Collaborative Wakeup in Clustered Ad Hoc Networks

Shan-Hung Wu, Chung-Min Chen, Ming-Syan Chen, *Fellow, IEEE*

***Abstract*—Clustering in wireless ad hoc networks has shown to be a promising technique to ensure the scalability and efficiency of various communication protocols. Since stations in these networks are usually equipped with batteries as the power source, it is critical to ensure the energy efficiency of a clustering scheme. The Quorum-based Power Saving (QPS) protocols are widely studied over the past years, as they render extensive energy conservation comparing to the IEEE 802.11 Power Saving (PS) mode. However, most existing QPS protocols adopt a symmetric design where each pair of stations in a network are guaranteed to discover each other. Observing that in clustered environments there is no need to insist on all-pair neighbor discovery, we propose an Asymmetric Cyclic Quorum (ACQ) system. The ACQ system guarantees the neighbor discovery between each member node and the clusterhead in a cluster, and between clusterheads in the network. We show that by taxing slightly more energy consumption on the clusterhead, the average energy consumption of stations in a cluster can reduce substantially than can be achieved by traditional QPS protocols. A novel construction scheme is proposed in this work, which assembles the ACQ system in $O(1)$ time. The constructing scheme is adaptive. Stations in a cluster can adjust their awake/sleep ratio collaboratively to strike the balance between energy efficiency and delay under various cluster conditions. Simulation results show that the ACQ system outperforms the previous studies up to 52% in energy efficiency, while introducing no extra worst-case latency.**

***Index Terms*—wireless communication, energy conservation, asymmetric quorum systems, network clustering, collaborative networks**

I. Introduction

The WIRELESS ad hoc networks have received a lot of attention recently. Clustering, as a means of topology control [1] in wireless ad hoc networks, has shown to be a promising technique to ensure the scalability and efficiency of various (e.g., routing) protocols [2], [3]. In contrast to the flat network structure, clustering offers a hierarchical view of network regions that facilitates reuse of resources (e.g., bandwidth, channel codes), localization of node dynamics, and coordination of transmissions [4]–[6]. Since each node in these networks is often equipped with batteries of limited capacity as the power source, energy conservation has long been a major interest in developing the clustering schemes [7]–[9].

Manuscript created October 1, 2010; revised February 15, 2011. This work was supported by the National Science Council, Taiwan, R.O.C. under the project no. 99-2218-E-007-023-.

S.-H. Wu is with the Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, ROC (e-mail: shwu@cs.nthu.edu.tw).

C.-M. is with Telcordia Technologies, Piscataway, NJ, USA (e-mail: chungmin@research.telcordia.com).

M. -S. Chen is with the Dept. of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC (e-mail: mschen@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/JSAC.2011.110908.

In IEEE 802.11 Distributed Coordination Function (DCF) [10], when a station (i.e., node) is not transmitting at PHY layer, it persists in *idle mode* and continuously listens for incoming transmissions. Studies [11], [12] observe that the energy cost of listening is only slightly lower than the cost of transmitting and receiving. If there are seldom transmissions destined to the station, *idle listening* would waste significant amount of energy. To address this problem, a *wakeup protocol* at MAC layer allows an idle station to save energy by entering the *sleep* (or *doze*) *mode*—to suspend the wireless module—and then wakeup periodically to check if there are pending transmissions. Since during wireless communication, the sender and receiver must both be awake to transmit and receive, the suspension should be exercised cautiously to ensure an overlap between awake periods.

Among all wakeup protocols, the Quorum-based Power Saving (QPS) protocols [13]–[17] are widely discussed over the past years. In a QPS protocol, the time axis on each station is divided evenly into *beacon intervals*. A station may stay awake or sleep during each beacon interval. Given an integer n, a quorum system defines a *cycle pattern*, which specifies the awake/sleep schedule during n continuous beacon intervals, for each station. Since the pattern repeats every n beacon intervals, we call n the *cycle length*. The merit of QPS protocols is that a station is required to remain awake only $O(\sqrt{n})$ beacon intervals every cycle, and that at least one of these awake beacon intervals is guaranteed to overlap with that of another station. QPS protocols render extensive energy efficiency as compared with IEEE 802.11 Power Saving (PS) mode [10].

However, in most existing QPS protocols the effect on power saving is limited by a theoretical bound. Specifically, given a cycle pattern of length n, a station is required to remain awake at least $\sqrt{n}$ beacon intervals to preserve an overlap [15]. The *duty cycle* of a station (i.e., portion of time a station must remain awake) can be no less than $O(\sqrt{n}/n) = O(1/\sqrt{n})$. Since the delay overhead increases proportionally to n and thus the value of n cannot be too large [13], this lower-bound of duty cycle seriously restricts the effectiveness of a QPS protocol.

In this paper, we propose a new quorum system, named Asymmetric Cyclic Quorum (ACQ) system, for clustered ad hoc networks whose effect on power saving is not restricted by the traditional bound of duty cycle. In clustered environments, a group of stations forms a *cluster*. A temporary *clusterhead* is selected in each cluster, which serves as a local coordinator of the cluster and is responsible for intra- and inter-cluster communication. Observe that each *member* (i.e., a regular node) in a cluster can simply rely on the clusterhead to forward its awake/sleep schedule or data, there is no need for a QPS protocol to insist on the overlap between *every pair* of stations. In other words, it is sufficient to promise that the awake period of each member in the cluster will overlap that of the clusterhead. The ACQ system defines two types of cycle patterns for members and the clusterhead respectively. These cycle patterns guarantee the overlap of awake beacon intervals between each member and the clusterhead in a cluster, and between all clusterheads in the network. In terms of duty cycles, the two types of cycle patterns are complementary to each other—when heavier a duty cycle is taxed on the clusterhead, *each member can have a lighter duty cycle below the traditional $O(1/\sqrt{n})$ bound*. This property fits well to the fact that clusterheads usually carry heavier load. Since members are the majority of nodes in a cluster, ACQ is able to give substantial reduction in average energy consumption.

To our best knowledge, the ACQ-based QPS protocol is the first wakeup protocol that capitalizes on the collaborative nature between nodes in a cluster to save energy. The ACQ system is a generalization of the cyclic quorum system [18]. To avoid exhaustive searching for ACQ (as did in the literature [18] to find the cyclic quorum system), we devise a novel constructing scheme that is able to assemble an ACQ system in $O(1)$ time. The proposed scheme is configurable. Different ACQs can be built that result in different distribution of energy consumption among members and the clusterhead. More importantly, the overlap is guaranteed even when stations obtain cycle patterns from different constructed ACQs. This allows each station to *adapt* its cycle patterns dynamically according to different cluster conditions (e.g., the remaining power of members or the clusterhead, or delay requirements of the current traffic, etc.). We also discuss how stations in a cluster can adapt their cycle patterns collaboratively to improve the overall energy efficiency. The adaptability, unlike other adaptive quorum systems which vary the cycle length [13], [14], does not impact the worst-case delay overhead and thus is valuable for applications (e.g., video/voice streaming) sensitive to delay or jitters. Experiment results show that ACQ is able to yield 36% and 52% improvement in energy efficiency as compared with CDS [17], [18] and AQEC [13] respectively given the same protocol design. Note ACQ improves the energy efficiency of clustering at MAC layer and thus is compatible with most existing clustering schemes at network and application layers.

The rest of this paper is organized as follows. Section II gives preliminaries and review of current QPS protocols. Section III formally defines the ACQ system. A constructing scheme of ACQ is introduced as well. We further extend ACQ to allow stations to adapt their cycle patterns in Section IV and discuss how stations can perform the adaptation collaboratively in response to various network conditions. In Section V, we evaluate the performance of ACQ in terms of energy efficiency, neighbor discovery time, and buffering delay. Section VI concludes the paper.

