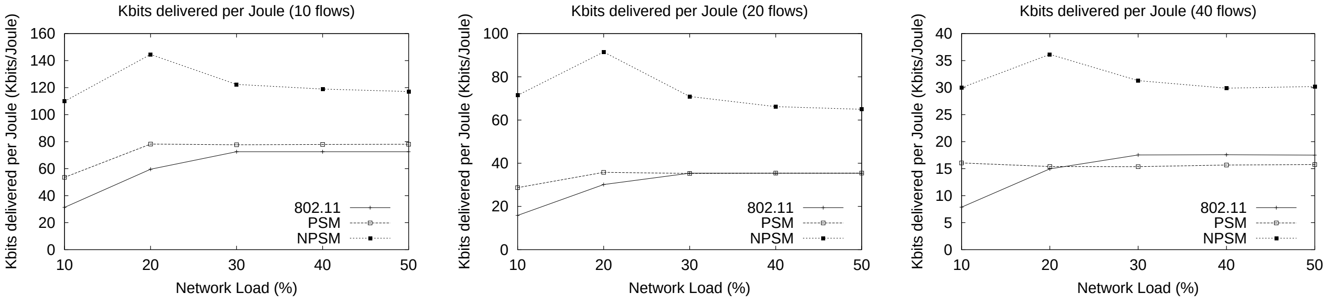
B.2 Wireless LAN: varying packet size

Fig. 6 and 7 show the aggregate throughput varying packet sizes in a wireless LAN with the network load of



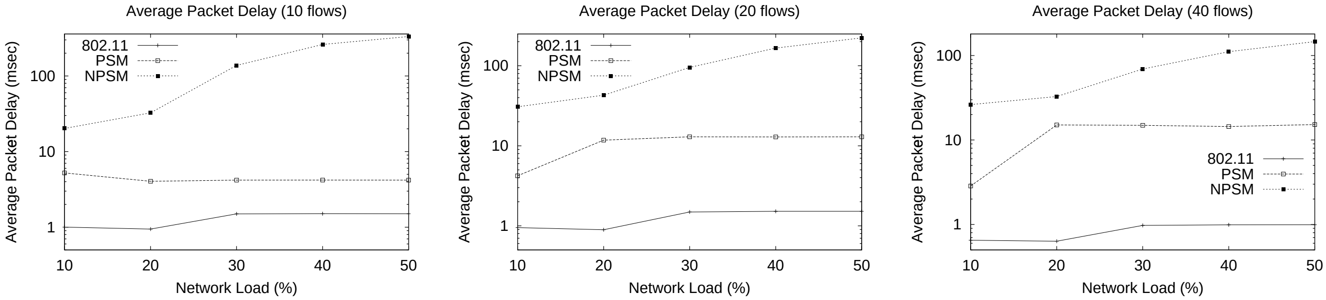
(a) 20 node network (b) 40 node network (c) 60 node network

Fig. 3. Aggregate throughput: wireless LAN with fixed packet size



(a) 20 node network (b) 40 node network (c) 60 node network

Fig. 4. Total data delivered per joule (Kbits/Joule): wireless LAN with fixed packet size



(a) 20 node network (b) 40 node network (c) 60 node network

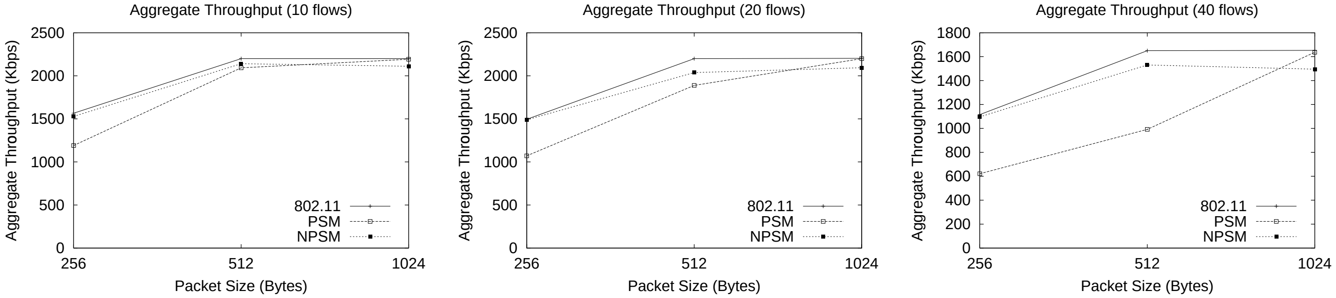
Fig. 5. Average packet delay: wireless LAN with fixed packet sizes (vertical axis uses log scale)

20% and 40%, respectively. The corresponding total data delivered per joule is shown in Fig. 8 and 9, respectively. Simulated packet sizes are 256, 512, and 1024 bytes.

