Heterogenous Quorum-based Wakeup Scheduling for Duty-Cycled Wireless Sensor Networks

Shouwen Lai

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Binoy Ravindran, Chair
Paul Plassmann
Y. Thomas Hou
Anil Vullikanti
Yaling Yang

May 5, 2009
Blacksburg, Virginia

Keywords: Quorum, Duty Cycle, MAC, Wireless Sensor Network

Heterogenous Quorum-based Wakeup Scheduling for Duty-Cycled Wireless Sensor Networks

Shouwen Lai

(ABSTRACT)

In duty-cycled wireless sensor networks, time is organized into consecutive small time slots. A node is either wholly awake in select slots (defined as slotted listening mode) or fractionally awake in every slot (defined as low power listening mode). In duty-cycled sensor networks which are not clock-synchronized, it is a non-trival problem to guarantee that two neighbor nodes discover each other within bounded latency.

In this dissertation proposal, we first present quorum-based asynchronous wakeup scheduling schemes for duty-cycled wireless sensor networks with slotted listening mode. The schemes organize time slots as quorum systems, and the slots in which a node will be awake are referred to as quorums. The goal is to ensure that two neighboring nodes that adopt such quorums as their wakeup schedules can hear each other at least once in bounded time slots. We propose two designs: *cyclic quorum system pair* (or *cqs-pair*) and *grid quorum system pair* (or *gqs-pair*).

The cqs-pair contains two cyclic quorum systems from which any two quorums will have a non-empty intersection. The cqs-pair design provides an optimal solution in terms of energy saving ratio for asynchronous wakeup scheduling. To quickly assemble a cqs-pair, we present a fast construction scheme which is based on the *multiplier theorem* and the (N, k, M, l)-difference pair defined by us. Regarding the gqs-pair, we prove that any two grid quorum systems will automatically form a gqs-pair. We analyze the performance of both designs, in terms of average discovery delay, quorum ratio, and energy saving ratio. We show that our designs achieve better trade-off between the average discovery delay and quorum ratio (and thus energy consumption) for different cycle lengths.

We also present rendezvous mechanisms for duty-cycled sensor networks with low power listening (LPL) mode. Our protocol is called Q-MAC, which combines quorum-based wakeup scheduling with low-power listening, to provide an asynchronous neighbor discovery, run-time configurable, and an ultra low duty cycle (i.e., 1%) solution for wireless sensor networks. Q-MAC provides configuration flexibility in duty cycle by selecting different pairwise quorums as preamble sampling schedules, which is different from the conventional approach of periodic preamble sampling as in B-MAC [1] and X-MAC [2] protocols. We show that Q-MAC can guarantee asynchronous neighbor discovery within bounded latency. Q-MAC's quorum-based wakeup scheduling is based on *cqs-pair*.

We implemented the proposed designs in a wireless sensor network platform consisting of Telosb motes. Our implementation-based measurements further validate the analytically-established performance trade-off of cqs-pair, gps-pair, and Q-MAC.

Based on these results, we propose several research directions for post-preliminary exam study. We propose to further improve the energy efficiency of quorum-based asynchronous wakeup scheduling mechanisms/protocols by asymmetric design. Another major direction is to develop cross-layer optimizations including that of routing, with such scheduling mechanisms/protocols. In addition, we also propose to develop capacity maximization solutions and support for efficient multicast and broadcast with asynchronous wakeup.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Duty-cycled Wireless Sensor Networks

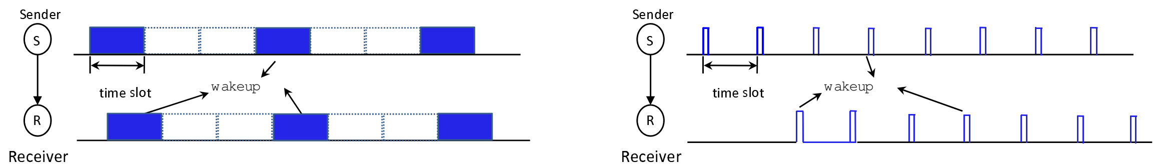
Wireless sensor networks (WSNs) [4] have recently received increased attention for a broad array of applications such as surveillance [5], environment monitoring [6], medical diagnostics [7], and industrial control [8]. As wireless sensor nodes usually rely on portable power sources such as batteries to provide the necessary power, their efficient power management is a critical issue.

It has been observed that idle energy plays an important role for saving energy in wireless sensor networks [9]. Most existing radios used in wireless sensor networks (i.e., CC2420 [10]) support different modes, like the transmit/receive mode, the idle mode, and the sleep mode. In the idle mode, the radio is not communicating but the radio circuitry is still turned on, resulting in energy consumption which is only slightly less than that in the transmitting or receiving states. Thus, a better way to optimize energy is to shut down the radio as much as possible in the idle mode [9], and it is desirable to have low duty cycle operation when the network is idle.

Supposing that time is arranged into consecutive and equal time slots, there are two modes for low duty cycle operation: slotted listening mode [11, 12] and low power listening mode [1]. In the slotted listening mode, a node is wholly awake in select slots and sleeps in the remaining slots when there is no data transmission or reception. In the low power listening (or LPL) mode, a node will be fractionally awake in every slot.

We define duty cycle as the percentage of active time of a node during its entire operational time. Generally, the duty cycle in the LPL mode is lower than that in the slotted listening mode.

Although low duty-cycled operation can increase energy efficiency in WSNs, neighbor discovery (or rendezvous) becomes more complex than that in non duty-cycled networks, since we cannot guarantee that two nodes are awake simultaneously.



(a) slotted listening (b) low power listening

Figure 1.1: Duty-cycled operation in radio of wireless sensor networks

1.2 Wakeup Scheduling and Rendezvous Mechanisms

In order to save more idle energy, it is necessary to introduce a wakeup scheduling mechanism in which a node sleeps in more slots in the idle state for duty-cycled WSNs in the presence of pending transmissions [13, 14]. The major objective of wakeup scheduling and corresponding rendezvous mechanism for neighbor discovery is to maintain network connectivity while reducing the idle state energy consumption. Existing wakeup scheduling mechanisms fall into three categories: on-demand wakeup, scheduled wakeup, and asynchronous wakeup.

In on-demand wakup mechanisms [15–18], out-of-band signaling or operational cycle is used to wake up sleeping nodes in an on-demand manner. For example, with the help of a paging signal, a node listening on a page channel can be woken up. As page radios can operate at lower power consumption, this strategy is very energy efficient. However, it suffers from increased implementation complexity.

In scheduled wakeup mechanisms [19–21], low-power sleeping nodes wake up at the same time, periodically, to communicate with one another. Examples include the S-MAC protocol [19] and the multi-parent schemes protocol [13]. In such mechanisms, all nodes maintain periodic sleep-listen schedules based on locally managed synchronization. Neighboring nodes form virtual clusters to set up a common sleep schedule.

The third category, asynchronous wakeup mechanisms [1,2,12,22,23], are also well studied. Compared to the scheduled rendezvous wakeup mechanism, asynchronous wakeup does not require clock synchronization. In this approach, each node follows its own wakeup schedule in the idle state, as long as the wakeup intervals among neighbors overlap. To meet this requirement, nodes usually have to wakeup more frequently than in the scheduled rendezvous mechanism. However, there are many advantages of asynchronous wakeup, such as easiness in implementation and low message overhead for communication. Furthermore, it can ensure network connectivity even in highly dynamic network environments.

The quorum-based wakeup scheduling paradigm, sometimes called quorum-based power saving (QPS) protocol [11, 24–26], has been proposed as a powerful solution for asynchronous wakeup scheduling in the slotted listening mode. In a QPS protocol, the time axis on each node is evenly divided into beacon intervals. Given an integer n, a quorum system defines a cyclic pattern, which specifies the awake/sleep scheduling pattern during n continuous beacon intervals for each node. We call n the *cycle length*, since the pattern repeats every n beacon intervals. A node may stay awake or sleep during each beacon interval. QPS protocols can guarantee that at least one awake interval overlaps between two adjacent (or neighboring) nodes, with each node being awake for only $O(\sqrt{n})$ beacon intervals.

Most previous works only consider *homogenous* quorum systems for asynchronous wakeup scheduling [12], which means that quorum systems for all nodes have the same *cycle length* and same pattern. However, many WSNs are increasingly heterogenous in nature—e.g., the network nodes are grouped into clusters, with each cluster having a high-power cluster head node and low-power cluster member nodes [27–32]. Thus, it is desirable that heterogenous sensor nodes (i.e., cluster-heads and cluster members) have *heterogenous* quorum-based wakeup schedules (or different *cycle lengths*). We describe two quorums from different quorum systems as *heterogenous quorums* in this proposal. If two adjacent nodes adopt heterogenous quorums as their wakeup schedules, then they have different cycle lengths and hence different wakeup patterns. Thus, the fundamental problem becomes how to guarantee the intersection property for heterogenous quorums and apply them to wakeup scheduling for WSNs with the slotted listening mode.

It is also necessary to investigate the asynchronous rendezvous mechanism in the LPL mode with quorum-based wakeup scheduling. This is because, neighbor discovery between any two nodes becomes more difficult when two neighbor nodes are not always simultaneously powered on in each time slot. The rendezvous mechanism in traditional MAC protocols such as B-MAC [1], X-MAC [2], and WISEMAC [22] for the LPL mode work as follows: in case of data transmission, the sender transmits long preambles or short strobed preambles that are long enough to notify a receiver who is also duty cycling to be awake; in the idle state, all nodes use low power listening by sampling the state of channels.

Although B-MAC and X-MAC achieve excellent performance for idle energy saving, they are less appealing in terms of flexible configuration. In particular, they assume fixed time slots (or channel checking interval) , i.e., 100ms or 200ms. However, once the application is deployed, it is almost not feasible to change parameters like the checking interval and the duty cycle of radio in the idle state. However, *run-time configuration* of these parameters is desirable for a variety of reasons.

First, many wireless sensor networks adopt a tiered topology [27, 28] in which there are heterogenous entities, i.e., cluster head and cluster members, which have different energy saving requirements. The election of different entities in a network is usually dynamic or through rotation [33, 34]. For example, a cluster head is selected based on its remaining energy. Such dynamic role changes require run-time configuration of the duty cycle of the radio.

Second, environmental changes, e.g., seasonal changes in wild fire monitoring, also require run-time configuration of duty cycles. For example, a lower duty cycle during the summer season and a higher duty cycle (which implies a shorter checking interval and lower discovery latency) during the winter season are often desirable. In addition, a mobile node may want to change its duty cycle parameters after joining a new network when the node does not have much a-priori knowledge of the network.

Thus, it is desirable to design a flexible asynchronous rendezvous mechanism in the LPL mode, for achieving both flexility of run-time configuration toward optimizing energy, as well as bounded neighbor discovery latency.

1.3 Summary of Current Research and Contributions

Our main goal in designing flexible asynchronous wakeup scheduling both in slotted listening mode and in the LPL mode is to maintain network connectivity. Here we use the term "connectivity" loosely, in the sense that a topologically connected network in our context may not be connected at any time; instead, all nodes are reachable from any node within a finite amount of time.

Towards that end, we have designed a heterogenous quorum-based wakeup scheduling mechanism for WSNs with slotted listening mode. We then extend the design to duty-cycled WSNs with LPL mode, which is more energy efficient and sacrifice little in neighbor discovery latency.

Our current research results and contributions are summarized as follows:

— *cqs-pair* [35]. We have developed a quorum-based asynchronous wakeup scheduling mechanism called *cyclic quorum system pair* (or *cqs-pair*). The cqs-pair mechanism guarantees that two adjacent nodes which adopt heterogenous cyclic quorums from such a pair as their wakeup schedules can hear each other at least once within one super cycle length (i.e., the larger cycle length in the cqs-pair).

We also developed a fast algorithm for constructing cqs-pairs, using the*multiplier theorem* [36] and the (N, k, M, l)-difference pair defined by us. Given a pair of cycle lengths (n and m, $n \leq m$), we show that the cqs-pair is an optimal design in terms of energy saving ratio. The fast construction scheme significantly improves the speed of finding an optimal quorum, in contrast to previous exhaustive methods [37].

We also analyze the performance of cqs-pair in terms of expected delay ($\frac{n-1}{2} < E(delay) < \frac{m-1}{2}$), quorum ratio, and energy saving ratio.

We also analyze the performance of cqs-pair in terms of expected delay, quorum ratio, and energy saving ratio, and derive analytical expressions that upper and lower bound these metrics, e.g., the expected delay is bounded by $\frac{n-1}{2} < E(delay) < \frac{m-1}{2}$, where n and m are the cycle lengths of two quorum systems in the cqs-pair.

With the help of the cqs-pair, we show that heterogenous WSNs can achieve better trade-off between energy consumption and average delay. For example, all cluster-heads and gateway nodes can select a quorum from the quorum system with smaller cycle length as their wake up schedules, to obtain smaller discovery delay. In addition, all members in a cluster can choose a quorum from the system with longer cycle length as their wakeup schedules, in order to save more idle energy.

— *gqs-pair* [38]. We have also developed another quorum-based asynchronous wakeup scheduling mechanism called the *grid quorum system pair* (or *gqs-pair*). In this design, each quorum system of the pair is a grid quorum system [37]. We prove that any two grid quorum systems can form a gqs-pair.

When compared with cqs-pair, gqs-pair has better performance in terms of average neighbor discovery latency. We show that for a gqs-pair with $\sqrt{n} \times \sqrt{n}$ grid and $\sqrt{m} \times \sqrt{m}$ grid, the average discovery delay is bounded within $\frac{(n-1)(\sqrt{n}+1)}{3\sqrt{n}} < E(Delay) < \frac{(m-1)(\sqrt{m}+1)}{3\sqrt{m}}$, while the quorum ratios are $\frac{2\sqrt{n}-1}{n}$ and $\frac{2\sqrt{m}-1}{m}$, respectively. With the help of the gqs-pair, WSNs can also achieve better trade-off between energy consumption and average delay.

— *Q-MAC* [39]. Based on the cqs-pair mechanism, we developed a quorum-based asynchronous duty cycling MAC protocol, called Q-MAC. We use the cqs-pair concept to design Q-MAC by combining dual preamble sampling to provide a flexible configuration solution for asynchronous low power listening mechanism at the MAC layer. In Q-MAC, a node will not follow periodic, duty cycled low-power listening (LPL) as in B-MAC or X-MAC, but instead follow a quorum-based LPL schedule. To use Q-MAC, a node can choose its quorum-based LPL schedule so long as the quorum schedule will intersect at least once with the quorum schedules of its neighboring nodes in bounded time.

We show that Q-MAC guarantees low duty cycle (e.g., 1%) and yet ensures the discovery of neighboring nodes within bounded delay. The primary advantage of Q-MAC is run-time configuration, which allows flexible adjustment of the power saving policy to reflect the available energy or different workloads in heterogenous networks. Although Q-MAC requires global agreement on some basic LPL parameters, such as the length of the channel checking interval and the preamble sampling period (equal to the awake time) in each interval, this does not preclude Q-MAC from independently choosing its quorum-based LPL pattern.

— *Implementation* [38, 39]. We implemented cqs-pair, gqs-pair, and Q-MAC in a wireless sensor network platform comprised of Telosb motes [40] running the TinyOS operating system. We implemented Q-MAC atop the LPL interface in TinyOS. The radio used by TelosB is the Chipcon CC2420, which is an 802.15.4 compliant device. We developed a variety of APIs that can be used by WSN applications to initiate the mechanisms and protocols. Our implementation did not require any modifications to upper layer protocols such as that for routing. Using this implementation, we experimentally evaluated the performance of the proposed mechanisms and compared them against existing solutions (e.g., B-MAC, X-MAC) in terms of energy-saving ratio, neighbor discovery latency, and deliver ratio. The experimental evaluation reveal that Q-MAC can achieve flexible tradeoffs between saving energy and bounding end-to-end transmission delay.

1.4 Summary of Proposed Post Preliminary-Exam Work

Based on these research results, we propose the following major research directions for the post-preliminary exam work:

• *Improved quorum-based asynchronous protocols.* We propose to investigate the possibility of further improving the energy efficiency of quorum-based asynchronous wakeup scheduling mechanisms and MAC protocols by *asymmetric design*. We observe that it is not necessary to always guarantee the quorum intersection property in the idle state since there is no data for transmission and nodes do not need to find out about each other in such cases. The intersection is only needed in case of data transmission via unicast or multicast. Several directions can be considered to reduce the quorum intersections. For example, we can consider the data transmission as a *write operation* and the idle listening as a *read operation*. Now, we can consider the concepts of read quorum, write quorum, and read-write quorum in a quorum group. In order to save energy, it is only necessary to guarantee the intersection property between some quorums, e.g., read quorums will not intersect with each other, but a read-quorum will intersect with a write quorum or a read-write quorum. Hence, if a node adopts the read quorum in the idle state, and switches to the write-quorum or the read-write quorum in case of data transmission, we can guarantee network connectivity, and meanwhile, provide higher energy efficiency.

We propose to design improved quorum-based asynchronous protocols, which are based on such quorum groups. An example design is a grid quorum group, i.e., a read quorum consisting of a column of elements in the grid, a write quorum consisting of a row of elements in the grid, and a read-write quorum consisting of a row plus a column of elements in the grid. We propose to design such protocols based on quorum groups to achieve better energy saving ratio and discovery latency, and that can be easily implemented for WSNs.

• *Cross-layer design with routing protocol adaption.* With quorum-based wakeup scheduling or LPL scheduling at the MAC layer, both the one-hop delivery latency and the end-to-end delay will be affected. Here, we refer to the delivery latency as the neighbor discovery latency after introducing the proposed quorum-based mechanism. If we denote the latency of a node discovering another node as the *cost* of a link connecting the two nodes, then the cost of the links will be continuously changing. In WSNs, with such dynamic “link costs’, how to find an optimal shortest routing path that yields the shortest end-to-end data delivery latency is a non-trivial problem. Traditional shortest path algorithms such as Bellman-Ford [41] and Dijkstra [42] cannot be used directly for finding such a shortest path because, at different temporal points, the cost of each link varies. Thus, the routing path that was built in the last time slot will not be valid in the current time slot anymore.

We propose to formally model this problem and solve it by developing variants of existing routing protocols, e.g., AODV [43], MintRoute [44], to construct a new routing protocol over quorum-based asynchronous MAC protocols. The proposed new routing protocol will be significantly different from conventional works in networks where the link cost was unchanged or changed slowly. It will have temporal adaptation feature such as that in [45], and will also need support from the underlying MAC layer. We propose to design such a protocol, establish its theoretical properties, implement it over a WSN platform of Telosb motes, and conduct experimental studies.

In addition to these major directions, we also propose the following minor research directions:

• *Capacity maximization.* Although quorum-based wakeup scheduling is energy efficient, it has the cost of additional neighbor discovery delay which may reduce the overall system transmission capacity. The additional neighbor discovery latency will not significantly affect applications with low traffic. But for traffic-intensive applications, such as data aggregation, the additional latency may negatively affect their performance. We propose to identify the factors that affect network capacities under quorum-based scheduling mechanisms. We propose to develop solutions to maximize the capacity, e.g., random quorum selection, practical backoff time setting, for applications with high traffic load such as data aggregation.

• *Broadcast/multicast support.* The dissertation proposal's quorum-based asynchronous wakeup scheduling mechanisms and MAC protocols support broadcast and multicast transmissions (see Chapter 5). However, these broadcast and multicast mechanisms have disadvantages. For example, a high number of RTS messages may be send out to trigger receivers to wake up to receive the broadcast/multicast data. A better solution may be to extend the cqs-pair concept to cqs m-pair in which m cyclic quorum systems have the heterogenous rotation closure property with one another. We propose to develop cqs m-pair-based asynchronous wakeup scheduling mechanisms to support broadcast and multicast.

