Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE @ DIRECT®

Computer Networks 43 (2003) 317–337

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks

Yu-Chee Tseng [a,*], Chih-Shun Hsu [b], Ten-Yueng Hsieh [a]

[a] Department of Computer Science and Information Engineering, National Chiao Tung University, Hsin-Chu 30050, Taiwan
[b] Department of Computer Science and Information Engineering, National Central University, Chung-Li 32054, Taiwan

Received 4 September 2002; received in revised form 10 January 2003; accepted 10 April 2003

Responsible Editor: L. Lenzini

Abstract

Power-saving is a critical issue for almost all kinds of portable devices. In this paper, we consider the design of power-saving protocols for *mobile ad hoc networks* (MANETs) that allow mobile hosts to switch to a low-power *sleep* mode. The MANETs being considered in this paper are characterized by unpredictable mobility, multi-hop communication, and no clock synchronization mechanism. In particular, the last characteristic would complicate the problem since a host has to predict when another host will wake up to receive packets. We propose three power management protocols, namely *dominating-awake-interval*, *periodically-fully-awake-interval*, and *quorum-based* protocols, which are directly applicable to IEEE 802.11-based MANETs. As far as we know, the power management problem for multi-hop MANETs has not been seriously addressed in the literature. Existing standards, such as IEEE 802.11, HIPERLAN, and bluetooth, all assume that the network is fully connected or there is a clock synchronization mechanism. Extensive simulation results are presented to verify the effectiveness of the proposed protocols.

Keywords: HIPERLAN; IEEE 802.11; Mobile ad hoc network (MANET); Power management; Power saving; Wireless communication

1. Introduction

Computing and communication anytime, anywhere is a global trend in today's development. Ubiquitous computing has been made possible by the advance of wireless communication technology and the availability of many light-weight, compact, portable computing devices. Among the various network architectures, the design of *mobile ad hoc network* (MANET) has attracted a lot of attention recently. A MANET is one consisting of a set of mobile hosts which can communicate with one another and roam around at their will. No base stations are supported in such an environment, and mobile hosts may have to communicate with each other in a *multi-hop* fashion. Applications of MANETs occur in situations like battlefields, major disaster areas, and outdoor assemblies. It is also a prospective candidate to solve the "last-mile" problem for broadband Internet service providers [1].

---

[*] Corresponding author.
E-mail address: yctseng@csie.nctu.edu.tw (Y.-C. Tseng).

1389-1286/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S1389-1286(03)00284-6

One critical issue for almost all kinds of portable devices supported by battery powers is *power saving*. Without power, any mobile device will become useless. Battery power is a limited resource, and it is expected that battery technology is not likely to progress as fast as computing and communication technologies do. Hence, how to lengthen the lifetime of batteries is an important issue, especially for MANET, which is supported by batteries only.

Solutions addressing the power-saving issue in MANETs can generally be categorized as follows:

- *Transmission power control:* In wireless communication, transmission power has strong impact on bit error rate, transmission rate, and inter-radio interference. These are typically contradicting factors. In [2], power control is adopted to reduce interference and improve throughput on the MAC layer. How to determine transmission power of each mobile host so as to determine the best network topology, or known as *topology control*, is addressed in [3–5]. How to increase network throughput by power adjustment for packet radio networks is addressed in [6].
- *Power-aware routing:* Power-aware routing protocols have been proposed based on various power cost functions [7–11]. In [7], when a mobile host's battery level is below a certain threshold, it will not forward packets for other hosts. In [10], five different metrics based on battery power consumption are proposed. Ref. [11] considers both hosts' lifetime and a distance power metric. A hybrid environment consisting of battery-powered and outlet-plugged hosts is considered in [8]. Two distributed heuristic clustering approaches for two multi-casting are proposed in [9] to minimizing the transmission power.
- *Low-power mode:* More and more wireless devices can support low-power sleep modes. IEEE 802.11 [12] has a power-saving mode in which a radio only needs to be awake periodically. HIPERLAN allows a mobile host in power-saving mode to define its own active period. An active host may save powers by turning off its equalizer according to the transmission bit rate. Comparisons are presented in [13] to study the power-saving mechanisms of IEEE 802.11 and HIPERLAN in ad hoc networks. Bluetooth [14] provides three different low-power modes: *sniff*, *hold*, and *park*. Other references include [15–21].

This paper studies the management of power-saving (PS) modes for IEEE 802.11-based MANETs and thus falls into the last category of the above classification. We consider MANETs which are characterized by multi-hop communication, unpredictable mobility, no plug-in power, and no clock synchronization mechanism. In particular, the last characteristic would complicate the problem since a host has to predict when another host will wake up to receive packets. Thus, the protocol must be asynchronous. As far as we know, the power management problem for multi-hop MANETs has not been addressed seriously in the literature. Existing standards, such as IEEE 802.11 and HIPERLAN, do support PS modes, but assume that the MANET is fully connected. Bluetooth also has low-power modes, but is based on a master–slave architecture, so time synchronization is trivial. The works [18,19] address the power-saving problem, but assume the existence of access points. A lot of works have focused on multi-hop MANETs on issues such as power-aware routing, topology control, and transmission power control (as classified above), but how to design PS mode is left as an open problem.

Two major challenges that one would encounter when designing power-saving protocols are: clock synchronization and the neighbor discovery. Clock synchronization in a multi-hop MANET is difficult since there is no central control and packet delays may vary due to unpredictable mobility and radio interference. PS modes are typically supported by letting low-power hosts wake up only in specific time. Without precise clocks, a host may not be able to know when other PS hosts will wake up to receive packets. Further, a host may not be aware of a PS host at its neighborhood since a PS host will reduce its transmitting and receiving activities. Such incorrect neighbor information may be detrimental to most current routing protocols because the route discovery procedure may incorrectly report that there is no route even when routes actually exist with some PS hosts in the middle. These problems will be discussed in more details in Section 2.

In this paper, we propose three asynchronous power management protocols for multi-hop MANETs, namely *dominating-awake-interval*, *periodically-fully-awake-interval*, and *quorum-based protocols*. We target ourselves at IEEE 802.11-based LAN cards. The basic idea is twofold. First, we enforce PS hosts sending more beacon packets than the original IEEE 802.11 standard does. Second and most importantly, we carefully arrange the wake-up and sleep patterns of PS hosts such that any two neighboring hosts are guaranteed to detect each other in finite time even under PS mode.

Based on our power-saving protocols, we then show how to perform unicast and broadcast in an environment with PS hosts. Simulation results are presented, which show that our protocols can save lots of powers when the traffic load is not high.

The rest of this paper is organized as follows. Preliminaries are given in Section 2. In Section 3, we present our power-saving protocols. Unicast, broadcast and routing protocols based on our power-saving mechanisms are in Section 4. Simulation results are presented in Section 5. Section 6 concludes this paper.

2. Preliminaries

In this section, we start with a general review on power-saving works, followed by detailed design of PS mode in IEEE 802.11. Then we motivate our work by pointing out some problems connecting to PS mode in multi-hop MANETs.

2.1. Reviews of power mode management protocols

Several power management protocols have been proposed for MANET in [15,16,20,21]. The power-aware multi-access protocol with signalling (PAMAS) [15] protocol allows a host to power its radio off when it has no packet to transmit/receive or any of its neighbors is receiving packets, but a separate signalling channel to query neighboring hosts' states is needed. Ref. [16] provides several sleep patterns and allows mobile hosts to select their sleep patterns based on their battery status and quality of service, but a special hardware, called *remote activated switch* (RAS), is required which can receive wakeup signals even when the mobile host has entered a sleep state. A connected-dominated-set-based power-saving protocol is proposed in [20]. Some hosts must serve as *coordinators*, which are chosen according to their remaining battery energies and the numbers of neighbors they can connect to. In the network, only coordinators need to keep awake; other hosts can enter the sleeping mode. Coordinators are responsible of relaying packets for neighboring hosts. With a similar idea, a grid-based energy-saving routing protocol is proposed in [21]. With the help of GPS, the area is partitioned in to small subareas called grids, in each of which only one host needs to remain active to relay packets for other hosts in the same grid.

A page-and-answer protocol is proposed in [18] for wireless networks with base stations. A base station will keep on sending paging messages whenever there are buffered packets. Each mobile host powers up periodically. However, there is no time synchronization between the base station and mobile hosts. On reception of paging messages, mobile hosts return acknowledgements, which will trigger the base station to stop paging and begin transmitting buffered packets. After receiving the buffered packets, mobile hosts return to power-saving mode, and the process repeats. When the system is too heavily loaded, the base station may spend most of its time in transmitting buffered packets, instead of paging messages. This may result in long packet delays for power-saving hosts. A theoretical analysis of [18] is in [22]. Several software power-control issues for portable computers are discussed in [17]. How to combine power management and power control for wireless cards is addressed in [19].

2.2. Power-saving modes in IEEE 802.11

IEEE 802.11 [12] supports two power modes: *active* and *power-saving* (PS) modes. The protocols for infrastructure networks and ad hoc networks are different. Under an infrastructure network, there is an access point (AP) to monitor the mode of each mobile host. A host in the active mode is fully powered and thus may transmit and receive at any time. On the contrary, a host in the PS mode only wakes up periodically to check for possible incoming packets from the AP. A host always notifies its AP when changing modes. Periodically, the AP transmits *beacon frames* spaced by a fixed *beacon interval*. A PS host should monitor these frames. In each beacon frame, a *traffic indication map* (TIM) will be delivered, which contains ID's of those PS hosts with buffered unicast packets in the AP. A PS host, on hearing its ID, should stay awake for the remaining beacon interval. Under the contention period (i.e., DCF), an awake PS host can issue a PS-POLL to the AP to retrieve the buffered packets. While under the contention-free period (i.e., PCF), a PS host will wait for the AP to poll it. Spaced by a fixed number of beacon intervals, the AP will send *delivery TIMs* (DTIMs) within beacon frames to indicate that there are buffered broadcast packets. Immediately after DTIMs, the buffered broadcast packets will be sent.