II. Preliminaries

In this section, we describe the clustered environments and review existing QPS protocols. Some terminologies and assumptions are specified as well that will be used throughout the text.

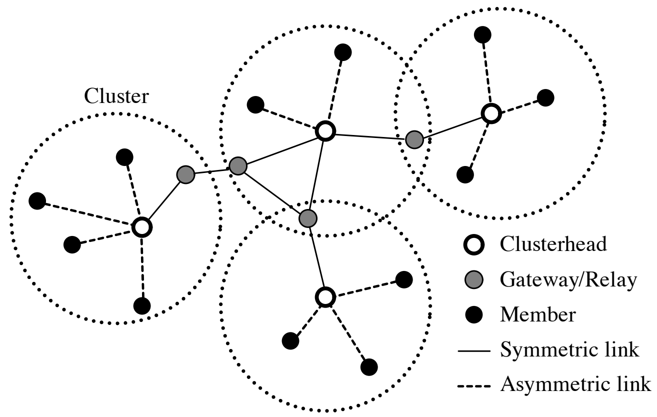

Fig. 1. Clustering in a wireless ad hoc network.

A. Clustering in Ad Hoc Networks

In a typical clustering scheme [3], [4], [7], [9], the nodes in a wireless ad hoc network are divided into *clusters*, as shown in Figure 1. Under the cluster structure, each node can be in one of the four states (or functions): the *clusterhead*, *gateway*, *member*, or *relay*. A clusterhead normally serves as a local coordinator for its cluster. It arranges intra-cluster transmissions and forwards data. A gateway is responsible for inter-cluster communications and forwards data between adjacent clusters. A member is an ordinary node that communicates only with the other hosts in the same cluster. In a cluster of diameter more than two hops, a relay forwards data between members and the clusterhead

As compared with the flat structure, clustering improves both the scalability and energy efficiency of a network due to the following benefits [7]–[9]. First, the geographically separated regions in different clusters facilitate spatial reuse of resource, such as bandwidth and codes, and increase system capacity [3], [4]. Second, the hierarchy localizes the node dynamics and gives a more stable view of the network topology [5], [19]. When a mobile host changes its attaching cluster, only nodes residing in the corresponding clusters need to update the routing information. Furthermore, clusterheads and gateways can normally form a virtual backbone [6], [20] that gives smaller network connectivity with better coordination. Retransmissions due to collisions or routing path losses can be reduced.

B. Quorum-based Power-Saving Protocols

We start reviewing existing wakeup protocols by considering first the IEEE 802.11 Power Saving (PS) mode [10]. As shown in Figure 2(a), the time axis on each PS station is divided evenly into *beacon intervals*. The duration of a beacon interval is denoted by $\overline{BI}$. In each beacon interval, a station is required to remain awake during the entire Announcement Traffic Indication Message window (*ATIM window*, whose duration is denoted by $\overline{AW}$). If a station, say H_1, intends to transmit data to the destination H_0, it first unicasts an *ATIM*

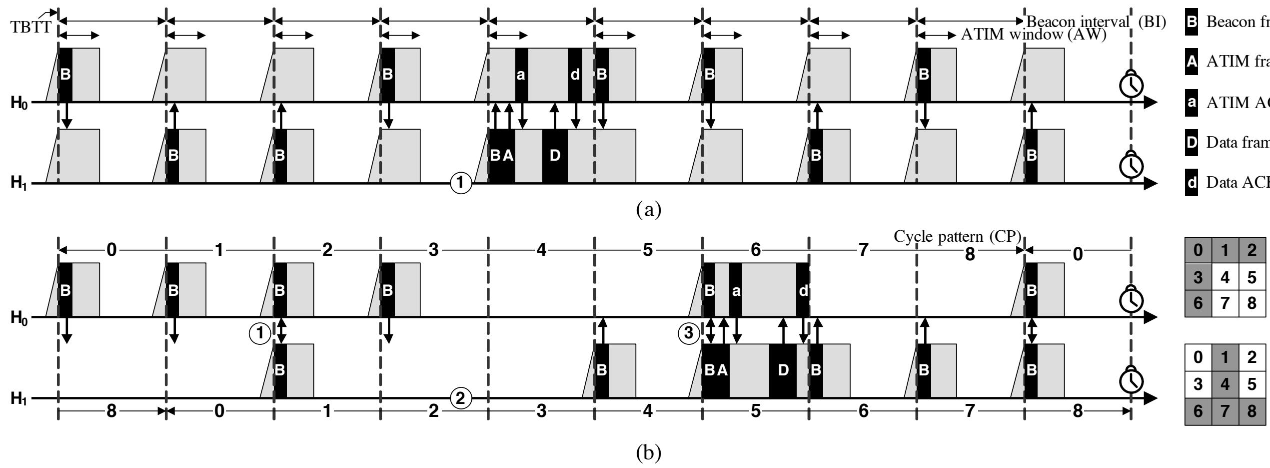

Fig. 2. Previous work. (a) IEEE 802.11 PS mode. (b) The grid-based QPS protocol.

frame to the host H_0 during an ATIM window (Figure 2(a)(1)). Since H_0 remains awake during every ATIM window, it receives the ATIM frame and sends back an acknowledgment. Both H_0 and H_1, after this ATIM notification procedure, keep awake for the entire beacon interval. Then the transmission of data can be started after the end of ATIM window. To avoid collisions, the data transmission follows the DCF (e.g., RTS, CTS, and random back-off) [10][1]. On the other hand, if there is no ATIM notifications, PS stations may enter the doze mode (that is, to sleep) after the ATIM window. Note $\overline{AW}$ is usually one fourth of $\overline{BI}$ as suggested in [10]. The energy consumption over ATIM windows is in-ignorable.

Apparently, the IEEE 802.11 PS mode functions only when the ATIM windows on all PS stations overlap. In other words, the Target Beacon Transmission Time (TBTT) needs to be aligned across all PS stations. To synchronize the timers, all stations contend to send a *beacon frame* carrying the clock information at the beginning of a beacon interval. Upon hearing the first beacon, each station synchronizes its timer with the contained information and cancels its own beacon transmission to avoid duplicate synchronization.

Based on the IEEE 802.11 PS mode, existing Quorum-based Power Saving (QPS) protocols can be generally classified into two categories: those do (*synchronous* QPS) or do not (*asynchronous* QPS) rely on timer synchronization. These two types of protocol are designed from different points of view. Asynchronous QPS protocols [15]–[17] *prolong* the awake periods on each station to ensure an overlap even when TBTT is not aligned. These protocols are useful to environments (e.g., sensor, vehicular networks) where clock synchronization is often costly or infeasible. On the other hand, synchronous QPS protocols [13], [14], [16] allow a station to *sleep more* without losing the overlap. These protocols achieve better energy efficiency. With recent developments in timer synchronization mechanisms for both single- [21] and multi-hop [22] wireless ad hoc networks, in this paper we assume that the clocks of stations are synchronized.

Next, we briefly summarize the grid-based QPS protocols [13], [15], [16] as they are relevant to our study. Figure 2(b) illustrates the awake/sleep schedules of two stations, H_0 and H_1, that are given by a grid quorum scheme with cycle length $n = 9$. A grid quorum scheme numbers every n continuous beacon intervals from 0 to $n-1$ and organizes them as an $\sqrt{n}\times\sqrt{n}$ array in a row-major manner. It defines a *quorum* as a set containing the numbers of beacon intervals along an arbitrary row and an arbitrary column in the array (e.g., $\{0,1,2,3,6\}$ or $\{1,4,6,7,8\}$ as shaded in Figure 2(b)). Each station, by using this scheme, is able to obtain its own quorum with a uniform *quorum size* (i.e., cardinality) $2\sqrt{n}-1$. For all beacon intervals whose numbers are specified in the quorum, a station remains awake during the ATIM windows (as in IEEE 802.11 PS mode). For the rest of beacon intervals, the station may sleep entirely during the periods. Since this schedule repeats every n beacon intervals, we call the repeating schedule *cycle pattern* (or *cycle* for short). The duration of a cycle is denoted by $\overline{CP}$. Note $\overline{CP} = n\overline{BI}$ and the duty cycle of a station is $(2\sqrt{n}-1)\overline{AW}/\overline{CP}$.