Since the RTS/CTS overhead per packet is identical regardless of the packet size, as the packet size increases in Fig. 6 and 7, the aggregate throughput of all schemes also increases. The aggregate throughput of PSM is lower than 802.11 or NPSM, especially when the network load is higher (please refer Fig. 7). As we explained in Fig. 3, the overhead of the ATIM window in PSM results in lower aggregate throughput, and this does not change with the packet size.

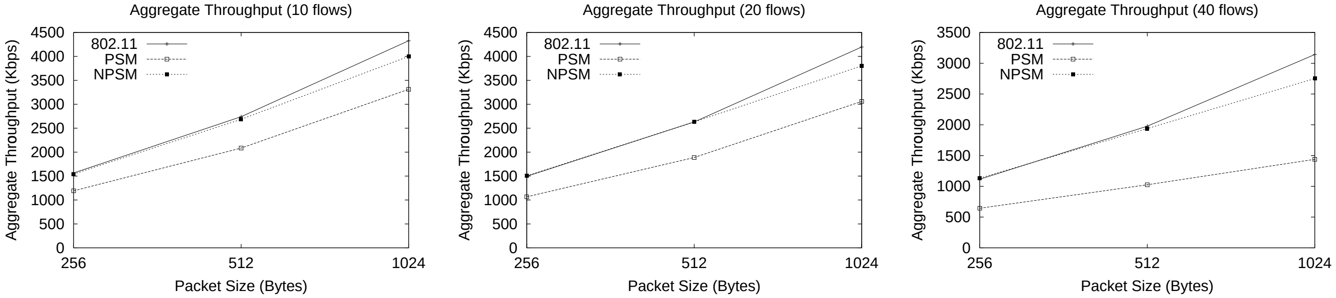
For NPSM with the packet size of 1024 bytes in Fig. 6, the aggregate throughput of NPSM is slightly lower than that of 802.11 or PSM. However, with a high network load in Fig. 7, NPSM performs much better than PSM due to the ATIM overhead of PSM.

In NPSM, as the packet size increases, the number of packets transmitted during the DATA window may be reduced because a large packet takes more time to be



(a) 20 node network (b) 40 node network (c) 60 node network

Fig. 6. Aggregate throughput: wireless LAN with different packet sizes (network load of 20%)



(a) 20 node network (b) 40 node network (c) 60 node network

Fig. 7. Aggregate throughput: wireless LAN with different packet sizes (network load of 40%)

transmitted. If the DATA window size is too small, the aggregate throughput of NPSM may degrade with a large packet size – specifically, if there is severe congestion during the DATA window, only a few nodes can transmit data and the other nodes will go to the doze state. However, this yields longer dozing period in NPSM. Therefore, when the data delivered per joule (Kbits/joule) is compared (see Fig. 8 and 9), NPSM performs the best among all schemes. Particularly, with the packet size of 1024 bytes, the gap between NPSM and 802.11 or PSM is much greater compared a packet size of 256 or 512 bytes. PSM performs better than 802.11 with a large packet size and a low network load as seen in Fig. 8. However, with a small packet size in Fig. 8 or with a high network load in Fig. 9, PSM does not save energy as compared to 802.11 because of the overhead of the RTS/CTS exchange as well as the overhead of the ATIM window (explained in Fig. 4). Note the curves for PSM and 802.11 in Fig. 9(b) are overlapped.