Under an ad hoc network, PS hosts also wake up periodically. ATIM frames, instead of TIM frames, are sent. The ATIM frame is a subtype of the management frame, which contains frame control, duration, destination address, source address, BSSID, sequence control, and FCS fields. However, the frame body field of the ATIM frame is null. The short interval that PS hosts wake up to transmit/receive ATIM frames to/from other hosts is called the *ATIM window*. It is assumed that hosts are fully connected and all synchronized, so the ATIM windows of all PS hosts will start at about the same time. In the beginning of each ATIM window, each host will contend to send a beacon frame. Any successful beacon serves as the purpose of synchronizing hosts' clocks. This beacon also inhibits other hosts from sending their beacons. To avoid collisions among beacons, a host should wait a random number of slots between 0 and $2 \times \mathrm{CW}_{\min} - 1$ before sending out its beacon.

After the beacon, a host with buffered unicast packets can send a directed ATIM to each of its intended receivers in PS mode during the ATIM window to inform the receivers that there are buffered packets for them. A directed ATIM should be acknowledged immediately. If no acknowledgement is received, the ATIM shall be retransmitted using the DCF access procedure. A station transmitting an ATIM shall remain awake for the entire current beacon interval. If a station has buffered mutlicast frames, it shall transmit an appropriate addressed *multi-cast ATIM*. Multicast ATIM frames shall not be acknowledged. Immediately following the ATIM window, a station shall begin transmitting buffered multi-cast frames. Following the transmission of multi-cast frames, unicast packets can then be sent based on the DCF access procedure. If a mobile host is unable to transmit its ATIM frame in the current ATIM window or has extra buffered packets, it should retransmit ATIMs in the next ATIM window. To protect PS hosts, only RTS, CTS, ACK, Beacon, and ATIM frames can be transmitted during the ATIM window.

Fig. 1 shows an example, where host A wants to transmit a packet to host B. During the ATIM window, an ATIM frame is sent from A to B. In response, B will reply with an ACK. After the ATIM window finishes, A can try to send out its data packet.

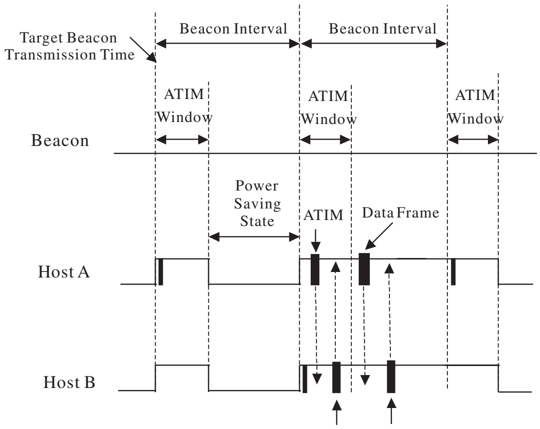


Fig. 1. An example of unicasting in an ad hoc networks with PS hosts.

2.3. Problem statement

The PS mode of IEEE 802.11 is designed for a single-hop (or fully connected) ad hoc network. When applied to a multi-hop ad hoc network, three problems may arise. All these will pose a demand of redesigning the PS mode for multi-hop MANET.

(A) *Clock synchronization:* Since IEEE 802.11 assumes that mobile hosts are fully connected, the transmission of a beacon frame can be used to synchronize all hosts' beacon intervals. So the ATIM windows of all hosts can appear at around the same time without much difficulty. In a multi-hop MANET, clock synchronization is a difficult job because communication delays and mobility are all unpredictable, especially when the network scale is large. Even if perfect clock synchronization is available, two temporarily partitioned subnetworks may independently enter PS mode and thus have different ATIM timing. With the clock-drifting problem, the ATIM windows of different hosts are not guaranteed to be synchronous. Thus, the ATIM window has to be redesigned.

(B) *Neighbor discovery:* In a wireless and mobile environment, a host can only be aware by other hosts if it transmits a signal that is heard by the others. For a host in the PS mode, both its chances to transmit and to hear others' signals are reduced. As reviewed above, a PS host must compete with other hosts to transmit its beacon. A host will cancel its beacon frame once it hears other's beacon frame. This may run into a dilemma that hosts are likely to have inaccurate neighborhood information when there are PS hosts. Thus, many existing routing protocols that depend on neighbor information may be impeded.

(C) *Network partitioning:* The above inaccurate neighbor information may lead to long packet delays or even network-partitioning problem. PS hosts with unsynchronized ATIM windows may wake up at different times and may be partitioned into several groups. These conceptually partitioned groups are actually connected. Thus, many existing routing protocols may fail to work in their route discovery process unless all hosts are awaken at the time of the searching process.

3. Power-saving protocols for MANET

In this section, we present three asynchronous power-saving protocols that allow mobile hosts to enter PS mode in a multi-hop MANET. According to the above discussion, we derive several guidelines in our design:

- *More beacons:* To prevent the inaccurate-neighbor problem, a mobile host in PS mode should insist more on sending beacons. Specifically, a PS host should not inhibit its beacon in the ATIM window even if it has heard others' beacons. This will allow others to be aware of its existence. For this reason, our protocols will allow multiple beacons in an ATIM window.
- *Overlapping awake intervals:* Our protocols do not count on clock synchronization. To resolve this problem, the wake-up patterns of two PS hosts must overlap with each other no matter how much time their clocks drift away.
- *Wake-up prediction:* When a host hears another PS host's beacon, it should be able to derive that PS host's wake-up pattern based on their time difference. This will allow the former to send buffered packets to the later in the future. Note that such prediction is not equal to clock synchronization since the former does not try to adjust its clock.

Based on the above guidelines, we propose three power-saving protocols, each with a different wake-up pattern for PS hosts. PS hosts' wake-up patterns do not need to be synchronous. For each PS host, it divides its time axis into a number of fixed-length intervals called beacon intervals. In each beacon interval, there are three windows called *active window*, *beacon window*, and *MTIM window*. During the active window, the PS host should turn on its receiver to listen to any packet and take proper actions as usual. The beacon window is for the PS host to send its beacon, while the MTIM window is for other hosts to send their MTIM frames to the PS host. Our MTIM frames serve the similar purpose as ATIM frames in IEEE 802.11; here we use MTIM to emphasize that the network is a multi-hop MANET. Excluding these three windows, a PS host with no packet to send or

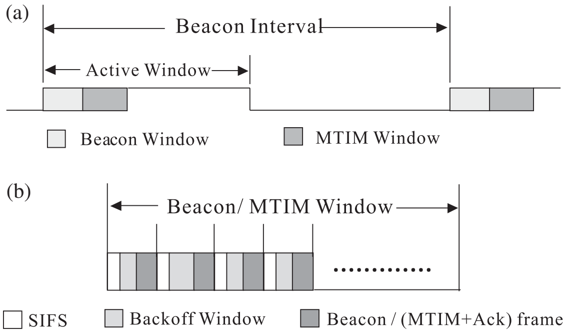


Fig. 2. Structure of a beacon interval: (a) active, beacon, and MTIM windows and (b) access procedure.

receive may go to the sleep mode. Fig. 2(a) shows an example structure of a beacon interval.

The following notations are used throughout this paper:

- *BI:* length of a beacon interval,
- *AW:* length of an active window,
- *BW:* length of a beacon window,
- *MW:* length of an MTIM window, where $\mathrm{MW} > \mathrm{BW}$.

We should comment at this point that the structure of a beacon interval may vary for different protocols (to be elaborated later). The illustration in Fig. 2(a) is only one of the several possibilities. In the beacon window (resp., MTIM window), hosts can send beacons (resp., MTIM frames) following the DCF access procedure. Each transmission must be led by a SIFS followed by a random delay ranging between 0 and $2 \times \mathrm{CW}_{\min} - 1$ slots. This is illustrated in Fig. 2(b).

3.1. Protocol 1: dominating-awake-interval

The basic idea of this approach is to impose a PS host to stay awake sufficiently long so as to ensure that neighboring hosts can know each otherand, if desire, deliver buffered packets. By "dominating-awake", we mean that a PS host should stay awake for at least about half of BI in each beacon interval. This guarantees any PS host's beacon window to overlap with any neighboring PS host's active window, and vice versa.

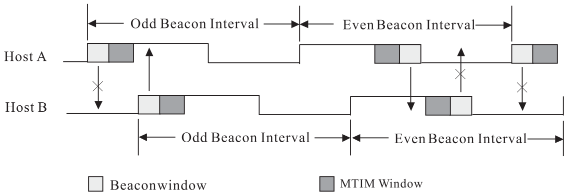


Fig. 3. Structures of odd and even intervals in the dominating-awake-interval protocol.

This protocol is formally derived as follows. When a host decides to enter the PS mode, it divides its time axis into fixed-length beacon intervals, each of length BI. Within each beacon, the lengths of all three windows (i.e., AW, BW, and MW) are constants. To satisfy the "dominating-awake" property, we enforce that $\mathrm{AW} \geqslant \mathrm{BI}/2 + \mathrm{BW}$. The sequence of beacon intervals is alternatively labeled as *odd* and *even* intervals. Odd and even intervals have different structures as defined below (see the illustration in Fig. 3):

- Each odd beacon interval starts with an active window. The active window is led by a beacon window and followed by an MTIM window.
- Each even beacon interval also starts with an active window, but the active window is terminated by an MTIM window followed by a beacon window.

It is not hard to see that by imposing the active window occupying at least half of each beacon interval, we can guarantee that two hosts' active windows always have some overlapping. However, why we have different structures for odd and even beacon intervals remains obscure. Let us consider Fig. 4, where beacon windows always appear at

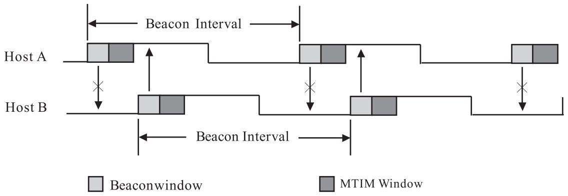


Fig. 4. An example where host B will always miss A's beacons.

the beginning of beacon intervals. In this case, host A can hear host B's beacons, but B always misses A's beacons. On the contrary, as Fig. 3 shows, A can hear B's beacons at odd intervals, and B can hear A's beacons at even intervals.

Earlier we imposed the condition $\mathrm{AW} \geqslant \mathrm{BI}/2 + \mathrm{BW}$. The following theorem provides a formal proof on the correctness of this protocol (proof available in Appendix A).