In QPS protocols, each station performs neighbor maintenance by remembering the awake/sleep schedules of nearby nodes[2]. Each beacon frame is required to carry information about the schedule (e.g., the adopted quorum and the current beacon interval, etc.) of the sending station. Unlike IEEE 802.11 PS mode where a station should cancel its own beacon transmission upon hearing the first beacon frame, every station should persist in beacon transmission even when others' beacons are heard in order to claim its own schedule.

As we can see in Figure 2(b), one grid-based quorum must intersect with another in two elements. This implies that the ATIM windows of any two stations must overlap twice per cycle. Once beacon frames are exchanged at an overlapped ATIM window (Figure 2(b)(1)), stations H_0 and H_1 are able to *discover* each other; that is, to receive one another's schedule

[1]The data transmission may proceed across multiple beacon intervals. When data transmission cannot complete within a single beacon interval (due to collisions or large data volume), H_1 can set the more-data bit (in the frame-control header field) true telling H_0 to remain awake during the entire successive beacon interval [10].

[2]We notice that a recent grid-based QPS protocol AQEC [13] does not perform neighbor maintenance (see QEC+ and AQEC+). This may lead to significant energy wastes on blindly sending the ATIM frames [14].

and predict its next coming of ATIM window. Suppose H_1 has data for H_0 (Figure 2(b)(2)), it buffers the data and waits for the ATIM window coming on H_0. After H_0 wakes-up (Figure 2(b)(3)), H_1 unicasts an ATIM frame to H_0 and starts the notification and data transmission procedures described previously in the IEEE 802.11 PS mode[3]. It is important to note that the grid quorum scheme ensures overlaps of ATIM windows *even when the numbering of beacon interval shifts between stations*. For example, in Figure 2(b) H_0's clock leads H_1's clock by one beacon interval. The quorum adopted by H_0, from H_1's point of view, becomes $\{0-1 \pmod 8, 1-1 \pmod 8, 2-1 (\bmod 8), 3-1 (\bmod 8), 6-1 (\bmod 8)\} = \{0,1,2,5,8\}$. We can easily verify that the rotated schedule of H_0 still overlaps twice per cycle with that of H_1. This property is called *rotation closure property*. Since the timer synchronization does not align the numbering semantics of beacon intervals across stations (it just aligns TBTT), only quorum schemes satisfying the rotation closure property can be used in a QPS protocol.

Given a cycle length n, existing quorum schemes satisfying the rotation closure property must have quorums of sizes larger than or equal to $\sqrt{n}$ [15]. This may seriously restrict the effectiveness of a QPS protocol as the duty cycle of a station can be no less than $O(\sqrt{n}/n) = O(1/\sqrt{n})$. Observe that in clustered environments, there is no need to insist the overlap of awake periods between *all* stations. We present a new quorum scheme whose effect on power saving is not limited by the traditional bound of duty cycle.

III. Asymmetric Cyclic Quorum System

This section introduces the ACQ system and its constructing scheme. The ACQ system defines two types of quorums: the *s-quorums* (*symmetric* quorums) and *a-quorums* (*asymmetric* quorums). In clustered environments, the clusterheads, gateways, and relays can use s-quorums to establish *symmetric links* (as shown by the solid lines in Figure 1) between themselves; while members can use a-quorums to establish *asymmetric links* (as shown by the long-dotted lines in Figure 1) to contact their clusterheads. Stations adopting s-quorums are able to discover each other as in conventional quorum systems. The ACQ system guarantees an overlap of ATIM window per cycle between these stations. Stations adopting a-quorums, however, can only discover stations with s-quorums. The ACQ system *does not* insist the intersection between a-quorums. We show that the cardinality of an a-quorum can be arbitrarily small (specifically, $O(1)$-sized). Therefore, the degree of power saving is expected to be substantially improved.

Next, we give formal definitions of an ACQ system.

A. Definitions and Fundamentals

Given a cycle length n, let $N = \{0, 1, \cdots, n-1\}$ be a universal set representing the numbers of beacon intervals in a cycle pattern. Consider the following definitions that is due to [18].

[3]The data transmission in a QPS protocol follows DCF as in the IEEE 802.11 PS mode to handle collisions.

Definition III.1 (cyclic set)**.** Let Q be a subset of N. Define $rotate(Q,i) = \{(q+i) \bmod n | \forall q \in Q\}$. The set $C(Q)$ is called a cyclic set (or cyclic group) of Q if $C(Q) = \{rotate(Q,i) | \forall i \in N\}$.

For example, suppose that $n = 9$ and $Q = \{0,1,2\}$. Then $C(Q) = \{\{0,1,2\},\{1,2,3\},\cdots,\{8,0,1\}\}$.

Definition III.2 (k-cyclic coterie)**.** Given a positive integer k, where $k \le n$. Let $X = \{Q_0, Q_1, \cdots, Q_{n-1}\}$ be a set of k-element subsets of N. The set X is called a *k-cyclic coterie* if and only if (a) $Q_i = rotate(Q_0, i)$; (b) $Q \cap Q' \neq \varnothing, \forall Q, Q' \in X$.

Conventionally, the cyclic coterie X is termed *quorum system*, and the elements of X (i.e., Q, the k-element subsets of N) are called *quorums*.

In this paper, we generalize the definition of a coterie to define an ACQ system.

Definition III.3 ((k,l)-cyclic bicoterie)**.** Given two positive integers k and l, where $k, l \le n$. Let $X = \{A_0, A_1, \cdots, A_{n-1}\}$ and $Y = \{S_0, S_1, \cdots, S_{n-1}\}$ be sets of k-element and l-element subsets of N respectively. The pair (X,Y) is called a *(k,l)-cyclic bicoterie* if and only if (a) $A_i = rotate(A_0, i)$ and $S_i = rotate(S_0, i)$; (b) $A \cap S \neq \varnothing, \forall A \in X, S \in Y$.

Note the pair (X,X) is a cyclic bicoterie if and only if X is a cyclic coterie.

Definition III.4 ((k,l)-ACQ system)**.** Given two positive integers k and l, where $k, l \le n$. Let X and Y be sets of nonempty subsets of N. The ordered pair (X,Y) is called a (k,l)-Asymmetric Cyclic Quorum (ACQ) system if and only if (a) (X,Y) is a (k,l)-cyclic bicoterie; (b) Y is an l-cyclic coterie.

The elements of X and Y (i.e., A and S, nonempty subsets of N) are called *a-quorums* and *s-quorums* respectively. The a-quorums have a quorum size k; while s-quorums have a quorum size l. Note the ACQ system is analogous to the *read-write quorum systems* used in replication management [23]. Different from those systems, ACQ satisfies the rotation closure property so it can be used by QPS protocols[4].

It can be shown that given a set Q, the cyclic group of Q, $C(Q)$, forms a k-cyclic coterie if and only if Q is a *k-difference set* [18]. We may obtain an analogous result when considering a pair (A,S).

Definition III.5 ((k,l)-difference pair)**.** Given two positive integers k and l, where $k, l \le n$. Let A and S be k-element and l-element subsets of N respectively. The ordered pair (A,S) is called a *(k,l)-difference pair* if for every $i \in N$, there exist (a,s), $a \in A$ and $s \in S$, such that $a - s \equiv i \bmod n$.

