# Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks

Yu-Chee Tseng, Chih-Shun Hsu, Ten-Yueng Hsieh

*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.**

## I. 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].

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 all supported by batteries.

Y. C. Tseng and T. Y. Hsieh are with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsin-Chu, Taiwan. E-mail: yctseng, tyhsieh@csie.nctu.edu.tw .

C. S. Hsu is with the Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan. E-mail: cshsu@axp1.csie.ncu.edu.tw .

This work is supported by the National Science Council of the Republic of China under Grant #NSC90-2213-E-009-049 and #NSC90-2213-E-009-154, and the Ministry of Education, the Republic of China, under grant 90-H-FA07-1-4(Learning Technology).

The authors would also like to thank the Lee and MTI Center for Networking Research at NCTU, Taiwan, for sponsoring this research.

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], [4], [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], [8], [9], [10], [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. Reference [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 multicasting 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. HyperLAN 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], [16], [17], [18], [19], [20], [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 HYPERLAN, 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 II.

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 send 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 and broadcast protocols based on our power-saving mechanisms are in Section 4. Simulation results are presented in Section 5. Section 6 concludes this paper.

## II. 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.

### *A. Reviews of Power Mode Management Protocols*

Several power management protocols have been proposed for MANET in [15], [16], [20], [21]. The *PAMAS* (Power Aware Multi-Access protocol with Signalling) [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. Reference [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].

### *B. Power-Saving Modes in IEEE 802.11*

IEEE 802.11 [12] supports two power modes: *active* and *power-saving (PS)*. 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), a 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. The short interval that PS hosts wake up 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 mobile host will contend to send a beacon frame. Any successful beacon serves as the purpose of synchronizing mobile

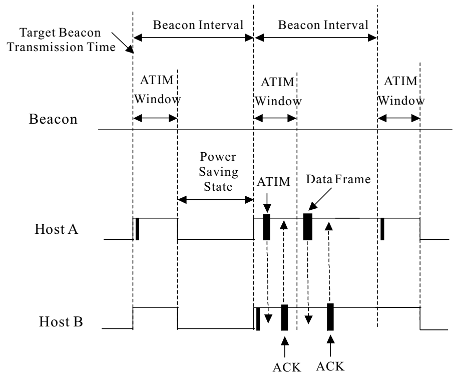


Fig. 1. An example of unicasting in an ad hoc networks with PS hosts.

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 $2 \times CW_{min} - 1$ before sending out its beacon.

After the beacon, a host with buffered unicast packets can send a direct ATIM frame to each of its intended receivers in PS mode. ATIM frames are also transmitted by contention based on the DCF access procedure. After transmitting an ATIM frame, the mobile host shall remain awake for the entire remaining period. On reception of the ATIM frame, the PS host should reply with an ACK and remains active for the remaining period. The buffered unicast packets should be sent based on the normal DCF access procedure after the ATIM window finishes. If the sender doesn't receive an ACK, it should retry in the next ATIM window. As for buffered broadcast packets, the ATIM frames need not be acknowledged. Broadcast packets then can be sent based on contention after the ATIM window finishes. 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.

Figure 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.

### C. *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 multihop 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 sub-networks 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 re-designed.

*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, not only is its chance to transmit reduced, but also its chance to hear others' signals. 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.

## III. 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 a 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 mutil-hop MANET. Excluding these three windows, a PS host with no packet to send or receive may go to the sleep mode. Figure 2(a)

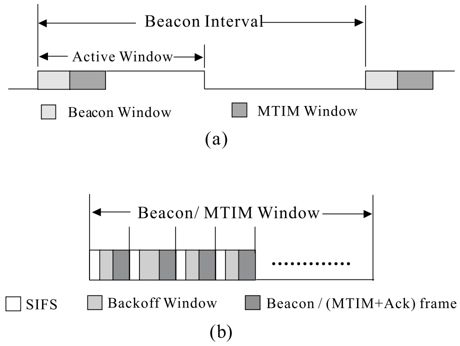


Fig. 2. Structure of a beacon interval: (a) active, beacon, and MTIM windows and (b) access procedure.