Theorem 1. *The dominating-awake-interval protocol guarantees that when* $\mathrm{AW} \geqslant \mathrm{BI}/2 + \mathrm{BW}$, *a PS host's entire beacon window always overlaps with any neighboring PS host's active window in every other beacon interval, no matter how much time their clocks drift away.*

The above theory guarantees that a PS host is able to receive all its neighbors' beacon frames in every two beacon intervals, if there is no collision in receiving the latter's beacons. Since the response time for neighbor discovery is pretty short, this protocol is suitable for highly mobile environments.

3.2. Protocol 2: periodically-fully-awake-interval

The previous protocol requires PS hosts keep active more than half of the time, and thus is not energy-efficient. To reduce the active time, in this protocol we design two types of beacon intervals: *low-power intervals* and *fully-awake intervals*. In a low-power interval, the length of the active window is reduced to the minimum, while in a fully-awake interval, the length of the active window is extended to the maximum. Since fully-awake intervals need a lot of powers, they only appear periodically and are interleaved by low-power intervals. So the energy required can be reduced significantly.

Formally, when a host decides to enter the PS mode, it divides its time axis into fixed-length beacon intervals of length BI. The beacon intervals are classified as *low-power* and *fully-awake* intervals. The fully-awake intervals arrive periodically every p intervals, and the rest of the intervals are low-power intervals. The structures of these beacon intervals are defined as follows:

- Each low-power interval starts with an active window, which contains a beacon window followed by an MTIM window, such that $\mathrm{AW} = \mathrm{BW} + \mathrm{MW}$. In the rest of the time, the host can go to the sleep mode.
- Each fully-awake interval also starts with a beacon window followed by an MTIM window. However, the host must remain awake in the rest of the time, i.e., $\mathrm{AW} = \mathrm{BI}$.

Intuitively, the low-power intervals are for a PS host to send out its beacons to inform others about its existence. The fully-awake intervals are for a PS host to discover who are in its neighborhood. It is not hard to see that a fully-awake interval always has overlapping with any host's beacon windows, no matter how much time their clocks drift away. By collecting other hosts' beacons, the host can predict when its neighboring hosts will wake up. Fig. 5 shows an example with $p = 4$ intervals. So hosts A's and B's beacons always have chances to reach the other's active windows. Proof of the following theorem is in Appendix B.

Theorem 2. *The periodically-fully-awake-interval protocol guarantees that a PS host's beacon windows overlap with any neighbor's fully-awake intervals in every p beacon intervals, no matter how much time their clocks drift away.*

Compared to the previous dominating-awake-interval protocol, which requires a PS host to stay awake more than half of the time, this protocol can save more power as long as $p > 2$. However, the response time to get aware of a newly appearing host could be as long as p beacon intervals. So this protocol is more appropriate for slowly

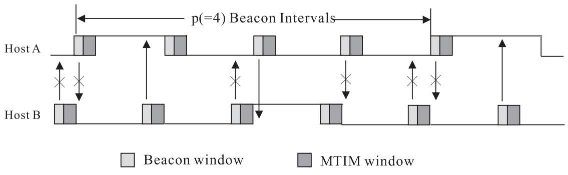


Fig. 5. An example of the periodically-fully-awake-interval protocol with fully-awake intervals arrive every $p = 4$ beacon intervals.

mobile environments. One way to reduce the response time is to decrease the value of p to fit one's need.

3.3. Protocol 3: quorum-based

In the previous two protocols, a PS host has to contend to send a beacon in each beacon interval. In this section, we propose a protocol based on the concept of *quorum*, where a PS host only needs to send beacons in $O(1/n)$ of the all beacon intervals. Thus, when transmission takes more powers than reception, this protocol may be more energy-efficient. The concept of quorums has been used widely in distributed system design (e.g., to guarantee mutual exclusion [23–26]). A quorum is a set of identities from which one has to obtain permission to perform some action [23]. Typically, two quorum sets always have non-empty intersection so as to guarantee the atomicity of a transaction. Here we adopt the concept of quorum to design PS hosts' wakeup patterns so as to guarantee a PS host's beacons can always be heard by others' active windows. This is why our protocol is named so.

The quorum structure of our protocol is as follows. The sequence of beacon intervals is divided into sets starting from the first interval such that each continuous n^2 beacon intervals are called a *group*, where n is a global parameter. In each group, the n^2 intervals are arranged as a 2-dimensional $n \times n$ array in a row-major manner. On the $n \times n$ array, a host can arbitrarily pick one column and one row of entries and these $2n - 1$ intervals are called *quorum intervals*. The remaining $n^2 - 2n + 1$ intervals are called *non-quorum intervals*.

Before proceeding, let us make some observation from the quorum structure. Given two PS hosts that are perfectly time-synchronized, it is not hard to see that their quorum intervals always have at least two intersecting beacon intervals (see the illustration in Fig. 6(a)). This is due to the fact that a column and a row in a matrix always have an intersection. Thus, two PS host may hear each other on the intersecting intervals. However, the above reasoning is not completely true since we do not assume that hosts are time-synchronized. For example, in Fig. 6(b) and (c), host A selects intervals on row 0 and column 1 as its quorum intervals from a 4×4 matrix, while host B selects intervals on row 2 and column 2 as its quorum intervals. When perfectly synchronized, intervals 2 and 9 are the intersections.

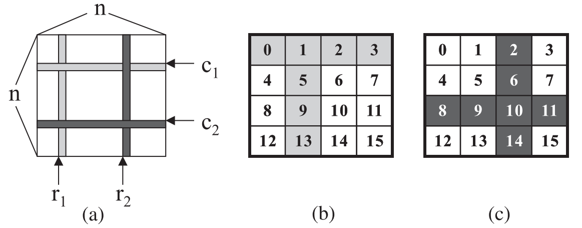


Fig. 6. Examples of the quorum-based protocol: (a) intersections of two PS hosts' quorum intervals, (b) host A's quorum intervals, and (c) host B's quorum intervals.

The structures of quorum and non-quorum intervals are formally defined below:

- Each quorum interval starts with a beacon window followed by an MTIM window. After that, the host must remain awake for the rest of the interval, i.e., AW = BI.
- Each non-quorum interval starts with an MTIM window. After that, the host may go to the sleep mode, i.e., we let AW = MW.

The protocol guarantees the following property (proof in Appendix C).

Theorem 3. *The quorum-based protocol guarantees that a PS host always has at least two entire beacon windows that are fully covered by another PS host's active windows in every n^2 beacon intervals.*

The quorum-based protocol has advantage in that it only transmits in $O(1/n)$ of the beacon intervals (on the contrary, the earlier two protocols have to transmit a beacon in every interval). In addition, it also keeps awake in $O(1/n)$ of the time. As long as $n \geqslant 4$, this amount of awaking time is less than 50%. So this protocol is more energy-efficient when transmission cost is high. The backside is that a PS host may learn its vicinity at lower speed and that it might be more expensive to implement.

Table 1
Characteristics of the proposed power-saving protocols

Protocol	Number of beacons	Active ratio	Neighbor sensitivity
Dominated-awake	1	$1/2 + \mathrm{BW}/\mathrm{BI}$	BI
Periodically-fully-awake	1	$1/p$	$p \times \mathrm{BI}/2$
Quorum-based	$(2n-1)/n^2$	$(2n-1)/n^2$	$(n^2/4) \times \mathrm{BI}$

3.4. Summary

Table 1 summarizes the characteristics of the three proposed power-saving protocols. "Number of beacon" indicates the average number of beacons that a host need to transmit in each beacon interval, "Active ratio" indicates the ratio of time that a PS host needs to stay awake while in the PS mode, and "neighbor sensitivity" indicates the average time that a PS host takes to hear a newly approaching neighbor's beacon. As Table 1 shows, the quorum-based protocol spends least power in transmitting beacons. The periodically-fully-awake-interval and the quorum-based protocols' active ratios can be quite small as long as p and n, respectively are large enough. The dominated-awake-interval protocol is most sensitive to neighbor changes, while the quorum-based protocol is least sensitive.

4. Unicast and broadcast protocols for power-saving hosts

This section discusses how a host sends packets to a neighboring PS host. Since the PS host is not always active, the sending host has to predict when the PS host will wake up, i.e., when the latter's MTIM windows will arrive. To achieve this, each beacon packet has to carry the clock value of the sending host so that other hosts can calculate their time differences. Table 2 summarizes when MTIM windows arrive in the proposed protocols, where m is any non-negative integer.

Since all the hosts in the MANET adopt the same power-saving protocol, the patterns of their MTIM windows are similar, except that their clocks might be different. Consider any two asynchronous mobile hosts A and B. Without loss of generality, let A's clock be faster than B's clock by ΔT μs. Whenever host A's MTIM window arrives, host A can predict that host B's MTIM window will arrive ΔT μs later; host B can predict that host A's MTIM window arrived ΔT μs before its own MTIM window arrives. If the quorum-based protocol is adopted, a host first predicts when another host's quorum interval starts and then can also predict the latter's MTIM windows in a similar manner.

After correctly predicting the receiving side's MTIM windows, the sending side can contend to send MTIM packets to notify the receiver during the receiver's MTIM window, after which the buffered data packet can be sent. Below, we discuss how unicast and broadcast are achieved.

4.1. Unicast

This is similar to the procedure in IEEE802.11's PS mode. During the receiver's MTIM window, the sender contends to send its MTIM packet to

Table 2
Timing of MTIM windows of the proposed protocols

Protocol	MTIM window's timing
Dominated-awake	$[(2m+1) \times \mathrm{BI} + \mathrm{BW}, (2m+1) \times \mathrm{BI} + \mathrm{BW} + \mathrm{MW}]$ (odd int.) $[2m \times \mathrm{BI} + \mathrm{BI}/2 - \mathrm{MW}, 2m \times \mathrm{BI} + \mathrm{BI}/2]$ (even int.)
Periodically-fully-awake	$[m \times \mathrm{BI} + \mathrm{BW}, m \times \mathrm{BI} + \mathrm{BW} + \mathrm{MW}]$
Quorum-based	$[m \times \mathrm{BI} + \mathrm{BW}, m \times \mathrm{BI} + \mathrm{BW} + \mathrm{MW}]$ (quorum int.) $[m \times \mathrm{BI}, m \times \mathrm{BI} + \mathrm{MW}]$ (non-quorum int.)

the receiver. The receiver, on receiving the MTIM packet, will reply an ACK after an SIFS and stay awake in the remaining of the beacon interval. After the MTIM window, the sender will contend to send the buffered packet to the receiver based on the DCF procedure.