[4]Due to the space limitation, we do not prove the rotation closure property of the ACQ system in this article. Interested readers may refer to [15], [18] for further information.

Consider an example where $n = 9$. Let $A = \{0, 3, 6\}$ and $S = \{0, 1, 2, 5\}$ be two subsets of N, $N = \{0, 1, \cdots, 8\}$, then (A, S) is a $(3, 4)$-difference pair, since

$$\begin{aligned}&0 \equiv 0-0,\ 1 \equiv 3-2,\ 2 \equiv 3-1,\\ &3 \equiv 3-0,\ 4 \equiv 6-2,\ 5 \equiv 6-1,\\ &6 \equiv 6-0,\ 7 \equiv 0-2,\ 8 \equiv 0-1 \pmod 9.\end{aligned}$$

We can also verify that (S, S) is a $(4, 4)$-difference pair.

Lemma III.6. *Given two positive integers k and l, where $k, l \le n$. Let A and S be k-element and l-element subsets of N respectively. The pair $(C(A), C(S))$ is a (k, l)-cyclic bicoterie if and only if (A, S) is a (k, l)-difference pair.*

The proof of this lemma is analogous to that given in [18] showing the relation between a k-cyclic coterie and a k-difference set, and is omitted here. Lemma III.6 implies that we may find an ACQ system by identifying two sets A and S such that both (A, S) and (S, S) are difference pairs. Consider the previous example where $n = 9$, $A = \{0, 3, 6\}$, and $S = \{0, 1, 2, 5\}$, we obtain

$$C(A) = \begin{array}{l}\{\{0,3,6\},\{1,4,7\},\{2,5,8\},\\ \{3,6,0\},\{4,7,1\},\{5,8,2\},\\ \{6,0,3\},\{7,1,4\},\{8,2,5\}\}\text{ and}\end{array}$$

$$C(S) = \begin{array}{l}\{\{0,1,2,5\},\{1,2,3,6\},\{2,3,4,7\},\\ \{3,4,5,8\},\{4,5,6,0\},\{5,6,7,1\},\\ \{6,7,8,2\},\{7,8,0,3\},\{8,0,1,4\}\}.\end{array}$$

We can easily verify that $(C(A), C(S))$ forms a $(3, 4)$-ACQ system. The sets A and S are called *generating sets* of a-quorums and s-quorums respectively.

It remains a challenging issue to efficiently assemble a cyclic coterie (and bicoterie) given an arbitrary value of n. Studies [17], [18] find an optimal cyclic coterie (that is, cyclic coterie with the smallest quorum size) by either using exhaustive searches [18] or assuming $n = k^2 + k + 1$, where k is the prime power [17]. Furthermore, the recent study [17] shows that the problem finding an optimal asymmetric block design can be reduced to the minimum vertex cover problem, which is NP-complete. This implies that the problem finding an optimal ACQ system is also NP-complete. In the next section, we propose an algorithm that is able to construct the ACQ system in $O(1)$ time over arbitrary value of n while giving nearly-optimal quorum sizes.

B. Constructing Scheme for ACQ

Given a cycle length n and two integers ϕ and δ, where $\phi \ge 1$ and $1 \le \phi + \delta \le n$. Let $p = \left\lceil \frac{n}{\phi+\delta} \right\rceil$, we define a generating set of a-quorums as follows.

$$A(\phi, \delta) = \{a_0, a_1, \cdots, a_{p-1}\}, \tag{1}$$

where $a_0 = 0$, $0 < a_i - a_{i-1} \le \phi + \delta$ for all $1 \le i \le p - 1$, and $0 < n - a_{p-1} \le \phi + \delta$. Basically, the difference between two successive elements in $A(\phi, \delta)$ is less than or equal to $\phi + \delta$. Note with the above definition, $A(\phi, \delta)$ is not unique. Let $q = \left\lceil \frac{n-2\delta+1}{2\phi} \right\rceil$, the generating set of s-quorums is given by

$$S(\phi, \delta) = \{0, 1, \cdots, \phi + \delta - 1, s_1, s_2, \cdots, s_{q-1}\}, \tag{2}$$

where $\phi + \delta - 1 < s_1 \le 2\phi + \delta - 1$, $0 < s_i - s_{i-1} \le \phi$ for all $2 \le i \le q - 1$, and $s_{q-1} \ge (n - 1)/2$. Essentially, $S(\phi, \delta)$ contains $\phi+\delta$ continuous elements from 0 to $\phi+\delta-1$, followed by $q - 1$ interspaced elements with mutual distances less than or equal to ϕ. Note $S(\phi, \delta)$ is not unique as well with the above definition. We call Eqs. (1) and (2) the *ACQ scheme*. Suppose $n = 9$, $\phi = 3$, and $\delta = 0$. By fixing $g_i - g_{i-1} = \phi+\delta$ and $s_i - s_{i-1} = \phi$ we may obtain $A(3, 0) = \{0, 3, 6\}$ and $S(3, 0) = \{0, 1, 2, 5\}$, as we have seen in the previous section.

Next, we show that $(C(A(\phi, \delta)), C(S(\phi, \delta)))$ forms an ACQ system.

Lemma III.7. *Given n, ϕ, and δ, where $\phi \ge 1$ and $1 \le \phi + \delta \le n$. Let $p = \left\lceil \frac{n}{\phi+\delta} \right\rceil$ and $q = \left\lceil \frac{n-2\delta+1}{2\phi} \right\rceil$. The pair $(A(\phi, \delta), S(\phi, \delta))$ is a $(p, \phi + \delta + q - 1)$-difference pair.*

Proof: For any integer d, $0 \le d \le n-1$, we find integers x and y in $A(\phi, \delta)$ and $S(\phi, \delta)$ respectively such that $x-y \equiv d \pmod n$. Let $A(\phi, \delta) = \{a_0, a_1, \cdots, a_{p-1}\}$.
Case 1: $d = 0$. By definition, 0 is included in both $A(\phi, \delta)$ and $S(\phi, \delta)$, therefore $x = y = 0$.
Case 2: $a_{i-1} < d \le a_i$ for some i, $1 \le i \le p-1$. By definition of $A(\phi, \delta)$ we have $a_i - a_{i-1} \le \phi + \delta$. Let $u = a_i - d$, we obtain $u \le \phi + \delta - 1$. By definition of $S(\phi, \delta)$, u must be included in $S(\phi, \delta)$. Note $u = a_i - d$ implies $a_i - u \equiv d \pmod n$. Therefore $x = a_i$ and $y = u$.
Case 3: $a_{p-1} < d \le n - 1$. By definition of $A(\phi, \delta)$ we have $n - a_{p-1} \le \phi + \delta$. Let $v = n - d$, we obtain $v \le \phi + \delta - 1$, and v must be included in $S(\phi, \delta)$. Note $v = n - d$ implies $0 - v \equiv d \pmod n$. Therefore $x = 0$ and $y = v$. ■

Lemma III.8. *Given n, ϕ, and δ, where $\phi \ge 1$ and $1 \le \phi + \delta \le n$. Let $q = \left\lceil \frac{n-2\delta+1}{2\phi} \right\rceil$. The pair $(S(\phi, \delta), S(\phi, \delta))$ is a $(\phi + \delta + q - 1, \phi + \delta + q - 1)$-difference pair.*