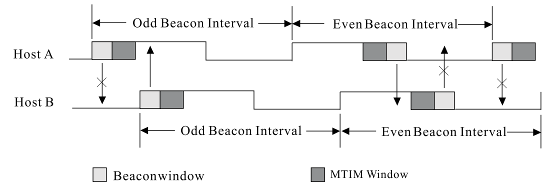


Fig. 3. Structures of odd and even intervals in the Dominating-Awake-Interval protocol.

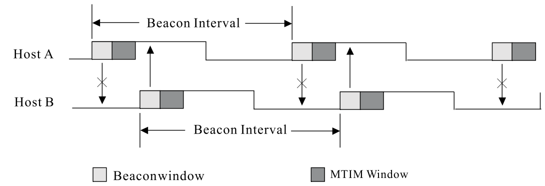


Fig. 4. An example where host B will always miss A's beacons.

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

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 Figure 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 $2 \times CW_{min} - 1$ slots. This is illustrated in Figure 2(b).

## A. 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 other and, 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.

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 $AW \geq BI/2 + BW$. The sequence of beacon intervals are alternatively labeled as *odd* and *even* intervals. Odd and even intervals have different structures as defined below (see the illustration in Figure 3).

- Each odd beacon interval starts with an active window. The active window is led by a beacon window 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's consider Figure 4, where beacon windows always appear at 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 Figure 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 $AW \geq BI/2 + BW$. The following theorem provides a formal proof on the correctness of this protocol.

*Theorem 1:* The Dominating-Awake-Interval protocol guarantees that when $AW \geq BI/2 + BW$, 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.

*Proof:* The detail of the proof is in [23] ■

The above proof 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.

## B. 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 $T$ 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 a MTIM window, such

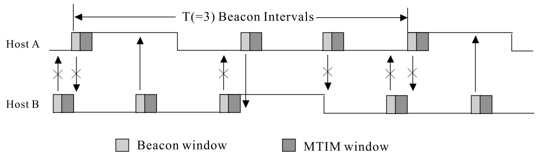


Fig. 5. An example of the Periodically-Fully-Awake-Interval protocol with fully-awake intervals arrive every $T = 3$ beacon intervals.

that $AW = BW + MW$. 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 a MTIM window. However, the host must remain awake in the rest of the time, i.e., $AW = BI$.

Intuitively, the low-power intervals is for a PS host to send out its beacons to inform others 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. Figure 5 shows an example with $T = 3$ intervals. So hosts $A$'s and $B$'s beacons always have chances to reach the other's active windows.

*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 $T$ 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 $T > 2$. However, the response time to get aware of a newly appearing host could be as long as $T$ beacon intervals. So this protocol is more appropriate for slowly mobile environments. One way to reduce the response time is to decrease the value of $T$ to fit one's need.

### C. 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 [24], [25], [26], [27]). A quorum is a set of identities from which one has to obtain permission to perform some action [24]. Typically, two quorum sets always have nonempty 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*.

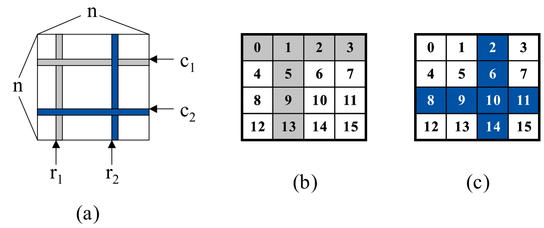


Fig. 6. Examples of the Quorum-Based Protocol (a)intersections of two PS hosts' quorum intervals, (b)host A's quorum intervals, and (c)host B's quorum intervals

Before proceeding, let's 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 Figure 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 (the formal proof is in Theorem 3). For example, in Figure 6(b) and (c), host $A$ selects intervals on row 0 and column 1 as its quorum intervals from a $4 \times 4$ 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.

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 $AW = MW$.

*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.

*Proof:* The detail of the proof is in [28] ■

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 $n \geq 4$, 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.

### D. Summary

Table I summarizes the characteristics of the three proposed power-saving protocols. "Number of beacons" 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 neighbor's beacon. As Table I 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 $T$ 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.