4.2. Broadcast

The situation is more complicated for broadcasting since the sender may have to deal with multiple asynchronous neighbors. To reduce the number of transmissions, we need to divide these asynchronous neighbors into groups and notify them separately in multiple runs. The steps are described below. Note that here the broadcast is not designed to be 100% reliable at the MAC layer (reliable broadcast may be supported at a higher layer).

When a source host S intends to broadcast a packet, it first checks the arrival time of the MTIM windows of all its neighbors. Then S picks the host, say Y, whose first MTIM window arrives earliest. Based on Y's first MTIM window, S further picks those neighbors whose MTIM windows have overlapping with Y's first MTIM window. These hosts, including Y, are groups together and S will try to notify them in one MTIM frame (note that such MTIM frames need not be acknowledged due to the unreliable assumption). After notifying these neighbors, the source S can contend to send its buffered broadcast packet to this group. Broadcast packets should be sent based on the DCF procedure too. After this transmission, S considers the rest of the neighbors that have not been notified yet in the previous MTIM and repeats the same procedure again to initiate another MTIM frame and broadcast packet. The process is repeated until all its neighbors have been notified.

A neighbor, on receiving an MTIM carrying a broadcast indication, should remain awake until a broadcast packet is received or a timeout value expires (here we recommend a timeout value equal to one beacon interval be used, but this can also be an adjustable parameter during system configuration).

We comment that most traditional on-demand routing protocols, such as DSR [27] and AODV [28], rely on broadcasting route request packets to discover new routes. If hosts' clocks are not well synchronized, some PS hosts may miss the request packets. With our broadcasting protocol, the route request packets can be received, though with some delays. At the destination side, unicast can be used to send the route reply packets. When PS hosts in the chosen route receive the route reply packet, it can go to the active mode.

5. Simulation experiments

To evaluate the performance of the proposed power-saving protocols, we have developed a simulator using C. In the simulations, we assume that the area size is 1000 m × 1000 m, and the transmission radius is 250 m (under such an environment, the average distance between hosts is around 3.2 hops). Hosts' transmission rate is 2 Mbits/s, and the battery power of each mobile host is 100 J. The MAC part basically follows the IEEE 802.11 standard [12], except the power management part. We adopt AODV as our routing protocol. The source and the destination of each route are randomly selected. Four parameters are tunable in our simulations:

- *Traffic load*: Routes are generated by a Poisson distribution with rate between 1 and 4 routs/s. For each route, 10 packets, each of size 1 kbytes, will be sent.
- *Mobility*: Host mobility follows the random way-point model. The pause time is set to 20 s. When moving, a host will move at a speed between 0 and 20 m/s.
- *Beacon interval*: The length of one beacon interval is 100–400 ms.
- *Number of hosts*: The total number of mobile hosts in the MANET is 50–200 hosts.

Basically, each simulation lasts for 100 s. However, when measuring the survival ratio of hosts in the MANET, the simulation will last until all the hosts have run out of energies. Each result is obtained from the average of 100 simulation runs. The confidence level shown in the figures is at 95% with the confidence interval of $(\overline{X} - 1.96\sigma/10,$

$\overline{X} - 1.96\sigma/10)$, where $\overline{X}$ is the mean and σ is the standard deviation of the samples. For simplicity, we assume that all hosts are in the PS mode. To make comparison, we also simulate an "always-active" scheme in which all hosts are active all the time.

Four performance metrics are used in the simulations:

- *Neighbor discovery time*: Average time to discover a newly approaching neighbor.
- *Survival ratio*: The number of surviving hosts over the total number of hosts. (A host is said to be surviving if its power is not exhausted yet.)
- *Route establishment probability*: The total number of successfully established routes over the total number of requests.
- *Route request/reply delay*: The time from the source host initiating the route request packet to the time the destination host receiving the packet and the time the reply is sent from the destination to the source.

The data packet transmission delay is not observed in our simulation, because when a route is established, all the hosts in the route should be switched to the active mode and thus the data packet transmission delay of our schemes should be similar to that of the "always-active" scheme. The power model in [29] is adopted, which is obtained by real experiments on Lucent WaveLAN cards. Table 3 summarizes the power consumption parameters used in our simulations. Sending/receiving a unicast/broadcast packet has a cost $P_{\text{base}} + P_{\text{byte}} \times L$, where P_{base} is the power consumption independent of packet length, P_{byte} is the power consumption per byte, and L is the packet length. When sending a packet of the same size, unicast consumes more power than broadcast because it needs to send and receive extra control frames (*RTS*, *CTS*, and *ACK*). The last two entries indicate the consumption when a host has no send/receive activity and is in the active mode and PS mode, respectively. As can be seen, staying in the active mode is much more energy-consuming. The traffic-related parameters are summarized in Table 4.

Table 3
Power consumption parameters used in the simulation

Unicast send	$454 + 1.9 \times L$ μJ/packet
Broadcast send	$266 + 1.9 \times L$ μJ/packet
Unicast receive	$356 + 0.5 \times L$ μJ/packet
Broadcast receive	$56 + 0.5 \times L$ μJ/packet
Idle	843 μJ/ms
Doze	27 μJ/ms

Table 4
Traffic-related parameters used in the simulation

Unicast packet size	1024 bytes
Broadcast packet size	32 bytes
Beacon window size	4 ms
MTIM window size	16 ms

In the following subsections, we show how beacon interval, mobility, traffic load, and host density affect the performance of the proposed power-saving protocols. For simplicity, the dominating-awake-interval protocol is denoted as D, the periodically-fully-awake-interval protocol with parameter p is denoted as P(p), the quorum-based protocol with parameter n is denoted as Q(n), and the "always active" scheme is denoted as AA.

5.1. Impact of beacon interval length

The length of beacon intervals has impact on hosts' sensitivity to environmental changes, power consumptions, route establishment probabilities, and route request/reply delays. However, these are contradicting factors. To observe its impact, we vary the beacon interval length between 100 and 400 ms. As Fig. 7 shows, longer beacon intervals only slightly increase the neighbor discovery time for schemes D and $P(5)$, but have more significant impact on schemes $Q(5)$. Overall, scheme D has the shortest neighbor discovery time, which is subsequently followed by $P(5)$ and $Q(5)$.

Fig. 8 shows that longer beacon intervals may lengthen the lifetime of the MANET, because the ratio of awake time for each host becomes smaller. However, longer beacon intervals may increase the broadcasting (and thus route discovery) cost. This will be shown in the next requirement. Overall, scheme $P(5)$ has the longest network lifetime, which is subsequently followed by $Q(5)$ and D.

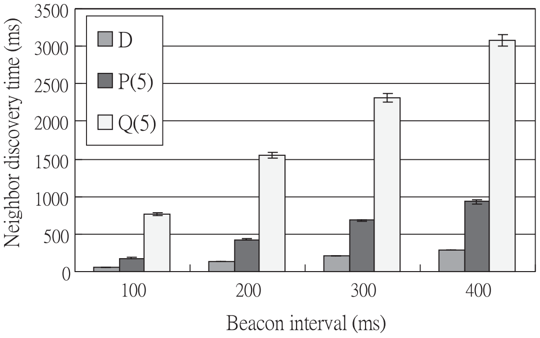


Fig. 7. Neighbor discovery time vs. beacon interval length (100 hosts, traffic load = 1 route/s, mobility = 5 m/s).

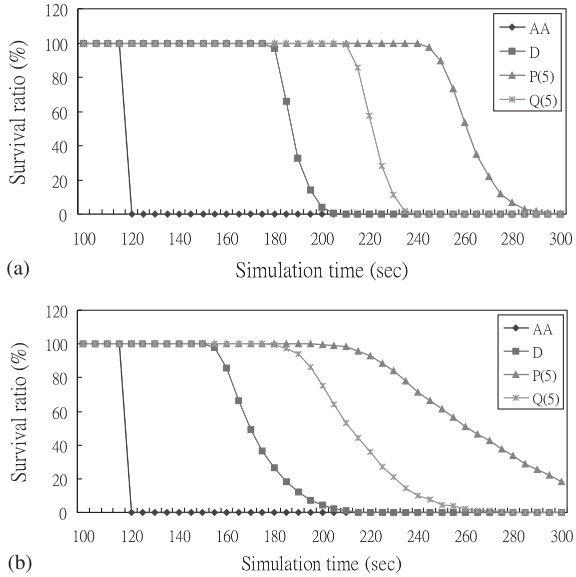


Fig. 8. Host survival ratio vs. beacon interval length: (a) beacon interval = 100 ms, and (b) beacon interval = 400 ms (100 hosts, traffic load = 1 route/s, moving speed = 5 m/s).

Fig. 9 shows the impact of beacon interval length over route request/reply delays. With longer beacon interval, it takes longer time to wake up hosts. For example, in the route discovery procedure, a host may have to send broadcasts to multiple groups of neighbors. The number of groups usually increases as the interval increases. As Fig. 9 shows, *Q*(5) has the least delays in route request and reply because it transmits less beacons and thus causes less contentions and collisions. Scheme *D* also has very low delay. Scheme *P*(5) has the highest delays in both route request and reply.

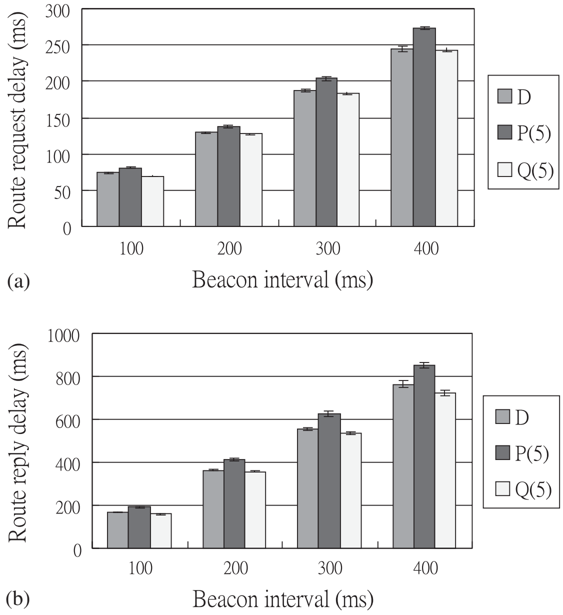


Fig. 9. Route request/reply delay vs. beacon interval length: (a) route request delay, and (b) route reply delay (100 hosts, traffic load = 1 route/s, moving speed = 5 m/s).

