

SCPS: A self-configuring power-saving protocol for wireless ad hoc networks[☆]

Shih-Chang Huang, Rong-Hong Jan^{*}, Wu Yang

Department of Computer Science, National Chiao Tung University, Hsinchu 30050, Taiwan, ROC

Received 5 July 2007; received in revised form 29 November 2007; accepted 18 January 2008

Available online 1 February 2008

Responsible Editor: V.R. Syrotiuk

Abstract

SCPS is a novel self-configuring power-saving protocol for wireless one-hop ad hoc networks. According to IEEE 802.11 WLAN standard, a station may enter a special *power-saving* (PS) mode. SCPS allows all stations in the PS mode to adjust their wakeup schedules whenever a station enters or exits the PS mode. The adjustment can balance the number of wakeup stations in each beacon interval so that the contention for transmission medium and the collisions in transmission will be ameliorated, which results in more efficient energy usage. Simulation results show that SCPS successfully balances the number of stations that wake up in each beacon interval, increases the sleep ratio, and reduces the collision probability. The combined effect reduces total energy consumption.

© 2008 Elsevier B.V. All rights reserved.

Keywords: Wireless ad hoc networks; Power-saving schedule; Self-configuration; Medium access control

1. Introduction

Wireless communication becomes an indispensable capability for mobile devices, whose tasks include communicating with other devices. One of the main problems in the mobile devices is the severely limited energy supplied by the necessarily small battery. Due to the slow progress in the devel-

opment of high-capacity batteries, power efficiency is crucial to the operation of the mobile devices.

Many efforts have been targeted toward the reduction of power consumption in wireless communication. Some previous research [1–4] adjusts the transmission rate of a mobile device according to the traffic. By switching to lower rate, a mobile device consumes less energy in issuing the packets. In [5–7], power-control mechanisms that adjust the transmission power based on the distance of the destination are proposed to save energy. New energy-aware routing protocols are also designed that distribute the work of packet relay to multiple mobile devices [8–10]. Lee et al. proposed a power-saving

[☆] This work was supported in part by National Science Council, Taiwan, ROC, under Grants NSC 95-2219-E-009-008, NSC 95-2752-E-009-005-PAE, and NSC 96-3114-P-001-002-Y.

^{*} Corresponding author. Tel.: +886 3 5731637; fax: +886 3 5721490.

E-mail address: rhjan@cis.nctu.edu.tw (R.-H. Jan).

method by adjusting the listen-interval¹ according to the TCP traffic [11].

Though the IEEE 802.11 standard [12], which is a specification for wireless local area networks (WLANs), provides a power-saving mechanism, many researchers have proposed power-saving extensions to IEEE 802.11. Some proposed a variable-length beacon-interval mechanism [13]; others flexibly adapt the ATIM window to save the power wasted on the idly listening [14,15]. All of these proposals focus on extending the sleep duration (in order to conserve energy) within a beacon interval. In addition, reducing or eliminating packet collisions [17–19] is yet another way to save energy. Stations will save their energy from the packet re-transmission.

The above mechanisms consider saving the energy within a beacon interval. Stations still need to wake up in every beacon interval. To save more energy, we need the mechanism that stations can sleep more beacon intervals instead of one [16].

Thus, in this paper, we propose a self-configuration power-saving (SCPS) protocol for one-hop ad hoc networks. As the mechanism of [16], our proposed SCPS can let stations sleep more than one beacon intervals instead of waking up at every TBTT (target beacon transmission time). Furthermore, SCPS provides more flexibility for stations to select the length of listen-interval. Each station can decide its own listen-interval as in the infrastructure mode of 802.11. Another key contribution of SCPS is that it attempts to evenly arrange stations to wake up in beacon intervals in order to reduce the medium-contention probability. The arrangement of wakeup schedules is done by individual stations when other stations enter or exit the PS mode.

The rest of this paper is organized as follows. In Section 2, the power-saving mechanism of IEEE 802.11 and previous research are reviewed. SCPS is discussed in Sections 3. In Section 4, we present the simulation results of SCPS. Finally, Section 5 gives the conclusion and the future work.

2. Related work

2.1. IEEE 802.11 WLANs power-saving mechanism

IEEE 802.11 has two operation modes—infrastructure and ad hoc. In the infrastructure mode,

¹ A *listen-interval* consists of one or more beacon intervals. This will be defined later.

each station can send its power management state to the associated access point (AP). Because all traffic for a station must go through its AP, the AP can buffer the packets for a station when that station is in the sleeping state. On the other hand, in the ad hoc mode, stations communicate with one another directly if they are within direct communication range. Because there is no central coordinator in the ad hoc mode, each station has to implement a distributed buffer system and uses *announcement traffic indication messages* (ATIMs) to inform other sleeping stations to receive data.

Fig. 1 illustrates the power-saving mechanism in the 802.11 ad hoc mode. A beacon interval is divided into two parts, the ATIM window and the data transmission window. Every station has to wake up during the ATIM window. If a station has buffered data for another station, it sends an ATIM frame to notify the receiver. After successfully transmitting the ATIM frames, the stations compete for the transmission medium to send their buffered data during the data transmission window.

On the other hand, stations that have no data to send or receive go to sleep at the end of the ATIM window. It will not wake up until the next beacon interval. All other stations will wake up during the entire data transmission window. For example, in Fig. 1, all three stations (A, B, and C) wake up in the first ATIM window. They do not send or receive any ATIM frames so they sleep at the end of the ATIM window. All three stations wake up again in the second ATIM window. Assume that station A has buffered data for station B at this time. It sends an ATIM frame to station B during the second ATIM window and sends the buffered data during the second data transmission window. Because station C does not have data to send or receive, it goes to sleep at the end of the second ATIM window.

2.2. Enhancements of the power-saving operations in IEEE 802.11 WLANs

The fix-length beacon interval limits the bandwidth flexibility while bandwidth demands of stations change randomly. In [13], Liu et al. proposed a variable-length beacon-interval mechanism. The length of the transmission window was determined by all the stations that have succeeded in ATIM frame transfer. So the length of a beacon interval can be dynamically adjusted according to the demands of stations.

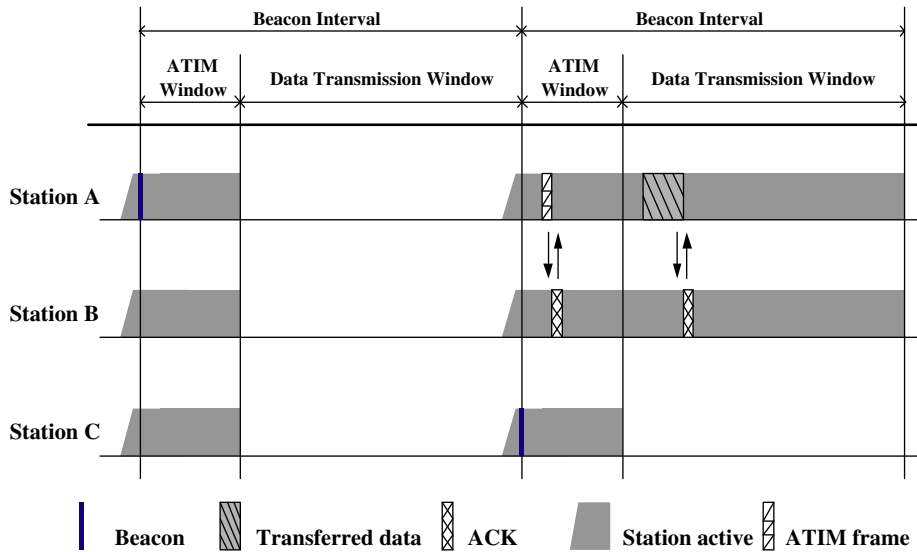


Fig. 1. The power-saving mechanism of IEEE 802.11 ad hoc mode.