1.5 Proposal Outline

The rest of the proposal is organized as follows: We overview past and related works and compare them with our work in Chapter 2. In Chapter 3, we outline the basic preliminaries of quorum-based power-saving protocols. The detailed design of heterogenous quorum systems pair (i.e., cqs-pair and gqs-pair) is discussed in Chapter 4. In this chapter, we present our cqs-pair construction scheme, and analyze the performance of cqs-pair and gqs-pair. We also describe our implementation for cqs-pair and gps-pair in Chapter 4, and report our experimental measurements. We present the design of Q-MAC and its performance in Chapter 5. We conclude, summarize our contributions, and describe the proposed post-exam work in detail in Chapter 6.

Chapter 2

Past and Related Work

Wakeup scheduling and corresponding neighbor discovery (rendezvous) mechanisms for wireless sensor networks can be broadly classified into three categories. We summarize and overview them as follows.

2.1 On-Demand Wakeup

The implementation of on-demand wakeup schemes [15, 18, 46] typically requires two different channels: a data channel and a wakeup channel for awaking nodes as and when needed. It is assumed that the nodes can be signaled and awakened at any point of time and then a message is sent to the node. This is usually implemented by employing two wireless interfaces. The first radio is used for data communication and is triggered by the second ultra low-power (or possibly passive) radio which is used only for paging and signaling. This allows for the immediate transmission of a signal on the wakeup channel if a packet transmission is in progress on the other channel, thus reducing the wakeup latency.

STEM [15] and its variation [16], and passive radio-triggered solutions [17] are examples of this class of wakeup methods. The drawback is the additional cost for the second radio. The STEM (Sparse Topology and Energy Management) work [15] uses two different radios for wakeup signals and data packet transmissions, respectively. The key idea is that a node remains awake until it has not received any message destined for it for a certain period of time. STEM uses separate control and data channels, and hence the contention among control and data messages is alleviated. The energy efficiency of STEM is dependent on that of the control channel.

Thus, although these methods can be optimal in terms of both delay and energy, they are not yet practical. The cost issues, currently limited available hardware options which results in limited range and poor reliability, and stringent system requirements prohibit the widespread use and design of such wakeup techniques.

2.2 Synchronized Rendezvous Wakeup

In this class [19, 20, 22, 47–49], the nodes follow deterministic (or possibly random) wakeup patterns. Time synchronization among the nodes in the network is generally assumed. These schemes require that all neighboring nodes wake up at the same time.

The simplest way is by using a Fully synchronized pattern, like that in the S-MAC protocol [19]. In this case, all nodes in the network wakeup at the same time according to a periodic pattern. S-MAC follows a virtual clustering approach to synchronize the nodes to a common wakeup scheme with a slotted structure. By regularly broadcasting SYNC packets at the beginning of a slot, neighboring nodes can adjust their clocks to the latest SYNC packet in order to correct relative clock drifts.

In a bootstrapping phase, nodes listen for incoming SYNC packets in order to join the wireless sensor networks, and join a virtual synchronization cluster. When hearing no SYNC's, a node starts alternating in its wake-up pattern and propagates its schedule with SYNC messages. A problem of the virtual clustering arises when several clusters evolve. Bordering nodes in-between two clusters have to adopt the wake-patterns of both clusters, which imposes twice the duty cycles to these nodes. An S-MAC slot consists in a listen interval and a sleep interval. The listen interval is fragmented into a synchronization window to exchange SYNC messages, and a second and third window dedicated to RTS-CTS exchange. Nodes with receiving a RTS traffic announcement will clear the channel with a CTS respective window, and stay awake during the sleep phase, whereas all other nodes will go back to sleep.

The slot length and duty cycle must be set in a fixed manner, which severely restrains latency and maximal throughput. This can be disadvantageous, as traffic can often be of bursty nature and the rate of traffic can vary over time.

A further improvement can be achieved by allowing nodes to switch off their radio when no activity is detected for at least a timeout value, like that in the T-MAC protocol [20]. In T-MAC, the listen interval ends when no activation event has occurred for a given time threshold. An activation event may be the sensing of any communication on the radio, the end-to-end transmission of a node's data transmission, overhearing a neighbor's RTS or CTS which may announce traffic destined to itself. One drawback of T-MAC's adaptive time-out policy is that nodes often go to sleep too early.

The disadvantages of scheduled rendezvous schemes include the complexity and the overhead for synchronization.

2.3 Asynchronous Wakeup

Asynchronous wakeup scheduling. B-MAC [1] is a CSMA-based technique that utilizes low power listening and an extended preamble for rendezvous. Nodes wake-up and sleep independently. If a sender wishes to transmit, it precedes the data packet with a preamble that is slightly longer than the sleep period of the receiver. During the awake period, a node samples the medium and if a preamble is detected, it remains awake to receive the data. With the extended preamble, a sender is assured that at some point during the preamble, the receiver will wake up, detect the preamble, and remain awake in order to receive the data. While B-MAC performs quite well in idle listening, it suffers from the overhearing problem, and the long preamble dominates the energy usage.

XMAC [2] and DPS-MAC [50] was proposed to improve B-MAC, in which a short preamble was proposed to replace the long preamble in B-MAC. Also, there is receiver information embedded in the short preamble to avoid the overhearing problem. The main disadvantage of B-MAC, X-MAC, and DPS-MAC is that it is difficult to reconfigure the protocols after deployment, and thus they lack flexibility.

quorum design. The concept of quorum systems, which are widely used in the design of distributed systems [51–56] for the application of data replicas, mutual exclusion and fault tolerance. A quorum system is a collection of sets such that the intersection of any two sets is always non-empty. There are two widely used quorum systems [37]: cyclic quorum system and grid quorum systems.

quorum-based wakeup scheduling [12, 57]. This was first introduced in [11] in the context of IEEE 802.11 ad hoc networks. The authors proposed three different asynchronous sleep/wakeup schemes that require some modifications to the basic IEEE 802.11 Power Saving Mode. More recently, Zheng *et al.* [12] took a systematic approach toward designing asynchronous wakeup mechanisms for ad hoc networks (which is also applicable for WSNs). They formulate the problem of generating wakeup schedules as a block design problem and derive theoretical bounds under different communication models. The basic idea is that each node is associated with a Wakeup Schedule Function (WSF) that is used to generate a wakeup schedule. For two neighboring nodes to communicate, their wakeup schedules must overlap regardless of their clock difference.

For the quorum-based asynchronous wakeup design, Luk and Wong [37] designed a cyclic quorum system using difference sets. However, they perform an exhaustive search to obtain a solution for each cycle length N, which is impractical when N is large.

Asymmetric quorum design [58]. In the clustered environment of sensor networks, it is not always necessary to guarantee all-pair neighbor discovery. The Asymmetric Cyclic Quorum (ACQ) system [58] was proposed to guarantee neighbor discovery between each member node and the clusterhead, and between clusterheads in a network. The authors also presented a construction scheme which assembles the ACQ system in $O(1)$ time to avoid exhaustive searching. ACQ is a generalization of the cyclic quorum system. The scheme is configurable for different networks to achieve different distribution of energy consumption between member nodes and the clusterhead.

However, it remains a challenging issue to efficiently design an asymmetric quorum system given an arbitrary value of n. One previous study [12] shows that the problem of finding an optimal asymmetric block design can be reduced to the minimum vertex cover problem, which is NP-complete. However, the ACQ [58] construction is not optimal in that the quorum ratio for symmetric-quorum is $\phi = \lceil \frac{n+1}{2} \rceil$ and the quorum ratio for asymmetric-quorum is $\phi' = \lceil \sqrt{\frac{n+1}{2}} \rceil$. Another drawback is that it cannot be a solution to the h-QPS problem since the two asymmetric-quorums cannot guarantee the intersection property.

Transport layer approach. Wang *et al.* [59] applied quorum-based wakeup scheduling at the transport layer which can cooperate with any MAC-layer protocol, allowing for the reuse of well-understood MAC protocols. The proposed technique saves idle energy by relaxing the requirement for end-to-end connectivity during data transmission and allowing the network to be disconnected intermittently via scheduled sleeping. The limitation of this work is that each node adopts the same grid quorum system as its wakeup schedule, and the quorum ratio is not optimal when compared with that of cyclic quorum systems.

Schedules based on Chinese Remainder Theorem. In [57], the authors present a mechanism called Disco which is a simple adaptation of the Chinese Remainder Theorem [60]. They show that Disco can ensure asynchronous neighbor discovery in bounded time, even if nodes independently set their own duty cycles. Another work [61] called C-MAC adopts similar mechanism for wakeup scheduling in WSNs.

In [62], Kuo *et. al.* adopt relative primes as the wakeup schedules for neighbor nodes in order to support multicast in asynchronous wakeup mechanisms. The main principle is the intersection property from Chinese Remainder Theorem [60]. The limitation of this mechanism is that the discovery latency is usually too long, i.e., over 100 slots for a $(13, 17)$ prime pair in [57], to satisfy the delay constraints of many WSN applications, which prevent their practical applications.

Chapter 3

Preliminaries

3.1 Network Model and Assumptions

We represent a multi-hop wireless sensor network by a directed graph $G(V, E)$, where V is the set of network nodes ($|V| = N$), and E is the set of edges. If node v_j is within the transmission range of node v_i, then an edge (v_i, v_j) is in E. We assume bidirectional links. The major objective of quorum-based wakeup scheduling is to maintain network connectivity regardless of clock drift. Here, we use the term “connectivity” loosely, in the sense that a topologically connected network in our context may not be connected at any time; instead, all the nodes are reachable from a node within a finite amount of time.

We also make the following assumptions: (1) Time axes is arranged as consecutive short time intervals or slots, and all slots have the same duration; (2) There is no time synchronization among nodes; thus the time slots in two nodes are not necessarily aligned; (3) In idle mode, at the beginning of a time interval, a node may or may not check the state of its channel, depending on its wakeup or LPL schedule; and (4) The overhead of turning on and shutting down radio is negligibly small.

As for the first assumption, the length of one time interval depends on application-specific requirements. For example, for a radio compliant with IEEE 802.15.4 [63, 64], the bandwidth is approximately 128kb/s and a typical packet size is less than 512KB. Given this, the slot length (i.e., the beacon interval) can be approximately $50ms$.

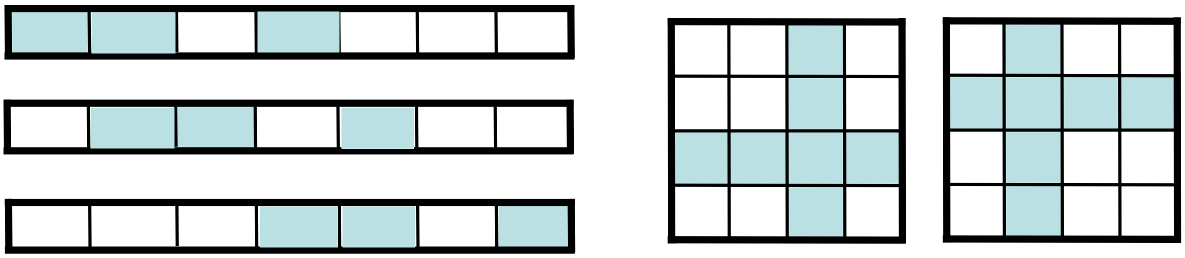


Figure 3.1: Cyclic Quorum System (left) and Grid Quorum System (right)

3.2 Quorum-based wakeup scheduling

3.2.1 Quorum system

We use the following definitions for quorum systems. Given a cycle length n, let $U = \{0, \cdots, n-1\}$ be an universal set.

Definition 1. *A quorum system $\mathcal{Q}$ under U is a superset of non-empty subsets of U, each called a quorum, which satisfies the intersection property:* $\forall G, H \in \mathcal{Q} : G \cap H \neq \emptyset$.

Definition 2. *Given an integer $i \geq 0$ and quorum G in a quorum system $\mathcal{Q}$ under U, we define* $G + i = \{(x + i) \, mod \, n : x \in G\}$.

Definition 3. *A quorum system $\mathcal{Q}$ under U is said to have the* rotation closure property *if* $\forall G, H \in \mathcal{Q}$, $i \in \{0, 1, ...n-1\}$*:* $G \cap (H + i) \neq \emptyset$.

There are two widely used quorum systems, *grid quorum system* and *cyclic quorum system*, that satisfy the *rotation closure property*.

Grid quorum system [37]. In a grid quorum system, shown in Figure 3.1, elements are arranged as a $\sqrt{n} \times \sqrt{n}$ array (square). A quorum can be any set containing a column and a row of elements in the array. The quorum size in a square grid quorum system is $2\sqrt{n} - 1$. An alternative is a "triangle" grid-based quorum in which all elements are organized in a triangle fashion. The quorum size in "triangle" quorum system is approximately $\sqrt{2}\sqrt{n}$.

Cyclic quorum system [37]. A cyclic quorum system is based on the ideas of cyclic block design and cyclic difference sets in combinatorial theory [36]. The solution set can be strictly symmetric for arbitrary n. For example, the set $\{1, 2, 4\}$ is a quorum from a cyclic quorum system with cycle length $= 7$. Figure 3.1 illustrates three quorums from a cyclic quorum system with cycle length 7.

3.2.2 Quorum-based Wakeup Scheduling

Previous work [24] has defined the **QPS** (quorum-based power-saving) problem as follows: Given an universal set $U = \{0, 1, ...n-1\}$ ($n > 2$) and a quorum system $\mathcal{Q}$ over U, two nodes that

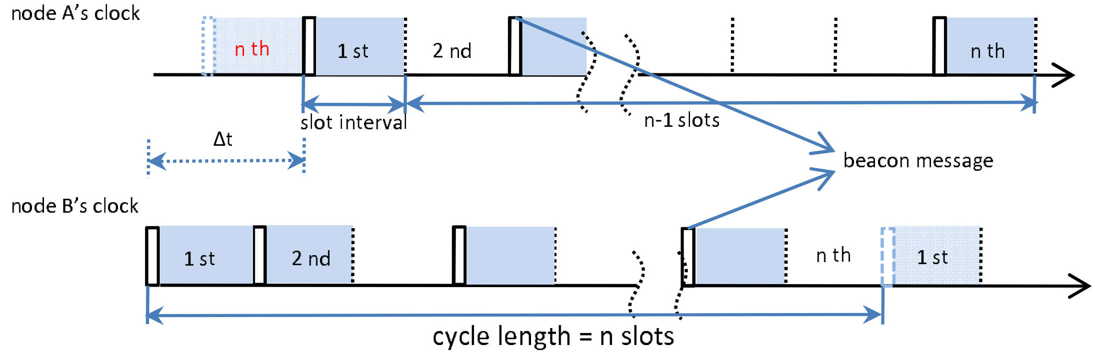


Figure 3.2: Neighbor discovery under partial overlap

select any quorum from $\mathcal{Q}$ as their wakeup schedules must have at least one overlap in every n consecutive time slots.

Theorem 1. *$\mathcal{Q}$ is a solution to the QPS problem if $\mathcal{Q}$ is a quorum system satisfying the* rotation closure property.

Theorem 2. *Both grid quorum systems and cyclic quorum systems satisfy the rotation closure property and can be applied as a solution for the QPS problem in wireless sensor networks.*

Proofs of Theorems 1 and 2 can be found in [24].

Since sensor nodes are subject to clock drift, the time slots are not exactly aligned to their boundaries in practical deployments. In most cases, two nodes only have *partial overlap* during a certain time interval. It has been shown that two nodes that adopt quorum-based wakeup schedules can discover each other even under *partial overlap*.

Theorem 3. *[12] If two quorums ensure a minimum of one overlapping slot, then with the help of a beacon message at the beginning of each slot, two neighboring nodes can hear each others' beacons at least once.*

Theorem 3's proof is presented in [12]. An illustration is given in Figure 3.2. Suppose that node A's quorum and node B's quorum intersect with each other in the first element and that the clock drift between the two nodes is Δt ($1\ slot < \Delta t < 2\ slots$). We can see that node A's 1^{st} beacon message in the current cycle (beacon messages are marked with solid lines) will be heard by node B during node B's 2^{nd} time slot interval in its current cycle. Meanwhile, node B's 2^{nd} beacon message in the current cycle will be heard by node A during its n^{th} time slot interval in the previous cycle (beacon messages are marked with dash lines).

This theorem ensures that two neighboring nodes can always discover each other within bounded time if all beacon messages are transmitted successfully. This property also holds true even in the case when two originally disconnected subsets of nodes join together as long as they use the same quorum system.

3.2.3 Heterogeneous Quorum-Based Wakeup Scheduling

We introduce the h-QPS (heterogeneous quorum-based power saving) problem in this section [35]. In WSNs, it is often desirable that different nodes wakeup according to heterogeneous quorum-based schedules. There are several reasons for this. First, many WSNs have heterogeneous nodes such as cluster-heads, gateways, and relay nodes [65]. They often have different requirements regarding average neighbor discovery delay and energy saving ratio. For cyclic quorum systems, generally, cluster-heads should wakeup based on a quorum system with small cycle length, and member nodes should wakeup based on a longer cycle length. Second, WSNs that are used in applications such as environment monitoring typically have seasonally-varying power saving requirements. For example, a sensor network for wild fire monitoring may require a larger energy saving ratio during winter seasons. Thus, they often desire variable cycle-length wakeups during different seasons.

We define the **h-QPS** problem as follows. Given two heterogeneous quorum systems $\mathcal{X}$ over $\{0, 1, \cdots, n-1\}$ and $\mathcal{Y}$ over $\{0, 1, \cdots, m-1\}$ ($n \leq m$), design a pair ($\mathcal{X}, \mathcal{Y}$) in order to guarantee that:

1. two nodes that select two quorums $G \in \mathcal{X}$ and $H \in \mathcal{Y}$ as their wakeup schedules, respectively, can hear each other at least once within every m consecutive slots; and
2. $\mathcal{X}$ and $\mathcal{Y}$ are solutions to QPS, individually.

A solution to the h-QPS problem is important toward ensuring connectivity when we want to dynamically change the quorum systems between all nodes. For example, suppose that all nodes in a WSN initially wakeup via a larger cycle length. When there is a need to reduce the cycle length (e.g., to meet a delay requirement or due to changing seasons), the sink node can send a request to the whole network gradually through some relay nodes. The connectivity between a relay node and the remaining nodes will be lost if the relay node first changes its wakeup schedule to a new quorum schedule, which cannot overlap with the original schedules of the remaining nodes.

Although grid quorum systems and cyclic quorum systems can be applied as a solution for the QPS problem, that does not necessarily mean that any pair of such systems can be a solution to the h-QPS problem. We will show this in Section 4.1.2.

3.2.4 Rendezvous Mechanism with Periodic LPL Scheduling

Previous works on low power listening (or LPL) adopt periodic preamble sampling mechanisms [1, 2] in which a node checks the state of its channel once every x time units, where x is usually $100ms$ or $200ms$. If the gain of the channel is less than a certain threshold level, it means that there is no activity from its neighbors and the node will go back to sleep.