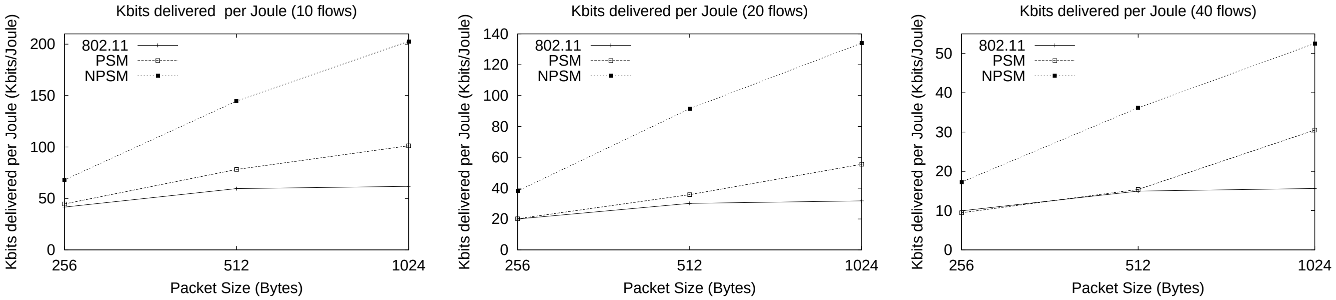
B.3 Multi-hop Network: varying the network load

We now present the simulation results for a multi-hop network. The simulation results for the multi-hop network are similar to those for the wireless LAN. However, since a packet travels four hops (on average) to reach a destination in our scenario, there is more energy consumption by intermediate nodes that forward packets to destination nodes.

Fig. 10 shows the simulation results for the multi-hop network. Similar to the simulation results in Fig. 3, all schemes perform comparably when the network load is low in Fig. 10(a). However, as the network load increases, the aggregate throughput with PSM and NPSM is lower than 802.11. However, NPSM performs better than PSM since it does not have the ATIM window overhead.

NPSM performs better than PSM and 802.11 with respect to the total data delivered per joule in Fig. 10(b). In contrast with the wireless LAN (Fig. 3), PSM generally performs better than 802.11 in a multi-hop network in Fig. 10(b). This is because in a multi-hop network, nodes that do not have any packet to forward can go to the doze state, resulting in energy savings.

Fig. 10(c) shows the average packet delay over all flows in the network. In Fig. 10(c), 802.11 without power saving mode performs the best among all schemes. As

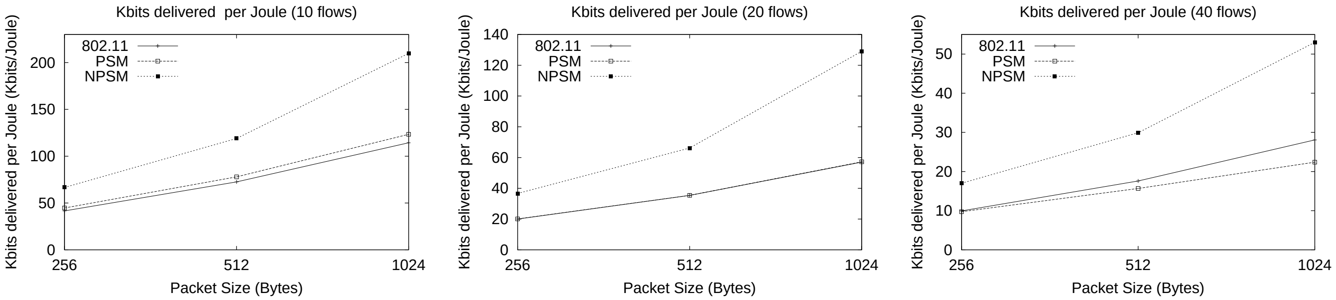


(a) 20 node network

(b) 40 node network

(c) 60 node network

Fig. 8. Total data delivered per joule: wireless LAN with different packet sizes (network load of 20%)