Similar to the fix-length beacon interval, in the 802.11, the duration of the ATIM window is fixed and is determined when the system starts up. A *fixed-duration* ATIM window results in low bandwidth utilization. Flexibly adapting the ATIM window to reduce the power wasted on idly listening is proposed [14,15].

In [14], if a station discovers that the channel has been idle for more than a predefined amount of time, that station will assume that all other stations are idle. At this time, the ATIM window ends and the data transmission window starts. In [15], Jung and Vaidya proposed a dynamic ATIM window adaptation mechanism called dynamic power-saving mode(DPSM). DPSM dynamically adjusts the ATIM window based on the number of transmission requests. This improves the sleep time and bandwidth utilization. However, stations still have to compete for the medium for transmitting ATIM frames and data frames.

In order to soothe the contention, several distributed mechanisms have been proposed [17–19]. They suggest that stations first compete for transmitting ATIM frames. During the competition an order for transmitting data frames is established. The contention during the data transmission window is therefore avoided.

The above approaches only consider the transmission schedule within a beacon interval or adjusts the ATIM window to fit the traffic load. Stations still have to wake up in *every* ATIM window even if they have no data to send or receive. In order to

avoid waking up in every beacon interval, Chao et al. proposed the quorum-based energy conservation protocol QEC [16]. Before entering the power-saving mode, each station creates a $n \times n$ grid. Each entry in this grid denotes a beacon interval. Each station randomly chooses one row and one column in this grid as its wakeup intervals. Therefore, any two stations will share at least two entries. They can communicate with each other during the beacon intervals denoted by the shared entries.

3. The SCPS approach

Because the number of wakeup stations in each beacon interval is not balanced, the QEC approach cannot avoid high packet collision rate. In order to balance the number of wakeup stations in beacon intervals, SCPS allows every station not only to decide its own listen-interval as in the 802.11 infrastructure mode but also to choose a suitable wakeup schedule. Each station wakes up once per listen-interval, rather than per beacon interval. Thus, a station wakes up less frequently.

In summary, by dynamically adjusting the wakeup schedule, SCPS conserves energy in two ways:

1. Stations sleep longer.
2. The numbers of wakeup stations during beacon intervals are balanced. This reduces the collision (and hence re-transmission) of packets.

We use Fig. 2 to explain the operations of SCPS. Assume that there are four stations, A, B, C, and D with listen-intervals 4, 2, 3, and 2, respectively. That is, station A wakes up every 4 beacon intervals, B wakes up every 2 beacon intervals, and so on. Assume that, in the 4th beacon interval, C wishes to send data to A. Because A does not wake up in this beacon interval, C cannot transmit the data to A at this time. C needs to wait until the 5th beacon interval in which A will wake up. If C knows A's wakeup schedule, C can sleep until the 5th beacon interval instead of waking up at every beacon interval to check A's availability. In SCPS, every station has a copy of the wakeup schedule of every other station that is in the PS mode.

SCPS has to deal with three issues. The first is synchronizing the timers. Because stations do not wake up in every beacon interval, stations' timers may become out of synchronization. The second is maintaining a consistent wakeup table (which is the collection of the wakeup schedules of all stations in the PS mode) among all stations. Because there is not a central coordinator in an ad hoc network, a distributed mechanism is needed to keep all the copies of the wakeup table consistent. The third is evenly arranging stations to wake up in beacon intervals in order to alleviate the contention.

3.1. The wakeup information of other stations

In SCPS, every station needs to maintain the wakeup information of all other stations, which is kept in the *Wakeup Information Table* (WIT).

Fig. 3 is an example of WIT. Each entry comprises four fields: station ID (SID), MAC address, listen-interval (ℓ), and wakeup count (WC). SID is a unique value chosen by the station to identify itself. The range of a SID is 0–127. When a station enters the PS mode, it announces its SID to the network. Other stations can bind this SID to the announcer's MAC address. The *listen-interval*, denote as ℓ , of a station is the number of beacon intervals for which the station sleeps between two adjacent wakeups. The wakeup count (WC) is the number of beacon intervals for which the station will sleep before the next wakeup. Note that WC cycles through from $\ell - 1$ down to 0.

Since an ad hoc network lacks a central coordinator, WIT maintenance is done during the beacon process. We add a new piece of information, called the *Station Wakeup Information* (SWI), in the beacon frames (Fig. 4). The station that wins the right to send a beacon frame broadcasts the SWI information, which contains the wakeup schedules of all stations in the PS mode. Thus, newly joined stations can obtain the wakeup schedules of other stations that are in the PS mode. We call the station

SID	MAC address	ℓ	WC
2	00-06-AE-EF-56-21	5	0
4	00-02-3B-D3-A6-21	4	3
8	00-06-4E-B7-65-67	4	0
5	00-03-E9-03-24-5F	4	2
9	00-01-D2-03-71-21	4	1

Fig. 3. The Wakeup Information Table (WIT).

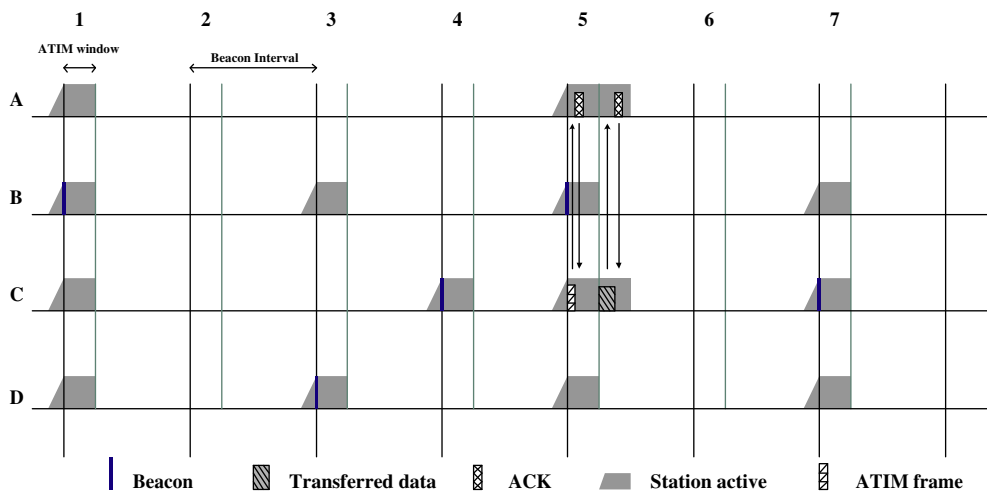


Fig. 2. The power-saving operation of SCPS.

that sends the beacon frame as the *beacon sender*. The SWI includes the following fields:

- PS (power-saving) status: This one-byte field indicates the state of the beacon sender. If the PS status is JOIN or LEAVE, it means that the beacon sender is going to enter or exit the PS mode, respectively; otherwise, its status is NORMAL.
- SID bitmap: The SID bitmap is similar to the virtual bitmap in the traffic indication map (TIM) in the 802.11 infrastructure mode. It consists of 128 bits. Each bit is tied to a station ID. When a SID is occupied by a station (which must be in the PS mode), the bit tied to the SID is 1; otherwise, the bit is 0. In practice, an one-hop ad hoc network comprises 25–40 stations. Thus, 128 bits should be enough.
- Wakeup Parameter Set (WPS): This set includes the listen-interval and wakeup counter of every station that is in the PS mode. The information is listed in the ascending order of the SIDs.