When a sender wants to send out data, it first sends out a long preamble [1] or multiple short strobed preambles, which contain the sender's identity [2]. When the desired receiver detects the short preamble, it will keep awake and will feed back an acknowledgment to the sender. After the

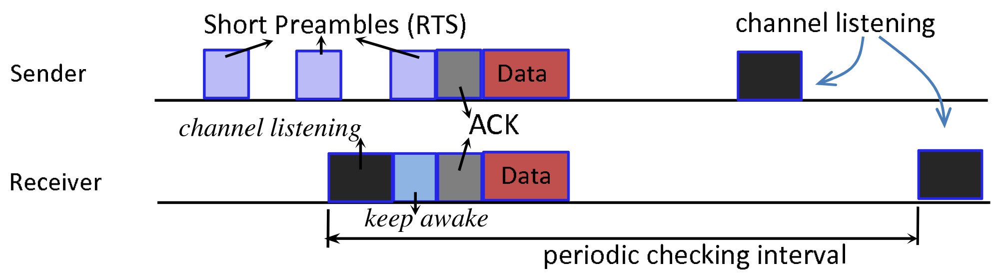


Figure 3.3: Neighbor discovery mechanism with periodic LPL scheduling in the X-MAC protocol

sender receives the acknowledgement, the actual data transmission will begin.

An illustration is given in Figure 3.3. In the idle state, both the sender and the receiver follow periodic LPL scheduling. Once the sender wants to transmit data, it sends out multiple short strobed preambles to trigger the receiver to wake up.

Chapter 4

Heterogenous Quorum-based Wakeup Scheduling

4.1 Heterogenous Quorum System Pair

4.1.1 Heterogeneous Rotation Closure Property

First, we define a few concepts to facilitate our presentation. Some definitions are extended from those in [37]. We will also use definitions from [36] to denote $\mathbb{Z}_n$ as a finite field of order n and $(\mathbb{Z}_n, +)$ as an Abelian Group.

Definition 4. *Let A be a set in $(\mathbb{Z}_n, +)$. For any integer $g \in \mathbb{Z}_n$, we define $A+g = \{(x+g) \, mod \, n : x \in A\}$. We further define a cyclic quorum system which contains $A, \ldots, A+n-1$ as $C(A, \mathbb{Z}_n)$.*

$A + g$ defines all elements in A that are roundly shifted by integer g in $\mathbb{Z}_n$. For example, if $A = \{1, 2, 4\}$ in $(\mathbb{Z}_7, +)$, then $A+4 = \{5, 6, 1\}$; we also have $C(A, \mathbb{Z}_7) = \{\{1, 2, 4\}, \{2, 3, 5\} \cdots, \{7, 1, 3\}\}$.

Definition 5. *(*p*-extension). Given two positive integers n and p, for a set $A = \{a_i | 1 \leq i \leq k, a_i \in \mathbb{Z}_n\}$, the* p-extension *of A is defined as A^p = $\{a_i + j * n | 1 \leq i \leq k, 0 \leq j \leq p-1, a_i \in \mathbb{Z}_n\}$. For a quorum system $\mathcal{Q} = \{A_1, \cdots, A_m\}$, the* p-extension *of $\mathcal{Q}$ is defined as $\mathcal{Q}^p = \{A_1^p, \cdots, A_m^p\}$.*

Example: Let $A = \{1, 2, 4\}$ in $(\mathbb{Z}_7, +)$. Now, $A^3 = \{1, 2, 4, 8, 9, 11, 15, 16, 18\}$ in $(\mathbb{Z}_{21}, +)$. If a quorum system $\mathcal{Q} = \{\{1, 2, 4\}, \{2, 3, 5\}, \{3, 4, 6\}\}$, then we have $\mathcal{Q}^2 = \{\{1, 2, 4, 8, 9, 11\}, \{2, 3, 5, 9, 10, 12\}, \{3, 4, 6, 10, 11, 13\}\}$.

Definition 6. (Heterogeneous rotation closure property). *Given two positive integers N and M where $N \leq M$ and $p = \lceil \frac{M}{N} \rceil$, consider two quorum systems $\mathcal{X}$ over the universal set $\{0, \cdots N-1\}$ and $\mathcal{Y}$ over the universal set $\{0, \cdots M-1\}$. The pair ($\mathcal{X}$,$\mathcal{Y}$) is said to satisfy the* heterogeneous rotation closure property *if :*

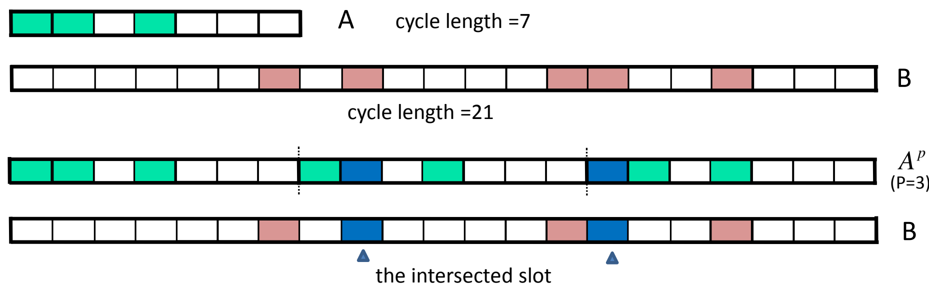


Figure 4.1: Heterogenous rotation closure property between two cyclic quorum systems: A with cycle length of 7 and B with cycle length of 21. A quorum from A's p-extension A^p will overlap with a quorum from B.

1. *$\forall G \in \mathcal{X}^p, H \in \mathcal{Y}, i \in N+$: $G \cap (H+i) \neq \emptyset$, and*
2. *$\mathcal{X}$ and $\mathcal{Y}$ satisfy the* rotation closure property *(Definition 1), respectively.*

Example: Let $A = \{1, 2, 4\}$ in $(\mathbb{Z}_7, +)$ and $B = \{1, 2, 4, 10\}$ in $(\mathbb{Z}_{13}, +)$. Consider two cyclic quorum systems $\mathcal{Q}_\mathcal{A} = C(A, \mathbb{Z}_7)$ and $\mathcal{Q}_\mathcal{B} = C(B, \mathbb{Z}_{13})$. Now, ${\mathcal{Q}_\mathcal{A}}^2 = C(\{1, 2, 4, 8, 9, 11\}, \mathbb{Z}_{14})$. We can verify that any two quorums from ${\mathcal{Q}_\mathcal{A}}^2$ and $\mathcal{Q}_\mathcal{B}$ must have non-empty intersection. Thus, the pair ($\mathcal{Q}_\mathcal{A}$,$\mathcal{Q}_\mathcal{B}$) satisfies the *heterogeneous rotation closure property*.

Lemma 1. *If two quorum systems $\mathcal{X}$ and $\mathcal{Y}$ satisfy the heterogeneous rotation closure property, then the pair $(\mathcal{X}, \mathcal{Y})$ is a solution to the h-QPS problem.*

Proof. According to Definition 6, if two quorum systems $\mathcal{X}$ and $\mathcal{Y}$ satisfy the heterogeneous rotation closure property, a quorum G from $\mathcal{X}$ and a quorum H from $\mathcal{Y}$ must overlap at least once within the larger cycle length of $\mathcal{X}$ and $\mathcal{Y}$. Thus, two nodes can hear each other if they select G and H as their wakeup schedules, respectively, based on Theorem 3. This implies that $(\mathcal{X}, \mathcal{Y})$ is a solution to the h-QPS problem. The lemma follows. □

Example: In Figure 4.1, there are two cyclic quorum systems $C(A, \mathbb{Z}_7)$ and $C(B, \mathbb{Z}_{21})$. Since they have different cycle lengths, we extend A's cycle by 3 ($3 = \lceil \frac{21}{7} \rceil$) times and denote its extension as A^p. Now, A^p will have an intersection with B within 21 time slot intervals. We can further verify that B and its rotations will overlap with A^p. Thus, $(C(A, \mathbb{Z}_7), C(B, \mathbb{Z}_{21}))$ has the *heterogeneous rotation closure property* and it can be a solution to the h-QPS problem.

4.1.2 Cyclic Quorum System Pair (Cqs-Pair)

In this section, we present one design of heterogenous quorum systems: *cqs-pair* which is based on the cyclic block design concept and cyclic difference sets in combinatorial theory [36].

We first review two definitions which were originally introduced in [37].

Definition 7. (N,k,λ)***- difference set****. A set D :* $\{a_1,...,a_k\}(\mod N)$*,* $a_i \in [0, N-1]$*, is called a* (N,k,λ)*-difference set if for every* $d \neq 0$ *, there are* exactly λ *ordered pairs* (a_i, a_j)*,* $a_i, a_j \in D$ *such that* $a_i - a_j \equiv d\ (modN)$*.*

Definition 8. ***Relaxed*** (N,k)***- difference set****. A set D :* $\{a_1,...,a_k\}(\mod N)$*,* $a_i \in [0, N-1]$*, is called a relaxed* (N,k)*-difference set if for every* $d \neq 0$*, there exists* at least *one ordered pair* (a_i, a_j)*,* $a_i, a_j \in D$ *such that* $a_i - a_j \equiv d\ (modN)$*.*

Example: The set $\{0,1,2,4,5,8,10\}$ is a $(15,7,3)$-difference set. The set $\{0,1,2,4\}$ is a relaxed (8,4)-difference set.

We will now introduce two new definitions that are closely related with the cqs-pair.

Definition 9. (N,k,M,l)***-difference pair.*** *Suppose* $N \leq M$ *and* $p = \lceil \frac{M}{N} \rceil$*. Suppose there are sets* $A : \{a_1, \cdots a_k\}$ *in* $(\mathbb{Z}_N, +)$ *and* $B : \{b_1, \cdots b_l\}$ *in* $(\mathbb{Z}_M, +)$*. The pair* (A, B) *is defined as a* (N,k,M,l)*-difference pair if* $\forall d \in \{0, \cdots, M-1\}$*, there exists at least one ordered pair* $b_i \in B$ *and* $a_j^p \in A^p$ *such that* $b_i - a_j^p \equiv\ d\ (mod\ M)$*.*

Consider an example where $N = 7$ and $M = 13$. Let $A = \{1,2,4\}$ and $B = \{1,3,6,7\}$ be two subsets in $(\mathbb{Z}_7, +)$ and $(\mathbb{Z}_{13}, +)$, respectively. Then (A, B) is a $(7,3,13,4)$-difference pair, since for A^2 ($\{1,2,4,8,9,11\}$) and B,

$$\begin{array}{ccccc} 1 \equiv 3-2 & 2 \equiv 6-4 & 3 \equiv 1-11 & 4 \equiv 6-2 & 5 \equiv 6-1 \\ 6 \equiv 7-1 & 7 \equiv 3-9 & 8 \equiv 6-11 & 9 \equiv 7-11 & 10 \equiv 1-4\ (mod\ 13) \\ 11 \equiv 6-8 & 12 \equiv 1-2 & 13 \equiv 1-1 & & \end{array}$$

Definition 10. ***(Heterogeneous cyclic coterie pair).*** *Given two groups of sets* $\mathcal{X} = \{A, A+1, \cdots, A+N-1\}$ *over* $\{0, \cdots, N-1\}$ *and* $\mathcal{Y} = \{B, B+1, \cdots, B+M-1\}$ *over* $\{0, \cdots, M-1\}$*, suppose* $N \leq M$ *and* $p = \lceil \frac{M}{N} \rceil$*. We call* $(\mathcal{X}, \mathcal{Y})$ *a* heterogeneous cyclic coterie pair *if:* $\forall (A+i)^p \subseteq \mathcal{X}^p$ *and* $(B+j) \subseteq \mathcal{Y}$*,* $(A+i)^p \cap (B+j) \neq \emptyset$*.*

Theorem 4. *Suppose* $A = \{a_1, \cdots, a_k\}$ *in* $(\mathbb{Z}_N, +)$ *and* $A_i = A+i$*;* $B = \{b_1, \cdots, b_k\}$ *in* $(\mathbb{Z}_M, +)$ *and* $B_i = B + j$*, where* $N \leq M$*. Given two groups of sets* $\mathcal{X} = \{A_i | 0 \leq i \leq N-1\}$ *and* $\mathcal{Y} = \{B_j | 0 \leq j \leq M-1\}$*, the pair* $(\mathcal{X}, \mathcal{Y})$ *is a* heterogeneous cyclic coterie pair *if and only if* (A, B) *is a* (N,k,M,l)*-difference pair.*

Proof. Sufficient Condition. Without loss of generality, we assume that $j > i$ regarding two sets B_i and A_j^p, where $p = \lceil \frac{M}{N} \rceil$. Consider the r^{th} element of B_i and s^{th} element of A_j^p, denoted by $b_{i,r}$ and $a_{j,s}^p$, respectively. We will now show that $b_{i,r} = a_{j,s}^p$.

Let the r^{th} element of B be b_r and the s^{th} element of A^p be a_s^p. Then $b_{i,r} - a_{j,s}^p = (b_r - a_s^p + i - j)\ mod\ M$. According to the definition of (N,k,M,l)-difference pair, there must be some r and s such that $b_r - a_s^p \equiv j - i\ (mod\ M)$. Therefore, we can always choose a pair of r and s such that $b_{i,r} - a_{j,s}^p \equiv 0\ (mod\ M)$. This implies that $B_j \cap A_i^P \neq \emptyset$.

Necessary Condition. We prove necessity by contradiction. Assume that $B_j \cap A_i^P \neq \emptyset$, but (A, B) is not a (N, k, M, l)-difference pair. Then, there exists a number $\in \{0, \cdots, M-1\}$, say t, for which $b_i - a_j^p \neq t \ (mod\ M), \forall i, j$.

Consider the r^{th} element of B_t and the s^{th} element of A^p. We have $b_{t,r} - a_s^p \equiv b_r - a_s^p + t \ (mod\ M)$. Since $B_t \cap A_i^P \neq \emptyset$, $b_{t,r} - a_s^p = 0$ for some r and s. This implies that $b_r - a_s^p \equiv t \ (mod\ M)$ for some r and s, which contradicts the derivation of $b_i - a_j^p \neq t \ (mod\ M) \ \forall i, j$ from the assumption. The theorem follows. □

If two groups of sets $\mathcal{X}$ and $\mathcal{Y}$ can form a *heterogeneous cyclic coterie pair*, then they have at least one intersection within the larger cycle length. But the pair does not guarantee that any two sets from the same group, $\mathcal{X}$ or $\mathcal{Y}$, also have an intersection.

Definition 11. ***cyclic quorum system pair (cqs-pair).*** *Given two quorum sets $\mathcal{X} = \{A, A+1, \cdots, A+N-1\}$ over $\{0, \cdots, N-1\}$ and $\mathcal{Y} = \{B, B+1, \cdots, B+M-1\}$ over $\{0, \cdots, M-1\}$, suppose $N \leq M$. We call $(\mathcal{X}, \mathcal{Y})$ a* cqs-pair *if*

1. *$(\mathcal{X}, \mathcal{Y})$ is a heterogeneous cyclic coterie pair; and*
2. *$\mathcal{X}$ and $\mathcal{Y}$ are cyclic quorum systems, respectively.*

Theorem 5. *Given two groups of sets $\mathcal{X} = \{A, A+1, \cdots, A+N-1\}$ and $\mathcal{Y} = \{B, B+1, \cdots, B+M-1\}$, where $A = \{a_1, \cdots, a_k\}$ in $(\mathbb{Z}_N, +)$ and $B = \{b_1, \cdots, b_l\}$ in $(\mathbb{Z}_M, +)$ $(N \leq M)$, the pair $(\mathcal{X}, \mathcal{Y})$ is a* cqs-pair *if and only if*

1. *(A, B) is a (N, k, M, l)-difference pair; and*
2. *A is a relaxed (N, k)-difference set and B is a relaxed (M, l)-difference set.*

Proof. If (A, B) is a (N, k, M, l)-difference pair, then $(\mathcal{X}, \mathcal{Y})$ is a heterogenous cyclic coterie pair. Further, if A and B are *relaxed difference sets*, respectively, then $\mathcal{X}$ and $\mathcal{Y}$ are cyclic quorum systems, respectively. Similarly, we can prove that the converse is also true. The theorem follows. □

Theorem 6. *The cyclic quorum system pair (*cqs-pair*) is a solution to the h-QPS problem.*

Proof. According to the definition of cqs-pair, a cqs-pair satisfies the heterogeneous rotation closure property. Thus, the cqs-pair can be a solution to the h-QPS problem according to Lemma 1. The theorem follows. □

Corollary 1. *Given a cyclic quorum system $\mathcal{X}$, $(\mathcal{X}, \mathcal{X})$ is a cqs-pair.*

Example 1: Let $A = \{1, 2, 4\}$ and $\mathcal{X} = C(A, \mathbb{Z}_7)$; $B = \{7, 9, 14, 15, 18\}$ and $\mathcal{Y} = C(B, \mathbb{Z}_{21})$. The pair $(\mathcal{X}, \mathcal{Y})$ is a cqs-pair. Also, both $(\mathcal{X}, \mathcal{X})$ and $(\mathcal{Y}, \mathcal{Y})$ are cqs-pairs.

Example 2: Let $A = \{3, 5, 6\}$ and $B = \{7, 9, 14, 15, 18\}$. The pair $(\mathcal{X}, \mathcal{Y})$ is *not* a cqs-pair, although $\mathcal{X}$ and $\mathcal{Y}$ are cyclic quorum systems, respectively.

Example 3: Let $A = \{1, 2, 4\}$ and $\mathcal{X} = C(A, \mathbb{Z}_7)$; $B = \{1, 2, 4\}$ in $(\mathbb{Z}_{14}, +)$ and $\mathcal{Y} = C(B, \mathbb{Z}_{14})$. (A, B) is a $(7, 3, 14, 3)$-difference pair. But $(\mathcal{X}, \mathcal{Y})$ is *not* a cqs-pair since B is not a relaxed difference set in $(\mathbb{Z}_{14}, +)$ and $\mathcal{Y}$ is not a cyclic quorum system.

4.1.3 Grid Quorum System Pair (Gqs-Pair)

Now, we introduce another design, *grid quorum system pair* (gqs-pair) of heterogenous quorum systems.

Definition 12. ***Grid quorum system pair (gqs-pair).*** *If a quorum in a grid quorum system contains one row and one column of elements, the* gqs-pair *is a pair consisting of any two such grid quorum systems.*

Lemma 2. *The* gqs-pair *satisfies the heterogeneous rotation closure property and can be a solution to the h-QPS problem.*

Proof. It has been proven in [24] that the grid quorum system satisfies the rotation closure property. Thus, we only need to prove that for two grid quorum systems $\mathcal{X}$ over $\{0, \cdots, n-1\}$ and $\mathcal{Y}$ over $\{0, \cdots, m-1\}$ $(n \leq m, p = \lceil \frac{m}{n} \rceil)$, $\forall G^p \in \mathcal{X}^p, H \in \mathcal{Y}, i \in \{0, \cdots M-1\}$, there is $G^p \cap (H+i) \neq \emptyset$ or $(G+i)^p \cap H \neq \emptyset$.

Consider a quorum G from $\mathcal{X}$ which contains all elements in the column c, namely c,$c+\sqrt{n}$,$\cdots$,$c+\sqrt{n}(\sqrt{n}-1)$, where $0 \leq c < \sqrt{n}$. Then, a quorum $(G+i)^p$ from the $p-extension$ of $\mathcal{X}$ contains elements, which has the largest distance of $\sqrt{n}$ between any two consecutive elements. $(G+i)^p$ must have an intersection with H since H contains a full row which has $\sqrt{m}$ $(\geq \sqrt{n})$ consecutive integers. Thus, the *grid quorum system pair* satisfies the heterogeneous rotation closure property and can be a solution to the h-QPS problem. The lemma follows. □

An illustration on the heterogeneous rotation closure property of the gqs-pair is given in Figure 4.2. There are two grid quorum systems in Figure 4.2, A with the size of 4×4 and B with the size of 6×6. Without considering clock drift, we can see that A's quorums will intersect with B's quorums in the 10^{th}, 3^{rd}, 7^{th}, and the 12^{th} slot.