## IV. Communication 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 as for other hosts to calculate their time differences. Table II summarizes when $MTIM$ windows arrive in the proposed protocols.

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

### A. Unicast

This is similar to the procedure in IEEE 802.11's PS mode. During the receiver's $MTIM$ window, the sender contends to send its $MTIM$ packet to the receiver. The receiver, on receiving the MTIM packet, will reply an ACK after 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.

### B. 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 this notification, $S$ 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. The process is repeated until all its neighbors have been notified.

A neighbor, on receiving a 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 of two beacon intervals be used, but this can also be a adjustable parameter during system configuration). The source $S$, after notifying all neighbors, can contend to send its buffered broadcast packet after the last neighbor's MTIM window passes. Broadcast packets should be sent based on the DCF procedure too.

## V. 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 transmission radius is 250 meters and the transmission rate is 2M bits/sec. The MAC part basically follows the IEEE 802.11 standard [12], except the power management part. We intend to model one central host with the possibility of multiple mobile hosts approaching or leaving it (in the experiment, we use four neighbors.) We use an “on-off” model to simulate the mobility of the surrounding hosts. Specially, in every 5 seconds, a surrounding host chooses to enter an “on” or an “off” state. An “on” state indicates that the host is within the central host's transmission range, while an “off” state indicates that it is out of the range. The choice is based on a probability distribution. Three parameters are tunable in our simulations:

- Traffic load: a Poisson distribution for unicast/broadcast with rate between 5 $\sim$ 30 packets/sec.
- “On” probability: a uniform distribution between 50% $\sim$ 100%.
- Beacon interval: length of one beacon interval between 100 ms $\sim$ 500 ms.

Each simulation lasts for 100 seconds. Each result is obtained from the average of 1000 simulation runs. 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.

Three performance metrics are used to evaluate our power-saving protocols:

- power consumption: the average power consumption per mobile host throughout one simulation run.
- power efficiency: the average power consumption for each successful packet transmission.
- neighbor discovery time: average time to discover a newly approaching neighbor.

The power model in [29] is adopted, which is obtained by real experiments on Lucent WaveLAN cards. Table III summarizes the power consumption parameters used in our simulations. Sending/receiving a unicast/broadcast packet has a cost $P_{base} + P_{byte} \times L$, 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 active mode is much more energy-consuming. The traffic-related parameters are in Table IV.

In the following subsections, we show how beacon interval, mobility, and traffic load affect the performance of the proposed power-saving protocols. For simplicity, the Dominating-Awake-Interval protocol is denoted as $D$, the Periodically-Fully-Awake-

TABLE I

CHARACTERISTICS OF THE PROPOSED POWER-SAVING PROTOCOLS

| Protocol | Number of beacons | Active ratio | Neighbor sensitivity |
|---|---|---|---|
| Dominated-Awake | 1 | $1/2 + BW/BI$ | $BI$ |
| Periodically-Fully-Awake | 1 | $1/T$ | $T \times BI/2$ |
| Quorum-Based | $(2n-1)/n^2$ | $(2n-1)/n^2$ | $(n^2/4) \times BI$ |

TABLE II

TIMING OF $MTIM$ WINDOWS OF THE PROPOSED PROTOCOLS.

| Protocol | $MTIM$ window's timing |
|---|---|
| Dominated-Awake | $[(2m+1) \times BI + BW, (2m+1) \times BI + BW + MW]$ (odd int.)<br>$[2m \times BI + BI/2 - MW, 2m \times BI + BI/2]$ (even int.) |
| Periodically-Fully-Awake | $[m \times BI + BW, m \times BI + BW + MW]$ |
| Quorum-Based | $[m \times BI + BW, m \times BI + BW + MW]$ (quorum int.)<br>$[m \times BI, m \times BI + MW]$ (non-quorum int.) |

TABLE III

POWER CONSUMPTION PARAMETERS USED IN THE SIMULATION

| | |
|---|---|
| Unicast send | $454 + 1.9 \times L\ \mu W$ |
| Broadcast send | $266 + 1.9 \times L\ \mu W$ |
| Unicast receive | $356 + 0.5 \times L\ \mu W$ |
| Broadcast receive | $56 + 0.5 \times L\ \mu W$ |
| Idle | $843\ \mu W/ms$ |
| Doze | $27\ \mu W/ms$ |