Proof: For any integer d, $0 \le d \le n - 1$, we find two integers x and y in $S(\phi, \delta)$ such that $x - y \equiv d \pmod n$. Let $S(\phi, \delta) = \{0, 1, \cdots, \phi + \delta - 1, s_1, s_2, \cdots, s_{q-1}\}$.
Case 1: $0 \le d \le \phi + \delta - 1$. We have $d - 0 \equiv d \pmod n$. By definition, both d and 0 are included in $S(\phi, \delta)$. Therefore $x = d$ and $y = 0$.
Case 2: $\phi + \delta - 1 < d \le s_1$. By definition $s_1 - (\phi + \delta - 1) \le (2\rho + \delta - 1) - (\phi + \delta - 1) = \phi$. Let $u = s_1 - d$, we obtain $u \le \phi - 1$. Since $\phi - 1 \le \phi + \delta - 1$, u must be included in $S(\phi, \delta)$. Note $u = s_1 - d$ implies $s_1 - u \equiv d \pmod n$. Therefore $x = s_1$ and $y = u$.
Case 3: $s_{i-1} < d \le s_i$ for some i, $2 \le i \le q - 1$. By definition $s_i - s_{i-1} \le \phi$. Let $v = s_i - d$, we obtain $v \le \phi - 1$. Since $\phi - 1 \le \phi + \delta - 1$, u must be included in $S(\phi, \delta)$. Note $v = s_i - d$ implies $s_i - v \equiv d \pmod n$. Therefore $x = s_i$ and $y = v$.
Case 4: $s_{q-1} < d \le n - 1$. By definition, $s_{q-1} \ge (n - 1)/2$. Let $d' = n - d$, we have $1 \le d' < n - s_{q-1}$. This leads to $1 \le d' < n - (n - 1)/2 = (n + 1)/2$. Since d' is an integer, $1 \le d' < (n + 1)/2$ implies $1 \le d' \le \lceil (n - 1)/2 \rceil$. Note $s_{q-1} \ge (n-1)/2$ implies $s_{q-1} \ge \lceil (n - 1)/2 \rceil$. Thus we have $1 \le d' \le s_{q-1}$. Applying the arguments as given in Cases 1, 2, and 3, we can find x' and y' in $S(\phi, \delta)$ such that $x' - y' \equiv d' \pmod n$. Since $d' = n-d$, we obtain $y' - x' \equiv d \pmod n$.

It follows that $x = y'$ and $y = x'$. ■

Theorem III.9. *Given n, ϕ, and δ, where $\phi \geq 1$ and $1 \leq \phi + \delta \leq n$. Let $p = \left\lceil \frac{n}{\phi+\delta} \right\rceil$ and $q = \left\lceil \frac{n-2\delta+1}{2\phi} \right\rceil$. The pair*

$$(C(A(\phi, \delta)), C(S(\phi, \delta)))$$

forms a $(p, \phi + \delta + q - 1)$-ACQ system.

Proof: This is a direct consequence of Lemmas III.7, III.8 and III.6. ■

By employing the ACQ scheme, each station is able to obtain either an a-quorum or s-quorum in $O(1)$ time. For ease of implementation, a station can simply choose $A(\phi, \delta)$ and $S(\phi, \delta)$ as its own a-quorum and s-quorum respectively.

C. Wakeup Protocols based on ACQ

In this section, we explain how ACQ can improve the energy efficiency of a wakeup protocol. Basically, given a quorum Q, a station sleeps and wakes-up by following the corresponding cycle pattern of Q described in Section II (e.g., beacon frames, neighbor maintenance, data transmission procedure, etc.). Suppose a network starts from a flat structure. Each PS station adopting ACQ can obtain an s-quorum first to ensure the all-pair neighbor discovery. After the network is clustered at either the network or application layer, members and clusterheads can obtain a-quorums and s-quorums respectively. Unlike stations in conventional QPS protocols whose duty cycle cannot be less than $O(1/\sqrt{n})$, the duty cycle of a member can be arbitrarily small. Consider an extreme case where $n = 9$, $\phi = 3$, and $\delta = 6$[5]. By fixing $g_i - g_{i-1} = \phi + \delta$ in A we have $S(3, 6) = \{0, 1, \cdots, 8\}$ and $A(3, 6) = \{0\}$. Here the clusterheads remain awake in the ATIM window of every beacon interval; however, each member is required to remain awake in only one ATIM window every n beacon intervals. The duty cycle of members is $O(1/n)$ below the conventional bound.

The design of an ACQ-based wakeup protocol can take into account several practical issues. First, un-guaranteed intersection between a-quorums does not imply that members are not able to directly communicate with each other. Note a clusterhead knows the schedule of each member in a cluster (through asymmetric links). By carrying the schedules of all members in beacon frames, the clusterhead allow members to obtain one another's schedule and predict the coming ATIM window at the receiving party. Second, under the situation that a cluster is forming or the clusterhead is lost, members can temporarily adopt s-quorums until a new clusterhead is elected. Last, the ACQ scheme may pose heavier duty cycles on stations using s-quorums, thereby inducing the fairness issue on energy consumption. This problem can be resolved by energy-aware and load-balanced clustering schemes [7]–[9], which trigger re-elections of clusterhead after an aging period or when the battery level of current clusterhead falls below certain threshold.

[5]We will discuss how to configure ϕ and δ properly in Section IV.

IV. Adaptive ACQ

Note that unlike stations in conventional QPS protocols whose duty cycle cannot be less than $O(1/\sqrt{n})$, the duty cycle of a member can be arbitrarily small under the condition where the offered load is light or data transmission is not likely to happen in the near future.

The ACQ scheme introduced so far ensures that all stations, by picking the *same* values of ϕ and δ, are able to communicate with each other (through either symmetric or asymmetric links). Observe that in Eqs. (1) and (2), different values of ϕ and δ may result in different quorum sizes and therefore different degrees of energy conservation. It is desirable that a parameter, ϕ or δ, is adjustable on each station. Next, we show that the values of δ can actually be different among stations without losing the network connectivity.

A. More Fundamentals

The Definition III.4 can be generalized as follows.

Definition IV.1 (($\widetilde{k}, \widetilde{l}$)-ACQ system)**.** Given two sets $\widetilde{k} = \{k_1, k_2, \cdots, k_m\}$ and $\widetilde{l} = \{l_1, l_2, \cdots, l_{m'}\}$, where $k_i, l_j \leq n$ for all $1 \leq i \leq m$ and $1 \leq j \leq m'$. Let X_i and Y_j be sets of nonempty subsets of N. The ordered pair $(\bigcup_{i=1}^{m} X_i, \bigcup_{j=1}^{m'} Y_j)$ is called a $(\widetilde{k}, \widetilde{l})$-Asymmetric Cyclic Quorum (ACQ) system if and only if (a) for all i and j, (X_i, Y_j) is a (k_i, l_j)-cyclic bicoterie; (b) for all j and j', $1 \leq j' \leq m'$, $(S_j, S_{j'})$ is an $(l_j, l_{j'})$-cyclic bicoterie.

In this extended ACQ system, the sizes of a-quorums and s-quorums are not uniform .

Theorem IV.2. *Given n, ϕ, δ_1, δ_2, $\cdots$, and δ_m, where $\phi \geq 1$ and $1 \leq \phi+\delta_1 \leq \phi+\delta_2 \leq \cdots \leq \phi+\delta_m \leq n$. Let $p_i = \left\lceil \frac{n}{\phi+\delta_i} \right\rceil$ and $q_i = \left\lceil \frac{n-2\delta_i+1}{2\phi} \right\rceil$ for all i, $1 \leq i \leq m$. The pair*

$$(\textstyle\bigcup_{i=1}^{m} C(A(\phi, \delta_i)), C(S(\phi, \delta_m)))$$

forms a $(\widetilde{p}, \phi + \delta_m + q_m - 1)$-ACQ system, where $\widetilde{p} = \{p_1, p_2, \cdots, p_m\}$.