4.2 Construction Scheme for Cqs-pair

It is straightforward to construct a *gqs-pair* since it contains two arbitrary grid quorum systems. Therefore, we only discuss the construction of *cqs-pair*, which is non-trivial. In previous works, exhaustive search has been used to find an optimal solution for the cyclic quorum design [37]. This is not practical when cycle length (n) is large. In this section, we first present a fast construction scheme for cyclic quorum systems and then apply it to the design of a cqs-pair.

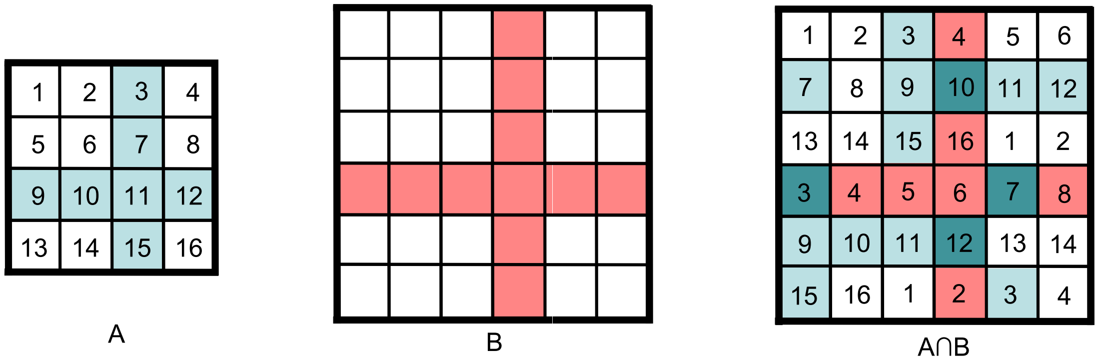


Figure 4.2: An example grid quorum system pair and its rotation closure property: grid quorum system A has a grid 4×4 and B has a grid 6×6. A quorum from A and a quorum from B overlap at 3 slots with B's cycle length.

4.2.1 Multiplier Theorem [3]

We introduce a few concepts to facilitate our presentation.

Definition 13. *Let D be a (v, k, λ)-difference set in an Abelian group $(G, +)$ of order v. For an integer m, we define*

$$mD = \{mx : x \in D\}$$

Then, m is called a multiplier of D if $mD = D + g$ for some $g \in G$. Also, we say that D is fixed by the multiplier m if $mD = D$.

Example: The set $D = \{0, 1, 5, 11\}$ is a $(13, 4, 1)$-difference set in $(\mathbb{Z}_{13}, +)$. Then, $3D = \{0, 2, 3, 7\} = D + 2$, and hence 3 is a multiplier of D.

Definition 14. ***Automorphism.*** *Suppose $(X, \mathcal{A})$ is a design. A transform function α is an automorphism of $(X, \mathcal{A})$ if*

$$[\{\alpha(x) : x \in A\} : A \in \mathcal{A}] = \mathcal{A}$$

Definition 15. ***Disjoint cycle representation:*** *The disjoint cycle representation of a set X is a group of disjoint cycles in which each cycle has the form $(x\ \alpha(x)\ \alpha(\alpha(x)) \cdots)$ for some $x \in X$.*

Suppose the automorphism is $x \mapsto 2x \ mod\ 7$. The disjoint cycle representation of $\mathbb{Z}_7$ is as follows: (0) (1 2 4) (3 6 5).

Theorem 7. ***(Multiplier Theorem).*** *Suppose there exists a (v, k, λ)-difference set D. Suppose also that the following four conditions are satisfied:*

1. *p is* prime*;*
2. *$gcd(p, v) = 1$;*
3. *$k - \lambda \equiv 0 \ (mod\ p)$; and*

4. $p > \lambda$.

Then p is a multiplier of D.

Theorem 8. *Suppose that m is a multiplier of a (v, k, λ)-difference set D in an Abelian group $(G, +)$ of order v. Then there exists a translate of D that is fixed by the multiplier m.*

The proofs of Theorem 7 and Theorem 8 are given in [36]. According to the Theorem of Singer Difference Set, there must exist a $(q^2+q+1, q+1, 1)$-difference set when q is a prime power. Thus, we only consider the $(q^2 + q + 1, q + 1, 1)$-design, where q is a prime power, in our construction scheme.

In the following, we first give an example to illustrate the application of the Multiplier Theorem for the construction of difference sets.

Example. We use the Multiplier Theorem to find a $(21, 5, 1)$-difference set. Observe that $p = 2$ satisfies the conditions of Theorem 7. Hence 2 is a multiplier of any such difference set. By Theorem 8, we can assume that there exists a $(21, 5, 1)$-difference set in $(\mathbb{Z}_{21}, +)$ that is fixed by the multiplier 2. Therefore, the *automorphism* is $\alpha(x) \mapsto 2x \ mod\ 21$. Thus, we obtain the disjoint cycle representation of the permutation defined by $\alpha(x)$ of $\mathbb{Z}_{21}$ as follows:

$$(0)\ \ (1\ 2\ 4\ 8\ 16\ 11)\ \ (3\ 6\ 12)\ \ (5\ 10\ 20\ 19\ 17\ 13)\ \ (7\ 14)\ \ (9\ 18\ 15)$$

The difference set we are looking for must consist of a union of cycles in the list above. Since the difference set has a size five, it must be the union of one cycle of length two and one cycle of length three. There are two possible ways to do this, both of which happen to produce the difference set:

$$(3\ 6\ 7\ 12\ 14)\ and\ \ (7\ 9\ 14\ 15\ 18)$$

With the Multiplier Theorem, we can quickly construct $(q^2+q+1, q+1, 1)$-difference sets, where q is a prime power. This mechanism significantly improves the speed of finding the optimal solution relative to the exhaustive method in [37].

After obtaining the difference sets, we use Theorem 5 to build a *cqs-pair*.

4.2.2 Verification Matrix

Armed with Theorem 4, we adopt a verification matrix to check the non-empty intersection property of two heterogeneous difference sets.

Suppose that $A = \{a_1, a_2, \cdots, a_k\}$ in $(\mathbb{Z}_N, +)$ and $B = \{b_1, b_2, \cdots, b_l\}$ in $(\mathbb{Z}_M, +)$ where $N \leq M$ and $p = \lceil \frac{M}{N} \rceil$. The verification matrix is defined as a $pk \times l$ matrix $\mathcal{M}_{l \times pk}$ whose element $m_{i,j}$ is equal to $(b_i - a_j^p)$ mod M, where $a_j^p \in A^p$, as shown below:

$$\mathcal{M}_{l\times pk} = \begin{bmatrix} b_1 - a_1^p & \cdots & b_1 - a_{pk}^p \\ \vdots & b_i - a_j^p & \cdots \\ b_l - a_1^p & \cdots & b_l - a_{pk}^p \end{bmatrix}$$

We can check whether (A, B) is a *heterogeneous cyclic coterie pair* by checking whether $\mathcal{M}_{l\times pk}$ contains all elements from 0 to $M-1$ or not. If the checking result is true, it means that:

$$\forall d \in \{0, \cdots, M-1\},\ \exists b_i \in B \ and \ a_j^p \in A^p,\ b_i - a_j^p \equiv\ d\ (mod\ M).$$

This indicates that (A, B) is a *heterogeneous cyclic coterie* based on Theorem 4. Otherwise, (A, B) is *not* a *heterogeneous cyclic coterie*. (An example of the verification matrix will be shown in Section 4.2.4.)

If two quorum systems $C(A_N, \mathbb{Z}_N))$ and $C(B_M, \mathbb{Z}_M))$ are cyclic quorum systems, respectively, we can verify whether the pair $[C(A_N, \mathbb{Z}_N)), C(B_M, \mathbb{Z}_M)]$ is a cqs-pair by checking whether or not the *verification matrix* constructed from A and B contains all elements from 0 to $M-1$.

4.2.3 Construction Algorithm

In our proposed algorithm for constructing a cqs-pair, we only consider cyclic quorum systems with a cycle length of $(q^2+q+1, q+1, 1)$, where q is a prime power. This is because, we can prove that when q is a prime power, there must exist a $(q^2+q+1, q+1, 1)$-difference set in $(\mathbb{Z}_{q^2+q+1,q+1,1}, +)$ [36].

We describe our algorithm for constructing a cqs-pair at a high-level of abstraction in Algorithm 1. The input of the algorithm is two numbers n and m, which satisfy $n = q^2+q+1$ and $m = r^2+r+1$ and where q and r are prime powers.

By employing our construction algorithm, for two different integers n and m that satisfy $n = q^2+q+1$ and $m = r^2+r+1$ (q and r being two prime powers, $n \le m$), it will take $O(n^2)$ and $O(m^2)$ time to build the *disjoint cycle representations*, respectively. After that, the algorithm will check $u \times v \times l \times pk \approx uvm^{3/2}n^{-1/2}$ elements, since $l \approx \sqrt{m}$ and $k \approx \sqrt{n}$, where u and v are numbers of $(n, k, 1)$-difference sets and $(m, l, 1)$-difference sets, respectively. Thus, the total time complexity is $O(uvm^{3/2}n^{-1/2} + m^2)$ for constructing a cqs-pair with input parameters n and m ($n \le m$).

4.2.4 A Complete Application Example

As an example, consider $n = 7$ and $m = 21$. By the Multiplier Theorem, we can easily obtain two $(7, 3, 1)$-difference sets $\{1, 2, 4\}$ and $\{3, 6, 5\}$ in $(\mathbb{Z}_7, +)$. Similarly, there are two $(21, 5, 1)$-difference sets, $\{3, 6, 7, 12, 14\}$ and $\{7, 9, 14, 15, 18\}$ in $(\mathbb{Z}_{21}, +)$. Let $A_7 = \{1, 2, 4\}$, $B_7 = \{3, 6, 5\}$, $A_{21} = \{3, 6, 7, 12, 14\}$, and $B_{21} = \{7, 9, 14, 15, 18\}$.

Totally, there are four pairs of $(7, 3, 1)$-difference sets and $(21, 5, 1)$-difference sets. First, we check

Algorithm 1 Constructing cqs-pair

Require: $n = q^2 + q + 1$ and $m = r^2 + r + 1$, q, r are prime powers
$n \leftarrow q^2 + q + 1$
$m \leftarrow r^2 + r + 1$
$p_a \leftarrow$ Multiplier of $(n, k, 1)$-difference set
$p_b \leftarrow$ Multiplier of $(m, l, 1)$-difference set
$\alpha_n(x) \leftarrow p_a \cdot x \ (mod\, n)$
$\alpha_m(x) \leftarrow p_b \cdot x \ (mod\, m)$
Construct the disjoint cycle representation for $\mathbb{Z}_n$ with $\alpha_n(x)$
Construct the disjoint cycle representation for $\mathbb{Z}_m$ with $\alpha_m(x)$
$u \leftarrow$#Num of unions of disjoint cycle being $(n, k, 1)$-difference set
$\{A_1, \cdots, A_u\} \leftarrow$ the set of unions of disjoint cycles being $(n, k, 1)$-difference set
$v \leftarrow$#Num of unions of disjoint cycle being $(m, l, 1)$-difference set
$\{B_1, \cdots, B_v\} \leftarrow$ the set of unions of disjoint cycles being $(m, l, 1)$-difference set
for i = 1 to u **do**
  for j = 1 to v **do**
    $M_{i,j} \leftarrow$ verification matrix (A_i, B_j)
    $\mathcal{X}_i \leftarrow C(A_i, \mathbb{Z}_n)$
    $\mathcal{Y}_j \leftarrow C(B_j, \mathbb{Z}_m)$
    if $M_{i,j}$ contains all elements from 0 to $m - 1$ **then**
      $(\mathcal{X}_i, \mathcal{Y}_j)$ is a cqs-pair
    else
      $(\mathcal{X}_i, \mathcal{Y}_j)$ is not a cqs-pair
    end if
  end for
end for

the pair $(C(A_7, \mathbb{Z}_7), C(A_{21}, \mathbb{Z}_{21}))$. The constructed verification matrix is as follows:

$$\begin{bmatrix} 2 & 1 & 20 & 16 & 15 & 13 & 9 & 8 & 6 \\ 5 & 4 & 2 & 19 & 18 & 16 & 12 & 11 & 9 \\ 6 & 5 & 3 & 20 & 19 & 17 & 13 & 12 & 10 \\ 11 & 10 & 8 & 4 & 3 & 1 & 18 & 17 & 15 \\ 13 & 12 & 10 & 6 & 5 & 3 & 20 & 19 & 17 \end{bmatrix}$$

We find that 7 and 14 are not in the matrix. Thus, the pair $(C(A_7, \mathbb{Z}_7), C(A_{21}, \mathbb{Z}_{21}))$ is *not* a cqs-pair. Similarly, we can check that $(C(B_7, \mathbb{Z}_7), C(B_{21}, \mathbb{Z}_{21}))$ is *not* a cqs-pair. But $(C(A_7, \mathbb{Z}_7), C(B_{21}, \mathbb{Z}_{21}))$ and $(C(B_7, \mathbb{Z}_7), C(A_{21}, \mathbb{Z}_{21}))$ are cqs-pairs, respectively.

The cqs-pair can be applied to WSNs for dynamically changing the quorum system (i.e., the cycle length) at each node, in order to meet end-to-end delay constraints and without loosing network connectivity. Table 4.1 shows the available pairs for cycle lengths ≤ 21.

Table 4.1: cqs-pair (for $n, m \leq 21$)

cycle length	7 $A_7 = \{1, 2, 4\}$ $B_7 = \{3, 5, 6\}$	13 $A_{13} = \{0, 1, 3, 9\}$ $B_{13} = \{0, 2, 6, 5\}$ $C_{13} = \{0, 4, 12, 10\}$ $D_{13} = \{0, 7, 8, 11\}$	21 $A_{21} = \{3, 6, 7, 12, 14\}$ $B_{21} = \{7, 9, 14, 15, 18\}$
7	$(C(A_7, \mathbb{Z}_7), C(A_7, \mathbb{Z}_7))$ $(C(B_7, \mathbb{Z}_7), C(B_7, \mathbb{Z}_7))$	$(C(A_7, \mathbb{Z}_7), C(A_{13}, \mathbb{Z}_{13}))$ $(C(A_7, \mathbb{Z}_7), C(B_{13}, \mathbb{Z}_{13}))$ $(C(B_7, \mathbb{Z}_7), C(C_{13}, \mathbb{Z}_{13}))$ $(C(B_7, \mathbb{Z}_7), C(D_{13}, \mathbb{Z}_{13}))$	$(C(A_7, \mathbb{Z}_7), C(B_{21}, \mathbb{Z}_{21}))$ $(C(B_7, \mathbb{Z}_7), C(A_{21}, \mathbb{Z}_{21}))$
13		$(C(A_{13}, \mathbb{Z}_{13}), C(A_{13}, \mathbb{Z}_{13}))$ $(C(B_{13}, \mathbb{Z}_{13}), C(B_{13}, \mathbb{Z}_{13}))$ $(C(C_{13}, \mathbb{Z}_{13}), C(C_{13}, \mathbb{Z}_{13}))$ $(C(D_{13}, \mathbb{Z}_{13}), C(D_{13}, \mathbb{Z}_{13}))$	$(C(B_{13}, \mathbb{Z}_{13}), C(A_{21}, \mathbb{Z}_{21}))$

4.3 Performance Analysis

4.3.1 Average Discovery Delay

We denote the *average discovery delay* as the time between data arrival and discovery of the adjacent receiver (i.e., the simultaneous wake-up of two nodes). Note that this metric does not include the time for delivering a message.

Suppose that the length of one time slot is 1.

Theorem 9. *The average discovery delay between two nodes that wakeup based on quorums from the same* cyclic quorum system *adopting the* $(n, k, 1)$*-difference set is:*

$$E(Delay) = \frac{n-1}{2}.$$

Proof. Let the k elements in $(n, k, 1)$-difference set be denoted as a_1, a_2, $\cdots$, a_k. If a node has a message that arrived during the i^{th} time slot, the expected delay (from data arrival to the simultaneous wake-up of two nodes) is $\frac{1}{k}(a_1 - i) \; mod \; n + \frac{1}{k}(a_2 - i) \; mod \; n + \cdots + \frac{1}{k}(a_k - i) \; mod \; n$. If a message has arrived, the probability of the message arriving during the i^{th} time slot is $\frac{1}{n}$. Thus, the expected delay (average delay) is:

$$\begin{aligned} E(Delay) = & \frac{1}{n}[\frac{1}{k}(a_1 - 1) \; mod \; n + \frac{1}{k}(a_2 - 1) \; mod \; n + \cdots + \frac{1}{k}(a_k - 1) \; mod \; n \\ & + \frac{1}{k}(a_1 - 2) \; mod \; n + \frac{1}{k}(a_2 - 2) \; mod \; n + \cdots + \frac{1}{k}(a_k - 2) \; mod \; n \\ & + \quad \cdots\cdots \\ & + \frac{1}{k}(a_1 - n) \; mod \; n + \frac{1}{k}(a_2 - n) \; mod \; n + \cdots + \frac{1}{k}(a_k - n) \; mod \; n] \\ = & \frac{1}{nk} \cdot (k \cdot 1 + k \cdot 2 + \cdots + k \cdot n - 1) = \frac{n-1}{2} \end{aligned}$$

The theorem follows. □

Corollary 2. *The average discovery delay between two nodes that wakeup based on a cqs-pair in which two* cyclic quorum systems *have cycle lengths* n *and* m *(*$n \leq m$*), respectively, is:*

$$\frac{n-1}{2} < E(Delay) < \frac{m-1}{2}.$$

Corollary 4 indicates that the average discovery delay between two nodes that adopt a cqs-pair is bounded. When the average one-hop delay constraint is D, we must meet $\frac{m-1}{2} \leq D$.

Theorem 10. *The average discovery delay between two nodes that wakeup based on quorums from the same* grid quorum system *with a grid of* $\sqrt{N} \times \sqrt{N}$ *elements is:*

$$E(Delay) = \frac{(N-1)(\sqrt{N}+1)}{3\sqrt{N}}.$$

Proof. Let $L = \sqrt{N}$. Suppose there are two grid quorum systems $\mathcal{Q}_a$ and $\mathcal{Q}_b$ that adopt $\sqrt{N} \times \sqrt{N}$ grid.

Given a quorum from $\mathcal{Q}_a$ (i.e., i_a^{th} row plus j_a^{th} column), and a quorum from $\mathcal{Q}_b$ (i.e., i_b^{th} row plus j_b^{th} column), when $i_b < i_a$, the discovery delay is:

$$delay = \begin{cases} (i_b - 1)L + j_a - 1, & j_b \neq j_a \\ j_a - 1, & j_b = j_a \end{cases}$$

When $i_b \geq i_a$, the discovery delay is:

$$delay = \begin{cases} (i_a - 1)L + j - 1, & j_b \neq j_a \\ j_a - 1, & j_b = j_a \end{cases}$$

The probability of a quorum in $\mathcal{Q}_b$ to select the i_b^{th} row and j_b^{th} column is $1/L^2$. Thus, when the quorum from $\mathcal{Q}_a$ contains the i_a^{th} row plus j_a^{th} column, the average discovery delay between $\mathcal{Q}_a$ and $\mathcal{Q}_b$ is:

$$D = \frac{1}{L}[(L-1)\frac{(i-2)(i-1)}{2} + (j-1)(i-2) + (i-\frac{1}{2})(L-i+1)(L-1)]$$

Therefore, the expected discovery delay (from data arrival to two nodes waking-up simultaneously) is:

$$\begin{aligned} E(Delay) &= \frac{1}{L^2}\sum_{i=1}^{L}\sum_{j=1}^{L} D && (4.1) \\ &= \frac{1}{L^3}\{\sum_{i=1}^{L}\frac{1}{L}[(L-1)\frac{(i-2)(i-1)}{2} + (j-1)(i-2) + (i-\frac{1}{2})(L-i+1)(L-1)]\} && (4.2) \\ &= \frac{L-1}{2L^2}\sum_{i=1}^{L}[(i-2)(i-1) + (i-2) + (2i-1)(L-i+1)] && (4.3) \\ &= \frac{L-1}{2L^2}\sum_{i=1}^{L}[(2L+1)i - i^2 - L - 1] && (4.4) \\ &= \frac{(L^2-1)(L+1)}{3L} && (4.5) \\ &= \frac{(N-1)(\sqrt{N}+1)}{3\sqrt{N}}. && (4.6) \end{aligned}$$

The theorem follows. □

Corollary 3. *The average discovery delay between two nodes that wakeup based on a gqs-pair in which two grid quorum systems adopt a grid of $\sqrt{n} \times \sqrt{n}$ and a grid of $\sqrt{m} \times \sqrt{m}$, respectively, is:*

$$\frac{(n-1)(\sqrt{n}+1)}{3\sqrt{n}} < E(Delay) < \frac{(m-1)(\sqrt{m}+1)}{3\sqrt{m}}$$

The proof for Corollary 3 is not difficult so that we omit it.

