An Asymmetric Quorum-based Power Saving Protocol for Clustered Ad Hoc Networks

Shan-Hung Wu[‡†♯] Chung-Min Chen[†] Ming-Syan Chen[‡]

[‡]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC
[†]Telcordia Applied Research Center, Taipei, Taiwan, ROC
[♯]Industrial Technology Research Institute, HsinChu, Taiwan, ROC
{brandonwu@research.telcordia.com, chungmin@research.telcordia.com, mschen@cc.ee.ntu.edu.tw}

Abstract

Clustering in Mobile Ad Hoc Networks (MANETs) has shown to be a promising technique to ensure the scalability and efficiency of various communication protocols. Since stations in MANETs are usually equipped with batteries as the power source, it is critical to ensure the energy efficiency of clustering schemes. The Quorum-based Power Saving (QPS) protocols render extensive energy conservation as compared with IEEE 802.11 Power Saving (PS) mode and are widely studied over the past years. However, most existing QPS protocols adopt a symmetric design where stations in the network are guaranteed to discover each other. Observing that in clustered environments there is no need to insist in 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. A construction scheme is presented in this work, which assembles the ACQ system in $O(1)$ time. 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. Simulation results show that the ACQ system outperforms the previous studies up to 52% in energy efficiency, while introducing no extra worst-case latency.

1 Introduction

The Mobile Ad Hoc Network (MANET) has received a lot of attention recently. Clustering, as a means of topology control [20] in MANETs, has shown to be a promising technique to ensure the scalability and efficiency of various (e.g., routing) protocols [9, 18]. 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 [10, 12, 22]. Since each node in a MANET 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 [2, 24, 25].

In IEEE 802.11 Distributed Coordination Function (DCF) [7], when a station (i.e., node) is not transmitting, it persists in *idle mode* and continuously listens for incoming transmissions. Studies [8, 16] observe that the energy cost of listening is only slightly lower than the cost of transmitting and receiving. Therefore, the best way for an idle station to save energy is to enter the *sleep* (or *doze*) *mode*—to suspend the wireless module. Since during wireless communication, both the sender and receiver must be awake to transmit and receive, suspension should be exercised cautiously to ensure an overlap between awake periods.

Among all possible solutions, the Quorum-based Power Saving (QPS) protocols [4, 6, 13, 23, 26] 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 [7].

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 [13]. 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 [4], 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 MANETs 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 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. The two types of cycle pattern are complementary—when heavier duty cycle is taxed on the clusterhead, *each* member can have lighter duty cycle below the traditional $O(1/\sqrt{n})$ bound. This property capitalizes the characteristics of clustered environments where clusterheads usually carry heavier load [10, 18]. Since members are the majority of nodes in a cluster, ACQ allows substantial reduction in average energy consumption.

To our best knowledge, the ACQ system is the first asymmetric quorum system. It is a generalization of the cyclic quorum system [19]. To avoid exhaustive searching for ACQ (as did in the literature [19] to find the cyclic quorum system), we present a constructing scheme that is able to assemble an ACQ system in $O(1)$ time. The proposed scheme is configurable. Different ACQs can be built for different networks that result in different distribution of energy consumption between members and the clusterhead. Experiment results show that ACQ is able to yield 36% and 52% improvement in energy efficiency as compared with CDS [19, 26] and AQEC [4] 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 2 gives preliminaries and review of current QPS protocols. Section 3 formally defines the ACQ system. A constructing scheme of ACQ is introduced thereafter. In Section 4, we evaluate the performance of ACQ in terms of energy efficiency, neighbor discovery time, and delay.

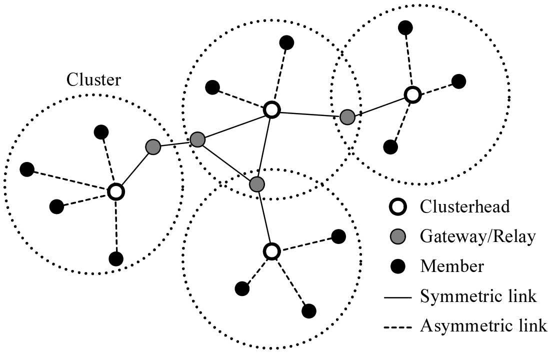


Figure 1. Clustering in a MANET.