In order to allow a station with new SWI information (such as those who want to enter or exit the PS mode) to preempt other stations when competing for the transmission medium, the rules for selecting a backoff window size is modified in SCPS. The station that wants to enter or exit the PS mode chooses its backoff window size randomly between 0 and $\rho/2$, where ρ is two times the minimum backoff window size specified in IEEE 802.11. Other stations choose their backoff window sizes randomly between $\rho/2$ and ρ .

Obviously, the WIT in each beacon frame consumes bandwidth. The size of a WIT is $1 + 16 +$

$2u$ bytes, where u is the number of stations in the PS mode. If the physical transmission rate is 1 Mbps, the extra time for transmitting the WIT is less than 2.2 ms when $u = 128$ stations, which is about 2% of the bandwidth if a beacon interval lasts for 100 ms. For a common network (containing 25–40 stations) the overhead is 0.54–0.78% of the bandwidth.

3.2. Timer synchronization

In order to synchronize stations' timers, we define the *Timer Synchronization Beacon Interval* (TSBI) as a beacon interval in which all stations will wake up and compete to serve as the beacon sender. All stations will synchronize their timers with that of the beacon sender. The beacon of TSBI is called as the *Timer Synchronization Beacon*. The number of beacon intervals between two consecutive Timer Synchronization Beacons is called the *Timer Synchronization Period* (TSP).

TSP affects the efficiency of power-saving operations and timer synchronization. Long TSP allows stations to sleep longer but also increases the failure probability on timer synchronization.

Table 1 shows the number of beacon intervals needed for the clock to drift for more than the duration of a DIFS with various clock accuracy ratio ranged from 0.001% to 0.01% in a 802.11b Direct Sequence Spread Spectrum (DSSS) physical layer. The worst clock accuracy ratio specified in 802.11 is 0.01%. Because a station has to wait for a DIFS period before it counts down the (random) backoff time, we require the clock skew should not exceed a DIFS period. For instance, if the clock accuracy ratio is 0.001%, all the clocks

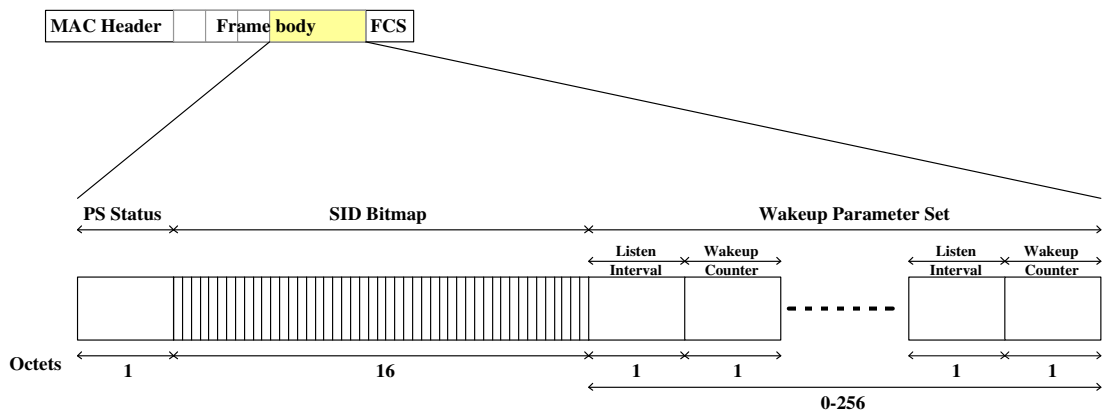


Fig. 4. The format of station wakeup information (SWI).

Table 1
The number of beacon intervals needed for the clock to drift for a DIFS

Accuracy	0.01%	0.005%	0.003%	0.002%	0.0016%	0.001%
Beacon intervals	2.50	5.00	8.33	12.50	15.00	25.00

should be synchronized every 25 (or less) beacon intervals.

Determining a suitable TSP needs experiments. In our simulation study, the clock accuracy ratio is assumed to be 0.001%. Thus, TSP is 25 beacon intervals. The reason will be given in the simulation section. Note that the TSBI is for the sole purpose of timer synchronization. The stations will not send any frames except the Timer Synchronization Beacon frame during a TSBI. After that, the stations go to sleep.

3.3. Maintaining a consistent wakeup table

In this section, we show how to maintain a consistent wakeup table while stations enter or exit the PS mode.

Before a new station, say S_j , enters the PS mode, it needs to be a beacon sender to send SWI to all other stations. Then, S_j stays wakeup until it receives a beacon frame with the NORMAL PS status. At that time, S_j knows that no station is entering or exiting the PS mode. It then builds a WIT table based on the SWI fields of the received beacon frame. In the next beacon interval, S_j chooses a SID, sets the PS status to JOIN, and prepares the SWI fields for the beacon frame. S_j uses $[0, \rho/2]$ as the backoff window size to compete for the medium for transmitting the beacon frame. Once S_j successfully sends out its beacon frame, it will serve as the beacon sender for the next k beacon intervals (starting from the current beacon interval). Note that k should be large enough to ensure that all stations in the PS mode will wake up and receive S_j 's beacon frame at least once during these k beacon intervals. k can be determined as follows:

$$k = \min(\ell_{\max}, \omega), \quad (1)$$

where ℓ_{\max} is the maximum listen-interval in the current WIT table and ω is the number of beacon intervals from the current beacon interval to the next TSBI. After k beacon intervals, all stations in the PS mode are notified and they can update their WIT tables. Other stations that also wish to be the beacon senders but fail to win the transmission med-

ium need to wait for the next beacon frame with the NORMAL PS status.

Similarly, when a station, say S_e , wants to exit the PS mode, it removes its wakeup schedule from the beacon frame, sets its PS status to LEAVE, and competes for the transmission medium with a back-off window size randomly chosen from $[0, \rho/2]$. After S_e sends its beacon frame, it continues to serve as the beacon sender for the next k beacon intervals to notify all other stations in the PS mode, where k is given in Eq. (1) above.

When a station wakes up and receives a beacon frame with a JOIN or LEAVE PS status, it uses the SWI information in the beacon frame to update its own WIT table. Because the MAC address of the beacon sender is included in the beacon frame, other stations can bind the beacon sender's SID to its MAC address.

Table 2 shows an example of a station entering and exiting the PS mode. We use English letters A , B , etc. to represent SIDs. In Table 2, $w_i(t) = 1$ if station i wakes up at beacon interval t ; $w_i(t) = 0$, otherwise. For each beacon interval, we mark the beacon sender with its PS status as a superscript. That is, $w_i(t) = 1^x$ indicates that, during interval t , station i is the beacon sender and the PS status is NORMAL (N), JOIN (J), or LEAVE (L).

Let $n(t)$ denote the number of stations waking up at beacon interval t . Then, $n(t)$ can be found by

$$n(t) = \sum_{i \in S} w_i(t),$$

where S is the set of all stations in PS mode.