TABLE IV

TRAFFIC-RELATED PARAMETERS USED IN THE SIMULATION

| | |
|---|---|
| Unicast packet size | 2048 bytes |
| Broadcast packet size | 256 bytes |
| Beacon window size | 8 ms |
| MTIM window size | 16 ms |

Interval protocol with parameter $T$ is denoted as $P(T)$, the Quorum-Based protocol with parameter $n$ is denoted as $Q(n)$, and the "always active" scheme is denoted as $AA$.

### A. Impact of Beacon Interval Length

The length of beacon intervals has impact on hosts' sensitivity to environmental changes and power consumptions. However, these are contradicting factors. To observe its impact, we vary the beacon interval length between 100 ms to 500 ms. As Figure 7 shows, longer beacon intervals only slightly increase the neighbor discovery time for schemes $D$ and $P(4)$, but have more significant impact on schemes $Q(4)$ and $Q(8)$. Overall, scheme $D$ has the shortest neighbor discovery time, which is subsequently followed by $P(4), Q(4)$, and $Q(8)$.

Figure 8 and Figure 9 show the power consumption at various beacon interval lengths for unicast and broadcast, respectively. In both cases, longer beacon intervals do incur less power consumption. From the curves, we would suggest the beacon interval be set at around 300 ms, since a larger interval will only contribute to a little more saving in power consumption. Overall, scheme $Q(8)$ has the smallest power consumption, which is subsequently followed by $P(4)$, $Q(4)$, $D$, and $AA$.

Comparing the above, we would conclude that $P(4)$ is the best choice since it minimizes neighbor discovery time as well as power consumption. Scheme $D$ is useful in highly mobile environment when power consumption is a less important issue, while scheme $Q(8)$ is very power-efficient when being applied to a low-mobility environment.

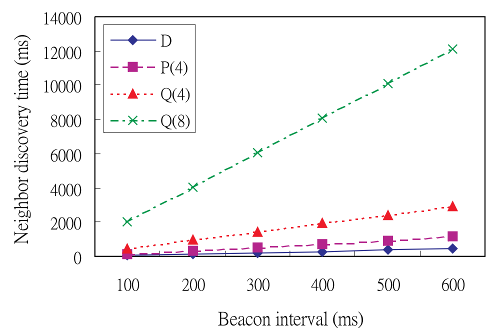


Fig. 7. Neighbor discovery time vs. beacon interval length. (traffic load = 10 packets, "on" probability = 80%)

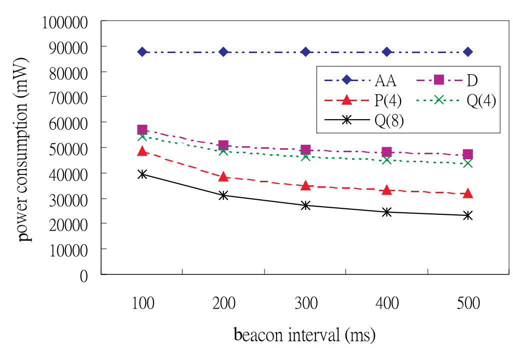


Fig. 8. Power consumption for unicast vs. beacon interval length. (traffic load = 10 packets/sec, "on" probability = 80%)

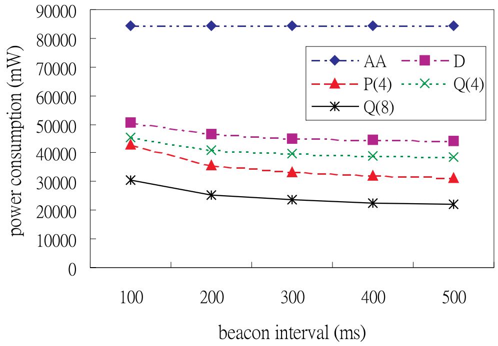


Fig. 9. Power consumption for broadcast vs. beacon interval length. (traffic load = 10 packets/sec, "on" probability = 80%)