2 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.

2.1 Clustering in Ad Hoc Networks

In a typical clustering scheme [2, 10, 18, 25], the mobile nodes in a MANET 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 [2, 24, 25]. First, the geographically separated regions in different clusters facilitate spatial reuse of resource, such as bandwidth and codes, and increase system capacity [10, 18]. Second, the hierarchy localizes the node dynamics and gives a more stable view of the network topology [5, 12]. 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 [15, 22] that gives smaller network connectivity with better coordination. Retransmissions due to collisions or routing path losses can be reduced.

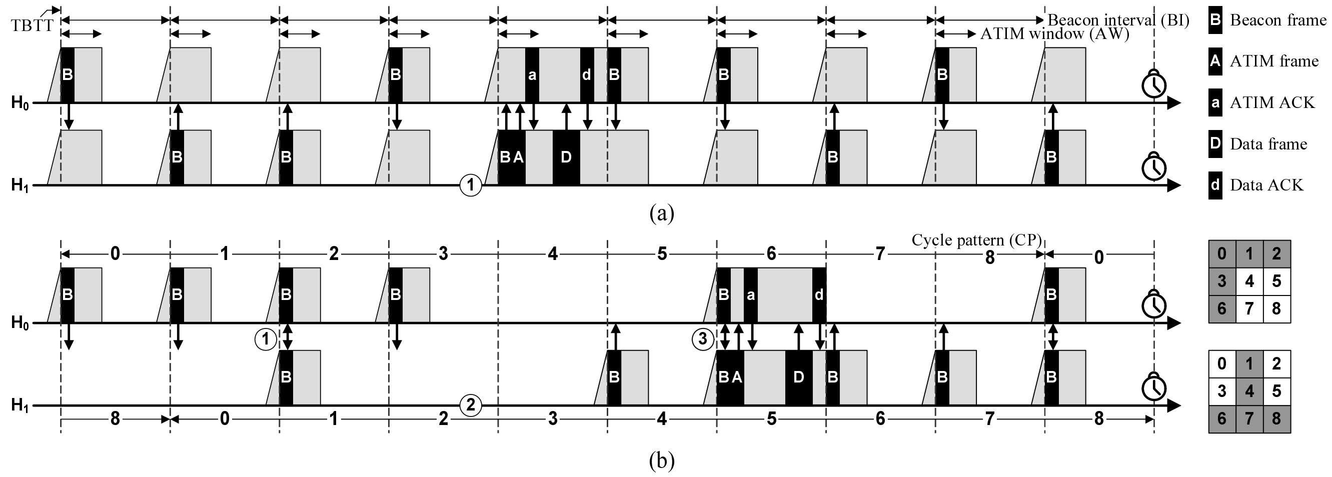


Figure 2. Previous work. (a) IEEE 802.11 PS mode. (b) The grid-based QPS protocol.

2.2 Quorum-based Power-Saving Protocols

We start reviewing existing power saving protocols by considering first the IEEE 802.11 Power Saving (PS) mode [7]. 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 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 acknowledgement. Both H_0 and H_1, after this ATIM notification procedure, keep awake for the entire beacon interval. The DCF (e.g., RTS, CTS, and random back-off) [7] can then be performed to transmit the data after the end of ATIM window. 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. 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, as suggested in [7], $\overline{AW}$ is one fourth of $\overline{BI}$. The energy consumption over ATIM windows is in-ignorable.

The IEEE 802.11 PS mode requires the Target Beacon Transmission Time (TBTT) on all PS stations to be aligned to ensure the overlap of ATIM windows. To synchronize 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.

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 [13, 23, 26] *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 [4, 6, 23] allow a station to *sleep more* without losing an overlap. These protocols achieve better energy efficiency. With recent developments in timer synchronization mechanisms for both single- [11] and multi-hop [21] MANETs, in this paper we assume that the clocks of stations are synchronized.