Fig. 10 shows the impact of beacon interval length on route establishment probability. Longer beacon intervals do decrease the probability. This is because longer beacon intervals will increase the time to deliver the route request packets to the destination, and unfortunately, at the time when the route reply packets are issued, the desired route may have become broken. In terms of route establishment probability, the differences between the three schemes are insignificant.

5.2. Impact of mobility

Mobility has a negative impact on survival ratio, route establishment probability, and route request/reply delays. To observe its effect, we vary hosts' moving speed between 0 and 20 m/s.

Fig. 11 shows the impact of mobility on survival ratio. Mobility will incur high energy consumption because hosts may spend more power on retransmitting packets. Among the four schemes, mobil-

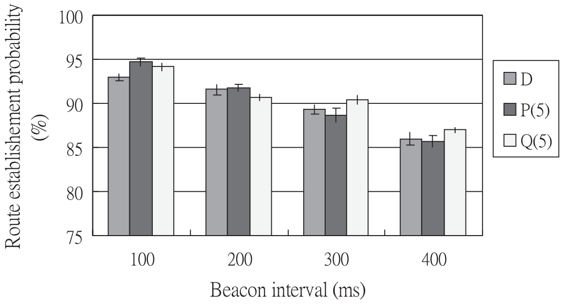


Fig. 10. Route establishment probability vs. beacon interval length (100 hosts, traffic load = 1 route/s, moving speed = 5 m/s).

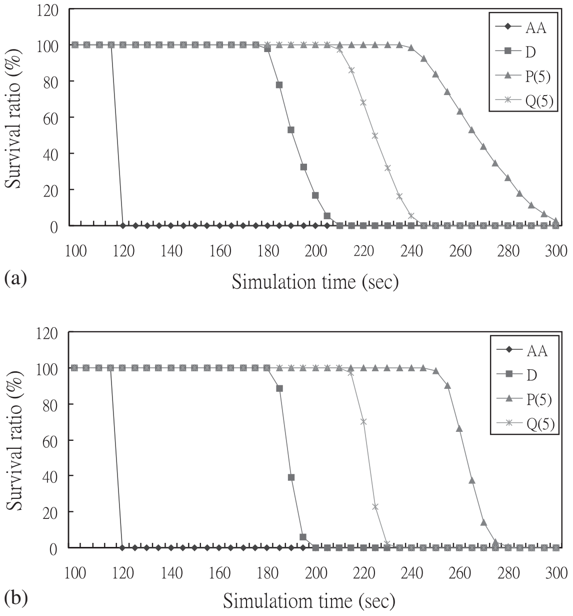


Fig. 11. Survival ratio vs. mobility: (a) moving speed = 0 m/s, and (b) moving speed = 20 m/s (beacon interval = 100 ms, 100 hosts, traffic load = 1 route/s).

ity has the least impact on AA because it takes the least extra power to retransmit unicast packets.

Fig. 12 shows that as the moving speed increases the route request/reply delays also increase. However, the impact is insignificant. In terms of transmission delay, scheme AA performs the best, which is subsequently followed by *Q*(5), *D* and then *P*(5). Scheme AA performs the best because the sender needs not to wait for the receiver to become active.

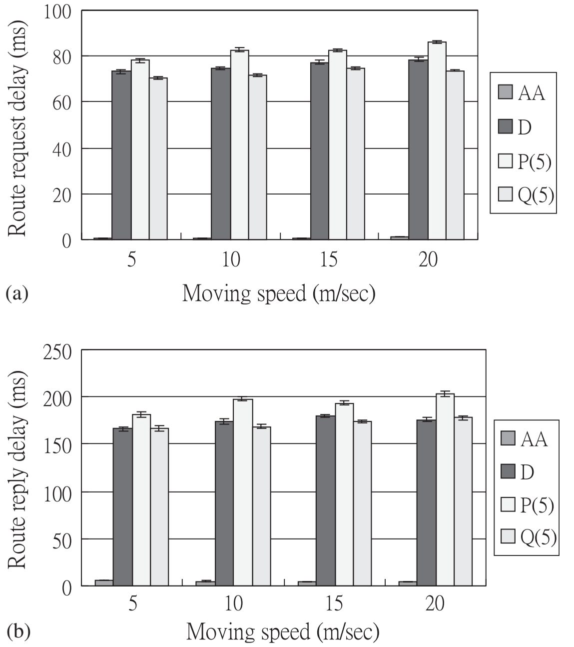


Fig. 12. Route request/reply delay vs. mobility: (a) route request delay, and (b) route reply delay (beacon interval = 100 ms, 100 hosts, traffic load = 1 route/s).

Fig. 13 shows the impact of mobility on the route establishment probability. Among the four schemes, the AA scheme performs the best and the *D* scheme performs the worst. *P*(5) and *Q*(5) are close to each other. Recall that the *D* scheme has the most accurate neighbor list. This simulation result indicates that the accuracy of neighbor list is not so important for the route establishment

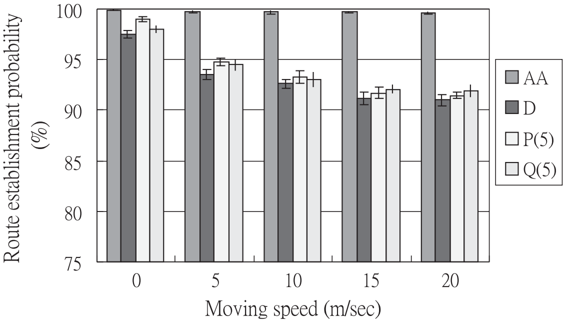


Fig. 13. Route establishment probability rate vs. mobility (beacon interval = 100 ms, 100 hosts, traffic load = 1 route/s).

probability because a host missing a route request may hear the same request from other neighbors, and route request is typically done by flooding.

5.3. Impact of traffic load

Traffic load also has a negative effect to survival ratio, route establishment probability, and route request/reply delays. To observe its impact, we vary the traffic load between 1 and 4 routes/s.

As Fig. 14 shows, when the traffic load becomes higher, the route establishment probability becomes lower because heavier traffic will cause more collisions and congestions. Among the four schemes, the AA scheme performs the best and the *Q*(5) scheme performs the worst. *D* and *P*(5) are close to each other. Fig. 15 shows the impact of traffic load on network lifetime. It is reasonable to see that higher traffic load will reduce the network lifetime.

5.4. Impact of host density

In this experiment, we fix the network size and vary the total number of hosts between 50 and 200. The impact on route establishment probability is in Fig. 16. We see that when the host density becomes higher, the route establishment probability somehow gets hurt. In a denser network, collision and congestion may become reasons that cause route establishment failure. Overall, AA performs the best and the *Q*(5) scheme performs the worst. *D* and *P*(5) are close to each other.

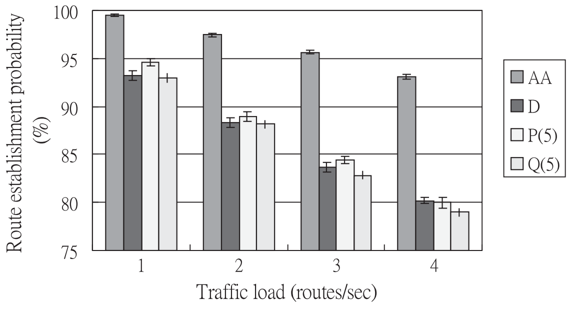


Fig. 14. Route establishment probability vs. traffic load (beacon interval = 100 ms, 100 hosts, mobility = 5 m/s).

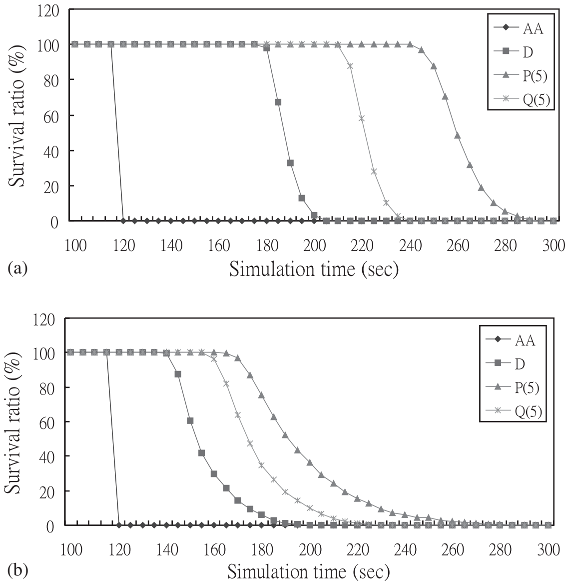


Fig. 15. Survival ratio vs. traffic load: (a) traffic load = 1 route/s, and (b) traffic load = 4 routes/s (beacon interval = 100 ms, 100 hosts, mobility = 5 m/s).

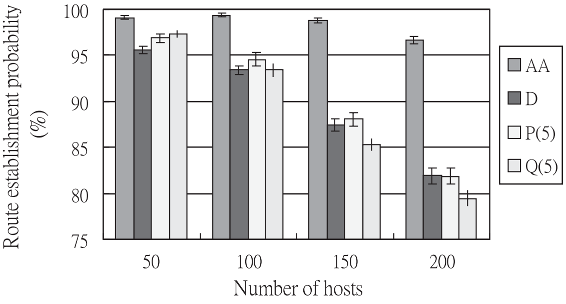
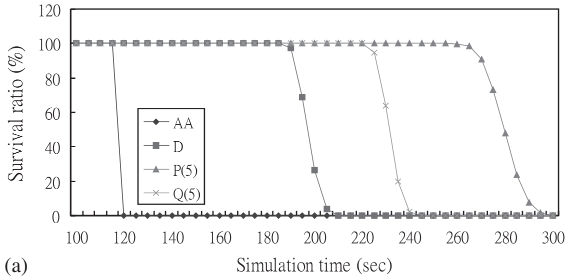


Fig. 16. Route establishment probability vs. host density (beacon interval = 100 ms, traffic load = 1 route/s, moving speed = 5 m/s).