There are 5 stations in this example. Assume that the 5th beacon interval is a TSBI. During the 1st beacon interval, 4 stations— A , B , C , and D —are in the PS mode. Their listen-intervals are 4, 3, 3, and 3, respectively. Assume that, at the 3rd beacon interval, station E wants to enter the PS mode with its listen-interval set to 4. Station E receives a beacon frame issued by station C with the NORMAL PS status. So it uses the SWI information in this beacon frame to update its WIT table. During the 4th beacon interval, station E adds its wakeup schedule to the beacon frame and sets the PS status to JOIN. It employs a short delay to win the opportu-

Table 2
A sequence of $w_i(t)$ and $n(t)$

t	1	2	3	4	5*	6	7	8	9	10	11	12
$w_A(t)$	0	1 ^N	0	0	0	1	0	0	0	1	0	0
$w_B(t)$	0	0	1	0	0	1	1 ^L	1 ^L	1 ^L	1 ^L	–	–
$w_C(t)$	0	0	1 ^N	0	0	1	0	0	1	0	0	1 ^N
$w_D(t)$	0	0	1	0	0	1 ^N	0	0	1	0	0	1
$w_E(t)$	–	–	1	1 ^J	1 ^J	0	1	0	0	0	1 ^N	0
$n(t)$	0	1	4	1	1	4	2	1	3	2	1	2

Note: 5* means the 5th beacon interval is a TSBI.

nity to serve as the beacon sender. It continues to serve as the beacon sender until the 5th beacon interval, which is the TSBI. Note that in this case $\ell_{\max} = 4$, $\omega = 2$, and that $k = \min(\ell_{\max}, \omega) = 2$.

Assume that, during the 5th beacon interval, station B wants to exit the PS mode. Because the received beacon frame is not in the NORMAL status, B keeps on monitoring the medium. During the 6th beacon interval, B detects that the PS status of the beacon frame is NORMAL. It competes for the medium during the 7th beacon interval. Station B continues to serve as the beacon sender until the 10th beacon interval. Stations that wake up and receive the beacon frames from B make use of the SWI information to update their respective WIT tables.

It is obvious that the number of wakeup stations in each beacon interval is not the same. If stations' wakeup schedules are properly arranged so that the number of wakeup stations in each beacon interval is roughly the same, the contention for the transmission medium will be reduced. Therefore, we hope to *balance* the number of wakeup stations in each beacon interval.

3.4. Balancing the number of wakeup stations

To balance the number of wakeup stations in each beacon interval, we need to adjust the WC of the sleeping stations when there is a station entering or exiting the PS mode. In the following subsec-

tions, we show how SCPS balances the number of stations.

3.4.1. A station enters the PS mode

Consider a wireless ad hoc network with six stations, A, B, C, D, E, and F. Assume they all are sleeping initially. Their listen-intervals are 4, 3, 3, 3, 3, and 4, respectively. We will use the sequence of $w_i(t)$'s and the total number of wakeup stations $n(t)$, for $t = 1, 2, \dots, 18$, in Table 3 as our example.

Assume that the 12th beacon interval is a TSBI and the stations' first wake up at the 4th, 3rd, 2nd, 1st, 1st, and 3rd beacon intervals, respectively. Assume that station J wants to enter the PS mode at beacon interval 4. Furthermore, assume station J's listen-interval is 3 and J randomly chooses a wakeup schedule $\langle 1, 0, 0 \rangle$ (this schedule means that J wakes up once every three beacon intervals). The row of $n(t)_{bf}$ shows the number of wakeup stations in each beacon interval before station J enters the PS mode and the row of $n(t)_{af}$ shows the numbers after station J enters the PS mode.

Before station J enters the PS mode, the maximum number of $n(t)$ in a beacon interval is 3, which occurs at beacon intervals 4th, 7th, and 16th, respectively. With the unfortunate choice of the wakeup schedule $\langle 1, 0, 0 \rangle$, the maximum number of $n(t)$ becomes 4 after station J enters the PS mode. It is because that its wakeup time includes the 4th, 7th, and 16th beacon intervals.

Table 3
A sequence of $w_i(t)$ and $n(t)$

t	1	2	3	4	5	6	7	8	9	10	11	12*	13	14	15	16	17	18
$w_A(t)$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
$w_B(t)$	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
$w_C(t)$	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0
$w_D(t)$	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	1	0	0
$w_E(t)$	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	1	0	0
$w_F(t)$	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0
$w_J(t)$	–	–	–	1 ^J	1 ^J	1 ^J	1 ^J	0	0	1	0	1	1	0	0	1	0	0
$n(t)_{bf}$	2	1	2	3	1	1	3	2	1	2	2	6	2	1	2	3	1	1
$n(t)_{af}$	2	1	2	4	2	2	4	2	1	3	2	7	3	1	2	4	1	1

Note: 12* means the 12th beacon interval is a TSBI.

On the other hand, if J chooses the wakeup schedule $\langle 0, 1, 0 \rangle$, the maximum of $n(t)$ is still 3, as shown in Table 4 (Only the stations' periodical wakeup schedules are show in it). Therefore, by choosing an appropriate wakeup schedule for J , the maximum number of competing stations at any interval can be minimized.

By observing the sequence of $n(t)_{bf}$ in Table 4, we may find that a pattern repeats every 12 beacon intervals, e.g.,

$$(n(4), n(5), \dots, n(15)) = (3, 1, 1, 3, 2, 1, 2, 2, 3, 2, 1, 2).$$

The length of this repeating pattern, r , must be a factor of the least common multiple (lcm) of the listen-intervals ℓ_i 's, where $i \in S$. For example, the listen-intervals of stations, A, B, C, D, E, F and J are 4, 3, 3, 3, 3, 4 and 3, respectively, in Table 4. Then

$$r = lcm\{4, 3, 3, 3, 3, 4, 3\} = 12.$$

In addition, for a station with listen-interval ℓ , there are exactly ℓ wakeup patterns:

$$(1, 0, 0, \dots), (0, 1, 0, \dots), (0, 0, 1, \dots), \dots, (0, 0, \dots, 1).$$

Consider station J of Table 4, whose listen-interval is 3. J will carry out the following computations to find out the maximum $n(t)$ for each pattern p , which is denoted as n_p^* :

Case 1. Assume $(w_J(4), w_J(5), \dots, w_J(15)) = (0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1)$. Then

$$\begin{aligned} (n(4), n(5), \dots, n(15)) &= (0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1) \\ &\quad + (3, 1, 1, 3, 2, 1, 2, 2, 3, 2, 1, 2) \\ &= (3, 1, 2, 3, 2, 2, 2, 2, 4, 2, 1, 3) \end{aligned}$$

and $n_1^* = \max\{n(4), n(5), \dots, n(15)\} = 4$.

Case 2. Assume $(w_J(4), w_J(5), \dots, w_J(15)) = (0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0)$. Then

$$\begin{aligned} (n(4), n(5), \dots, n(15)) &= (0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0) \\ &\quad + (3, 1, 1, 3, 2, 1, 2, 2, 3, 2, 1, 2) \\ &= (3, 2, 1, 3, 3, 1, 2, 3, 3, 2, 2, 2) \end{aligned}$$

and $n_2^* = \max\{n(4), n(5), \dots, n(15)\} = 3$.

Case 3. Assume $(w_J(4), w_J(5), \dots, w_J(15)) = (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0)$. Then

$$\begin{aligned} (n(4), n(5), \dots, n(15)) &= (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0) \\ &\quad + (3, 1, 1, 3, 2, 1, 2, 2, 3, 2, 1, 2) \\ &= (4, 1, 1, 4, 2, 1, 3, 2, 3, 3, 1, 2) \end{aligned}$$

and $n_3^* = \max\{n(4), n(5), \dots, n(15)\} = 4$.