4.3.2 Quorum Ratio and Energy Conservation

We define **quorum ratio**, denoted ϕ, as the proportion of the beacon intervals that is required to be awake in each cycle. Correspondingly, the energy conservation ratio of a node is $1 - \phi$.

If the wakeup schedule of a node is a cyclic quorum system which is based on a $(n, k, 1)$-difference set, its quorum ratio is $\frac{k}{n}$. In [12], the authors have proved that a $(q^2 + q + 1, q + 1, 1)$-difference set exists and that, it is an optimal design for a given quorum size $q + 1$. The optimal quorum ratio is $\phi = \frac{q+1}{q^2+q+1}$ for such a cyclic quorum system.

For a cqs-pair, the *quorum ratios* for systems in the pair which are based on (N, k, M, l)-difference pair are:

$$\phi_1 = \frac{\sqrt{4N-3}-1}{2N} \text{ and } \phi_2 = \frac{\sqrt{4M-3}-1}{2M}$$

respectively. Here, we only consider $(q^2 + q + 1, q + 1, 1)$-design in the cqs-pair construction scheme.

For a grid quorum system with $\sqrt{n} \times \sqrt{n}$ grid, the *quorum ratio* is:

$$\phi = \frac{2\sqrt{n}-1}{n}$$

and the corresponding energy saving ratio is:

$$1 - \phi = 1 - \frac{2\sqrt{n}-1}{n}.$$

Recalling the average discovery delay in Section 4.3.1, we can observe that there is a trade-off between the average delay and the quorum ratio. Larger the cycle length of a quorum system, larger is the discovery delay, but smaller is the quorum ratio.

4.4 Implementations of Cqs-pair and Qqs-pair

We implemented heterogenous quorum systems in a WSN platform comprised of Telosb motes [66]. There are three key issues in converting the Cqs-pair and Qqs-pair concepts into practical implementations. The first key issue is to ensure that two nodes can discover each other in the presence of clock drift. The second one is that a node should keep awake if there is pending data for receiving or for transmitting. The third issue is how to support multicast or broadcast.

4.4.1 Beacon Messages

Previous work on the implementation of QPS protocol over IEEE 802.11 adopts the concept of ATIM (Ad hoc Traffic Indication Map) windows [11], in which a node can optionally enter the sleep mode if it receives no ATIM frame in an ATIM window.

In our implementation, we do not use the notion of ATIM windows. We define the time interval that a node is scheduled to be awake as an *active slot*, and the time interval that the node is scheduled to sleep as a *silent slot*. In an *active slot*, a node has to transmit its own *beacon message* to inform its neighbors about its wakeup status, and listen to *beacons* from other nodes for which it may have buffered packets that are waiting for transmission.

In our scheme, to ensure the correctness of the protocol, a node remains awake throughout its entire active slot. It may be possible for nodes to be only partially awake during their active slots – such optimizations can be considered in future works. In a *silent slot*, a node will shut down its radio.

The beacon message contains three fields:{*indic, node_id, time_stamp*}. The *indic* field can have only two types of value: indic=0, which indicates that the message is a beacon message, and indic = 1, which indicates that the message is data. The *node_id* field is used to distinguish among different nodes. The *time_stamp* field is used to identify whether or not two beacon messages are identical.

4.4.2 Power Management

The goal of power management is to facilitate effective communication while saving as much energy as possible. In our power management scheme, a node determines its desirable *communication schedule*, i.e., when it should go to sleep or wake up. The relationship between the wakeup schedule and the communication schedule devised by a power management policy for a sender is illustrated in Figure 4.3.

At the MAC-layer, we propose a reservation mechanism for communication on top of the proposed quorum-based heterogenous wakeup scheduling scheme (cqs-pair or gqs-pair). Each node has two states, *idle mode* and *active mode*. In the *idle mode*, a node will follow its wakeup schedule to wake-up or to sleep. We also call this mode as power saving mode. Once there is data for receiving or for transmitting, the node will enter into the *active mode* as shown in Figure 4.3.

In the active mode, a sender maintains a table of timers for all its neighbors. The timers are triggered once the sender receives beacon messages from the neighbors. The initial value of each timer is one time slot. The sender will also record its own wakeup schedule via a timer. If both the sender and the receiver is in an *active slot*, then they can communicate. If the sender enters into a *silent slot* but there are more packets for transmission and the receiver is still in an *active slot*, then the sender will keep awake in its next slot. If there are more packets for transmission, but the receiver will enter a *silent slot*, then the sender will send a *keep-awake* message to the receiver at the end of the transmission of the current packet. The receiver that is being requested to stay awake will then send back an acknowledgment, indicating its willingness to remain awake in its next slot.

The power management scheme at the receiver side is simpler than that at the sender side. In active mode, if there is no *keep-awake* message, the receiver will continue communication until the end of its current slot interval; otherwise, it will keep awake in its next slot.

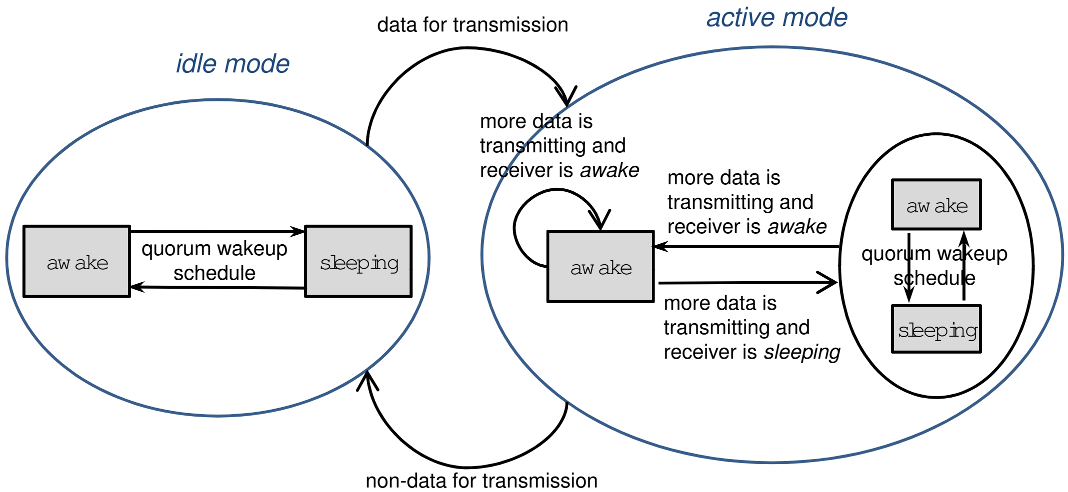


Figure 4.3: Power Management at the Transmitter Side: Communication Schedule and Wakeup schedule

4.4.3 Support for Multicast and Broadcast

Quorum-based asynchronous wakeup protocols cannot guarantee that more than one receiver is awake when a sender wishes to multicast or broadcast.

There are multiple ways to support multicast and broadcast. One method is to adopt relatively prime frequencies among all nodes for wakeup scheduling. This method does not need synchronization between the sender and all the receivers. The sender only needs to notify m receivers to wake-up via the pairwise relative primes p_1, p_2,..., p_m, respectively. Then each receiver generates its new wakeup frequency based on the received frequency. Through Chinese Remainder Theorem [62, 67], it can be proven that the m receivers must wakeup simultaneously at the I^{th} beacon interval ($0 \leq I \leq p_1 \times p_2 ... \times p_m$). The sender can then transmit a multicast/broadcast message at this interval.

Another way to multicast/broadcast is by using synchronization over quorum-based wakeup schedules. The sender can book-keep all neighbors' schedules, and synchronize their schedules so that neighboring nodes wake up in the same set of slots with the use of Lamport's clock synchronization algorithm [68]. When all nodes are awake simultaneously, the senders then send a message to multiple neighbors simultaneously.

The first mechanism has the advantage that no synchronization is needed between a sender and multiple receivers. But it cannot bound the average delay. The second approach can bound the average delay but it needs book-keeping and synchronization over asynchronous wakeup schedules.

For multicast/broadcast, we set a threshold L. If the number of multicast packets exceed the threshold L, the sender will send a Multicast-Notify to all neighbor receivers, requesting them to stay awake. Otherwise the sender will send the multicast data to each receiver one by one, by unicasting. The value of the threshold L depends on the configuration of time slot lengths and packet lengths.

To reduce the time of waiting before actual transmission, the Multicast-Notify message contains a field to notify all receivers on how long they should wait. The value of this field is the time between when the message is sent and when all receivers are awake.

4.5 Performance Evaluation

We evaluated the performance of our schemes through numerical studies and by real implementation over a WSN platform of Telosb motes [40]. In our experiments, a set of nodes was deployed. The radio range was configured to 10 meters for each node. There was one sink node which acted as the base station. The sink node communicated with a laptop computer (through a wireless USB serial port), which recorded performance measurements. The detailed radio parameters such as data rates default to the data sheet of TelosB [40].

We built our wakeup scheduling schemes over the basic CSMA/CA protocol. We used MintRoute [44] as the routing protocol for end-to-end transmission. Traffic load was generated by a Poisson distribution [69] with rates in the range 10-100 packets/sec. Each packet only contains one Active Message whose size is defined in TinyOS 2.0 [66].

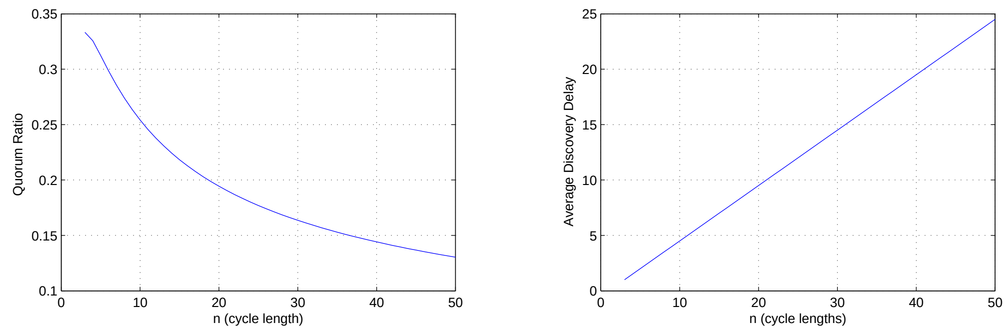
Two important performance metrics were measured in our experimental study: (1) quorum ratio and energy saving ratio; and (2) average neighbor discovery delay.

4.5.1 Performance Trade-off

We first evaluated the quorum ratio and average neighbor discovery delay by numerical analysis.

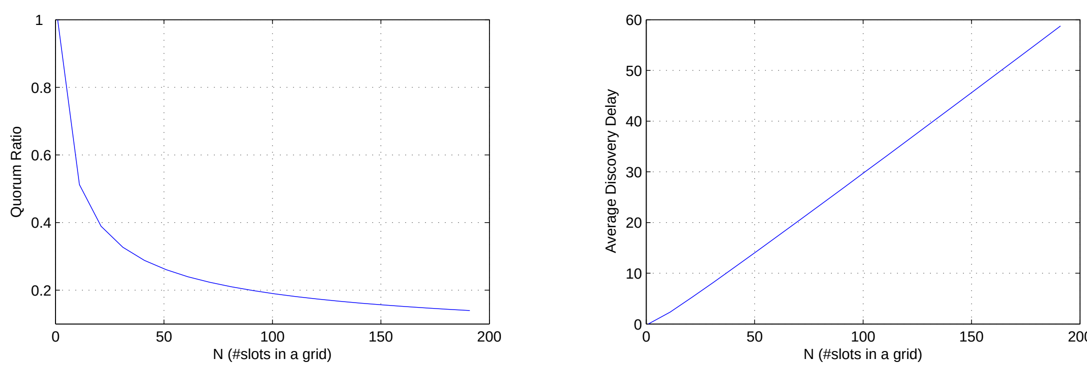
The performance of a *cyclic quorum system* is shown in Figure 4.4. There is a trade-off between quorum ratio and average discovery delay since they have reverse changing trends under increasing cycle lengths.

The performance of a *grid quorum system* is shown in Figure 4.5. The grid quorum system's quorum ratio is bigger than that of the cyclic quorum system with identical cycle length, but the average discovery delay is approximately $2/3$ of that of the corresponding cyclic quorum system.



(a) Quorum Ratio (b) Average Discovery Delay

Figure 4.4: Quorum Ratio and Average Discovery Delay for Cyclic Quorum Systems (Numerical Results)



(a) Quorum Ratio (b) Average Discovery Delay

Figure 4.5: Quorum Ratio and Average Discovery Delay for Grid Quorum Systems (Numerical Results)

4.5.2 Impact of Heterogeneity

For heterogenous quorum-based wakeup scheduling, like cqs-pair or gqs-pair, the cycle lengths of two quorum systems are different. We evaluated the impact of heterogeneity of two different cycle lengths on the average discovery delay between two neighbor nodes.

For this set of experiment, we focused on the cqs-pair since it is an optimal design. We fixed the traffic load between two nodes at $10\ packets/sec$ in the experiment.

We varied the cycle lengths of two neighbor nodes in two (different) quorum systems in a cqs-pair. The cycle length of one node was varied from 7 to 58. The neighbor node, which used a counterpart cyclic quorum system, had its cycle length varied from 7 to 21. We do not show the impact on the energy consumption ratio when the cycle lengths of cqs-pair were varied, since the energy consumption ratio of a node is mainly dependent upon its own cycle length, which has already been evaluated in Section 4.5.1.

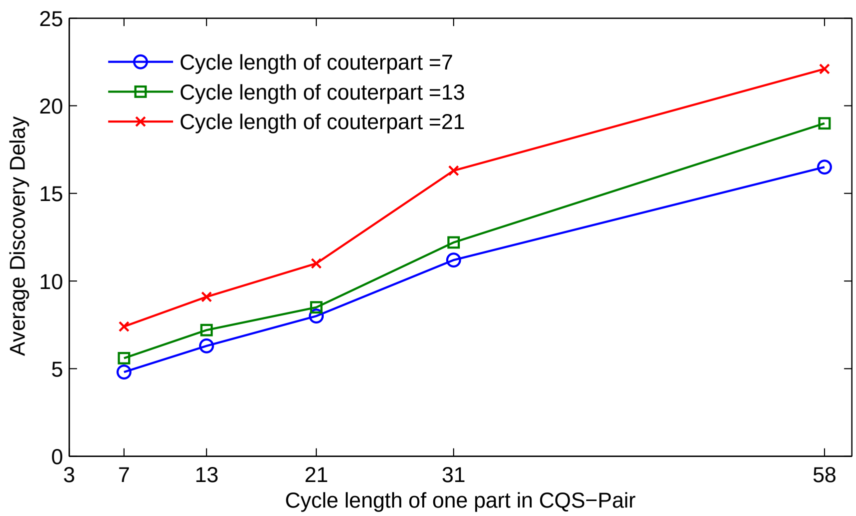


Figure 4.6: Impact of Heterogeneity

Figure 4.6 shows how the average discovery delay (standard deviation $< 2\%$) changes with different cqs-pairs. When one part in a pair keeps its cycle length constant and the counterpart increases its cycle length, the average discovery delay between them almost increases linearly.

4.5.3 Impact of traffic load

In this section, we report our experiments on measuring the impact of traffic load on the performance of cqs-pair and comparisons with the basic CSMA/CA MAC protocol. We varied the traffic load from 10 packets/sec to 100 packets/sec in the experiments. The cycle lengths of cyclic quorum systems in the cqs-pair were chosen among 7, 13 and 21.

Figure 4.7 shows how the energy consumption ratios of nodes adopting different cyclic quorum systems increase under increasing traffic load between two neighboring nodes. The rationale is that higher traffic loads will cause a node to increase its wakeup time ratio in our implementation. When the traffic load is low, the impact is insignificant because a node will maintain its current wakeup schedule, without adding more wakeup slots into its communication schedule for transmitting or for receiving packets.

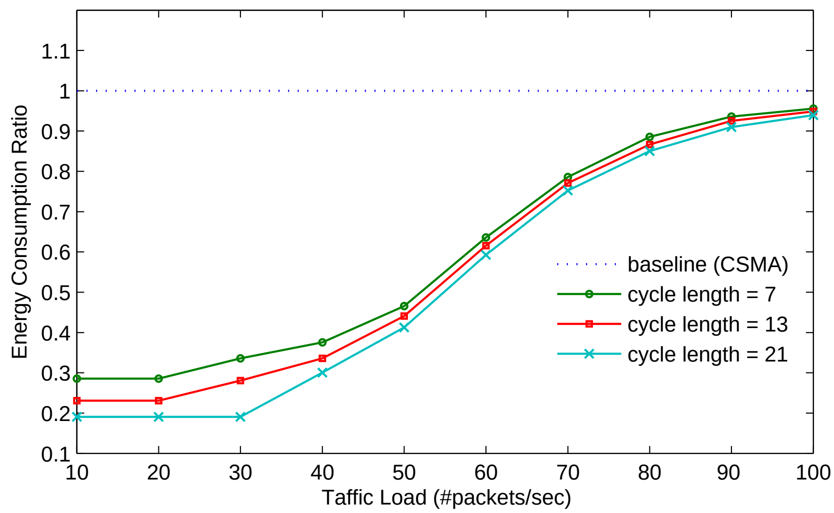


Figure 4.7: Impact of Traffic Load: Energy Consumption Ratio

Figure 4.8 shows that the average discovery delay standard deviation $< 1\%$ slot) decreases with the increasing of traffic load. This is because, the communication schedule of a node will have more active slots, when compared with its quorum-based wakeup schedule during high traffic load. With more slots staying awake, the average discovery delay between two neighboring nodes will be significantly reduced.