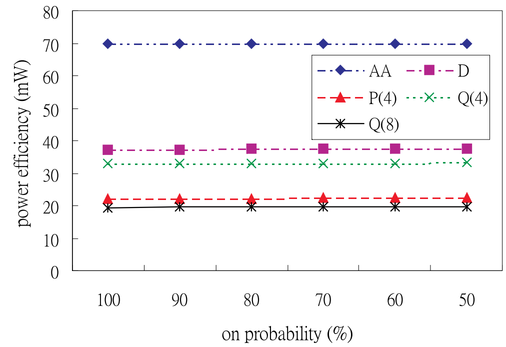


Fig. 11. Power efficiency for broadcast vs. "on" probability. (traffic load = 10 packets/sec, beacon interval = 300 ms)

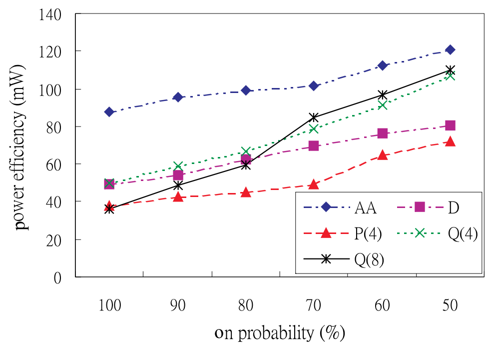


Fig. 10. Power efficiency for unicast vs. "on" probability. (traffic load = 10 packets/sec, beacon interval = 300 ms)

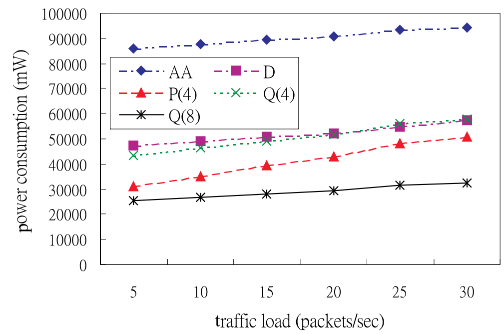


Fig. 12. Power consumption for unicast vs. traffic load. ("on" probability = 80%, beacon interval = 300 ms)

### B. Impact of Mobility

We use an "on-off" model to simulate the mobility of the surrounding hosts. Intuitively, a lower "on" probability implies higher mobility. When the mobility is high, a PS host may not be able to keep accurate neighborhood information, leading to a higher chance of sending useless "orphan" packets and thus wasting powers.

Figure 10 shows the impact of "on" probability on power efficiency. When the "on" probability is less than 70%, the power efficiency will increase sharply for all schemes. This is because inaccuracy neighborhood information mislead a host keeping on polling a missing host by sending many $MTIM$ packets. The efficiency of $Q(8)$ and $P(4)$ are the best at higher "on" probability. However, at lower "on" probability, $P(4)$ will outperform $Q(8)$ (because $Q(8)$'s sensitivity to neighborhood change will reduce).

The simulation result for broadcasting is show in Figure 11. The major difference is that the power efficiency is quite independent of the "on" probability. The reason is that broadcast is unreliable; based on our assumption, a broadcast packet is counted as successful as long as some neighbors are there to receive the packet. Thus, there are less useless transmissions. The trend is similar — $Q(8)$ performs the best, which is subsequently followed by $P(4), Q(4), D$, and $AA$.

### C. Impact of Traffic Load

In this experiment, we vary the traffic load to observe the effect. Figure 12 shows the power consumption for unicast traffic. A higher traffic load incurs higher power consumption, which is reasonable since hosts have less chance to sleep. Figure 13 shows the power consumption for broadcast traffic. The increase of power consumption due to increase of traffic load is almost unnoticeable because the broadcast packets being injected are quite small.

To observe from a different angle, Figure 14 and Figure 15 show the power efficiency at different traffic loads. A higher load makes transmitting a packet less costly for both unicast and broadcast traffics because multiple packets may be transmitted in one beacon interval. At lower traffic load, the idle time for hosts increases, thus wasting more power. Again, $P(4)$ and $Q(8)$ are most power-efficient, which are followed subsequently by $Q(4), D$, and $AA$.