Finally, station J chooses a pattern with the minimum n_x^* as its schedule. In this example, $n_2^* = \min\{n_1^*, n_2^*, n_3^*\} = 3$. J chooses the second case as its schedule.

Formally, we define the *wakeup scheduling problem* (WSP) as follows: Assume station J wants to enter the PS mode. Given a set of m sleeping stations at beacon interval t , assume each station i has the wakeup schedule $\langle w_i(t+1), w_i(t+2), w_i(t+3), \dots, w_i(t+\ell_i) \rangle$. We want to assign a wakeup schedule $\langle w_J(t+1), w_J(t+2), w_J(t+3), \dots \rangle$ to station J such that $\max\{n(t+k) \mid k = 1, 2, \dots\}$ is minimized, where $n(t+k)$ is defined as $\sum_{i \in S \cup \{J\}} w_i(t+k)$, for $k = 1, 2, \dots$.

The WSP problem may be solved with the following method:

1. Determine the length of the repeating pattern: $r = lcm\{\ell_i \mid i \in S \cup \{J\}\}$.
2. Generate the set of all possible wakeup patterns:

$$\begin{aligned} W &= \{\langle w_J^a(t+1), w_J^a(t+2), \dots, w_J^a(t+r) \rangle \mid a \\ &= 1, 2, \dots, \ell_J\} \end{aligned}$$

Table 4
A sequence of $w_i(t)$ and $n(t)$ for station J with first wakeup time $t = 5$

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$w_A(t)$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
$w_B(t)$	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
$w_C(t)$	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
$w_D(t)$	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
$w_E(t)$	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
$w_F(t)$	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0
$w_J(t)$	–	–	–	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
$n(t)_{bf}$	2	1	2	3	1	1	3	2	1	2	2	3	2	1	2	3	1	1
$n(t)$	2	1	2	3	2	1	3	3	1	2	3	3	2	2	2	3	2	1

- For each wakeup pattern $\langle w_j^a(t+1), w_j^a(t+2), \dots, w_j^a(t+r) \rangle$, where $a = 1, 2, \dots, \ell_j$, compute

$$n^{(a)} = \max \left\{ n_k \mid n_k = \sum_{i \in S} w_i(t+k) + w_j^a(t+k), k = 1, 2, \dots, r \right\}.$$

- Choose a wakeup pattern $\langle w_j^*(t+1), w_j^*(t+2), \dots, w_j^*(t+r) \rangle$ such that $n^* = \min\{n^{(a)} \mid a = 1, 2, \dots, \ell_j\}$. That is, the wakeup pattern $\langle w_j^*(t+1), w_j^*(t+2), \dots, w_j^*(t+r) \rangle$ balances the total number of wakeup stations $n(s)$, for $s = t+1, t+2, \dots$

The time complexity for computing a minimum sequence depends on the least common multiple of the listen-intervals. Assume that R is the current least common multiple and the listen-interval of a newly joined station is L . The time complexity for the new station to examine all possible patterns is $O(R^*L)$ and that for finding out the minimum (that is, n^*) is $O(L)$. Thus, the total time complexity is $O(R^*L) + O(L) = O(R^*L)$.

3.4.2. A station exits the PS mode

When a station, say S_e , exits the PS mode, the number of wakeup stations in each beacon interval may become unbalanced. Table 5 illustrates such a scenario. The number of wakeup stations, shown

in the row of $n(t)$ in Table 5, is not balanced after station C exits the PS mode at the 5th beacon interval. This is because stations D and E wake up at the same beacon interval. If one of them rearranges its wakeup sequence, the number of wakeup stations in each beacon interval can become balanced again. The unbalanced phenomenon may become more severe if many stations exit the PS mode but no station enters the PS mode in a period of time. Thus, it is necessary to re-balance the number of wakeup stations after some station exits the PS mode.

Note that all stations have to wake up in the TSBI to synchronize their timers. Thus, TSBI is the *right* time to re-balance the number of wakeup stations. Thus, we embed the re-balance mechanism in TSBI.

At TSBI, say beacon interval $t + 1$, each station i solves the WSP problem as if it wants to enter the PS mode again and calculates the optimal value n_i^* . Each station i can also calculate the optimal values n_j^* (where $j \in S - \{i\}$) for all other stations j . Comparing n_i^* with other n_j^* 's, station i can learn if n_i^* is the smallest. The station with the least n_j^* wins the right to serve as the beacon sender. It can re-balance the number of wakeup stations in a beacon interval as well as synchronize all stations' timers. In case there are more than one station with the same least n^* , the one with the smallest station ID wins.

Table 5
A sequence of $w_i(t)$ and $n(t)$ for station C exiting the PS mode at time $t = 5$

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$w_A(t)$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
$w_B(t)$	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
$w_C(t)$	0	1	0	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–
$w_D(t)$	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
$w_E(t)$	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
$w_F(t)$	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
$n(t)$	2	1	2	3	0	1	3	1	1	2	1	3	2	0	2	3	0	1

Table 6
A sequence of $w_i(t)$ and $n(t)$ when station C exits the PS mode at time $t = 5$

t	1	2	3	4	5	6	7	8	9	10	11	12*	13	14	15	16	17	18
$w_A(t)$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
$w_B(t)$	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
$w_C(t)$	0	1	0	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–
$w_D(t)$	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	1	0
$w_E(t)$	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
$w_F(t)$	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
$n(t)$	2	1	2	3	0	1	3	1	1	2	1	2	1	1	2	2	1	1

Note: 12* means the 12th beacon interval is a TSBI. Re-balancing occurs at $t = 12$.

Table 6 shows an example of the re-balance mechanism. Assume that at the 5th beacon interval, station *C* exits the PS mode. All stations in the PS mode learn *C* exited the PS mode after the 8th beacon interval. The 12th beacon interval is a TSBI. All stations wake up and solve the WSP problems. Then, every station calculates $n_A^* = 3$, $n_B^* = 3$, $n_D^* = 2$, $n_E^* = 2$, and $n_F^* = 3$. Station *D* wins the right to serve as the beacon sender in the 12th beacon interval. Its wakeup schedule ($w_D(12)$, $w_D(13)$, $w_D(14)$, ...) is set to (0, 0, 1, ...) Note that the maximum number of wakeup stations in each beacon interval after the 12th becomes 2 rather than 3.

4. Simulation results

4.1. Performance metrics and environment setup

The following four metrics are studied in the simulation of SCPS:

1. *The number of wakeup stations in the ATIM window of a beacon interval.* Generally, the probability of a collision during the competition for the transmission medium is influenced by the number of wakeup stations that contend during the ATIM window. If we can carefully control the number of wakeup stations, the collision probability can be reduced.
2. *Sleep ratio.* Sleep ratio is a common index for evaluating power-saving mechanisms. The sleep ratio is defined as the ratio of the amount of time that a station sleeps to the total amount of time. The longer a station sleeps, the less energy it consumes. The sleep ratio is influenced by the traffic generation rate. Thus, we adjust the traffic generation rate in our simulation to observe the variance of the sleep ratio.
3. *The average queuing delay.* Usually, increasing the sleep time also increases the queuing delay. A longer listen-interval results in more efficient energy usage but it also increases the packets' queuing time. This metric shows that the average queuing delay for different listen-intervals. The delay is calculated from the time when a packet is put into the queue until it is sent out.
4. *The average packet drop ratio.* A station drops its queued packets if the packets cannot be transmitted before the expiration time or they collide with other stations' packets while they are transmit-

ted. Dropped packets need to be re-sent. This wastes energy. Hence, the average packet drop ratio is a good index of energy efficiency.