4.6 Conclusions

In this chapter, we presented a theoretical approach for heterogeneous asynchronous wakeup scheduling in wireless sensor networks. We first defined the h-QPS problem—i.e., given two cycle lengths n and m ($n < m$), how to design a pair of heterogeneous quorum systems to guarantee that two adjacent nodes that select heterogenous quorums from the pair as their wakeup schedules can hear each other at least once in every m consecutive time slots. We proposed two designs for heterogeneous asynchronous wakeup scheduling: the cyclic quorum system pair (cqs-pair) and the grid quorum system pair (gqs-pair). We also presented a fast construction scheme to assemble a cqs-pair. In our construction scheme, we first quickly construct an $(n, k, 1)$-difference set and

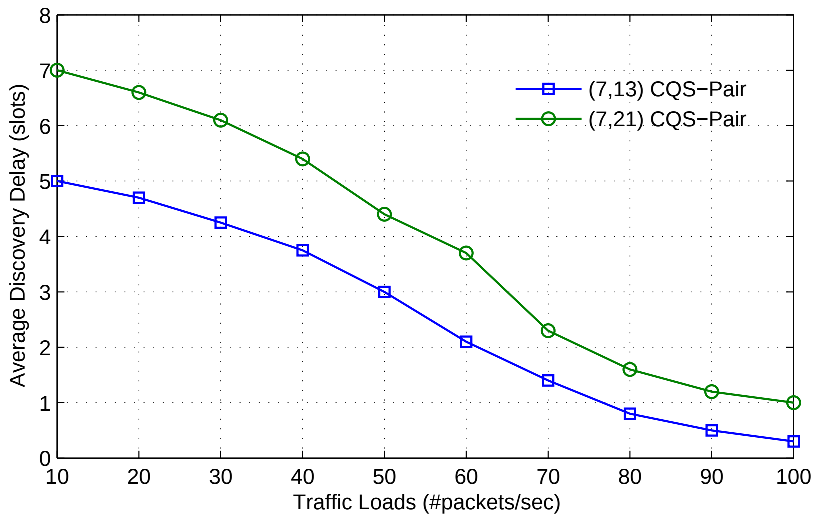


Figure 4.8: Impact of Traffic Load: average discovery delay

an $(m, l, 1)$-difference set. Based on two difference sets A in $(\mathbb{Z}_n, +)$ and B in $(\mathbb{Z}_m, +)$, we can construct a cqs-pair $(C(A, \mathbb{Z}_n), C(B, \mathbb{Z}_m))$ when A and B can form a (n, k, m, l)-difference pair.

The performance of a cqs-pair and a gqs-pair were analyzed in terms of average delay, quorum ratio, and energy saving ratio. We show that the average delay between two nodes that wakeup via heterogenous quorums from a cqs-pair is bounded between $\frac{n-1}{2}$ and $\frac{m-1}{2}$, and the quorum ratios of the two quorum systems in the pair are optimal, respectively, given their cycle lengths n and m. For a gqs-pair with $\sqrt{n} \times \sqrt{n}$ grid and $\sqrt{m} \times \sqrt{m}$ grid, the average discovery delay is bounded within $\frac{(n-1)(\sqrt{n}+1)}{3\sqrt{n}} < E(Delay) < \frac{(m-1)(\sqrt{m}+1)}{3\sqrt{m}}$, while the quorum ratios are $\frac{2\sqrt{n}-1}{n}$ and $\frac{2\sqrt{m}-1}{m}$, respectively.

Chapter 5

Q-MAC: Asynchronous Rendezvous over Quorum-based LPL Scheduling

We extend the quorum-based wakeup scheduling in slotted listening mode to that in the LPL mode. Our work is called Q-MAC, a MAC protocol that combines quorum-based wakeup scheduling with low-power listening (LPL), to provide an asynchronous neighbor discovery, run-time configurable, ultra low duty cycle (i.e. 1%) solution for wireless sensor networks. Q-MAC provides configuration flexibility in duty cycle by selecting different pairwise quorums as preamble sampling schedules, which is different from the conventional approach of periodical preamble sampling in B-MAC [1] and X-MAC [2] protocols. Furthermore, Q-MAC can guarantee asynchronous neighbor discovery within bounded latency. Q-MAC's quorum-based wakeup scheduling is based on the concept of *cqs-pair*, which contains two cyclic quorum systems with the closure intersection property.

5.1 Basic Designs

We now use the cqs-pair concept to design the Q-MAC protocol by combining dual preamble sampling to provide a flexible configuration solution for asynchronous low power listening mechanism at the MAC layer. We assume that the checking intervals (time slot) are identical for all nodes in a network in the Q-MAC design.

5.1.1 Quorum-based LPL Scheduling and Communication Control

Previous works on low power listening (or LPL) adopt periodic preamble sampling [1, 2] mechanisms in which a node checks the state of its channel once every x time units, where x is usually 100ms or 200ms. If the gain of the channel is less than a certain threshold level, it means that there are no activity from its neighbor and the node will go back to sleep.

In order to allow flexible run-time configuration of duty cycle, we combine quorum-based scheduling with the LPL mechanism. In the design of Q-MAC, a node will not periodically check the channel state via preamble sampling as in B-MAC. Rather, the sampling pattern will follow a cqs-based schedule. If a cqs adopts $(n, k, 1)$-difference set design, the corresponding sampling schedule will be to check k times in every n consecutive checking intervals based on the $(n, k, 1)$-difference set. We denote the intervals in which the channel state will be checked as *quorum intervals*, and the intervals in which the channel state will not be checked as *non-quorum intervals*, to differentiate the checking interval from those in B-MAC or X-MAC.

An example is given in Figure 5.1(a). When the cqs has a cycle length of 7, for any consecutive 7 intervals, a node will only make short preamble sampling in the 1^{st}, 2^{nd}, and 4^{th} intervals if the quorum is $\{1, 2.4\}$. The 1^{st}, 2^{nd}, and 4^{th} intervals are also referred as *quorum intervals*.

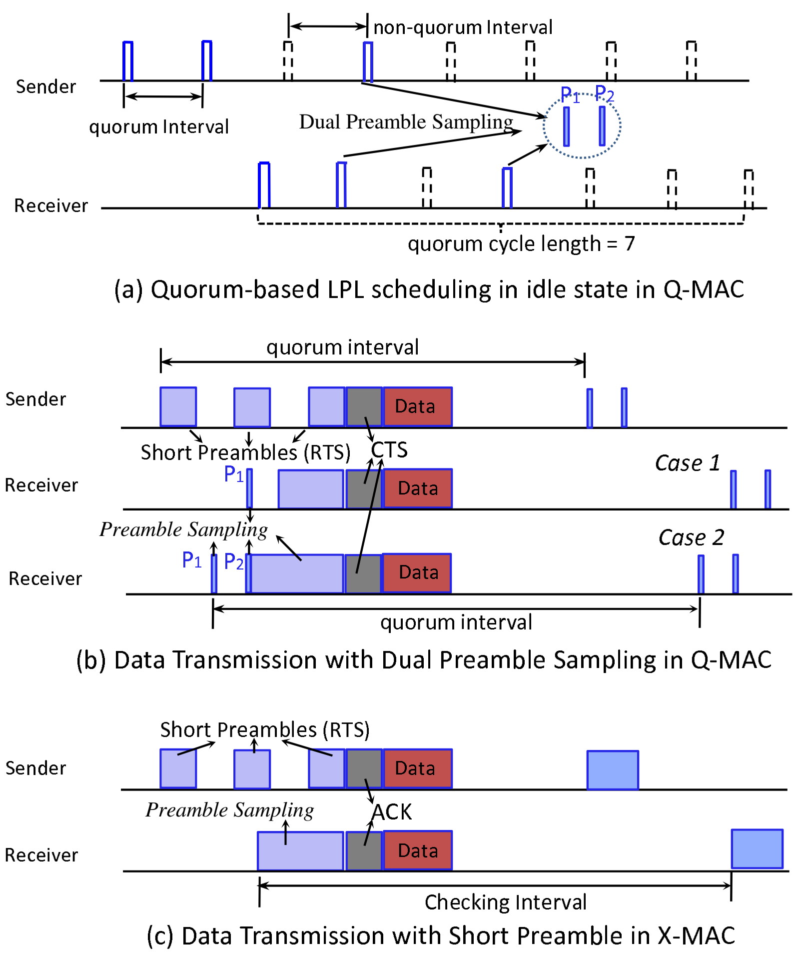


Figure 5.1: Quorum-based Asynchronous Rendezvous Scheme in Q-MAC and Mechanism of X-MAC

If two nodes select quorum systems satisfying the rotation closure property, we can guarantee that two nodes will have at least one overlapped *quorum interval*.

Most standard LPL based duty-cycled MAC protocols adopt long preambles, which suffer from energy waste and long discovery delays. Short strobed preambles [2] overcome the disadvantages of long preamble lengths. To further reduce the idle listening time, we adopt a dual preamble sampling mechanism which is similar to that in [50]. The design of the communication control of Q-MAC is as follows.

In idle state, each node checks the channel state by *dual preamble sampling* in two separate periods P_1 and P_2 at the beginning of each quorum interval in idle state, as illustrated in Figure 5.1(b).

When there is data for transmission, a sender in Q-MAC sends out a series of short preambles, which include the receiver ID. We denote the short preamble message as *RTS*. Then the sender waits for the acknowledgement, denoted as *CTS*, from the receiver for a short time period. Neighbor nodes check their channel state in P_1 and P_2. If the channel is not free in P_1 (i.e., case 1 in Figure 5.1 (b)), nodes enter into the IDLE mode for a period of time and proceed to receive a RTS messages to determine whether it is the receiver. The period P_2 is needed when the channel is found to be free in P_1 (i.e., case 2 in Figure 5.1 (b)). If a node is the receiver, the node will immediately send back an acknowledgement message (CTS) to the sender.

If the sender does not receive a CTS message after sending a RTS message, it will continue to send out RTS messages until the current quorum interval is expired. The sender will continue to send out RTS messages in the next quorum interval. If the coming interval is not a quorum interval, the sender will keep its SLEEP state until the arrival of a quorum interval.

Compared with the short strobe preamble scheme in XMAC, shown in Figure 5.1(c), our mechanism is more energy efficient due to two aspects:

1. There are less preamble sampling for quorum-based LPL scheduling, when compared with periodic preamble sampling; and
2. Dual preamble sampling will consume less energy since there is a period of IDLE state between the two short sampling periods P_1 and P_2.

In a cyclic quorum system based on the $(n, k, 1)$-difference set design, we denote n as the quorum cycle length and k as the quorum size. We list all parameters and their default values for Q-MAC design in Table 5.1.

We determine the value of $T_{P_1-P_2}$ as follows:

$$T_{P_1-P_2} = \min\{T_{CTS}, T_{RTS} - 2T_s\}$$

In our implementation, we set $T_{RTS} = T_{RTS} + 2T_s$. Thus, $T_{P_1-P_2}$ can be simply stated as:

$$T_{P_1-P_2} = T_{CTS}$$

According to measurements in [1], we set the default value for preamble sampling (T_s) as 3 milliseconds when a node performs a sequence of operations to startup radio and set the radio in the

Table 5.1: Parameters for Q-MAC design

Notation	Parameter	Default
n	quorum cycle length	1,3,7,13
k	quorum size	1,2,3,4
T_i	(non-)quorum interval (s)	$100E-3$
T_{RTS}	time duration of a RTS message	-
T_{CTS}	time duration of CTS message	-
T_s	duration for channel sampling(s)	$3E-3$
T_{P_1}	minimum period P_1	T_s
T_{P_2}	minimum period P_2	T_s
$T_{P_1-P_2}$	time interval between P_1 and P_2	-

RX mode. Thus the total time duration for the dual preamble sampling is 6 milliseconds.

5.2 Advanced design

5.2.1 Broadcast Support

Quorum-based asynchronous wakeup protocols cannot guarantee that more than one receivers are awake at the same time when a sender wishes to broadcast. Thus how to support broadcast is a non-trivial issue in the Q-MAC design.

One straightforward way is to replace broadcast by multiple unicasts. However, this approach is not efficient when there are many (e.g., 10) neighbors for a sender. Another way is by using synchronization over quorum-based wakeup schedules. The sender can book-keep all neighbors' schedules, and synchronize their schedules so that neighboring nodes wake up in the same set of intervals with the use of Lamport's clock synchronization algorithm [68]. When all nodes are awake simultaneously, the senders then send a message to multiple neighbors simultaneously. The disadvantage of this approach is that it needs book-keeping and time synchronization.

We propose a more efficient way to support broadcast in Q-MAC. In our design, the sender do not need to time synchronize all its neighbors and do not need to use multiple unicast messages. In our approach, a sender uses limited number of RTS messages to notify all neighbors that are keeping awake. Then the sender broadcastes data in its final *quorum interval* of the current quorum cycle. The rationale is that all neighbors of the sender must overlap their quorum interval, partially or wholly, with one of the sender's *quorum interval* within one quorum cycle due to the *heterogenous rotation closure property*.

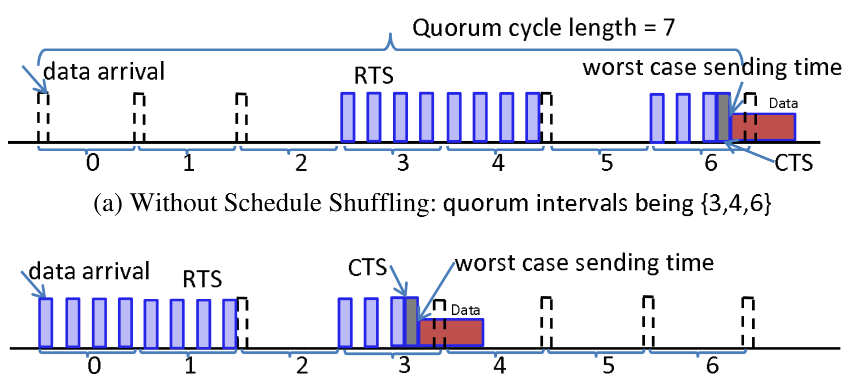
When the sender and receiver adopt heterogenous quorum systems, the sender should temporally switch its quorum schedule to the quorum schedule in the system with larger quorum cycles. For example, if they adopt a $(7, 13)$-pair, then the sender should temporally switch to the cqs with cycle length of 13 for broadcast.

To implement this, we add a field in the RTS message to inform all neighbors that are keeping awake for receiving a broadcast message. The number of RTS messages that are send out before the actual broadcasting is equal to the quorum size of the sender's schedule multiplying the number of RTS messages in a single quorum interval.

5.2.2 Quorum Schedule Shuffling

In order to reduce the discovery latency in Q-MAC, we adopt the mechanism of quorum schedule shuffling, as an *optional* feature of Q-MAC. The shuffling means that a node will restart its quorum-based preamble sampling schedule. The purpose of shuffling is to schedule a *quorum interval* as soon as possible so that the sender can send out RTS messages immediately.

The rationale is as follows: the worst-case discovery latency is equal to the distance between the first quorum intervals and the last quorum intervals. The value of such distance depends on the quorum selection, i.e., it is 4 intervals for the quorum $\{0, 1, 3\}$, but is 7 for quorum $\{3, 4, 6\}$ as illustrated in Figure 5.2. If we adopt quorum schedule shuffling, we can always guarantee that the worst case distance is the shortest one, when compared with scheduling without shuffling.



(a) Without Schedule Shuffling: quorum intervals being {3,4,6}

(b) With Schedule Shuffling: quorum intervals being {0,1,3}

Figure 5.2: Worst case data transmission in Q-MAC without shuffling and with shuffling

Quorum schedule shuffling will not increase the duty cycle since it only reschedules the start time of a quorum interval without introducing any additional quorum intervals.

5.2.3 Application in Two Tiered Topology

In tiered topology for wireless sensor networks, there are heterogenous entities, i.e., cluster head and cluster members. Usually, the cluster members do not need to communicate with each others, and the most of data transmission is from cluster members to cluster head, i.e., in target tracking or environment monitoring applications.

To apply Q-MAC into two tiered topology, we use cqs-pair for the quorum-based LPL scheduling. The cluster head should adopt a cqs with smaller quorum cycle as its LPL schedule, and the cluster members should adopt a cqs with bigger quorum cycle as its LPL schedule.

We will show in Section 5.3 that the discovery latency between smaller quorum cycles is shorter. Thus by adopting smaller quorum cycles between all cluster heads which may act as virtual routers in WSNs, the end-to-end transmission latency will be shorten. However, the energy consumption in idle state for cluster head will be more than that of cluster members due to bigger quorum ratio of cluster head. To compensate such imbalance of energy consumption, the rotation mechanism [33] can be applied by rotating the roles of all nodes.

When there is data for broadcasting, the cluster head can temporally switch to the quorum system with larger quorum cycle in the pair in order to reduce the number of RTS messages that are send out before data broadcasting, as mentioned in Section 5.2.1.

5.3 Performance Modeling

We first analyze the performance for homogenous cqs in Q-MAC. This analysis is the basis for the performance analysis of heterogenous cqs in Q-MAC.

5.3.1 Homogenous Cyclic Quorum Systems

Suppose the length of a quorum interval is T_i and and the quorum cycle length is n for a cqs.

Duty Cycle in idle state: Suppose the duty cycle in a single quorum interval is $\alpha = \frac{T_s}{T_i}$ where T_s is the duration for channel sampling, and the quorum system is based on the design of (n, k, l)-difference set.

The quorum ratio (proportion of the quorum size in a whole quorum cycle) is $\frac{\sqrt{4n-3}-1}{2n}$ based on [35]. The overall *duty cycle* for such a system in idle state is:

$$\phi = \frac{\sqrt{4n-3}-1}{2n} \cdot \alpha \tag{5.1}$$

If $\alpha = 5\%$, the duty cycle is 2.15% for a cqs with quorum cycle length of 7, and 1.53% for the a cqs with quorum cycle length of 13. The relationship between quorum ratio and quorum cycle length is shown in Figure 5.3.1.

Discovery Latency: We denote the *discovery latency* as the time between data arrival and discovery of the adjacent receiver (i.e., receiving the CTS message from the receiver). Note that this metric does not include the time for delivering a message.

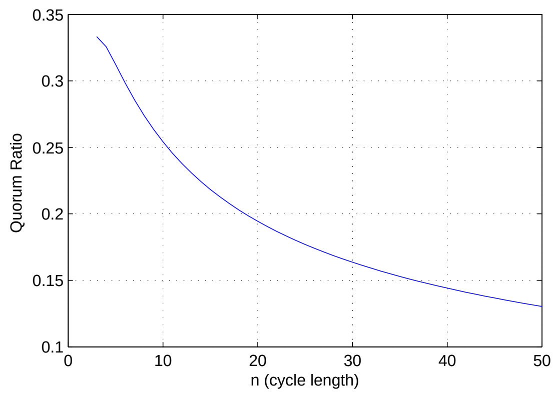


Figure 5.3: Quorum ratio for different cycle lengths

Theorem 11. *Without quorum schedule shuffling, the worst case discovery latency between two nodes that wakeup based on same* cyclic quorum systems *adopting the* $(n, k, 1)$*-difference set is:* $D = n \cdot T_i$.

Since same quorum systems must form a cqs-pair, there must have an intersection for the two schedule. As shown in Figure 5.2, the worst cast intersection may happen in the last interval in a quorum cycle which has the discovery latency of $n \cdot T_i$. We skip the detailed proof.

Theorem 12. *Without quorum schedule shuffling, the average case discovery latency between two nodes that wakeup based on same* cyclic quorum systems *adopting the* $(n, k, 1)$*-difference set is:* $D = \frac{n}{2} \cdot T_i$.

Proof. Let the k elements in $(n, k, 1)$-difference set be denoted as a_1, a_2, $\cdots$, a_k. If a node has a message that arrived during the i^{th} time interval, the expected latency for two quorum systems having the first intersection is $\frac{1}{k}(a_1 - i)\ mod\ n + \frac{1}{k}(a_2 - i)\ mod\ n + \cdots + \frac{1}{k}(a_k - i)\ mod\ n$.