We briefly summarize the grid-based QPS protocols [4, 13, 23] 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 letting beacon frames carry information about the schedule (e.g., the adopted quorum and the current beacon interval, etc.)[1]. 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 its 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 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. 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 $\{0-1(\bmod 8), 1-1(\bmod 8), 2-1(\bmod 8), 3-1(\bmod 8), 6-1(\bmod 8)\} = \{0,1,2,5,8\}$. 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 timer synchronization does not align the numbering semantics of beacon intervals, 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 $\sqrt{n}$ [13]. 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.

3 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.

The design of ACQ takes 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 [2, 24, 25], which trigger re-elections of clusterhead if the battery level of current clusterhead falls below certain threshold.

Next, we give formal definitions of an ACQ system.

3.1 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 [19].

Definition 3.1 (cyclic set) *Let X be a subset of N. Define $rotate(X,i) = \{(x+i) \bmod n | \forall x \in X\}$. The set $C(X)$ is called a cyclic set (or cyclic group) of X if $C(X) = \{rotate(X,i) | \forall i \in N\}$.*

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

---

[1] We notice that a recent grid-based QPS protocol AQEC [4] does not perform neighbor maintenance (see QEC+ and AQEC+). This may lead to significant energy wastes on blindly sending the ATIM frames [6].

Definition 3.2 (k-cyclic coterie) *Given a positive integer k, where $k \leq n$. Let $X = \{X_0, X_1, \cdots, X_{n-1}\}$ be a set of k-element subsets of N. The set X is called a k-cyclic coterie if and only if (a) $X_i = rotate(X_0, i)$; (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., k-element subsets of N) are called *quorums*.

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

Definition 3.3 ((k,l)-cyclic bicoterie) *Given two positive integers k and l, where $k, l \leq n$. Let $X = \{X_0, X_1, \cdots, X_{n-1}\}$ and $Y = \{Y_0, Y_1, \cdots, Y_{n-1}\}$ 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) $X_i = rotate(X_0, i)$ and $Y_i = rotate(Y_0, i)$; (b) $Q \cap Q' \neq \varnothing, \forall Q \in X, Q' \in Y$.*

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

Definition 3.4 ((k,l)-ACQ system) *Given two positive integers k and l, where $k, l \leq n$. Let A and S 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 A and S (i.e., 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 [3]. Different from those systems, ACQ satisfies the rotation closure property so it can be used by QPS protocols[2].

It has been shown that given a set X, the cyclic group of X, $C(X)$, forms a k-cyclic coterie if and only if X is a k-*difference set* [19]. We may obtain an analogous deduction when given a pair (X, Y).

Definition 3.5 ((k,l)-difference pair) *Given two positive integers k and l, where $k, l \leq n$. Let X and Y be k-element and l-element subsets of N respectively. The ordered pair (X, Y) is called a (k,l)-difference pair if for every $i \in N$, there exist (x, y), $x \in X$ and $y \in Y$, such that $x - y \equiv i \pmod{n}$.*

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

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

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

Lemma 3.1 *Given two positive integers k and l, where $k, l \leq n$. 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 [19] showing the relation between a k-cyclic coterie and a k-difference set, and is omitted here. Lemma 3.1 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

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

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

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 [19, 26] find an optimal cyclic coterie (that is, cyclic coterie with the smallest quorum size) by either using exhaustive searches [19] or assuming $n = k^2 + k + 1$, where k is the prime power [26]. Furthermore, the recent study [26] 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.

3.2 Constructing Scheme for ACQ

In this section, we present an algorithm originally used for replication management [17] that is able to construct the ACQ system in $O(1)$ time. This scheme takes arbitrary values of n as the input and returns a-quorums and s-quorums of nearly-optimal sizes.

Given a cycle length n and an integer ϕ, where $1 \leq \phi \leq n$. Let $p = \left\lceil \frac{n}{\phi} \right\rceil$, we define a generating set of a-quorums as follows.

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

---

[2] 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 [13, 19] for further information.