We compare these four metrics in SCPS, 802.11 PS mode (denoted as PSM in the following figures), DPSM [15], and QEC [16] with simulation. The beacon interval is fixed at 100 ms. The ATIM window size for DPSM ranges from 2 to 26 ms. For the other two protocols, the ATIM window size is 25 ms. Two different grid sizes are implemented for QEC: QEC2 (with a 2×2 grid) and QEC4 (with a 4×4 grid). In QEC2, the average listen-interval is $\frac{4}{3}$ beacon intervals, that is, a station wakes up three times every four beacon intervals. Similarly, the average listen-interval is $\frac{16}{7}$ beacon intervals for QEC4.

In SCPS, the length of each station's listen-interval is randomly chosen from 1 to 4 beacon intervals. For a fair comparison, we choose two configurations for SCPS:

SCPSa Half of the stations wake up every beacon interval. The remaining stations wake up once every two beacon intervals. The average listen-interval is, thus, $\frac{4}{3}$ beacon intervals. SCPSa will be compared against QEC2.

SCPSb Three fourths of the stations wake up once every two beacon intervals. The remaining one fourth of the stations wake up once every four beacon intervals. The average listen-interval is, thus, $\frac{16}{7}$ beacon intervals. SCPSb will be compared against QEC4.

The remaining simulation parameters are as follows: The physical data transmission rate is 11 Mbps. The constant bit rate (CBR), 4 packets/s, is used for traffic generation. The size of a packet is 8000 bits. A packet is dropped if its queuing time is more than 1.6 s, which is 4 times the maximum listen-interval. The maximum buffer size of each station is 20 k bytes. The total simulation time was 1800 s. The performance metrics are averaged over 1000 runs.

The clock accuracy ratio in our simulation is assumed to be 0.001%. In QEC4, two stations will synchronize their clocks when they wake up at the same beacon interval. The difference between two consecutive clock synchronizations of two stations is at most 15 beacon intervals. According to Table 1, we have to use clocks with accuracy ratio no

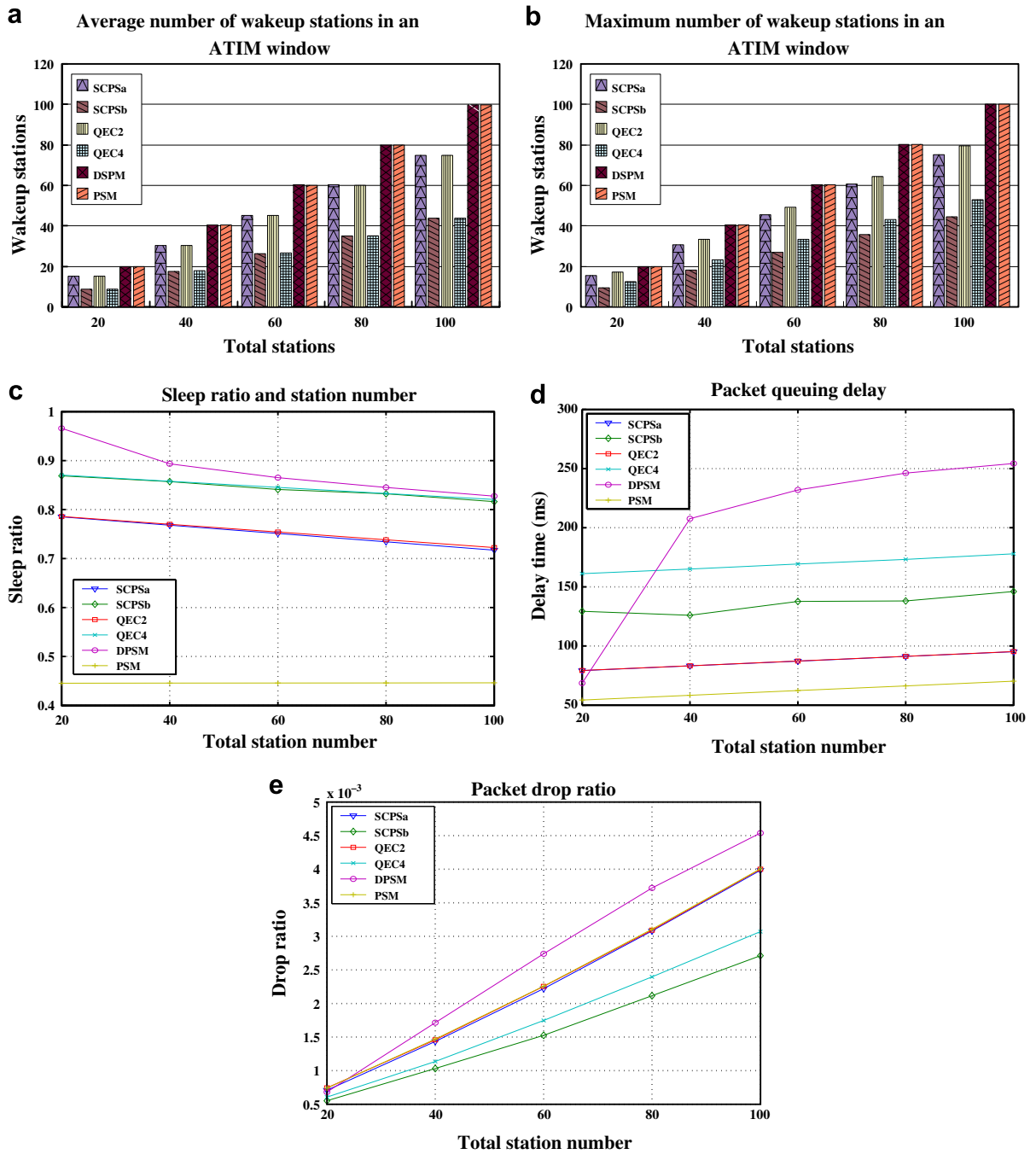


Fig. 5. The influence of the numbers of stations.

worse than 0.0016%.² In our simulation, clock accuracy ratio is fixed at 0.001% and, hence, TSP is 25 beacon intervals.

² Note that the smaller the numerical value of the clock accuracy ratio is, the more accurate the clock will be.

4.2. Experimental results

We first inspect the influence of the number of stations in the PS state, which is shown in Fig. 5. Fig. 5a shows the average number of wakeup stations in an ATIM window. Because in both 802.11

PSM and DPSM, all stations in the PS state need to wake up in every beacon interval, the number of wakeup stations is equal to the total number of stations in the PS state. On the other hand, for QEC and SCPS, the average numbers of wakeup stations in a beacon interval obtained from the experiments are identical. This is consistent with the probability distribution.

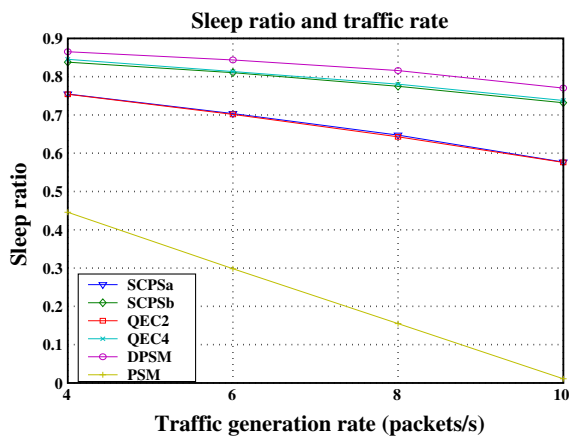
Fig. 5b shows the maximum number of wakeup stations in an ATIM window. Consider the case of 100 total stations. The maximum number of wakeup stations in QEC4 is 52.46 while that in SCPSb is 44.37. It is obvious that SCPSb incurs lighter contention than QEC4 in the worst case. Similarly for QEC2 vs. SCPSa.