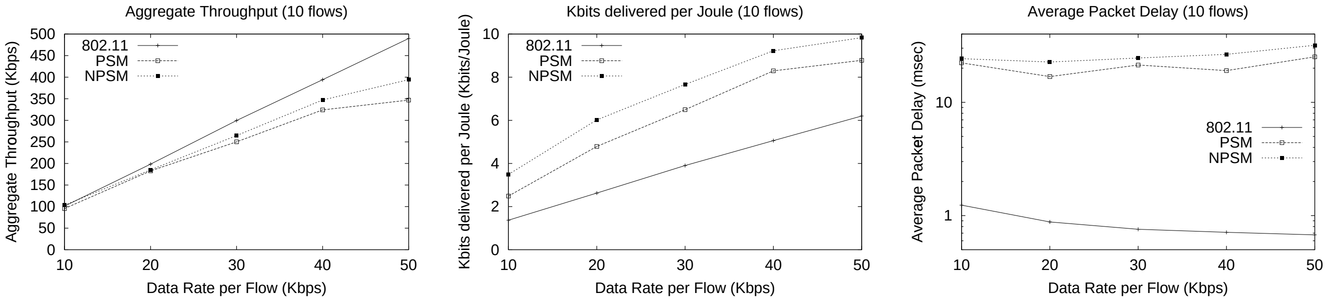


(a) 20 node network

(b) 40 node network

(c) 60 node network

Fig. 9. Total data delivered per joule: wireless LAN with different packet sizes (network load of 40%) – the curves for PSM and 802.11 are overlapped in (b).



(a) Aggregate throughput

(b) Total data delivered per joule (Kbits/Joule)

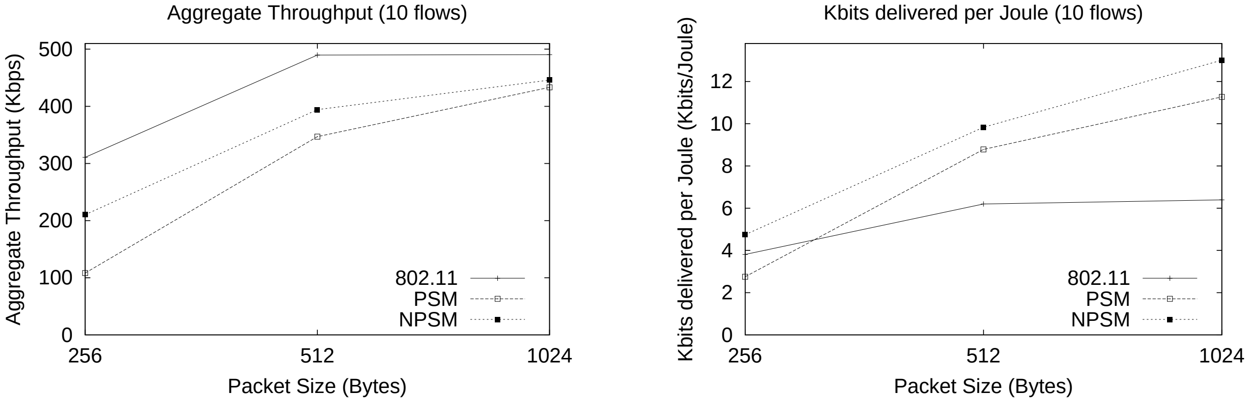
(c) Average packet delay

Fig. 10. Multi-hop network: 50 nodes with 10 flows (fixed packet size). Vertical axis uses log scale in (c).

the network load increases the delay for NPSM gets longer. In Fig. 10(c), the packet delay for PSM is slightly shorter than NPSM, but the aggregate throughput of PSM is worse than that of NPSM (see Fig. 10(a)); recall that lost packets are not considered for packet delay. There is a trade-off between energy savings and packet delay. NPSM gives a long dozing time (more energy savings, in Fig. 10(b)) with the cost of a longer packet delay.

B.4 Multi-hop Network: varying packet size

Fig. 11 shows the simulation result for a multi-hop network with various packet sizes. Each flow generates traffic at the rate of 50 Kbps. In Fig. 11(a), NPSM generally performs better than PSM. For the total data delivered per joule in Fig. 11(b), PSM with the packet size of 256 bytes performs worse than 802.11 due to the low aggregate throughput (Fig. 11(a)). NPSM performs the best for reasons explained in the wireless LAN case.



(a) Aggregate throughput (b) Total data delivered per joule (Kbits/Joule)

Fig. 11. Multi-hop network: 50 nodes with 10 flows (50 Kbps per flow, with different packet sizes)

V. CONCLUSION

We have presented a new power saving MAC protocol, NPSM. The aggregate throughput of IEEE 802.11 PSM degrades as the network load increases due to the time overhead of the ATIM window. Also, in PSM, extra energy is consumed by transmitting ATIM and ATIM-ACK frames during the ATIM window.