Theorem IV.3. *Given n, ϕ, δ_1, δ_2, $\cdots$, and δ_m, where $\phi \geq 1$, and $1 \leq \phi+\delta_1 \leq \phi+\delta_2 \leq \cdots \leq \phi+\delta_m \leq n$. Let $p_i = \left\lceil \frac{n}{\phi+\delta_i} \right\rceil$ and $q_i = \left\lceil \frac{n+1}{2(\phi+\delta_i)} \right\rceil$ for all i, $1 \leq i \leq m$. The pair*

$$(C(A(\phi, \delta_1)), \textstyle\bigcup_{i=1}^{m} C(S(\phi, \delta_i)))$$

forms a $(p_1, \widetilde{k})$-ACQ system, where $\widetilde{k} = \{(\phi+\delta_1+q_1-1), (\phi+\delta_2+q_2-1), \cdots, (\phi+\delta_m+q_m-1)\}$.

The proofs are omitted due to the space limitation. Theorems IV.2 and IV.3 are generalizations of Theorem III.9. The ACQ scheme, with these extensions, can further guarantee that:

- The clusterheads (or gateways) are able to discover each other regardless of the values of δ they choose;
- Each member is able to discover its clusterhead as long as the value of δ on the member does not exceed that on the clusterhead.

Since the size of a-quorums and s-quorums are fine-tunable (by varying δ) on each station, the ACQ scheme offers enhanced flexibility in power saving as compared with traditional quorum systems. Note δ is adjustable given a *fixed* cycle length n. This means that as δ varies, both the neighbor discovery time and data buffering delay are bounded by $O(n)$ beacon intervals.

In the following section, we investigate how to obtain a proper configuration for parameters ϕ and δ.

B. Collaborative Adaptation

In the ACQ scheme, ϕ and δ are global- and local-parameters respectively. The value of ϕ must remain the *same* among all stations in a network; while the value of δ can be dynamically *adapted* on each station according to various network conditions.

We first obtain a proper configuration of ϕ by assuming $\delta = 0$. Let $|X|$ denote the cardinality of a set X. Consider the following corollaries.

Corollary IV.4. *Given n, ϕ, where $1 \leq \phi \leq n$. Let $A(\phi, 0)$ and $S(\phi, 0)$ be generating sets returned by the ACQ scheme. If $|A(\phi, 0)| \leq |S(\phi, 0)|$, then $\phi > \sqrt{(n-1)/2}$.*

Corollary IV.5. *Given n, ϕ, where $1 \leq \phi \leq n$. Let $S(\phi, 0)$ be a generating set of s-quorums in the ACQ scheme. $S(\phi, 0)$ has the minimal size $2\left\lceil\sqrt{(n+1)/2}\right\rceil - 2$ or $2\left\lceil\sqrt{(n+1)/2}\right\rceil - 1$, when $\phi = \left\lceil\sqrt{(n+1)/2}\right\rceil$.*

The proofs are omitted due to the space limitation. Observe that in clustered environments, a clusterhead (or gateway) often carries heavier duty than a member does. Therefore, the size of an s-quorum should be no smaller than that of an a-quorum. Applying Corollaries IV.4 and IV.5, we may obtain $\phi \geq \left\lceil\sqrt{(n+1)/2}\right\rceil$ as reasonable configurations.

To decide the value of δ, consider a scenario where a clusterhead intends to transmit data to one of its members. Assume the data arrive at each beacon interval with equal probability. Let V be a random variable denoting the data buffering time (in number of beacon intervals) for the member. The mean of V can be expressed by $E[V] = \frac{1}{n}\sum_{i=0}^{n-1}\{(\phi+\delta) - [i \bmod (\phi+\delta)]\} \approx (\phi+\delta+1)/2$. Suppose $n = 9$ and $\phi = 3$, we may obtain the following statistics for $\delta = 0, 3, 6$.

δ	A	S	$\lvert A\rvert/\lvert S\rvert$	$E[V]$
0	$\{0, 3, 6\}$	{0-2, 5}	3/4	2
3	$\{0, 6\}$	{0-5}	2/6	4
6	$\{0\}$	{0-8}	1/9	5

As we can see, the value of δ controls the distribution of energy consumption. The larger the value of δ, the more awake periods are "transferred" from a member to the clusterhead. Due to this fact, the policies in deciding δ are diverse on these two stations.

A member tends to increase the value of δ as large as possible to give better energy efficiency. Nevertheless, when δ is increased, the average data buffering time $E[V]$ grows as well, which may impact the performance of a QPS protocol. Under such a trade-off, we may obtain a proper configuration of δ by either using the load-threshold technique [13] or power-management protocols [24], [25]. With the load-threshold technique, the value of δ can be adjusted according the receiving load on each station. In the power management protocol, the value of δ shrinks when certain events (e.g., replies to route discovery packets) are monitored, which provide good hints for upcoming data transmissions.

The clusterhead tends to shrink the value of δ as small as possible to give better energy efficiency. By Theorem IV.2, the value of δ chosen by the clusterhead must be larger than or equal to those chosen by all its members. Therefore, a proper configuration should be no smaller than $\delta_{\max}$, the maximal value of δ among the members. By employing the load-thresholds or power management protocols, the clusterhead is able to obtain a value δ_{net} suitable for the current network condition. Combining these, the clusterhead may pick $\max\{\delta_{\max}, \delta_{net}\}$ as a proper configuration.

Based on the fact that members are usually the majority of nodes in a cluster, the above collaborative adaptation scheme improves the average energy efficiency of a cluster by allowing each member to sleep more (under the condition where the offered load is light or data transmission is not likely to happen in the near future) without losing the connectivity to its clusterhead. Note that unlike stations in conventional QPS protocols whose duty cycle cannot be less than $O(1/\sqrt{n})$, the duty cycle of a member can be arbitrarily small.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of ACQ by considering both its theoretical properties and simulation results. We conduct our simulation on top of the *ns-2* simulator [26]. To see the improvement purely resulted from MAC layer, we consider the singleton topology consisting of a two-hop cluster in the network. A stationary node is delegated as the clusterhead and there is no re-election of the clusterhead. There are 20 stationary members uniformly distributed in the cluster, each with half-duplex wireless channel of rate 2 Mbps. The transmission range of a node is 100 meters. The clusterhead broadcasts schedules of all members with its beacon frames. Any two members falling apart from their coverage rely on the clusterhead to forward data. The durations of ATIM window and beacon interval are set 25 and 100 ms respectively [10]. The packet size is 256 bytes and each station is supplied with Poisson traffic with rate varying from 5 to 25 KBytes per second. The power consumption rate for transmit, receive, idle, and sleep modes in the wireless module are set 1650, 1400, 1150, and 45 mW respectively [27]. Each node has initial energy 500 J. We also implement the timer synchronization mechanism [10].

We define two theoretical metrics to evaluate the performance of ACQ:

Quorum ratio. Proportion of the beacon intervals that required to be awake in each cycle. This metrics equals $|Q|/n$, where $|Q|$ and n denote the quorum size and cycle length respectively.

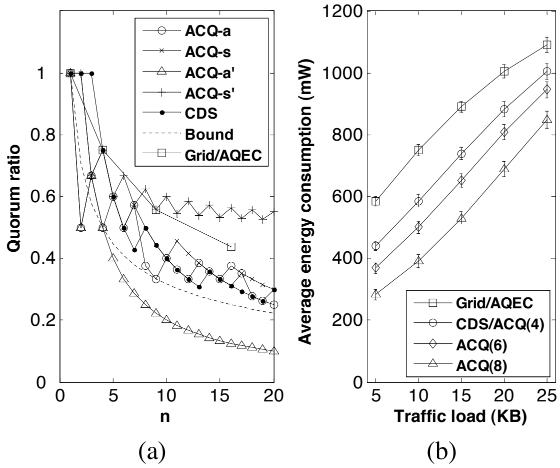

Fig. 3. Quorum ratio and average energy consumption rate.