If a message has arrived in the sender, the probability of the message arriving during the i^{th} time slot is $\frac{1}{n}$. Thus, the expected delay latency for two quorum systems having the first intersection is:

$$\begin{aligned}
E(Delay) == &\frac{1}{n}[\frac{1}{k}(a_1-1)\, mod\, n + \frac{1}{k}(a_2-1) \mod n + \cdots \\
&+ \frac{1}{k}(a_k-1)\, mod\, n \\
&+ \frac{1}{k}(a_1-2)\, mod\, n + \frac{1}{k}(a_2-2)\, mod\, n + \cdots \\
&+ \frac{1}{k}(a_k-2)\, mod\, n + \quad \cdots\cdots \\
&+ \frac{1}{k}(a_1-n)\, mod\, n + \frac{1}{k}(a_2-n)\, mod\, n + \cdots \\
&+ \frac{1}{k}(a_k-n)\, mod\, n] \\
&= \frac{1}{nk}\cdot(k\cdot 1 + k\cdot 2 + \cdots + k\cdot n - 1) \\
&= \frac{n-1}{2}(\cdot T_i)
\end{aligned} \tag{5.2}$$

After the first intersection, the average delay for the sender receiving acknowledgement (CTS) from the receiver is $1/2$ interval. Thus the total expected latency is $\frac{n}{2}\cdot T_i$. The theorem follows. □

Theorem 13. *With quorum schedule shuffling, the average discovery delay between two nodes that wakeup based on quorums from the same* cyclic quorum system *adopting the* $(n, k, 1)$*-difference set is:*

$$E(Delay) = \frac{n+1}{4}\cdot T_i$$

Due to space limitation, we skip the proof for Theorem 13. The proof is very similar as that for the Theorem 12.

The average discovery delay is $3.5T_i$ for cqs with quorum cycle length of 7 without quorum schedule shuffling, and $3.25T_i$ for the cqs with quorum cycle length of 13 with shuffling.

5.3.2 Heterogenous Cyclic Quorum Systems

If two nodes adopt cqs-pair as their LPL schedules, the duty cycle will separately follow the Equation 5.1 in idle states. For example, in a two tiered topology, if the cluster head adopts the $(3, 2, 1)$-difference set for the design of its LPL schedule, and the cluster members use $(13, 4, 1)$-difference sets for the design of their schedules, and $\alpha = 5\%$, the actual duty cycle will be 3.33% and 1.53%, respectively.

Regarding the discovery latency, without quorum scheduling shuffling, we have:

Corollary 4. *Without quorum schedule shuffling, the average discovery latency between two nodes that wakeup based on a cqs-pair in which two* cyclic quorum systems *have cycle lengths n and m ($n \leq m$), respectively, is:*

$$\frac{n}{2} \cdot T_i < E(Delay) < \frac{m}{2} \cdot T_i$$

Corollary 4 indicates that the average discovery latency between two nodes that adopt a cqs-pair is bounded. When the average one-hop delay constraint is D, we must meet $\frac{m}{2}T_i \leq D$.

Corollary 5. *With quorum schedule shuffling, the average discovery latency between two nodes that wakeup based on a cqs-pair in which two* cyclic quorum systems *have cycle lengths n and m ($n \leq m$), respectively, is:*

$$\frac{n+1}{4} \cdot T_i < E(Delay) < \frac{m+1}{4} \cdot T_i$$

We skip the proofs for Corollary 4 and 5 since they follow the Theorem 12 and 13 respectively.

5.4 Experimental Results

In order to evaluate and demonstrate the correctness and benefits of Q-MAC, we evaluated the performance of our scheme through experimental studies in real implementation over a WSN platform of Telosb motes [40] which was developed at the University of California at Berkeley. we have implemented the protocol on top of LPL interface in the TinyOS 2.0 [66]. The radio used by the TelosB is the Chipcon CC2420, which is an 802.15.4 compliant device.

An application initiates the protocol by calling the APIs `setInterval(nx_uint32_t)` and `setQuorum(nx_uint16_t,nx_uint16_t)`. The function `setInterval(nx_uint32_t)` sets the duration for a quorum interval which is equal to T_i (default value is 100ms). The function `setQuorum(nx_uint16_t,nx_uint16_t)` sets the quorum-based pattern for LPL scheduling. For example, if quorum is based on a $(7, 3, 1)$-difference set design, then the input parameters for the function are 7 and 3.

We choose two competitors for the purpose of comparison: X-MAC [2] and Disco [57], which both are asynchronous MAC protocols. We do not compare with B-MAC, since X-MAC is the improved version of B-MAC. However, we do not implement the adaptive algorithm-part of X-MAC. This simplification will not result in any loss of comparison fairness, since we are mainly concerned with how Q-MAC achieves better performance in terms of duty cycle and discovery latency.

5.4.1 Experimental Configuration

For our experimental evaluation, the values of the dual preamble sampling periods (T_{P_1} and T_{P_2}) were set to 3 ms which is based on the measurement results in [1]. The duration of the preamble containing the RTS message (T_{RTS}) was set to 13ms. The duration of the preamble containing the CTS message (T_{CTS}) was set to 7ms. Thus, the gap between P_1 and P_2 ($T_{P_1-P_2}$) was set to 7 ms (13- 2*3=7). We summarize the configuration of Q-MAC in our experimental study in Table 5.2.

Table 5.2: Experimental Parameters

Channel bandwidth	250kbps
Data packet size	100bytes
RTS/CTS message size	16bytes
Preamble sampling duration T_s (ms)	3
$T_{P_1-P_2}$	7

In our experiments, we chose a chain topology containing 6 nodes. The radio range was configured to 10 meters (power level at -25 dBm) for each node according to the Telosb datasheet [40]. There was one source node which generated traffic at one end of the chain. There was one sink node which acted as the base station. The sink node communicated with a laptop computer (through a wireless USB serial port), which was used for monitoring and recording the performance measurements. The other nodes acted as routing points. We choose chain topology because there is no overhearing problem in Q-MAC so that it was not important for us to show the impact of network density on performance in terms of duty cycles and discovery latencies.

We used the MintRoute [44] as the routing protocol. The traffic load was generated by a constant bit rate (CBR) mode with variable rates in the range of 0.1 $\sim$ 1 packets/second to mimic the low traffic pattern in wireless sensor networks [70]. Each data packet only contains one Active Message whose size is defined in TinyOS 2.0 [66].

Three important performance metrics were measured in our experimental study: (1) duty cycles which reflect the energy saving ratio; and (2) neighbor discovery latency; and (3) packet delivery ratio.

5.4.2 Duty Cycle

In this section, we demonstrate the energy efficiency of Q-MAC. As the energy consumption of the radio is low in idle mode and that the processor consumes an extremely small amount of energy in comparison with the radio, we only evaluated the duty cycle of radio in our experiment, rather than the real energy consumption, to measure energy efficiency.

For Q-MAC, we set the awake time for the dual preamble sampling as $3ms$ for each sampling duration, and the checking interval as $100ms$. We used (7,7)-pair for LPL scheduling, which has a duty cycle of $\frac{3}{7} \cdot \frac{2*3}{100} = 2.5\%$, without traffic load. For X-MAC, in order to make a fair comparison, we set the duration of short strobed preamble as 15ms which is same as that in [2], and the checking interval as 600ms (duty cycle=$\frac{15}{600} = 2.5\%$). For the Disco protocol, we used the $(37, 43)$ balanced primes and symmetric pairs ($\frac{1}{37} + \frac{1}{43} = 5\%$, so that the average duty cycle was 2.5% for a single node) and the slot length was 10ms. Thus, the duty cycles for all mechanisms without traffic were identical.

The experiment was repeated with varying CBR traffics from 0.1 $\sim$ 1 packets/second. The transmissions were sequential so as to avoid contention. The results (standard deviation $< 2\%$ for 10 times of experiments) are shown in Figure 5.4. As shown in the figure, the duty cycle increases slowly with increasing traffic loads in Q-MAC and X-MAC. Also, the duty cycle of Disco does not change much, since its duty cycle was mainly determined by the prime selection.

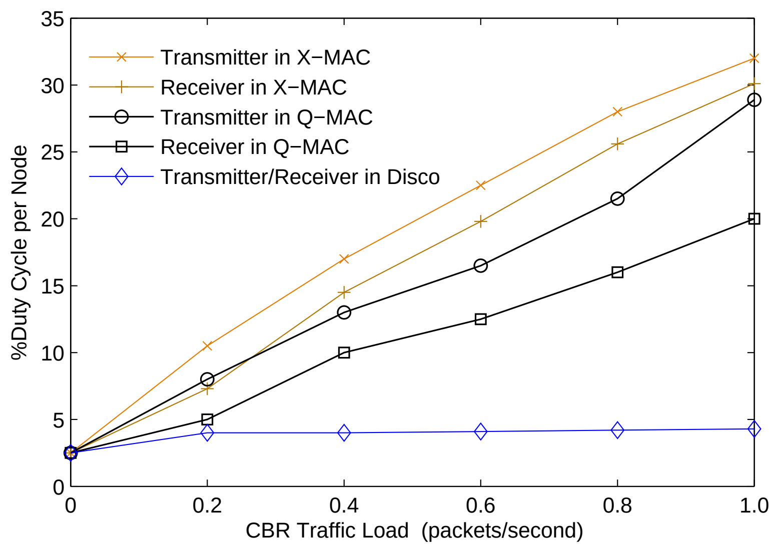


Figure 5.4: Duty cycles of senders and receivers under different traffic load

Observe that Q-MAC achieves slightly lower duty cycle when compared with X-MAC. This is because, Q-MAC uses dual preamble sampling which consumes less energy when compared with one single short preamble sampling (15 ms). In addition, with the same duty cycle configuration in the idle state, Q-MAC adopts a shorter checking interval (100ms vs 600 ms), and hence sends out less number of RTS messages before receiving a CTS message from the receiver.

Note that the duty cycle of Q-MAC is higher than that of Disco in Figure 5.4. However, Disco achieves lower duty cycle at the expense of longer discovery latency and lower discovery ratio (we illustrate this in Section 5.4.3).

5.4.3 Discovery Latency

Discovery latency refers to the delay between data arrival and discovery of the adjacent receiver (i.e., via receiving the CTS message from the receiver). The distribution of discovery latencies, as well as the worst-case latency, are both important metrics in evaluating the performance of an asynchronous MAC protocol.

In our experiment, we first measured the one-hop discovery latency between a sender and a receiver. The protocol configurations for Q-MAC, X-MAC, and Disco were the same as that in Section 5.4.2 to ensure that they had identical duty cycles in the idle state.

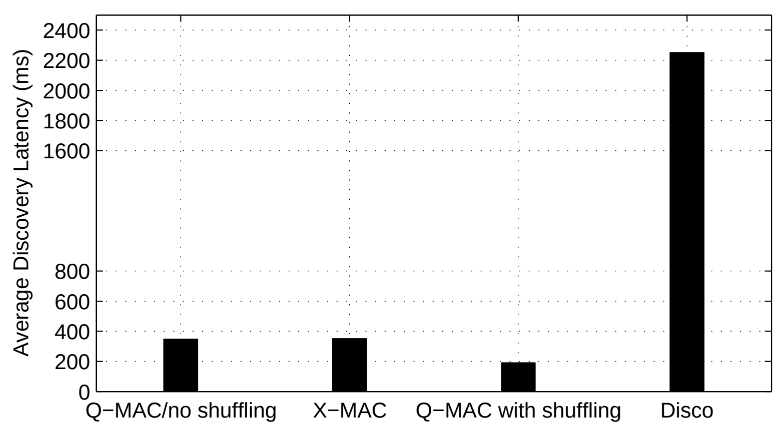


Figure 5.5: One-hop discovery latency

In Figure 5.5, we observe that Q-MAC without shuffling can achieve almost the same latency (standard deviation < 90 ms for repeating experiment 10 times) as X-MAC. Although Q-MAC may need more quorum intervals for a node to intersect with its neighbor's schedule, Q-MAC uses smaller checking intervals (e.g., 100ms), which can still guarantee lower discovery latencies. With the help of quorum schedule shuffling, we observe that the discovery latency is further reduced for Q-MAC. For Disco, its discovery latency is significantly high, as it takes more than hundreds of slots for neighbor discovery.

We also measured the discovery latency for different cqs-pairs. In this set of experiments, we did not change the checking interval (100ms) for all the cqs-pairs. Figure 5.6 shows the impact of different cqs-pair settings on discovery latency. From the figure, we observe that the discovery latency is almost proportional to the sum of the quorum cycle lengths of the pairwise quorum systems in a cqs-pair.

Finally, we measured the end-to-end transmission delay under the chain topology which contained 6 nodes. We generated one packet every second at one end of the chain, and transmitted the packet over multi-hops to different destinations. We adopted the $(7, 7)$-pair in Q-MAC.

In Figure 5.7, we show that the end-to-end delay is increased with the hop count. Our experiments

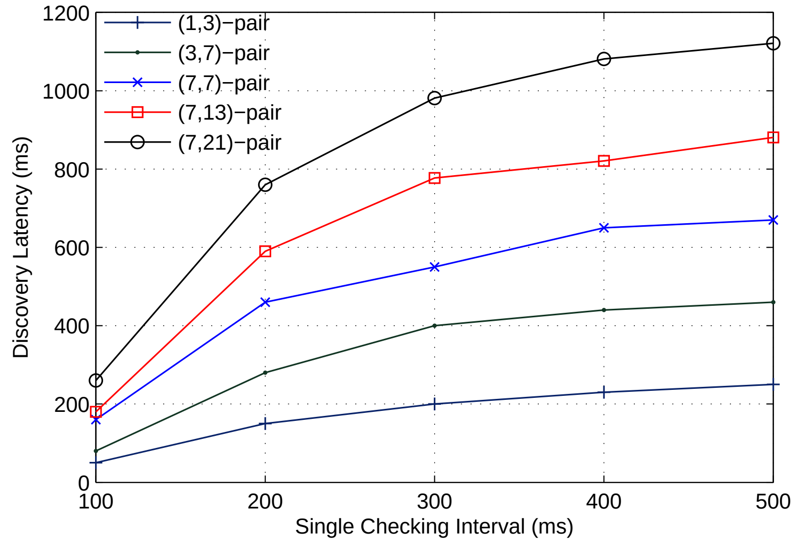


Figure 5.6: Discovery latency for different cqs-pairs

result also show that Q-MAC with schedule shuffling reduces end-to-end delay by approximately 30% when compared with X-MAC, under the same duty cycle in the idle state (2.5%). This is because Q-MAC uses shorter checking interval (100ms vs 600ms). We do not compare with Disco since Q-MAC significantly outperforms Disco in terms of end-to-end transmission delay. Q-MAC has a more tight bound for discovery latency. In our experiments, the bound for Q-MAC was 7 checking intervals, but Disco consumed almost thousands of slots for end-to-end transmission regarding $(37, 43)$ balanced primes.

5.4.4 Delivery Ratio

We define delivery ratio as the number of packets received by a receiver divided by the number of packets sent by the sender. We use the chain network topology described in Section 5.4.3 to evaluate the delivery ratio. The checking interval was set to 100ms for Q-MAC and 600ms for X-MAC. We also varied the traffic load from 0.1packets/second to 1packet/second.

From Figure 5.8, we observe similar performances for Q-MAC and X-MAC since both the two protocols use the acknowledgement mechanism i.e., the sender sending the preamble with the receiver ID and the receiver acknowledging the receipt of the preamble (RTS). In contrast, Disco achieves a low delivery ratio because the long discovery latency results in dropping of packets in the local buffer.

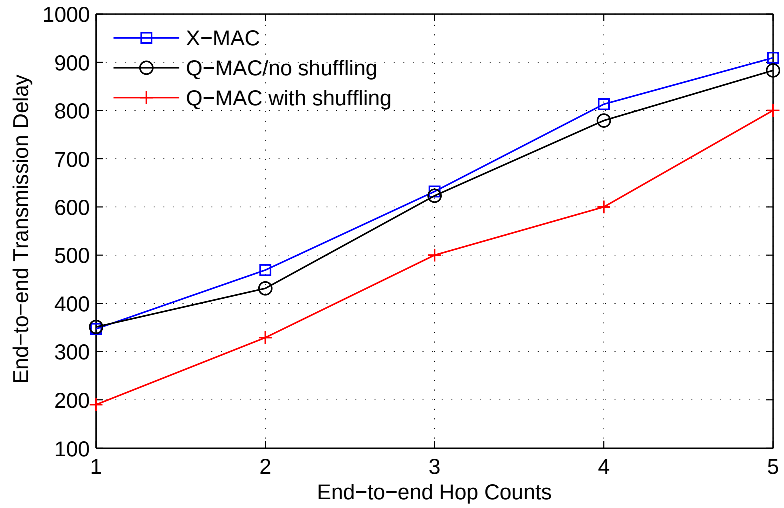


Figure 5.7: End-to-end transmission delay

5.4.5 Broadcast Latency

Since quorum-based scheduling in Q-MAC cannot guarantee that all neighbor nodes have a quorum interval that intersects simultaneously, a sender has to wait for an additional time period before all neighbor nodes become awake, for sending out broadcast data, as explained in Section 5.2.1. We refer to the time between when the broadcasting data of the sender is ready and when all of the neighbor nodes are awake as the broadcast latency.

We did not compare the broadcast latency with X-MAC and Disco since the latter two mechanisms do not explicitly present how to support broadcast. In our experiment, we varied the checking interval from 100ms to 500ms for all cqs-pairs. As shown in Figure 5.9, the broadcast latency of Q-MAC is almost linearly increased with the increase of the checking interval for different cqs-pairs.

5.5 conclusion

In this chapter, we presented a flexible solution for MAC protocol design in wireless sensor networks. The motivation was to provide a run-time configurable, ultra-low power, asynchronous neighbor discovery MAC protocol for WSN applications such as target tracking and environment monitoring.

Our solution — the Q-MAC protocol — is easy to implement: nodes select a pair of cyclic quorum systems, referred to as a cqs-pair, as their preamble sampling schedules such that they satisfy the rotation closure property. This run-time configurable protocol achieves faster discovery than other

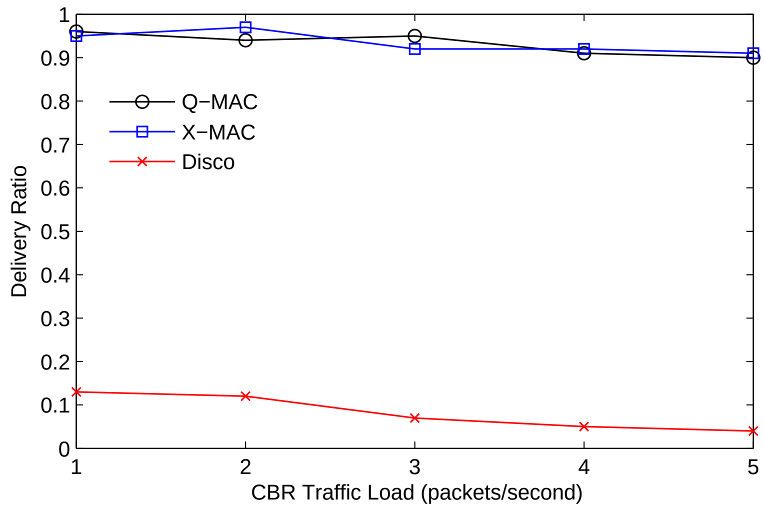


Figure 5.8: Delivery Ratio

asynchronous protocols for a given duty cycle, allows nodes to independently select their own duty cycles, provides a provable upper bound on discovery latency, and performs well in practice. The construction of a cqs-pair is based on the Multiplier Theorem and has low time complexity.