The NPSM removes the ATIM window overhead from PSM in IEEE 802.11 in order to increase channel capacity for data transmission and reduce the energy consumption. Removing the ATIM window makes more sense when the channel bandwidth is high, where a packet transmission occurs quickly. Simulation results confirm that NPSM gives better aggregate throughput and energy savings as compared to PSM. Since nodes can enter the doze state in NPSM, the average packet delay of NPSM is longer than IEEE 802.11 without using power saving mode. NPSM saves energy at the cost of increasing the packet delay.

REFERENCES

[1] M. Adamou and S. Sarkar, “A Framework for Optimal Battery Management for Wireless Nodes,” in *INFOCOM 2002*, June 2002.

[2] C. Bisdikian, “An Overview of the Bluetooth Wireless Technology,” *IEEE Communications Magazine*, vol. 39, pp. 86–94, December 2001.

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks,” in *MOBICOM 2001*, July 2001.

[4] C. F. Chiasserini and R. R. Rao, “Energy Efficient Battery Management,” in *INFOCOM 2000*, March 2000.

[5] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, “Conserving Transmission Power in Wireless Ad Hoc Networks,” in *ICNP'01*, November 2001.

[6] P. J. M. Havinga and G. J. M. Smit, “Energy-efficient TDMA Medium Access Control Protocol Scheduling,” in *Proc. Asian International Mobile Computing Conference (AMOC 2000)*, November 2000.

[7] P. Johansson, R. Kapoor, M. Kazantzidis, and M. Gerla, “Rendezvous Scheduling in Bluetooth Scatternets,” in *ICC 2002*, vol. 1, pp. 318–324, April 2002.

[8] E.-S. Jung and N. H. Vaidya, “An Energy Efficient MAC Protocol for Wireless LANs,” in *INFOCOM 2002*, June 2002.

[9] A. Kamerman and L. Monteban, “WaveLAN-II: A High-Performance Wireless LAN for the Unilicensed Band,” *Bell Labs Technical Journal*, vol. 2, summer 1997.

[10] Lucent, *IEEE802.11 WaveLAN PC Card - User's Guide, p.A-1*.

[11] J. P. Monks, V. Bharghavan, and W. mei W. Hwu, “A Power Controlled Multiple Access Protocol for Wireless Packet Networks,” in *INFOCOM 2001*, April 2001.

[12] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, “Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW protocol,” in *European Wireless 2002*, February 2002.

[13] P. Nuggehalli, V. Srinivasan, and R. R. Rao, “Delay Constrained Energy Efficient Transmission Strategies for Wireless Devices,” in *INFOCOM 2002*, June 2002.

[14] T. Simunic, H. Vikalo, P. Glynn, and G. D. Micheli, “Energy Efficient Design of Portable Wireless Systems,” in *International Symposium on Low Power Electronics and Design (ISLPED)*, July 2000.

[15] S. Singh, M. Woo, and C. S. Raghavendra, “Power-aware Routing in Mobile Ad Hoc Networks,” in *MOBICOM '98*, October 1998.

[16] A. Spyropoulos and C. Raghavendra, “Energy Efficient Communications in Ad Hoc Networks Using Directional Antennas,” in *INFOCOM 2002*, June 2002.

[17] M. Stemm and R. 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] The CMU Monarch Project., “The CMU Monarch Project's Wireless and Mobility Extensions to NS.”

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

[20] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, “Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks,” in *INFOCOM 2002*, June 2002.

[21] J. E. Wieselthier, G. Nguyen, and A. Ephremides, “Energy-Limited Wireless Networking with Directional Antennas: The Case of Session-Based Multicasting,” in *INFOCOM 2002*, June 2002.

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

[23] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed Energy Conservation for Ad Hoc Routing,” in *MOBICOM 2001*, July 2001.

[24] W. Ye, J. Heidemann, and D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in *INFOCOM 2002*, June 2002.

[25] R. Zheng and R. Kravets, “On-demand Power Management for Ad Hoc Network,” Tech. Rep. UIUCDCS-R-2002-2280, Department of Computer Science, University of Illinois at Urbana-Champaign, 2002.

[26] H. Zhu, G. Cao, G. Kesidis, and C. Das, “An Adaptive Power-Conserving Service Discipline for Bluetooth,” in *ICC 2002*, vol. 1, pp. 303–307, April 2002.