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

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

where $\phi - 1 < s_1 \leq 2\phi - 1$, $0 < s_i - s_{i-1} \leq \phi$ for all $2 \leq i \leq q-1$, and $s_{q-1} \geq (n-1)/2$. Essentially, $S(\phi)$ contains ϕ continuous elements from 0 to $\phi - 1$, followed by $q - 1$ interspaced elements with mutual distances less than or equal to ϕ. Note $S(\phi)$ is not unique as well with the above definition. We call Eqs. (1) and (2) the *ACQ scheme*. Suppose $n = 9$ and $\phi = 3$. By fixing $g_i - g_{i-1} = \phi$ 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 $(C(A(\phi)), C(S(\phi)))$ forms an ACQ system.

Lemma 3.2 *Given n and ϕ, where $1 \leq \phi \leq n$. Let $p = \left\lceil \frac{n}{\phi} \right\rceil$ and $q = \left\lceil \frac{n+1}{2\phi} \right\rceil$. The pair $(A(\phi), S(\phi))$ is a $(p, \phi + q - 1)$-difference pair.*

Proof. *For any integer d, $0 \leq d \leq n-1$, we find integers x and y in $A(\phi)$ and $S(\phi)$ respectively such that $x - y \equiv d \pmod{n}$. Let $A(\phi) = \{a_0, a_1, \cdots, a_{p-1}\}$.*

Case 1: $d = 0$. By definition, 0 is included in both $A(\phi)$ and $S(\phi)$, therefore $x = y = 0$.

Case 2: $a_{i-1} < d \leq a_i$ for some i, $1 \leq i \leq p-1$. By definition of $A(\phi)$ we have $a_i - a_{i-1} \leq \phi$. Let $u = a_i - d$, we obtain $u \leq \phi - 1$. By definition of $S(\phi)$, u must be included in $S(\phi)$. Note $u = a_i - d$ implies $a_i - u \equiv d \pmod{n}$. Therefore $x = a_i$ and $y = u$.

Case 3: $a_{p-1} < d \leq n - 1$. By definition of $A(\phi)$ we have $n - a_{p-1} \leq \phi$. Let $v = n - d$, we obtain $v \leq \phi - 1$, and v must be included in $S(\phi)$. Note $v = n - d$ implies $0 - v \equiv d \pmod{n}$. Therefore $x = 0$ and $y = v$. ■

Lemma 3.3 *Given n and ϕ, where $1 \leq \phi \leq n$. Let $q = \left\lceil \frac{n+1}{2\phi} \right\rceil$. The pair $(S(\phi), S(\phi))$ is a $(\phi + q - 1, \phi + q - 1)$-difference pair.*

Proof. *For any integer d, $0 \leq d \leq n-1$, we find two integers x and y in $S(\phi)$ such that $x - y \equiv d \pmod{n}$. Let $S(\phi) = \{0, 1, \cdots, \phi - 1, s_1, s_2, \cdots, s_{q-1}\}$.*

Case 1: $0 \leq d \leq \phi - 1$. We have $d - 0 \equiv d \pmod{n}$. By definition, both d and 0 are included in $S(\phi)$. Therefore $x = d$ and $y = 0$.

Case 2: $\phi - 1 < d \leq s_1$. By definition $s_1 - (\phi - 1) \leq (2\phi - 1) - (\phi - 1) = \phi$. Let $u = s_1 - d$, we obtain $u \leq \phi - 1$. By definition u must be included in $S(\phi)$. Note $u = s_1 - d$ implies $s_1 - u \equiv d \pmod{n}$. Therefore $x = s_1$ and $y = u$.

Case 3: $s_{i-1} < d \leq s_i$ for some i, $2 \leq i \leq q-1$. By definition $s_i - s_{i-1} \leq \phi$. Let $v = s_i - d$, we obtain $v \leq \phi - 1$. By definition u must be included in $S(\phi)$. Note $v = s_i - d$ implies $s_i - v \equiv d \pmod{n}$. Therefore $x = s_i$ and $y = v$.