Fig. 17 shows that a higher node density will bring down the network lifetime. We note that although the traffic load is the same, the broadcast cost to discover routes will become higher. When a route request is issued, not only more hosts will help searching for routes, but also the broadcast cost in each individual host will become higher as the network is denser.



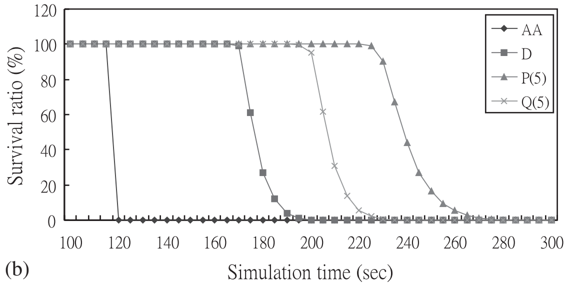


Fig. 17. Survival ratio vs. node density: (a) 50 hosts, and (b) 200 hosts (beacon interval = 100 ms, traffic load 1 route/s, mobility = 5 m/s).

6. Conclusions

In this paper, we have addressed the power management problem in a MANET, which is characterized by unpredictable mobility, multi-hop communication, and no clock synchronization. We have pointed out two important issues, the *neighbor discovery* problem and the *network-partitioning* problem, which may occur if one directly adopts the PS mode defined in the IEEE 802.11 protocol. As far as we know, the power-saving issues, particularly for multi-hop MANETs, have not been addressed seriously in the literature. In this paper, we have proposed three power-saving protocols for IEEE 802.11-based, multi-hop, unsynchronized MANETs. Simulation results have shown that our power-saving protocols can save lots of energies with reasonable route establishment probability. Among the three proposed protocols, the dominating-awake-interval protocol is most energy-consuming but has the shortest neighbor discovery time, while the periodical-fully-awake-interval protocol is most energy-saving but has the longer route discovery delays. The quorum-based protocol consumes more energies than the periodical-fully-awake-interval protocol, but it transmits fewer beacon frames than the other two protocols and has shorter route discovery delays. We believe that the proposed protocols can be applied to current IEEE 802.11 wireless LAN cards easily with little modification.

Acknowledgements

This work is co-sponsored by the MOE Program for Promoting Academic Excellence of Universities under grant numbers A-91-H-FA07-1-4 and 89-E-FA04-1-4.

Appendix A. Proof of Theorem 1

It suffices to prove the theorem under the condition $\mathrm{AW} = \mathrm{BI}/2 + \mathrm{BW}$. Consider any two asynchronous mobile hosts A and B. Without loss of generality, let A's clock be faster than B's clock by $\Delta T = k \times \mathrm{BI} + \Delta t$, where $0 \leqslant \Delta t < \mathrm{BI}$ and $k \geqslant 0$ is an integer. In the following derivation, let us use A's clock as a reference to derive B's clock. Assuming that n and m are non-negative integers, we can derive the following timings for A and B.

- A's active windows: $[n \times \mathrm{BI}, n \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW}]$.
- A's beacon windows in odd intervals: $[(2m+1) \times \mathrm{BI}, (2m+1) \times \mathrm{BI} + \mathrm{BW}]$.
- A's beacon windows in even intervals: $[2m \times \mathrm{BI} + \mathrm{BI}/2, 2m \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW}]$.
- B's active windows: $[n \times \mathrm{BI} + \Delta T, n \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} + \Delta T]$.
- B's beacon windows in odd intervals: $[(2m+1) \times \mathrm{BI} + \Delta T, (2m+1) \times \mathrm{BI} + \mathrm{BW} + \Delta T]$.
- B's beacon windows in even intervals: $[2m \times \mathrm{BI} + \mathrm{BI}/2 + \Delta T, 2m \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} + \Delta T]$.

Below, we prove that for host A, in every pair (odd and even) of beacon intervals, there is at least one entire beacon window overlapping with host B's active window, and vice versa. This is done by showing that the former's beacon window will start later than the later's active window, and terminates earlier than the later's active window. We separate from two cases.

- *Case* 1: $0 \leqslant \Delta t \leqslant \mathrm{BI}/2$
  1. Consider B's odd ($(2m+1)$th) interval. We prove that B's beacon window will be covered by A's $(2m+1+k)$th active window. The following derivation shows that B's beacon window starts later than A's active window:

     $$\begin{aligned}(2m+1+k) \times \mathrm{BI} &\leqslant (2m+1+k) \times \mathrm{BI} + \Delta t \\ &= (2m+1) \times \mathrm{BI} + k \times \mathrm{BI} + \Delta t \\ &= (2m+1) \times \mathrm{BI} + \Delta T,\end{aligned}$$

     and the following shows that B's beacon window ends earlier than A's active window:

     $$\begin{aligned}(2m+1) \times \mathrm{BI} + \mathrm{BW} + \Delta T &= (2m+k+1) \times \mathrm{BI} + \mathrm{BW} + \Delta t \\ &\leqslant (2m+k+1) \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW}.\end{aligned}$$

  2. Consider A's even ($(2m)$th) interval. We prove that A's beacon window will be covered by B's $(2m-k)$th active window. The following derivation shows that A's beacon window starts later than B's active window:

     $$\begin{aligned}(2m-k) \times \mathrm{BI} + \Delta T &= (2m-k) \times \mathrm{BI} + k \times \mathrm{BI} + \Delta t \\ &\leqslant 2m \times \mathrm{BI} + \mathrm{BI}/2,\end{aligned}$$

     and the following shows that A's beacon window ends earlier than B's active window:

     $$\begin{aligned}2m \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} &\leqslant 2m \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} + \Delta t \\ &= (2m-k) \times \mathrm{BI} + \mathrm{BI}/2 \\ &\quad + \mathrm{BW} + k \times \mathrm{BI} + \Delta t \\ &= (2m-k) \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} + \Delta T.\end{aligned}$$

- *Case* 2: $\mathrm{BI}/2 < \Delta t < \mathrm{BI}$, assume that $\Delta t = \mathrm{BI}/2 + \Delta d, 0 < \Delta d < \mathrm{BI}/2$
  1. Consider B's even ($(2m)$th) interval. We prove that B's beacon window will be covered by A's $(2m+1+k)$th active window. The following derivation shows that B's beacon window starts later than A's active window:

     $$\begin{aligned}(2m+1+k) \times \mathrm{BI} &< (2m+1+k) \times \mathrm{BI} + \Delta d \\ &= 2m \times \mathrm{BI} + \mathrm{BI}/2 + k \times \mathrm{BI} \\ &\quad + (\mathrm{BI}/2 + \Delta d) \\ &= 2m \times \mathrm{BI} + \mathrm{BI}/2 + \Delta T,\end{aligned}$$

     and the following shows that B's beacon window ends earlier than A's active window:

     $$\begin{aligned}2m \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} + \Delta T &= 2m \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} + k \times \mathrm{BI} \\ &\quad + (\mathrm{BI}/2 + \Delta d) \\ &= (2m+1+k) \times \mathrm{BI} + \mathrm{BW} + \Delta d \\ &< (2m+1+k) \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW}.\end{aligned}$$

  2. Consider A's odd ($(2m+1)$th) interval. We prove that A's beacon window will be covered by B's $(2m-k)$th active window. The following derivation shows that A's beacon window starts later than B's active window:

     $$\begin{aligned}(2m-k) \times \mathrm{BI} + \Delta T &= (2m-k) \times \mathrm{BI} + k \times \mathrm{BI} + \Delta t \\ &< (2m+1) \times \mathrm{BI},\end{aligned}$$

     and the following shows that A's beacon window ends earlier than B's active window:

     $$\begin{aligned}(2m+1) \times \mathrm{BI} + \mathrm{BW} &< 2m \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} + (\mathrm{BI}/2 + \Delta d) \\ &= (2m-k) \times \mathrm{BI} + \mathrm{BI}/2 \\ &\quad + \mathrm{BW} + k \times \mathrm{BI} + (\mathrm{BI}/2 + \Delta d) \\ &= (2m-k) \times \mathrm{BI} + \mathrm{BI}/2 + \mathrm{BW} + \Delta T.\end{aligned}$$

Appendix B. Proof of Theorem 2

Consider any two asynchronous mobile hosts A and B. Without loss of generality, let A's clock be faster than B's clock by $\Delta T = k \times \mathrm{BI} + \Delta t$, where

$0 \leqslant \Delta t < \mathrm{BI}$ and k is a non-negative integer. In the following derivation, let us use A's clock as a reference to derive B's clock. Assuming that n is a non-negative integer and the fully-awake intervals arrive periodically every p intervals, we can derive the following timings for A and B.

- A's active windows: $[n \times p \times \mathrm{BI}, (n \times p + 1) \times \mathrm{BI} + \mathrm{BW} + \mathrm{MW}]$.
- A's beacon windows: $[n \times \mathrm{BI}, n \times \mathrm{BI} + \mathrm{BW}]$.
- B's active windows: $[n \times p \times \mathrm{BI} + \Delta T, (n \times p + 1) \times \mathrm{BI} + \mathrm{BW} + \mathrm{MW} + \Delta T]$.
- B's beacon windows: $[n \times \mathrm{BI} + \Delta T, n \times \mathrm{BI} + \mathrm{BW} + \Delta T]$.

Below, we prove that for host A, in every p beacon intervals, there is at least one entire beacon window overlapping with host B's active window, and vice versa. This is done by showing that the former's beacon window will start later than the later's active window, and terminates earlier than the later's active window.