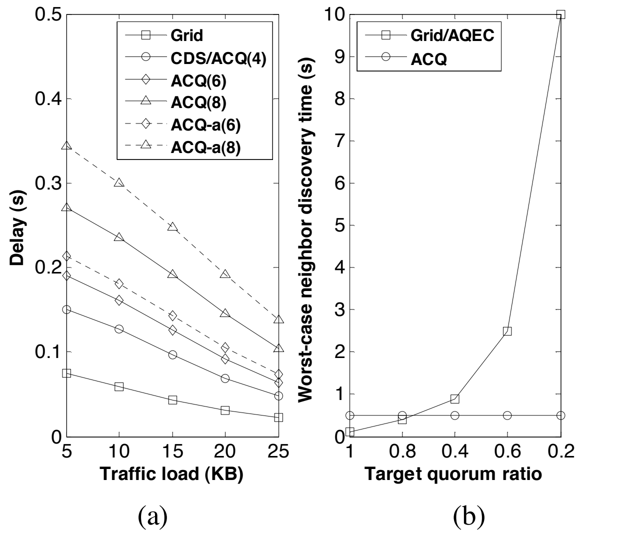

Fig. 4. Average buffering delay and worst-case neighbor discovery time.

Worst-case neighbor discovery time. The maximum amount of time for a station to discover the schedule of its new neighbor.

In addition to theoretical analysis, we run experiments in terms of the following metrics:

Average energy consumption. Energy consumption rate of a station during the experiment.

Average delay. The time between data arrival on a station and the reception at the next hop. Note this metrics includes data buffering time on a sending station.

Average lifetime: The time between the start of simulation and exhaustion of energy on a station.

We compare the ACQ scheme with the CDS [17], [18], Grid [15], [16], and AQEC [13] quorum schemes. To focus on the scheme comparison, we employ the same protocol design as introduced in Section II for all works such that they differ from each other only in the awake/sleep schedules. The default cycle length is 16 beacon intervals and $\phi = \left\lceil \sqrt{(n+1)/2} \right\rceil$ by Corollaries IV.4 and IV.5.

A. Quorum Ratio

We first explore the quorum ratio of different schemes by varying the cycle length n from 1 to 20. Two configuration sets for ACQ, ACQ-a/s and ACQ-a'/s', are employed where $\delta = 0$ and $\delta = \lceil (n+1)/2 \rceil - \phi$ respectively. As shown in Figure 3(a), for small δ the quorum ratios of ACQ are very close to that of CDS and approaches the theoretical bound $1/\sqrt{n}$ [15]. By increasing δ, ACQ is able to produce a-quorums with quorum ratios merely 33% of that given by CDS (when $n = 20$) and below the theoretical bound. Notice that the quorum ratios of s-quorums becomes higher as δ increases. However, since members (i.e., a-quorum users) are the majority of nodes in cluster environments, the average quorum ratio should be very close to that returned by a-quorums. This allows substantial reduction in energy consumption.

We also notice that Grid and AQEC cannot produce quorums given *arbitrary* cycle lengths. The sparse configuration density in these schemes may result in limited applicability to various network conditions (e.g., node mobility, delay requirements, etc.).

B. Energy Conservation

In this section we evaluate the average energy consumption rate by varying the loads from 5 to 25 KBytes per second. The cycle length is fixed to 16 beacon intervals in all schemes. This implies that the worst-case latency is 16 beacon intervals since stations meet at least once per cycle. We consider three configuration sets for ACQ, where $\phi + \delta = 4$, 6, and 8. The experimental results are illustrated in Figure 3(b). Note the performance of CDS are very close to that of ACQ when $\phi + \delta = 4$. Therefore, we present their results with the same line. Under all loads, ACQ yields better energy efficiency, and the improvement becomes significant as δ increases. In particular, at the load 5 KBytes per second, ACQ achieves 36% and 52% reduction in average energy consumption as compared with Grid and AQEC respectively, while rendering the same worst-case latency.

The error bars in Figure 3(b) depict the standard deviation of power consumption of nodes. We can see that the variance increases as the traffic loads become heavier. This is due to the higher chance of contention.

C. Delay

In this section, we investigate the average delay encountered during each packet transmission. The load varies from 5 to 25 KBytes per second. As we can see in Figure 4(a), the delays are under 300 ms in all schemes, and decreases as the load becomes heavier. This is because that under high loads, the ATIM notification and data transmission procedures (with more-data bit set true) become frequent and cause a receiving station to remain awake most of time. This reduces the data buffering time on sending stations.

Note the lines ACQ-a(6) and ACQ-a(8) reveal the delay encountered when sending packets to only the members. They are higher than the overall average. We notice that there are around 35% of the transmissions are destined to the clusterhead because of data forwarding (which happens when members are not within their mutual coverage). Since

TABLE I
AVERAGE LIFETIME OF NODES (IN SECONDS).

Scheme	ACQ(4)	ACQ(6)	ACQ(8)	ACQ-a
Members	618	739	801	823
Clusterhead	363	354	347	348

s-quorums require the clusterhead to be awake more as δ increases, the overall delay is reduced.

Recently, the IEEE 802.11e standard has defined an extension of the IEEE 802.11 Power Saving mode, named Automatic Power Saving Delivery (APSD), that allows a power saving node to provide the QoS guarantee by waking up within a beacon interval whenever it needs to transmit data and at the same time pull all buffered data destined to itself. It is important to note that the wakeup protocols we discussed in this paper are compatible to the APSD (specifically, Unscheduled APSD, U-APSD). When APSD is used, the lines shown in Figure 4(a) indicate the delay of traffics without the QoS guaranteed only. On the other hand, the QoS guaranteed traffics can be transmitted without being effected by a power saving protocol, since a station can become awake actively. A more detailed performance of U-APSD can be found in [28], [29].

D. Neighbor Discovery Time

In this section, we study the worst-case neighbor discovery time given that the target quorum ratio must be met. We consider the target quorum ratios 0.2, 0.4, $\cdots$, and 1. Note in conventional quorum schemes, the resulting quorum ratio can be tuned by varying the cycle length n. On the other hand, ACQ is able to produce different quorum ratios given different values of δ. We set a fixed cycle length $n = 5$ for ACQ. The performance results are shown in Figure 4(b). While ACQ offers stable neighbor discovery time as δ varies, the neighbor discovery time given by traditional schemes increases exponentially when the desired quorum ratio is small. This is because that stations are guaranteed to meet only once per cycle. We may conclude that ACQ offers more stable performance in terms of worst-case latency. This property is valuable for highly dynamic networks where stations need to discover each other within a short period.

E. Adaptive ACQ

In this section, we study the effects of adaptive ACQ by considering the average lifetime of members and the clusterhead. We offer each station variable load, distributed uniformly within $(0, 25]$ KBytes per second. Members adapt their own δ dynamically using the load-threshold technique [13], and the clusterhead adapts its δ by following the collaborative adaptation procedure described in Section IV. The cycle length is fixed to 16 beacon intervals, $\phi = 3$, and we consider three configuration sets for non-adaptive ACQ, where $\phi + \delta = 4$, 6, and 8. The results are shown in Table I. For simplicity, we allow the clusterhead to function even when its energy level falls down to 0 in order to collect the lifetime of members. Therefore, the results are optimistic without taking the cost for re-electing the clusterhead into account.

As we see, the lifetime of the clusterhead is considerably shorter than that of the members. One reason is that the ACQ casts more energy consumption onto the clusterhead. Another, and more dominating reason is that the clusterhead needs to keep awake frequently by following the wakeup protocol to forward the members' data. This makes the impact of ACQ less significant on the clusterhead.

On the other hand, the adaptive ACQ (ACQ-a) improves the energy efficiency of members by allowing them to sleep more under the light traffic. In particular, the ACQ-a gives 25%, 10%, and 3% improvement in energy efficiency as compared to ACQ(4), ACQ(6), and ACQ(8) respectively. It there is an aging period that triggers the re-election of clusterhead at the network or application layer, this implies that the lifetime of the network can also be prolonged.