## VI. 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 in such an environment if one directly adopts the power-saving (PS) mode defined in the IEEE 802.11 protocol. As far

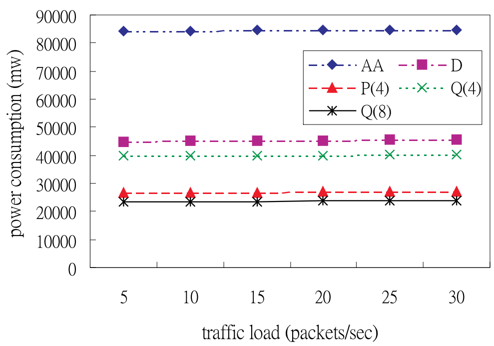


Fig. 13. Power consumption for broadcast vs. traffic load. ("on" probability = 80%, beacon interval = 300 ms)

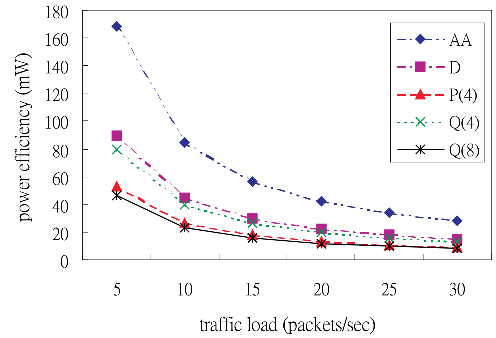


Fig. 15. Power efficiency for broadcast vs. traffic load. ("on" probability = 80%, beacon interval = 300 ms)

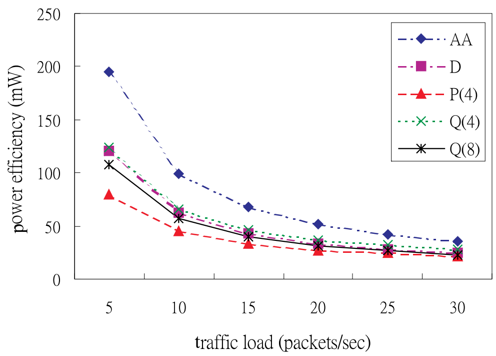


Fig. 14. Power efficiency for unicast vs. traffic load. ("on" probability = 80%, beacon interval = 300 ms)

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. The protocols can each guarantee an upper bound on packet delay if there is no collision in the beacon window (but collision is inevitable in any random-access network). Simulation results have shown that our power-saving protocols can save lots of power with reasonable neighbor discovery time. Among the three proposed protocols, the Dominating-Awake-Interval protocol is most power-consuming but has the lowest neighbor discovery time, while the Quorum-based protocol is most power-saving but has the longest neighbor discovery time. They are appropriate for highly mobile and lowly mobile environments, respectively. The Periodical-Fully-Awake-Interval protocol can balance both power consumption and neighbor discovery time, and thus may be used in most typical environments. We believe that the proposed protocols can be applied to current IEEE 802.11 wireless LAN cards easily with little modification.

## 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, and J. P. Sheu, "Intelligent Medium Access for Mobile Ad Hoc Networks with BusyTones and Power Control," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1647–1657, Sep 2000.

[3] L. Hu, "Topology Control for Multihop Packet Radio Networks," *IEEE Transactions on Communications*, vol. 41, pp. 1474–1481, Oct 1993.

[4] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment," *IEEE INFOCOM*, pp. 404–413, 2000.

[5] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," *IEEE INFOCOM*, pp. 1388–1397, 2001.

[6] C. F. Huang, Y. C. Tseng, S. L. Wu, and J. P. Sheu, "Increasing the Throughput of Multihop Packet Radio Networks with Power Adjustment," *International Conference on Computer, Cummunication, and Networks*, 2001.

[7] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "A Distributed Contention Control Mechanism for Power Saving in random-access Ad-Hoc Wireless Local Area Networks," *Proc. of IEEE International Workshop on Mobile Multimedia Communications*, pp. 114–123, 1999.