- Consider B's $(n \times p - k)$th interval. We prove that B's beacon window will be covered by A's $(n \times p)$th active window. The following derivation shows that B's beacon window starts later than A's active window:

  $$\begin{aligned} &n \times p \times \mathrm{BI} \\ &\quad \leqslant (n \times p - k) \times \mathrm{BI} + k \times \mathrm{BI} + \Delta t \\ &\quad = (n \times p - k) \times \mathrm{BI} + \Delta T, \end{aligned}$$

  and the following shows that B's beacon window ends earlier than A's active window:

  $$\begin{aligned} &(n \times p - k) \times \mathrm{BI} + \mathrm{BW} + \Delta T \\ &\quad = n \times p \times \mathrm{BI} + \mathrm{BW} + \Delta t \\ &\quad < (n \times p + 1) \times \mathrm{BI} + \mathrm{BW} + \mathrm{MW}. \end{aligned}$$

- Consider A's $(n \times p + k + 1)$th interval. We prove that A's beacon window will be covered by B's $(n \times p)$th active window. The following derivation shows that A's beacon window starts later than B's active window:

  $$\begin{aligned} n \times p \times \mathrm{BI} + \Delta T &= (n \times p + k) \times \mathrm{BI} + \Delta t \\ &< (n \times p + k + 1) \times \mathrm{BI}, \end{aligned}$$

  and the following shows that A's beacon window ends earlier than B's active window:

  $$\begin{aligned} &(n \times p + k + 1) \times \mathrm{BI} + \mathrm{BW} \\ &\quad < (n \times p + 1) \times \mathrm{BI} + \mathrm{BW} + \mathrm{MW} + k \times \mathrm{BI} \\ &\qquad + \Delta t \\ &\quad = (n \times p + 1) \times \mathrm{BI} + \mathrm{BW} + \mathrm{MW} + \Delta T. \end{aligned}$$

Appendix C. Proof of Theorem 3

Consider any two asynchronous mobile hosts A and B. Without loss of generality, let A's clock be faster than B's clock by $\Delta T = k \times \mathrm{BI} + \Delta t$, where $0 \leqslant \Delta t < \mathrm{BI}$ and $k \geqslant 0$ is an integer.

Assume that A picks row r_1 and column c_1 as its quorum intervals, while B picks row r_2 and column c_2 as its quorum intervals, where $0 \leqslant r_1, c_1, r_2, c_2 < n$.

In the following derivation, let us use A's clock as a reference to derive B's clock. Assume that a, x, and y are non-negative integers, $0 \leqslant x, y < n$. We can derive the following timings for A and B.

- A's active window in its chosen row: $[(a \times n^2 + r_1 \times n) \times \mathrm{BI}, (a \times n^2 + (r_1 + 1) \times n) \times \mathrm{BI} + \mathrm{MW}]$.
- A's active window in its chosen column: $[(a \times n^2 + y \times n + c_1) \times \mathrm{BI}, (a \times n^2 + y \times n + c_1 + 1) \times \mathrm{BI} + \mathrm{MW}]$.
- A's beacon window in its chosen row: $[(a \times n^2 + r_1 \times n + x) \times \mathrm{BI}, (a \times n^2 + r_1 \times n + x) \times \mathrm{BI} + \mathrm{BW}]$.
- A's beacon window in its chosen column: $[(a \times n^2 + y \times n + c_1) \times \mathrm{BI}, (a \times n^2 + y \times n + c_1) \times \mathrm{BI} + \mathrm{BW}]$.
- B's active window in its chosen row: $[(a \times n^2 + r_2 \times n) \times \mathrm{BI} + \Delta T, (a \times n^2 + (r_2 + 1) \times n) \times \mathrm{BI} + \mathrm{MW} + \Delta T]$.
- B's active window in its chosen column: $[(a \times n^2 + y \times n + c_2) \times \mathrm{BI} + \Delta T, (a \times n^2 + y \times n + c_2 + 1) \times \mathrm{BI} + \mathrm{MW} + \Delta T]$.
- B's beacon window in its chosen row: $[(a \times n^2 + r_2 \times n + x) \times \mathrm{BI} + \Delta T, (a \times n^2 + r_2 \times n + x) \times \mathrm{BI} + \mathrm{BW} + \Delta T]$.

- B's beacon window in its chosen column: $[(a^2 + y \times n + c_2) \times \mathrm{BI} + \Delta T, (a \times n^2 + y \times n + c_2) \times \mathrm{BI} + \mathrm{BW} + \Delta T]$.

Below, we prove that for host A, in every n^2 beacon intervals, there are at least two beacon windows overlapping with host B's active window, and vice versa. This is done by showing that two of the former's beacon windows will start later than the later's active windows, and terminate earlier than the later's active windows. Assume that $c_1 - k = -p \times n + u, c_2 + k = q \times n + v$, where p, q, u, and v are non-negative integers, $0 \leqslant v$, $u < n$.

- We prove that one of B's beacon window in its chosen column will be covered by A's active window in its chosen row.

Consider B's $(a \times n^2 + (r_1 - q) \times n + c_2)$th interval in its chosen column. We prove that B's beacon window will be covered by A's active window in its chosen row. The following derivation shows that B's beacon window starts later than A's active window:

$$\begin{aligned}
&(a \times n^2 + r_1 \times n) \times \mathrm{BI} \\
&\quad \leqslant (a \times n^2 + (r_1 - q) \times n + q \times n + v) \times \mathrm{BI} + \Delta t \\
&\quad = (a \times n^2 + (r_1 - q) \times n + c_2 + k) \times \mathrm{BI} + \Delta t \\
&\quad = (a \times n^2 + (r_1 - q) \times n + c_2) \times \mathrm{BI} + \Delta T,
\end{aligned}$$

and the following shows that B's beacon window ends earlier than A's active window:

$$\begin{aligned}
&(a \times n^2 + (r_1 - q) \times n + c_2) \times \mathrm{BI} + \mathrm{BW} + \Delta T \\
&\quad = (a \times n^2 + (r_1 - q) \times n + c_2 + k) \times \mathrm{BI} + \mathrm{BW} \\
&\qquad + \Delta t \\
&\quad = (a \times n^2 + r_1 \times n + v) \times \mathrm{BI} + \mathrm{BW} + \Delta t \\
&\quad \leqslant (a \times n^2 + (r_1 + 1) \times n) \times \mathrm{BI} + \mathrm{MW}.
\end{aligned}$$

- We prove that one of B's beacon window in its chosen row will be covered by A's active window in its chosen column.

Consider B's $(a \times n^2 + r_2 \times n + u)$th interval in its chosen row and A's $(a \times n^2 + (r_2 + p) \times n + c_1)$th interval in its chosen column. We prove that B's beacon window will be covered by A's active window. The following derivation shows that B's beacon window starts later than A's active window:

$$\begin{aligned}
&(a \times n^2 + (r_2 + p) \times n + c_1) \times \mathrm{BI} \\
&\quad \leqslant (a \times n^2 + (r_2 + p) \times n + c_1 - k) \times \mathrm{BI} \\
&\qquad + k \times \mathrm{BI} + \Delta t \\
&\quad = (a \times n^2 + (r_2 + p) \times n - p \times n + u) \times \mathrm{BI} + \Delta T \\
&\quad = (a \times n^2 + r_2 \times n + u) \times \mathrm{BI} + \Delta T,
\end{aligned}$$

and the following shows that B's beacon window ends earlier than A's active window:

Since $u + k = p \times n + c_1 \Rightarrow$

$$\begin{aligned}
&(a \times n^2 + r_2 \times n + u) \times \mathrm{BI} + \mathrm{BW} + \Delta T \\
&\quad = (a \times n^2 + r_2 \times n + u + k) \times \mathrm{BI} + \mathrm{BW} + \Delta t \\
&\quad = (a \times n^2 + (r_2 + p) \times n + c_1) \times \mathrm{BI} + \mathrm{BW} + \Delta t \\
&\quad < (a \times n^2 + (r_2 + p) \times n + c_1 + 1) \times \mathrm{BI} + \mathrm{MW}.
\end{aligned}$$

- We prove that one of A's beacon window in its chosen column will be covered by B's active window in its chosen row.

—if $(0 < u < n) \Rightarrow$

Consider A's $(a \times n^2 + (r_2 + p) \times n + c_1)$th interval in its chosen column. We prove that A's beacon window will be covered by B's active window in its chosen row. The following derivation shows that A's beacon window starts later than B's active window:

Since $k = p \times n + c_1 - u \Rightarrow$

$$\begin{aligned}
&(a \times n^2 + r_2 \times n) \times \mathrm{BI} + \Delta T \\
&\quad = (a \times n^2 + r_2 \times n + k) \times \mathrm{BI} + \Delta t \\
&\quad = (a \times n^2 + (r_2 + p) \times n + c_1 - u) \times \mathrm{BI} + \Delta t \\
&\quad < (a \times n^2 + (r_2 + p) \times n + c_1) \times \mathrm{BI},
\end{aligned}$$

and the following shows that A's beacon window ends earlier than B's active window:

Since $p \times n + c_1 = k + u \Rightarrow$

$$\begin{aligned}
&(a \times n^2 + (r_2 + p) \times n + c_1) \times \mathrm{BI} + \mathrm{BW} \\
&\quad \leqslant (a \times n^2 + r_2 \times n + k + u) \times \mathrm{BI} + \mathrm{BW} + \Delta t \\
&\quad = (a \times n^2 + r_2 \times n + u) \times \mathrm{BI} + \mathrm{BW} + \Delta T \\
&\quad < (a \times n^2 + (r_2 + 1) \times n) \times \mathrm{BI} + \mathrm{MW} + \Delta T.
\end{aligned}$$

—if $(u = 0) \Rightarrow$

Consider A's $(a \times n^2 + (r_2 + p + 1) \times n + c_1)$th interval in its chosen column. We prove that A's beacon window will be covered by B's active window in its chosen row. The following derivation shows that A's beacon window starts later than B's active window:

Since $k = p \times n + c_1 \Rightarrow$

$$\begin{aligned}&(a \times n^2 + r_2 \times n) \times \mathrm{BI} + \Delta T\\&\quad= (a \times n^2 + r_2 \times n + k) \times \mathrm{BI} + \Delta t\\&\quad= (a \times n^2 + (r_2 + p) \times n + c_1) \times \mathrm{BI} + \Delta t\\&\quad\leqslant (a \times n^2 + (r_2 + p + 1) \times n + c_1) \times \mathrm{BI},\end{aligned}$$

and the following shows that A's beacon window ends earlier than B's active window:

$$\begin{aligned}&(a \times n^2 + (r_2 + p + 1) \times n + c_1) \times \mathrm{BI} + \mathrm{BW}\\&\quad\leqslant (a \times n^2 + (r_2 + 1) \times n + k) \times \mathrm{BI} + \mathrm{BW} + \Delta t\\&\quad< (a \times n^2 + (r_2 + 1) \times n) \times \mathrm{BI} + \mathrm{MW} + \Delta T.\end{aligned}$$

- We prove that one of A's beacon window in its chosen row will be covered by B's active window in its chosen column.