VI. CONCLUSIONS

In this paper, we proposed an asymmetric quorum system, ACQ, for clustered environments. The ACQ is able to produce two complementary types of quorums: the a-quorums and s-quorums. When heavier duty cycle is taxed on the s-quorum users (e.g., clusterheads, relays, or gateways), a-quorum users (e.g., members) can have lighter duty cycle below the traditional $O(1/\sqrt{n})$ bound. We devised a construction scheme for the ACQ system, named ACQ scheme, which assembles a-quorums and s-quorums in $O(1)$ time. The parameter δ of the ACQ scheme is adaptive on each station according to various network conditions. We studied how nodes in a cluster can adapt their quorums collaboratively to improve the overall energy efficiency. From extensive simulation results, ACQ exhibits a superior performance as compared with traditional quorum schemes.

REFERENCES

[1] R. Rajaraman, "Topology control and routing in ad hoc networks: A survey," *ACM SIGACT News*, vol. 33, no. 2, pp. 66–73, 2002.

[2] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[3] C. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, 1997.

[4] T. Hou and T. Tsai, "An access-based clustering protocol for multihop wireless ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 7, pp. 1201–1210, 2001.

[5] A. Iwata, C. Chiang, G. Pei, M. Gerla, and T. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1369–1379, 1999.

[6] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing ad hoc routing with dynamic virtual infrastructures," in *Proc. INFOCOM*, 2001, pp. 1763–1772.

[7] A. Amis and R. Prakash, "Load-balancing clusters in wireless ad hoc networks," in *Proc. ASSET*, 2000, pp. 25–32.

[8] J. Wu, F. Dai, and M. G. amd I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," *IEEE/KICS Journal of Communications and Networks*, vol. 4, no. 1, pp. 59–70, 2002.

[9] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, 2004.

[10] I. . L. S. Committee, *Wireless LAN Medium Access Control and Physical Layer Specifications*, 1999.

[11] L. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. INFOCOM*, 2001, pp. 1548–1557.

[12] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," *Wireless Networks*, vol. 6, no. 4, pp. 263–277, 2000.

[13] C. Chao, J. Sheu, and I. Chou, "An adaptive quorum-based energy conserving protocol for ieee 802.11 ad hoc netowrks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 560–570, 2006.

[14] Z. Chou, "A randomized power management protocol with dynamic listen interval for wireless ad hoc networks," in *Proc. Vehicular Technology Conference (VTC-Spring)*, 2006, pp. 1251–1255.

[15] J. Jiang, Y. Tseng, C. Hsu, and T. Lai, "Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks," *Mobile Networks and Applications*, vol. 10, no. 1-2, pp. 169–181, 2005.

[16] Y. Tseng, C. Hsu, and T. Hsieh, "Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks," in *Proc. INFOCOM*, 2002, pp. 200–209.

[17] R. Zheng, J. Hou, and L. Sha, "Optimal block design for asynchronous wake-up schedules and its applications in multihop wireless networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1228–1241, 2006.

[18] W. Luk and T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *Proc. ICDCS*, 1997, pp. 100–106.

[19] W. Chen, N. Jain, and S. Singh, "Anmp: Ad hoc network management protocol," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1506–1531, 1999.

[20] U. Kozat, G. Kondylis, B. Ryu, and M. Marina, "Virtual dynamic backbone for mobile ad hoc networks," in *Proc. ICC*, 2001, pp. 250–255.

[21] L. Huang and T. Lai, "On the scalability of ieee 802.11 ad hoc networks," in *Proc. MobiHoc*, 2002, pp. 173–182.

[22] J. Sheu, C. Chao, and C. Sun, "A clock synchronization algorithm for multi-hop wireless ad hoc networks," in *Proc. ICDCS*, 2004, pp. 574–581.

[23] P. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[24] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

[25] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," in *Proc. INFOCOM*, 2003, pp. 481–491.

[26] "The network simulator," *http://www.isi.edu/nsnam/ns*.

[27] E. Jung and N. Vaidya, "An energy efficient mac protocol for wireless lans," in *Proc. INFOCOM*, 2002, pp. 1756–1764.

[28] X. Pérez-Costa, D. Camps-Mur, and A. Vidal, "On distributed power saving mechanisms of wireless LANs 802.11 e U-APSD vs 802.11 power save mode," *Computer Networks*, vol. 51, no. 9, pp. 2326–2344, 2007.

[29] S. Mangold, S. Choi, G. Hiertz, O. Klein, and B. Walke, "Analysis of IEEE 802.11 e for QoS support in wireless LANs," *Wireless Communications, IEEE*, vol. 10, no. 6, pp. 40–50, 2004.

Shan-Hung Wu Shan-Hung Wu received the B.S. degree from the Department of Information Management, National Central University, Jhongli, Taiwan, and the M.S. and Ph.D. degrees from the Department of Computer Science and the Department of Information and Electrical Engineering, respectively, National Taiwan University, Taipei, Taiwan. He is currently an assistant professor in the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Before joining the National Tsing Hua University, he was a senior research scientist at Telcordia Technologies. His research interests include wireless networks, mobile data management, database systems, and data mining.

Chung-Min Chen Chung-Min Chen received the Ph.D. degree in computer science from the University of Maryland, College Park and the BS degree in computer science from the National Taiwan University. He is currently an executive director and senior scientist at Telcordia Technologies. His research interests include data management, network management and their applications.

Ming-Syan Chen Ming-Syan Chen received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, and the M.S. and Ph.D. degrees in Computer, Information and Control Engineering from The University of Michigan, Ann Arbor, MI, USA, in 1985 and 1988, respectively. Dr. Chen is now a Distinguished Professor in Electrical Engineering Department at National Taiwan University. He was a research staff member at IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA from 1988 to 1996, the Director of Graduate Institute of Communication Eng. from 2003 to 2006, and also the President/CEO of Institute for Information Industry (III) in Taiwan from 2007 to 2008. His research interests include database systems, data mining, mobile computing systems, and multimedia networking, and he has published more than 260 papers in his research areas.

In addition to serving as program committee members in many conferences, Dr. Chen served as an associate editor of IEEE Transactions on Knowledge and Data Engineering (TKDE) from 1997 to 2001, is currently on the editorial board of Very Large Data Base (VLDB) Journal, Knowledge and Information Systems (KAIS) Journal, and International Journal of Electrical Engineering, and is a Distinguished Visitor of IEEE Computer Society for Asia-Pacific from 1998 to 2000, and also from 2005 to 2007 (invited twice). He served as the international vice chair for INFOCOM 2005, program chair of PAKDD-02, program co-chair of MDM-03, program vice-chair of IEEE ICDE-08, CEC/EEE-06, ICDE-06, ICDCS-05, ICPP-03, and VLDB-2002, and many other program chairs and co-chairs. He was a keynote speaker on Web data mining in International Computer Congress in 1999 and IEEE ISM in 2007, a tutorial speaker on Web data mining in DASFAA-1999 and on parallel databases in the 11th IEEE ICDE in 1995 and also a guest co-editor for IEEE TKDE on a special issue for data mining in December 1996. He holds, or has applied for, eighteen U.S. patents and seven ROC patents in the areas of data mining, Web applications, interactive video playout, video server design, and concurrency and coherency control protocols. He is a recipient of the NSC (National Science Council) Distinguished Research Award, Pan Wen Yuan Distinguished Research Award, Teco Award, Honorary Medal of Information, and K.-T. Li Research Penetration Award for his research work, and also the Outstanding Innovation Award from IBM Corporate for his contribution to a major database product. He also received numerous awards for his research, teaching, inventions and patent applications. Dr. Chen is a Fellow of ACM and also a Fellow of IEEE.