[8] J. H. Ryu and D. H. Cho, "A New Routing Scheme Concerning Power-Saving in Mobile Ad-Hoc Networks," *Proc. of IEEE International Conference on Communications*, vol. 3, pp. 1719–1722, 2000.

[9] J. H. Ryu, S. Song, and D. H. Cho, "A Power-Saving Multicast Routing Scheme in 2-tir Hierarchical Mobile Ad-Hoc Networks," *Proc. of IEEE Vehicular Technology Conference*, vol. 4, pp. 1974–1978, 2000.

[10] S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proc. of the International Conference on Mobile Computing and Networking*, pp. 181–190, 1998.

[11] I. Stojmenovic and X. Lin, "Power-aware Localized Routing in Wireless Networks," *Proc. of IEEE International Parallel and Distributed Processing Symposium*, pp. 371–376, 2000.

[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, and A. Wolisz, "Power-Saving Mechanisms in Emerging Standards for Wireless LANs: The MAC Level Perspective," *IEEE Persinal Communications*, pp. 40–48, Jun 1998.

[14] J. C. Haartsen, "The Bluetooth Radio System," *IEEE Persinal Communications*, pp. 28–36, Feb 2000.

[15] S. Singh and C. S. Raghavendra, "Power Efficient MAC Protocol for Multihop Radio Networks," *Proc. of IEEE International Personal, Indoor and Mobile Radio Communications Conference*, pp. 153–157, 1998.

[16] C. F. Chiasserini and R. R. Rao, "A Distributed Power Management Policy for Wireless Ad Hoc Networks," *IEEE Wireless Communications and Networking Conference*, pp. 1209–1213, 2000.

[17] J. R. Lorch and A. J. Smith, "Software Strategies for Portable Computer Energy Management," *IEEE Personal Communications*, pp. 60–73, Jun 1998.

[18] A. K. Salkintzis and C. Chamzas, "An In-Band Power-Saving Protocol for Mobile Data Networks," *IEEE Transactions on Communications*, vol. 46, pp. 1194–1205, Sep 1998.

[19] T. Simunic, H. Vikalo, P. Glynn, and G. D. Micheli, "Energy Efficient Design of Portable Wireless Systems," *Proc. of the International Symposium on Low Power Electronics and Design*, pp. 49–54, 2000.

[20] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Proc. of the International Conference on Mobile Computing and Networking*, pp. 85–96, 2001.

[21] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *Proc. of the International Conference on Mobile Computing and Networking*, pp. 70–84, 2001.
[22] A. K. Salkintzis and C. Chamzas, "Performance Analysis of a Downlink MAC Protocol with Power-Saving Support," *IEEE Transactions on Vehicular Technology*, vol. 49, pp. 1029–1040, May 2000.
[23] C. S. Hsu, "The Correctness of the Dominating-Awake-Interval Protocol," *http://axp3.csie.ncu.edu.tw/*$\sim cshsu$*/proof1.html*, 2001.
[24] H. Garcia-Molina and D. Barbara, "How to Assign Votes in a Distributed Systems," *Journal of the ACM*, vol. 32, pp. 841–860, Oct 1985.
[25] D. Agrawal and A. El Abbadi, "An Efficient and Fault-Tolerance Algorithm for Distributed Multual Exclusion," *ACM Transactions on Computer Systems*, vol. 9, pp. 1–20, Feb 1991.
[26] A. Kumar, "Hierarchical Quorum Consensus: A New Algorithm for Managing Replicated Data," *IEEE Transactions on Computers*, vol. 40, pp. 996–1004, Sep 1991.
[27] Y. C. Kuo and S. T. Huang, "A Geometric Approach for Constructing Coteries and k-Coteries," *IEEE Transactions on Parallel and Distributed Systems*, vol. 8, pp. 402–411, Apr 1997.
[28] C. S. Hsu, "The Correctness of the Quorum-Based Protocol," *http://axp3.csie.ncu.edu.tw/*$\sim cshsu$*/proof3.html*, 2001.
[29] L. M. Feeney and M. Nilsson, "Investigating the Energy Consumption of Wireless Network Interface in an Ad Hoc Networking Environment," *IEEE INFOCOM*, pp. 1548–1557, 2001.