—if $(0 \leqslant v < n - 1) \Rightarrow$

Consider A's $(a \times n^2 + r_1 \times n + v + 1)$th interval in its chosen row and B's $(a \times n^2 + (r_1 - q) \times n + c_2)$th interval in its chosen column. We prove that A's beacon window will be covered by B's active window. The following derivation shows that A's beacon window starts later than B's active window:

$$\begin{aligned}&(a \times n^2 + (r_1 - q) \times n + c_2) \times \mathrm{BI} + \Delta T\\&\quad= (a \times n^2 + (r_1 - q) \times n + q \times n + v) \times \mathrm{BI} + \Delta t\\&\quad= (a \times n^2 + r_1 \times n + v) \times \mathrm{BI} + \Delta t\\&\quad< (a \times n^2 + r_1 \times n + v + 1) \times \mathrm{BI},\end{aligned}$$

and the following shows that A's beacon window ends earlier than B's active window:

$$\begin{aligned}&(a \times n^2 + r_1 \times n + v + 1) \times \mathrm{BI} + \mathrm{BW}\\&\quad= (a \times n^2 + (r_1 - q) \times n + q \times n + v + 1) \times \mathrm{BI}\\&\qquad + \mathrm{BW}\\&\quad< (a \times n^2 + (r_1 - q) \times n + c_2 + k + 1) \times \mathrm{BI}\\&\qquad + \mathrm{MW} + \Delta t\\&\quad= (a \times n^2 + (r_1 - q) \times n + c_2 + 1) \times \mathrm{BI} + \mathrm{MW}\\&\qquad + \Delta T.\end{aligned}$$

—if $(v = n - 1) \Rightarrow$

Consider A's $(a \times n^2 + r_1 \times n)$th interval in its chosen row and B's $(a \times n^2 + (r_1 - q - 1) \times n + c_2)$th interval in its chosen column. We prove that A's beacon window will be covered by B's active window. The following derivation shows that A's beacon window starts later than B's active window:

$$\begin{aligned}&(a \times n^2 + (r_1 - q - 1) \times n + c_2) \times \mathrm{BI} + \Delta T\\&\quad= (a \times n^2 + (r_1 - q - 1) \times n + q \times n + v) \times \mathrm{BI}\\&\qquad + \Delta t\\&\quad= (a \times n^2 + (r_1 - 1) \times n + v) \times \mathrm{BI} + \Delta t\\&\quad< (a \times n^2 + r_1 \times n) \times \mathrm{BI},\end{aligned}$$

and the following shows that A's beacon window ends earlier than B's active window:

$$\begin{aligned}&(a \times n^2 + r_1 \times n) \times \mathrm{BI} + \mathrm{BW}\\&\quad= (a \times n^2 + (r_1 - q - 1) \times n + q \times n + v + 1)\\&\qquad \times \mathrm{BI} + \mathrm{BW}\\&\quad< (a \times n^2 + (r_1 - q - 1) \times n + c_2 + k + 1) \times \mathrm{BI}\\&\qquad + \mathrm{MW} + \Delta t\\&\quad= (a \times n^2 + (r_1 - q - 1) \times n + c_2 + 1) \times \mathrm{BI}\\&\qquad + \mathrm{MW} + \Delta T.\end{aligned}$$

References

[1] Nokia, Wireless broadband access—Nokia rooftop solution, Nokia Network References, http://www.wbs.nokia.com/solution/index.html, 2001.

[2] S.L. Wu, Y.C. Tseng, J.P. Sheu, Intelligent medium access for mobile ad hoc networks with BusyTones and power control, IEEE Journal on Selected Areas in Communications 18 (2000) 1647–1657.

[3] L. Hu, Topology control for multihop packet radio networks, IEEE Transactions on Communications 41 (1993) 1474–1481.

[4] R. Ramanathan, R. Rosales-Hain, Topology control of multihop wireless networks using transmit power adjustment, in: IEEE INFOCOM, 2000, pp. 404–413.

[5] R. Wattenhofer, L. Li, P. Bahl, Y.M. Wang, Distributed topology control for power efficient operation in multihop wireless ad hoc networks, in: IEEE INFOCOM, 2001, pp. 1388–1397.

[6] C.F. Huang, Y.C. Tseng, S.L. Wu, J.P. Sheu, Increasing the throughput of multihop packet radio networks with power adjustment, in: International Conference on Computer, Communication, and Networks, 2001.

[7] J. Gomez, A.T. Campbell, M. Naghshineh, C. Bisdikian, A distributed contention control mechanism for power saving in random-access ad-hoc wireless local area networks, in: Proceedings of IEEE International Workshop on Mobile Multimedia Communications, 1999, pp. 114–123.

[8] J.H. Ryu, D.H. Cho, A new routing scheme concerning power-saving in mobile ad-hoc networks, in: Proceedings of IEEE International Conference on Communications, vol. 3, 2000, pp. 1719–1722.

[9] J.H. Ryu, S. Song, D.H. Cho, A power-saving multicast routing scheme in 2-tir hierarchical mobile ad-hoc networks, in: Proceedings of IEEE Vehicular Technology Conference, vol. 4, 2000, pp. 1974–1978.

[10] S. Singh, M. Woo, C.S. Raghavendra, Power-aware routing in mobile ad hoc networks, in: Proceedings of the International Conference on Mobile Computing and Networking, 1998, pp. 181–190.

[11] I. Stojmenovic, X. Lin, Power-aware localized routing in wireless networks, in: Proceedings of IEEE International Parallel and Distributed Processing Symposium, 2000, pp. 371–376.

[12] LAN MAN Standards Committee of the IEEE Computer Society, IEEE Std 802.11-1999, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE, 1999.

[13] H. Woesner, J.P. Ebert, M. Schlager, A. Wolisz, Power-saving mechanisms in emerging standards for wireless LANs: the MAC level perspective, IEEE Personal Communications (June 1998) 40–48.

[14] J.C. Haartsen, The bluetooth radio system, IEEE Personal Communications (February 2000) 28–36.

[15] S. Singh, C.S. Raghavendra, Power efficient MAC protocol for multihop radio networks, in: Proceedings of IEEE International Personal, Indoor and Mobile Radio Communications Conference, 1998, pp. 153–157.

[16] C.F. Chiasserini, R.R. Rao, A distributed power management policy for wireless ad hoc networks, in: IEEE Wireless Communication and Networking Conference, 2000, pp. 1209–1213.

[17] J.R. Lorch, A.J. Smith, Software strategies for portable computer energy management, IEEE Personal Communications (1998) 60–73.

[18] A.K. Salkintzis, C. Chamzas, An in-band power-saving protocol for mobile data networks, IEEE Transactions on Communications 46 (1998) 1194–1205.

[19] T. Simunic, H. Vikalo, P. Glynn, G.D. Micheli, Energy efficient design of portable wireless systems, in: Proceedings of the International Symposium on Low Power Electronics and Design, 2000, pp. 49–54.

[20] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in: Proceedings of the International Conference on Mobile Computing and Networking, 2001, pp. 85–96.

[21] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: Proceedings of the International Conference on Mobile Computing and Networking, 2001, pp. 70–84.

[22] A.K. Salkintzis, C. Chamzas, Performance analysis of a downlink MAC protocol with power-saving support, IEEE Transactions on Vehicular Technology 49 (2000) 1029–1040.

[23] H. Garcia-Molina, D. Barbara, How to assign votes in a distributed systems, Journal of the ACM 32 (1985) 841–860.

[24] D. Agrawal, A. El Abbadi, An efficient and fault-tolerance algorithm for distributed mutual exclusion, ACM Transactions on Computer Systems 9 (1991) 1–20.

[25] A. Kumar, Hierarchical quorum consensus: a new algorithm for managing replicated data, IEEE Transactions on Computers 40 (1991) 996–1004.

[26] Y.C. Kuo, S.T. Huang, A geometric approach for constructing coteries and *k*-coteries, IEEE Transactions on Parallel and Distributed Systems 8 (1997) 402–411.

[27] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imelinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic, Boston, 1996, pp. 153–179.

[28] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.

[29] L.M. Feeney, M. Nilsson, Investigating the energy consumption of wireless network interface in an ad hoc networking environment, in: IEEE INFOCOM, 2001, pp. 1548–1557.

Yu-Chee Tseng received his B.S. and M.S. degrees in Computer Science from the National Taiwan University and the National Tsing-Hua University in 1985 and 1987, respectively. He worked for the D-LINK Inc. as an engineer in 1990. He obtained his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. From 1994 to 1996, he was an Associate Professor at the Department of Computer Science, Chung-Hua University. He joined the Department of Computer Science and Information Engineering, National Central University in 1996, and has become a Full Professor since 1999. Since Aug. 2000, he has become a Full Professor at the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan. He has served as a Program Chair in the *Wireless Networks and Mobile Computing Workshop*, 2000 and 2001, as an Associate Editor for *The Computer Journal*, as a Guest Editor for *ACM Wireless Networks* special issue on "Advances in Mobile and Wireless Systems", as a Guest Editor for *IEEE Transactions on Computers* special on "Wireless Internet", as a Guest Editor for *Journal of Internet Technology* special issue on "Wireless Internet: Applications and Systems", as a Guest Editor for*Wireless Communications and Mobile Computing* special issue on "Research in Ad Hoc Networking, Smart Sensing, and Pervasive Computing", as an Editor for *Journal of Information Science and Engineering*, and as a Guest Editor for *Telecommunication Systems* special issue on "Wireless Sensor Networks". He received the Outstanding Research Award, 2001–2002, from the National Science Council, ROC. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. He is a Senior Member of the IEEE.

Chih-Shun Hsu received his B.S. degree in computer education from National Taiwan Normal University, Taiwan, in 1990, and the M.S. degree in computer science from National Taiwan University, Taiwan, in 1992. He joined the faculty of the Department of Information Management, Nanya Institute of Technology, Taiwan, as an instructor in 1996. He is currently a Ph.D. candidate of the Department of Computer Science and Information Engineering, National Central University, Taiwan. His current research interests include wireless communications and mobile computing.

Ten-Yueng Hsieh received his B.S and M.S. degrees in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, in 2000 and 2002, respectively. His current research interests include mobile computing, wireless internetworking, telecommunication, and voice over IP.