Case 4: $s_{q-1} < d \leq n-1$. By definition, $s_{q-1} \geq (n-1)/2$. Let $d' = n - d$, we have $1 \leq d' < n - s_{q-1}$. This leads to $1 \leq d' < n - (n-1)/2 = (n+1)/2$. Since d' is an integer, $1 \leq d' < (n+1)/2$ implies $1 \leq d' \leq \lceil (n-1)/2 \rceil$. Note $s_{q-1} \geq (n-1)/2$ implies $s_{q-1} \geq \lceil (n-1)/2 \rceil$. Thus we have $1 \leq d' \leq s_{q-1}$. Applying the arguments as given in Cases 1, 2, and 3, we can find x' and y' in $S(\phi)$ 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 3.1 *Given n and ϕ, where $1 \leq \phi \leq n$. Let $p = \left\lceil \frac{n}{\phi} \right\rceil$ and $q = \left\lceil \frac{n+1}{2\phi} \right\rceil$. The pair*

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

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

Proof. *This is a direct consequence of Lemmas 3.2, 3.3 and 3.1.* ■

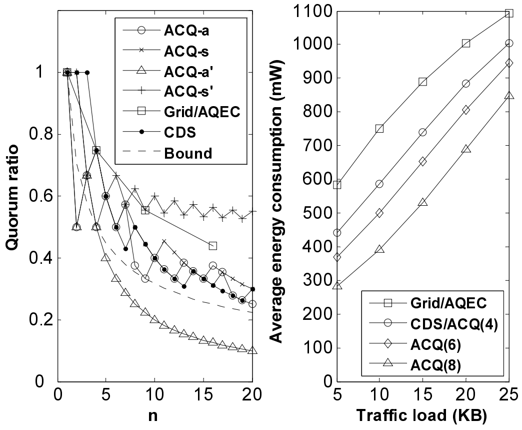
By employing the ACQ scheme, each station is able to obtain either an a-quorum or s-quorum in $O(1)$ time. Note that for ease of implementation, a station can simply choose $A(\phi)$ and $S(\phi)$ as its own a-quorum and s-quorum respectively. We explore the effect of ϕ in the next section.

4 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 [1]. To see the improvement purely resulted from MAC layer, we consider the singleton topology consisting of a two-hop cluster in the network. 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 [7]. The mean 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 rates for transmit, receive, idle, and sleep modes in the wireless module are set 1650, 1400, 1150, and 45 mW respectively [14]. We also implement the timer synchronization mechanism [7].

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

Quorum ratio. Proportion of the beacon intervals that



(a) (b)

Figure 3. Quorum ratio and average energy consumption rate.

required to be awake in each cycle. This metrics equals $|Q|/n$, where $|Q|$ and n denote the quorum size and cycle length respectively.

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 and reception in each hop. Note this metrics includes data buffering time on a sending station.

We compare the ACQ scheme with the CDS [19, 26], Grid [13, 23], and AQEC [4] quorum schemes. To focus on the scheme comparison, we employ the same protocol design as introduced in Section 2 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 $\phi = 4$.

4.1 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 $\phi = \left\lceil \sqrt{(n+1)/2} \right\rceil$ and $\phi' = \lceil (n+1)/2 \rceil$ respectively. As shown in Figure 3(a), for small ϕ the quorum ratios of ACQ are very close to that of CDS and approaches the theoretical bound $1/\sqrt{n}$ [13]. By increasing ϕ, we are able to "transfer" quorum ratio from a-quorums to s-quorums. The generating set $A(\phi')$ is able to produce a-quorums with quorum ratio merely 33% of that given by CDS (when $n = 20$) and below the theoretical bound. Notice that the quorum ratio of s-quorums becomes higher as ϕ 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 scalability since the cycle length is expected to be linearly (and unitarily) configurable for various network conditions (e.g., node mobility, delay requirements, etc.).

4.2 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 $\phi = 4$, 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 $\phi = 4$. Therefore, we present their results with the same line. Under all loads, ACQ yields better energy efficiency, and the improvement becomes significant as ϕ 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.