Fig. 5c shows the relationship of the sleep ratio and the total number of stations. As expected, the sleep ratio in every protocol decreases as the number of stations grows. The original IEEE 802.11 protocol (PSM) has the worst (that is, lowest) sleep ratio. Its sleep ratio is about 45%. In QEC and SCPS, the sleep ratios are more than 72%. The sleep ratio in SCPSa is almost the same as that in QEC2. Similarly, the sleep ratios in SCPSb and QEC4 are also almost identical. For DPSM, because its initial ATIM window is short (2 ms), DPSM has the best sleep ratio. However, as the number of stations increases, the sleep ratio in DPSM decreases rapidly. This is due to the fact that the narrow bandwidth provided by short ATIM windows cannot satisfy the transmission requests as there are more and more stations. Every station has to increase its ATIM window size and, hence, the sleep ratio decreases.

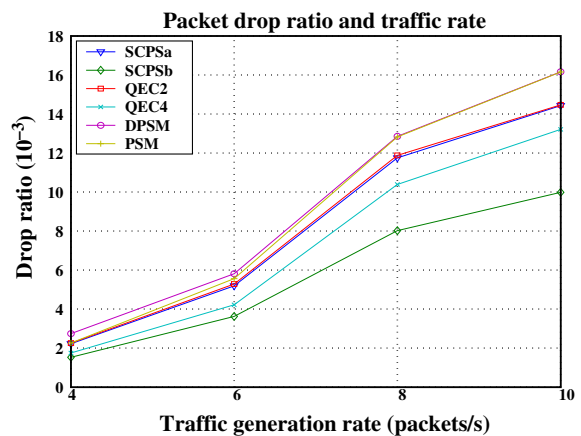
Fig. 5d shows the average queuing delay of a packet. When there are more stations in the network, the competition for the medium gets hotter. Thus, the queuing delay becomes longer. Among the tested protocols, PSM has the shortest delay because stations wake up more often. In DPSM, because of the slow adaptation of the ATIM window size, the queuing delay grows rapidly as the number of stations exceeds 40. This results from the short initial ATIM window, which limits the number of ATIM frames that can be announced and causes the packets congested in the buffer. Consequently, the queuing delay becomes longer. SCPSa has almost the same queuing delay as QEC2 (their curves completely overlap and are hard to distinguish in Fig. 5d). The delay in SCPSb is shorter than that in QEC4 by 30 ms. SCPSb reduces 20% of the queuing delay of QEC4 but its sleep ratio is almost the same as that in QEC4.

Fig. 5e shows the number of dropped packets. With the constant traffic generation rate of 4 packets per second, the total number of transmitted packets per station is 7200 packets (4 packets/s \times 1800 s). Because of the short initial ATIM window in DPSM, the number of ATIM frames that can be announced is very limited. Thus, many packets are dropped even though there is still available bandwidth in the data transmission window. Furthermore, all stations have to wake up in every beacon interval in DPSM. This causes higher probability of collision. So more packets are dropped in DPSM. DPSM has the highest packet drop ratio.

The packet drop ratios of SCPSa, QEC2, and PSM are almost the same. The drop ratio in SCPSb



(a) Sleep ratio vs. traffic rate



(b) Packet dropped vs. traffic rate

Fig. 6. Influence of traffic generation rates (60 stations; packet size : 8000 bits).

is less than that in QEC4 because the number of wakeup stations in SCPSb is more balanced. By minimizing the maximum number of wakeup stations in every beacon interval, SCPSb reduces the number of packet collisions by alleviating the medium contention.

In Fig. 6, we observe the impact of the traffic generation rates. In Fig. 6a, the sleep ratios in all tested protocols decrease as the traffic generation rate increases. Because stations which transmit or receive data in a data transmission window need to stay awake during the whole beacon interval, 802.11 PSM has the worst sleep ratio. The stations almost cannot sleep when the traffic generation rate is higher than 10 packets/s.

Fig. 6b gives the ratio of dropped packets in each approach. Without distributing the wakeup stations evenly among all beacon intervals, DPSM and 802.11 PSM drop more packets. Furthermore, because of the slow adaptation of the ATIM window size, DPSM drops even more packets than 802.11 PSM while traffic rate is less than 6 packets/s.

Fig. 7 shows the impact of packet sizes. While the packet size is 8000 bits, the bandwidth capacity can still satisfy the requests of all stations. Packets are dropped due to packet collision. As the packet size increases to 16,000 bits or more, the bandwidth capacity cannot satisfy all requests. In this case, packets are dropped because packets wait too long in the queue. The number of dropped packets is the same for all tested protocols (because we use the same traffic generation rate and the same bandwidth).

The throughput and sleep ratio for SCPSb are given in Fig. 8. The throughput ratio is defined as

the ratio of the number of successfully transmitted packets over the total number of generated packets. The packet size is 16,000 bits. As expected, the sleep ratio increases and the throughput ratio decreases while a listen-interval lasts longer. For stations with listen-interval equal to 1 beacon interval, the sleep ratio is only 30% but the throughput ratio is higher than 95%. On the other hand, for stations with listen-interval equal to 4, the sleep ratio approximates 80%, but the throughput ratio only 55%.

Finally, Fig. 9 shows the average waiting time that a station spends on switching to the PS mode in SCPSb. The X-axis is the number of stations that switch to the PS mode every 40 beacon intervals.

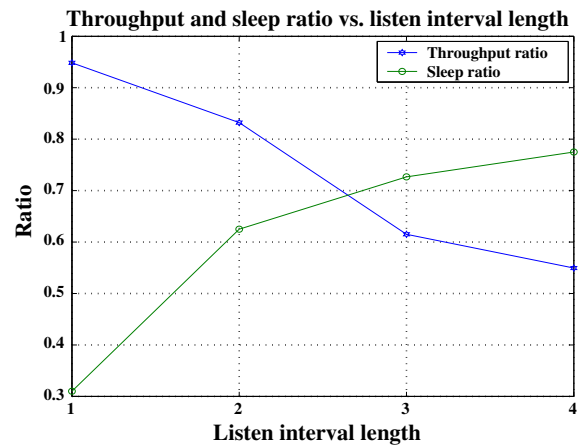


Fig. 8. Throughput and sleep ratio as functions of the listen-interval in SCPSb (60 station; traffic generation rate: 8 packets/s).

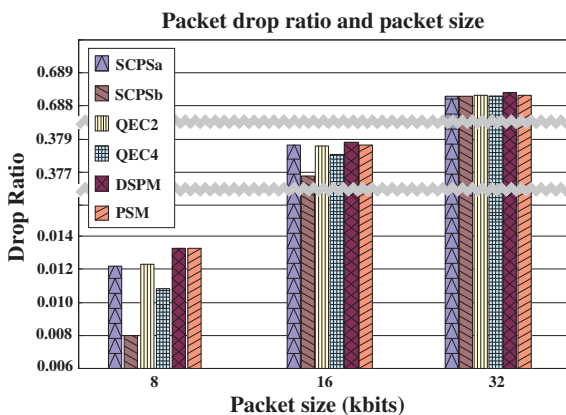


Fig. 7. Influence of packet sizes (60 station; traffic generation rate : 8 packets/s).