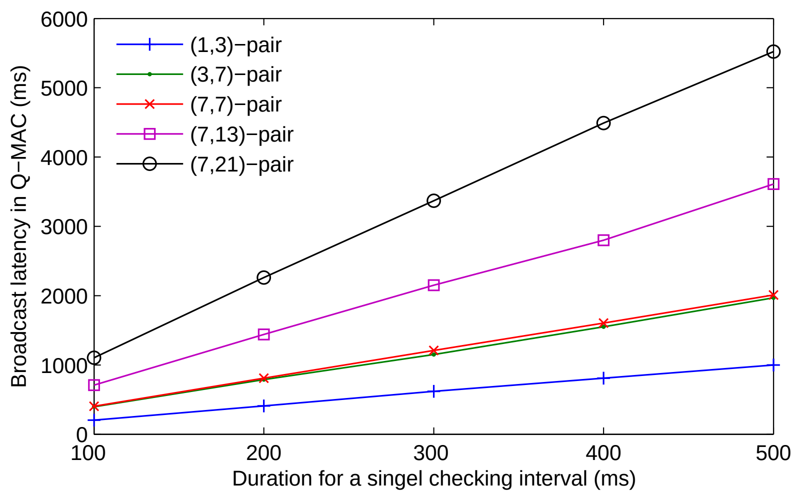


Figure 5.9: Broadcast latency with schedule shuffling

Chapter 6

Conclusions, Contributions and Proposed Post Preliminary Exam Work

Our works is motivated by saving idle energy consumption for wireless sensor network. It is based on the preservation that idle energy consumption is not neglectful for most of applications in WSNs, like environment monitoring [6], target tracking [5] etc,.

The proposed solutions were inspired by the quorum-based power saving mechanism [11]. Our solutions, cqs-pair and ggs-pair, provide a general framework in which we can satisfy the heterogenous energy saving requirement and meanwhile guarantee neighbor discovery latency. We show that the generalization is a non-trivial extension from past works.

Furthermore, we applied the heterogenous wakeup scheduling into low power listening (LPL) mode in WSNs and proposed the Q-MAC protocol. We addressed the rendezvous issues and proposed quorum-based LPL scheduling which is different from previous periodic LPL listening pattern. Our mechanism provide a run-time configurable, ultra-low power, asynchronous neighbor discovery MAC protocol for WSN, which has more merits than B-MAC [1] and X-MAC [2] in terms of idle energy saving and end-to-end data delivery delay.

6.1 Contributions

In this proposal, we studied a number of issues. In particular, our main goal is to maintain network connectivity in duty-cycled network with quorum-based wakeup scheduling. Here we use the term "connectivity" loosely, in the sense that a topologically connected network in our context may not be connected at any time; instead, all nodes are reachable from a node within a finite amount of time.

Towards that end we have conducted a series of studies. Specifically, for the design rationale (described in Chapter 4), we first attempt to design a heterogenous quorum-based wakeup schedule for WSNs with slotted listening mode, and then extend the works to duty-cycled WSNs with LPL mode which is more energy efficient and sacrifice little in neighbor discovery latency. The current research status and contributions are concluded as follows.

We proposed *cyclic quorum system pair* [35] (or *cqs-pair*) which guarantees that two adjacent nodes which adopt heterogenous cyclic quorums from such a pair as their wakeup schedules, can hear each other at least once within one super cycle length (i.e., the larger cycle length in the cqs-pair). In [35], we also presented a fast algorithm for constructing cqs-pairs, using the*multiplier theorem* [36] and the (N, k, M, l)-difference pair defined by us. Given a pair of cycle lengths (n and m, $n \leq m$), we show that the cqs-pair is an optimal design in terms of energy saving ratio. The fast construction scheme significantly improves the speed of finding an optimal quorum, in contrast to previous exhaustive methods [37]. We also analyze the performance of cqs-pair in terms of expected delay ($\frac{n-1}{2} < E(delay) < \frac{m-1}{2}$), quorum ratio, and energy saving ratio. With the help of the cqs-pair, wireless sensor networks can achieve better trade-off between energy consumption and average delay. For example, all cluster-heads and gateway nodes can pick up a quorum from the quorum system with smaller cycle length as their wake up schedule, to get smaller discovery delay. In addition, all members in a cluster can choose a quorum from the system with longer cycle length as their wakeup schedules, in order to save more idle energy.

Our another research work on quorum-based wakeup scheduling is *gqs-pair* [38]. In this work, we presented another design for heterogenous quorum system pair. The design is called the *grid quorum system pair* (or *gqs-pair*) in which each quorum system of the pair is a grid quorum system [37]. we prove that any two grid quorum systems can form a gqs-pair. Comparing with cqs-pair, gqs-pair has better performance in terms of average neighbor discovery latency. We show that for a gqs-pair with $\sqrt{n} \times \sqrt{n}$ grid and $\sqrt{m} \times \sqrt{m}$ grid, the average discovery delay is bounded within $\frac{(n-1)(\sqrt{n}+1)}{3\sqrt{n}} < E(Delay) < \frac{(m-1)(\sqrt{m}+1)}{3\sqrt{m}}$, while the quorum ratios are $\frac{2\sqrt{n}-1}{n}$ and $\frac{2\sqrt{m}-1}{m}$, respectively. With the help of the gqs-pair, WSNs can also achieve trade-off between energy consumption and average delay.

In the followup work, we further apply the quorum-based wakeup scheduling into low power listening mode for duty-cycled wireless sensor network. We proposed *Q-MAC* [39], a quorum-based asynchronous duty cycling MAC protocol. In Q-MAC, a node will not follow periodic, duty cycled low-power listening (LPL) as in B-MAC or X-MAC, but a quorum-based LPL scheduling. Q-MAC can guarantee low duty cycle (e.g. 1%) and yet still ensure the discovery of neighbor nodes within bounded delay. The primary advantage is the run-time configuration, which allow flexible adjustment on power saving policy to reflect the available energy or different workloads in heterogenous networks. Although Q-MAC requires global agreement on some basic LPL parameters, like the length of channel checking interval and preamble sampling period (equal to the awake time) in each interval, this does not preclude Q-MAC from independently choosing its quorum-based LPL patterns.

We use the cqs-pair [35] concept to design the Q-MAC protocol by combining dual preamble sampling to provide a flexible configuration solution for asynchronous low power listening mechanism at the MAC layer. To use Q-MAC, a node can choose its quorum-based LPL schedule so long as the quorum schedule will intersect at least once with the quorum schedules of its neighbor nodes in bounded time.

The experimental evaluation from real implementation of Q-MAC illustrates that Q-MAC can achieve flexible tradeoffs between saving energy and bounding end-to-end transmission delay.

We implemented cqs-pair, gqs-pair and Q-MAC in a wireless sensor network platform comprised of Telosb [40] motes and tested their performance in terms of energy-saving ratio, neighbor discovery latency, and deliver ratio. The work is presented in [38, 39]. For Q-MAC, we implemented the protocols on top of the LPL interface in TinyOS. The radio used by TelosB is the Chipcon CC2420, which is an 802.15.4 compliant device. We developed a variety of APIs which an application initiates the protocol. Our implementation do not need to modify the upper layer routing protocols.

6.2 Post Preliminary Exam Work

Based on our current direction of research, we propose the following works with the top priority for our post preliminary exam work:

1. Investigate the possibility of further improving energy efficiency via asymmetric design;
   We observe that it is not necessary to guarantee intersection in idle state since there is data for transmission and nodes do not need to find out each other in such case. The intersection is only needed for data transmission via unicast or multicast. Let's define the data transmission as *write operation*, the idle listening as *read operation*.
   Thus it is possible to propose the concepts of read quorum, write quorum, read-write quorum in a quorum group. In order to save energy, it is only necessary to guarantee the intersection property between any two quorums, i.e., read quorums will not intersect with each other, but a read-quorum will intersect with a write quorum or a read-write quorum. Hence if a node adopts read quorum in idle state, and switch to write-quorum or read-write quorum in case of data transmission, we can guarantee the network connective, and meanwhile, provide higher energy efficiency.
   For the coming step, we may design a new protocol which is based such quorum group. An example design may be grid quorum group, i.e., a read quorum consists of an column of element in a grid, a write quorum consists of an row of elements in the grid, a read-write quorum consists of an row plus an column of elements in the grid. We will design a protocol based on quorum groups to achieve better energy saving ratio and discovery latency, and are easily implemented for WSNs.
2. Cross-layer design with routing protocol adaption;
   With quorum-based rendezvous mechanism in MAC layer, both the one-hop delivery latency and the end-to-end delay will be affected. Here, we refer the delivery latency mainly as the neighbor discovery latency after introducing the quorum-based mechanism in the proposal. If we denote the latency of one node discovering another node as the cost for a link which

connecting the two nodes, then the link cost will be continuously changing.
In wireless sensor network with dynamic “link cost”, how to find an optimal shortest routing path for end-to-end data delivering is a non-trivial issue. Traditional Bellman-Ford algorithm or Dijkstra’s algorithm cannot be used directly for finding such an shortest path because in different slots, the cost of each link is varying, so that the built-up routing path in last time slot will not be valid in current time slot anymore.
We plan to model the problem formally, and try to make adaption from existing routing protocols, i.e., AODV, MintRoute, etc,. to build up a new routing protocol over quorum-based, asynchronous MAC protocol which is our previous work. The new routing protocol will be highly different from conventional works in networks where the link cost was unchanged or changed slowly. It has temporal adaption feature, and need the support from the under layer. We will design such a protocol and implement it over Telosb platform.

Besides, we are also interesting to solve the following two problems,

1. Capacity maximization;
   Although the quorum-based wakeup scheduling is energy efficient, it brings the cost of additional neighbor discovery delay which will reduce the overall system transmission capacity. The additional neighbor discovery latency do not have much affects on applications with low traffic. But for traffic-intensive applications, like data aggregation, the additional latency is not neglectful.
   In the post preliminary-exam work, we will find the factors that affect the capacities in our quorum-based scheduling mechanism, and propose some solutions to maximize the capacity, i.e., random quorum selection, practical backoff time setting, for the application with high traffic load such as data aggregation [71].
2. Broadcast/multicast support;
   Although we have proposed some solutions to support broadcast/multicast in previous work, there are still some disadvantages, like too many RTS messages are send out for triggering receiver for waking up to receive the broadcast/multicast data.
   A better solution in the future may be to extend the cqs-pair to cqs m-pair in which m cyclic quorum systems have the heterogenous rotation closure property with one another. The behind rationale can be based on Chinese Remainder Theorem.

Bibliography

[1] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (Sensys)*, 2004, pp. 95–107.

[2] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems (Sensys)*, 2006, pp. 307–320.

[3] D. G. Center and D. M. Gordon, "The prime power conjecture is true for," *Electronic J. Combinatorics*, vol. 1, p. 6, 1994.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[5] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. A. Stankovic, and T. Abdelzaher, "Achieving real-time target tracking usingwireless sensor networks," in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS06)*, 2006, pp. 37–48.

[6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA02:)*, 2002, pp. 88–97.

[7] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, and A. Wood, "Wireless sensor networks for in-home healthcare:," in *Proceeding of High Confidence Medical Devices, Software, and Systems (HCMDSS05)*, 2005, pp. 2–3.

[8] G. Platt, M. Blyde, S. Curtin, and J. Ward, "Distributed wireless sensor networks and industrial control systems - a new partnership," in *Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors (EmNets05:)*, Washington, DC, USA, 2005, pp. 157–158.

[9] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *IEEE Conference on. Computer Communications (INFOCOM)*, 2001, pp. 1548–1557.

[10] T. I. (TI), “Cc2420 data sheet,” http://focus.ti.com/lit/ds/symlink/cc2420.pdf.

[11] C. H. Y. Tseng and T. Hsieh, “Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks,” in *IEEE Conference on. Computer Communications (INFOCOM)*, 2002, pp. 200 – 209.

[12] R. Zheng, J. C. Hou, and L. Sha, “Asynchronous wakeup for ad hoc networks,” in *Proceedings of the ACM symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2003, pp. 35–45.

[13] A. Keshavarzian, H. Lee, and L. Venkatraman, “Wakeup scheduling in wireless sensor networks,” in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2006, pp. 322–333.

[14] E.-Y. Lin and J. M. Rabaey, “A comprehensive study of power-efficient rendezvous schemes for wireless sensor networks,” Ph.D. dissertation, University of California, Berkeley, 2005.

[15] C. Schurgers, S. G. V. Tsiatsis, and M. Srivastava, “Topology management for sensor networks: Exploiting latency and density,” in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.

[16] M. J. Miller and N. H. Vaidya, “Power save mechanisms for multi-hop wireless networks,” in *Proceedings of the First International Conference on Broadband Networks (BROADNETS)*, 2004, pp. 518–526.

[17] L. Gu and J. Stankovic, “Radio-triggered wake-up capability for sensor networks,” in *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2004, pp. 27–37.

[18] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, “Dw-mac: a low latency, energy efficient demand-wakeup mac protocol for wireless sensor networks,” in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc08:)*, 2008, pp. 53–62.

[19] W. Ye, J. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 12, pp. 493–506, 2004.

[20] T. Dam and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” in *The First ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2003.

[21] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, “Towards optimal sleep scheduling in sensor networks for rare-event detection,” in *Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN)*, 2005.

[22] A. El-Hoiydi and J.-D. Decotignie, "Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks," in *Proceedings of the Ninth International Symposium on Computers and Communications (ISCC)*, 2004, pp. 244–251.

[23] N. Vasanthi and S. Annadurai, "Aws: asynchronous wakeup schedule to minimize latency in wireless sensor networks," in *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*, vol. 1, June 2006, pp. 7–.

[24] C. H. J.R. Jiang, Y.C. Tseng and T. Lai, "Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks," *ACM Journal on Mobile Networks and Applications (MONET)*, 2005.

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

[26] Z.-T. Chou, "A randomized power management protocol with dynamic listen interval for wireless ad hoc networks," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 3, 2006, pp. 1251–1255.

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

[28] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys)*, 2004, pp. 214–226.

[29] A. Amis and R. Prakash, "Load-balancing clusters in wireless ad hoc networks," in *Proceedings of 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, 2000, pp. 25–32.

[30] T. Hou and T. Tsai, "A access-based clustering protocol for multihop wireless ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 7, pp. 1201–1210, Jul 2001.

[31] C. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 15, no. 7, pp. 1265–1275, Sep 1997.

[32] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 366–379, Oct.-Dec. 2004.

[33] R. Misra and C. Mandal, "Rotation of cds via connected domatic partition in ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 4, pp. 488–499, 2009.

[34] J. Ma and L. Ni, "Low-overhead dominating set based algorithms for maximizing lifetime in wireless sensor networks," in *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS2007)*, Oct. 2007, pp. 1–9.

[35] S. Lai, B. Zhang, B. Ravindran, and H. Cho, "Cqs-pair: Cyclic quorum system pair for wakeup scheduling in wireless sensor networks." in *International Conference on Principles of Distributed Systems (OPODIS)*, vol. 5401. Springer, 2008, pp. 295–310.

[36] D. R. Stinson, *Combinatorial Designs: Constructions and Analysis*. SpringerVerlag., 2003.

[37] W. Luk and T. Huang, "Two new quorum based algorithms for distributed mutual exclusion," in *International Conference on Distributed Computing Systems (ICDCS)*, 1997, pp. 100 – 106.

[38] S. Lai, B. Ravindran, and H. Cho, *submitted to IEEE Transaction on, Computers (TC)*, 2009.

[39] S. Lai, B. Zhang, B. Ravindran, and H. Cho, "A run-time configurable low power mac protocol for heterogenous sensor networks," in *submitted to ACM Mobicom 2009*, 2009.

[40] Crossbow, "Telosb datasheet," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[41] D. P. Bertsekas, "A simple and fast label correcting algorithm for shortest paths," *Networking*, vol. 23, no. 7, pp. 703–709, 1993.

[42] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[43] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA99)*, Feb 1999, pp. 90–100.

[44] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems (Sensys)*, 2003, pp. 14–27.

[45] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos, P. V. Krishnamurthy, and O. Ercetin, "A framework for distributed spatio-temporal communications in mobile ad hoc networks," in *Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, April 2006, pp. 1–13.

[46] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," in *IEEE Computer and Communications Societies (Infocom)*, vol. 1, 2003, pp. 481–491.

[47] E.-Y. Lin, J. Rabaey, and A. Wolisz, "Power efficient rendezvous schemes for dense wireless sensor networks," in *IEEE International Conference on Communications (ICC)*, vol. 7, June 2004, pp. 3769–3776.

[48] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN05).* Piscataway, NJ, USA: IEEE Press, 2005, p. 4.

[49] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *24th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom05)*, vol. 4, March 2005, pp. 2470–2481.

[50] H. Wang, X. Zhang, F. Abdesselam, and A. Khokhar, "Dps-mac: An asynchronous mac protocol for wireless sensor networks," vol. 7, June 2007, pp. 393–404.

[51] C. J. Colbourn, J. H. Dinitz, and D. R. Stinson, "Quorum systems constructed from combinatorial designs," *Information and Computation*, vol. 169, no. 2, pp. 160–173, 2001.

[52] J. L. M. Nuno Preguica, "Revisiting hierarchical quorum systems," in *Proceedings of the The 21st International Conference on Distributed Computing Systems (ICDCS01).* Washington, DC, USA: IEEE Computer Society, 2001, p. 264.

[53] M. Maekawa, "A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems," *ACM Transaction on Computer System*, vol. 3, no. 2, pp. 145–159, 1985.

[54] D. Malkhi and M. Reiter, "Byzantine quorum systems," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing (STOC97).* New York, NY, USA: ACM, 1997, pp. 569–578.

[55] D. Malkhi, M. Reiter, and R. Wright, "Probabilistic quorum systems," in *Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing (PODC97)*, 1997, pp. 267–273.

[56] M. Naor and A. Wool, "The load, capacity, and availability of quorum systems," *SIAM Journal on Computing*, vol. 27, no. 2, pp. 423–447, 1998.

[57] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM conference on Embedded network sensor systems (Sensys 08)*, 2008, pp. 71–84.

[58] S. H. Wu, C. M. Chen, and M. S. Chen, "An asymmetric quorum-based power saving protocol for clustered ad hoc networks," in *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS)*, 2007.

[59] Y. Wang, C. Wan, M. Martonosi, and L. Peh, "Transport layer approaches for improving idle energy in challenged sensor networks," in *Proceedings of the 2006 SIGCOMM workshop on Challenged networks (SIGCOMM Workshops)*, 2006, pp. 253 – 260.

[60] H. S. Z. I. Niven and H. L. Mongomery, *Introduction to the Theory of Numbers*. John Wiley & Sons, 1991.

[61] Y. Chen and Y. Lin, "C-mac: An energy-efficient mac scheme using chinese-remainder-theorem for wireless sensor networks," in *IEEE International Conference on Communications (ICC 07)*, June 2007, pp. 3576–3581.

[62] Y. Kuo and C. Chen, "Crt-mac: A power-saving multicast protocol in the asynchronous ad hoc networks," in *IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC)*, 2008, pp. 332 – 337.

[63] I. C. Society, "Ieee standard 802.15.4a-2007," August 31 2007.

[64] IEEE, "Ieee 802.15 $wpan^{TM}$ task group 4 (tg4)," http://www.ieee802.org/15/pub/TG4.html/.

[65] J. Blum, M. Ding, A. Thaeler, and X. Cheng, "Connected dominating set in sensor networks and manets," *Handbook of Combinatorial Optimization (2005)*, pp. 329 – 369, 2005.

[66] TinyOS, "Tinyos community forum," http://www.tinyos.net/.

[67] R. L. R. Thomas H. Cormen, Charles E. Leiserson and C. Stein, *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill., 2001.

[68] L. Lamport, "Time,clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, 1978.

[69] I. Demirkol, F. Alagoz, H. Delic, and C. Ersoy, "Wireless sensor networks for intrusion detection: packet traffic modeling," *Communications Letters, IEEE*, vol. 10, no. 1, pp. 22–24, Jan 2006.

[70] I. Demirkol, C. Ersoy, and F. Alagoz, "Mac protocols for wireless sensor networks: A survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.

[71] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCSW 02)*, 2002, pp. 575–578.