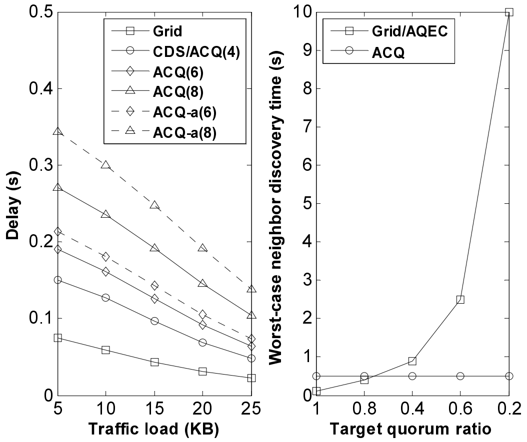
4.3 Average Delay

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 only the delay encountered when sending packets to 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. Since s-quorums require the clusterhead to be awake more as ϕ increases, the overall delay is reduced.

4.4 Neighbor Discovery Time

In this section, we study the worst-case neighbor discovery time given that the target quorum ratio must be met.



(a) (b)

Figure 4. Average delay and worst-case neighbor discovery time.

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 when different values of ϕ are set. We fix the cycle length $n = 5$ for ACQ. The performance results are shown in Figure 4(b). While ACQ offers stable neighbor discovery time as ϕ 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.

References

[1] The network simulator. *http://www.isi.edu/nsnam/ns*.

[2] A. Amis and R. Prakash. Load-balancing clusters in wireless ad hoc networks. In *Proc. of ASSET*, 2000.

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

[4] C. Chao, J. Sheu, and I. Chou. An adaptive quorum-based energy conserving protocol for ieee 802.11 ad hoc netowrks. *IEEE Transactions on Mobile Computing*, 5(5):560–570, 2006.

[5] W. Chen, N. Jain, and S. Singh. Anmp: Ad hoc network management protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1506–1531, 1999.

[6] Z. Chou. A randomized power management protocol with dynamic listen interval for wireless ad hoc networks. In *Proc. of Vehicular Technology Conference (VTC-Spring)*, 2006.

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

[8] L. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proc. of INFOCOM*, 2001.

[9] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[10] T. Hou and T. Tsai. An access-based clustering protocol for multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 19(7):1201–1210, 2001.

[11] L. Huang and T. Lai. On the scalability of ieee 802.11 ad hoc networks. In *Proc. of MobiHoc*, 2002.

[12] A. Iwata, C. Chiang, G. Pei, M. Gerla, and T. Chen. Scalable routing strategies for ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1369–1379, 1999.

[13] 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*, 10(1-2):169–181, 2005.

[14] E. Jung and N. Vaidya. An energy efficient mac protocol for wireless lans. In *Proc. of INFOCOM*, 2002.

[15] U. Kozat, G. Kondylis, B. Ryu, and M. Marina. Virtual dynamic backbone for mobile ad hoc networks. In *Proc. of ICC*, 2001.

[16] R. Kravets and P. Krishnan. Application-driven power management for mobile communication. *Wireless Networks*, 6(4):263–277, 2000.

[17] C. Lin, G. Chiu, and C. Cho. A new quorum-based scheme for managing replicated data in distributed systems. *IEEE Transactions on Computers*, 51(12):1442–1447, 2002.

[18] C. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, 1997.

[19] W. Luk and T. Wong. Two new quorum based algorithms for distributed mutual exclusion. In *Proc. of ICDCS*, 1997.

[20] R. Rajaraman. Topology control and routing in ad hoc networks: A survey. *ACM SIGACT News*, 33(2):66–73, 2002.

[21] J. Sheu, C. Chao, and C. Sun. A clock synchronization algorithm for multi-hop wireless ad hoc networks. In *Proc. of ICDCS*, 2004.

[22] P. Sinha, R. Sivakumar, and V. Bharghavan. Enhancing ad hoc routing with dynamic virtual infrastructures. In *Proc. of INFOCOM*, 2001.

[23] Y. Tseng, C. Hsu, and T. Hsieh. Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks. In *Proc. of INFOCOM*, 2002.

[24] 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*, 4(1):59–70, 2002.

[25] O. Younis and S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, 2004.

[26] 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*, 5(9):1228–1241, 2006.