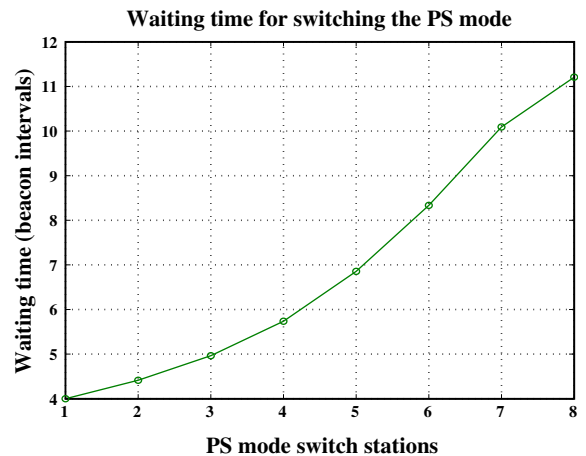


Fig. 9. The average waiting time for a station to switch to the PS mode in SCPSb.

Because a station switching to the PS mode needs to serve as the beacon sender for a fixed number of listen-intervals and because a NORMAL beacon frame must be detected before another station may switch to the PS mode, at most 8 stations may switch to the PS mode every 40 beacon intervals. From Fig. 9, we can see that the waiting time increases as the number of switching stations grows. In the worst case, stations have to wait three times the maximum listen-interval.

5. Conclusions and future work

We proposed a flexible power-saving scheduling protocol, SCPS, for ad hoc networks. By providing each station a flexible mechanism for adjusting its listen-interval, SCPS can balance the number of wakeup stations during each beacon interval without the need of a coordinator. Simulation results show that, when the number of wakeup stations is balanced, packets dropped due to collision are effectively reduced.

Currently, SCPS is applied to one-hop ad hoc networks only. For extension to multi-hop networks, we would need an efficient mechanism to propagate the WIT information to those stations that are not one-hop reachable. Furthermore, timer synchronization in multi-hop networks will become more difficult.

References

- [1] G. Razzano, L. Andreani, R. Cusani, Wireless LANs: an adaptive algorithm to reduce power consumption, in: Proceedings of the IEEE International Conference on Communications (ICC 2003), vol. 2, 2003, pp. 1096–1100, May.
- [2] S.T. Sheu, J. Chen, MR²RP: the multi-rate and multi-range routing protocol for ad hoc wireless networks, in: Proceedings of the 7th International Symposium on Computers and Communications, 2002, pp. 435–440, July.
- [3] C.S. Hsu, J.P. Sheu, Y.C. Tseng, Minimize waiting time and conserve energy by scheduling transmissions in IEEE 802.11-based ad hoc networks, in: Proceedings of the 10th International Conference on Telecommunications, vol. 1, 2003, pp. 393–399, February.
- [4] J. Jelitto, A.N. Barreto, H.L. Truong, Power and rate adaptation in IEEE 802.11a wireless LANs, in: Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference, vol. 1, 2003, pp. 413–417, April.
- [5] C.A.S. Oliveira, P.M. Pardalos, A distributed optimization algorithm for power control in wireless ad hoc networks, in: Proceedings of the 18th International Parallel and Distributed Processing Symposium, 2004, pp. 177–185, April.
- [6] A. Sheth, R. Han, Adaptive power control and selective radio activation for low-power infrastructure-mode 802.11 LANs, in: Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops, 2003, pp. 812–818, May.
- [7] S. Gitzenis, N. Bambos, Efficient data prefetching for power-controlled wireless packet networks, in: Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004, pp. 64–73, August.
- [8] R.P. Mann, K. Namuduri, R. Pendse, Energy aware routing protocol (EARP) for ad-hoc wireless sensor networks, in: Proceedings of the IEEE Global Telecommunications Conference Workshops 2004 (GlobeCom Workshops 2004), 2004, pp. 419–427, November.
- [9] D. Kim, J.J. Garcia-Luna-Aceves, K. Obraczka, J.-C. Cano, P. Manzoni, Routing mechanisms for mobile ad hoc networks based on the energy drain rate, IEEE Transactions on Mobile Computing 2 (2) (2003) 161–173.
- [10] T. Park, K.G. Shin, Optimal tradeoffs for location-based routing in large-scale ad hoc networks, IEEE/ACM Transactions on Networking 13 (2) (2005) 398–410.
- [11] J.R. Lee, S.W. Kwon, D.H. Cho, Adaptive beacon listening protocol for a TCP connection in slow-start phase in WLAN, IEEE Communications Letters 9 (9) (2005) 853–855.
- [12] IEEE 802.11 Working Group. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
- [13] M. Liu, M.T. Liu, A power-saving scheduling for IEEE 802.11 mobile ad hoc network, in: Proceedings of the IEEE International Conference on Computer Networks and Mobile Computing (ICCNMC 2003), 2003, pp. 238–245, October.
- [14] S.L. Wu, P.C. Tseng, An energy efficient MAC protocol for IEEE 802.11 WLANs, in: Proceedings of the IEEE 2nd Annual Conference on Communication Networks and Services Research, 2002, pp. 137–145, May.
- [15] E.S. Jung, N.H. Vaidya, An energy efficient MAC protocol for wireless LANs, in: Proceedings of the IEEE Twenty-First Annual Joint Conference of the Computer and Communications Societies (INFOCOM 2002), vol. 3, 2002, pp. 1756–1764, June.
- [16] C.M. Chao, J.P. Sheu, I.C. Chou, An adaptive quorum-based energy conserving protocol for IEEE 802.11 ad hoc networks, IEEE Transactions on Mobile Computing 5 (5) (2006) 560–570.
- [17] V. Rajendran, G.L. Aveses, J. J. K. Obraczka, Energy-efficient, application-aware medium access for sensor networks, in: Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems, 2005, pp. 1–8, November.
- [18] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, Energy-efficient MAC: Energy-efficient collision-free medium access control for wireless sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, 2003, pp. 181–192, November.
- [19] B. Hohlt, L. Doherty, E. Brewer, Oral presentation session III: energy efficient design: Flexible power scheduling for sensor networks, in: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, 2004, pp. 205–214, April 2004.



Shih-Chang Huang received the B.S. degree in Computer Science and Engineering from Tatung University, Taiwan, in 1997, and received M.S. degree in Computer Science from National Tsing Hua University, Taiwan, in 1999. Since 2002, he has been working toward the Ph.D., degree in Computer and Information Science at National Chiao Tung University, Taiwan. His research interests include wireless network and

wireless sensor network.



Rong-Hong Jan received the B.S. and M.S. degrees in Industrial Engineering, and the Ph.D., degree in Computer Science from National Tsing Hua University, Taiwan, in 1979, 1983, and 1987, respectively. He joined the Department of Computer Science, National Chiao Tung University, in 1987, where he is currently a Professor. During 1991–1992, he was a Visiting Associate Professor in the Department of Computer Science,

University of Maryland, College Park, MD. His research inter-

ests include wireless networks, mobile computing, distributed systems, network reliability, and operations research.



Wuu Yang received his B.S. degree in computer science from National Taiwan University in 1982 and the M.S. and Ph.D., degrees in computer science from University of Wisconsin at Madison in 1987 and 1990, respectively. Currently he is a professor in the National Chiao-Tung University, Taiwan, Republic of China. Dr. Yang's current research interests include Java and network security, programming languages and

compilers, and attribute grammars. He is also very interested in the study of human languages and human intelligence.