Asynchronous Wakeup for Ad Hoc Networks*

Rong Zheng, Jennifer C. Hou and Lui Sha
Department of Compute Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
E-mail: {*zheng4,jhou,lrs*}*@cs.uiuc.edu*

ABSTRACT

Due to the slow advancement of battery technology, power management in wireless networks remains to be a critical issue. Asynchronous wakeup has the merits of not requiring global clock synchronization and being resilient to network dynamics. This paper presents a systematic approach to designing and implementing asynchronous wakeup mechanisms in ad hoc networks. The optimal wakeup schedule design can be formulated as a block design problem in combinatorics. We propose a neighbor discovery and schedule bookkeeping protocol that can operate on the optimal wakeup schedule derived. Two power management policies, i.e. slot-based power management and on-demand power management, are studied to overlay desirable communication schedule over the wakeup schedule mandated by the asynchronous wakeup mechanism. Simulation studies indicate that the proposed asynchronous wakeup protocol is quite effective under various traffic characteristics and loads: energy saving can be as high as 70%, while the packet delivery ratio is comparable to that without power management.

Categories and Subject Descriptors

C.2.1 [**Computer-communication Networks**]: Network architecture and design—*distributed networks, wireless communication*

General Terms

Algorithms, theory

Keywords

Asynchronous wakeup, power management, block design and ad hoc networks

1. INTRODUCTION

With the proliferation of portable computing platforms and small wireless devices, ad hoc wireless networks have received more and more attention as a means of data communication among devices regardless of their physical locations. As wireless devices usually rely on portable power sources such as batteries to provide the necessary power, power management in wireless networks has become a crucial issue. It has been observed that energy is not always consumed by active communication in ad hoc networks [6]. Experimental results have shown that the energy consumed by wireless devices in the idle state is only slightly less than that in the transmitting or receiving states. As a result, an important technique to reduce power consumption in ad hoc networks is to place nodes in the low-power sleep mode whenever possible.

Most wireless devices, including IEEE 802.11 compatible NICs and bluetooth devices, can operate at different power consumption modes. In the low-power state (or sleep state), part of the radio circuitry (for example, the equalizer) is shut down to save power. Regardless of the power management policies used to determine which set of nodes should be set to sleep [4, 24, 26] and when, it is necessary to have MAC or physical layer support, called the *wakeup mechanism*, for wakening sleeping nodes in the presence of pending transmission. The major objective of the wakeup mechanism is to maintain network connectivity while reducing the idle state energy consumption.

Existing wakeup mechanisms fall into three categories: on-demand wakeup, scheduled rendezvous, and asynchronous wakeup. In on-demand wakeup mechanisms, out-band signaling is used to wake up sleeping nodes in an on-demand manner. For example, with the use of radio frequency identifier (RFID) [13], a high-power base node can awaken lower-power RFID tag. As RFID radios can operate at power consumption of roughly three orders of magnitude lower than typical commercial radios operating in the Mbps range, it has been proposed in [13] to apply RFID to embedded systems. Although the RFID technology is relatively mature and of low cost and complexity, it is highly asymmetric and sensitive to the interference over a large range of frequencies in uncontrolled environments. This makes the RFID technology an ill-fit for multi-hop ad hoc environments. In [17], the authors proposed a wake-on-wireless technique that uses a separate control channel with lower-power radio operating at a frequency band that is different from the one used for the data channel. The main concern of this type of approaches is that the transmission range of radios op-

---

*This research is supported in part by NSF under grant number ANIR-0221357 and MURI/AFOSR under contract number F49620-00-1-0330

MobiHoc'03, June 1–3, 2003, Annapolis, Maryland, USA.

erating at different frequency bands or using different modulation schemes are usually different. For example, in [17] the low-power radio operates at 915MHz ISM band with a transmission range of about 332 ft in free space and 30 ft indoor while the IEEE 802.11 cards operates at 2.4 GHz with transmission range upto 1750 ft. Therefore, (static or dynamic) power control is required to ensure the consistency among two channels.

In scheduled rendezvous wakeup mechanisms, low-power sleeping nodes wake up at the same time periodically to communicate with one another. This is the mechanism used in IEEE 802.11 power saving mode (PSM). As all the participating nodes have to synchronize their clocks (so that a common wakeup time period can be scheduled), this mechanism is most appropriate for single-hop networks in which all the nodes can hear one another. The scheduled rendezvous wakeup mechanism may not be well-suited in multi-hop ad hoc networks, as distributed clock synchronization is a nontrivial issue [5]. It does not work well in the presence of network dynamics. For example, two partitioned sub-networks may have non-overlapping rendezvous points. As a result, even when they are geographically adjacent to each other, the time for neighbor discovery is unbounded.

Asynchronous wakeup is first studied in [20]. The authors investigate several wakeup schedules and suggest necessary modification to IEEE 802.11 PSM to support asynchronous wakeup. As compared to the scheduled rendezvous wakeup mechanisms, asynchronous wakeup does not require clock synchronization. Each node follows its own wakeup schedule in idle states, 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 mechanisms. On the flip side, asynchronous wakeup is easier to implement and can ensure network connectivity even in highly dynamic networks. In other words, asynchronous wakeup trades energy consumption for the robustness of network connectivity. A key challenge is to derive schedules that have minimum idle state energy consumption with bounded neighbor discovery latency.

In this paper, we take a systematic approach to designing asynchronous wakeup mechanisms in ad hoc networks. We intend to address the following fundamental questions related to asynchronous wakeup,

1. Given a desirable delay for neighbor discovery, what is the minimum percentage of time a node has to be awake?
2. Does there exist an optimal schedule that can achieve such minimum value?
3. How to design a wakeup protocol using the optimal schedule?
4. How can power management be performed with asynchronous wakeup?

To answer the former two questions, we first formulate the problem of generating wakeup schedules as a block design problem in combinatorics. We then derive the theoretical limit of the wakeup schedule and give an optimal solution that can achieve minimum idle state energy consumption with bounded neighbor discovery latency. To answer the latter two questions, we study, after the theoretical base is laid, two protocol design issues: (i) efficient implementation of the wakeup schedule and (ii) power management using asynchronous wakeup. In the former issue, we devise a neighbor discovery and neighbor schedule bookkeeping protocol that can operate without requiring slot boundaries be aligned. The protocol is also resilient to packet collision and network dynamics. In the latter issue, we consider two design choices: slot-based power management and on-demand power management, to determine how a node transitions among different power management modes. In slot-based power management, power management modes are managed slot by slot based on the number of buffered packets for a particular neighbor, while in on-demand power management the transition between power management states are triggered by the presence/lack of certain communication events. In both cases, a desirable communication schedule can be overlaid over the wakeup schedule.

To verify the effectiveness of the design, we implement our proposed schemes using IEEE 802.11 MAC (without the power management component) in *ns-2*. Simulation studies indicate that our wakeup schedule design guarantees that any two neighboring nodes can detect each other in finite time without global clock synchronization. In conjunction with the power management policies, the proposed wakeup protocol can achieve communication efficiency comparable to the case without power management while saving energy up to 70%.

The rest of this paper is organized as follows. In Section 2, we formally define the problem of generating wakeup schedules as a block design problem and derive the theoretical results. In Section 3 we devise a wakeup protocol to discover neighbors in bounded time based on the optimal wakeup schedule. In Section 4, we propose two power management schemes operating on top of the wakeup protocol and provide a qualitative comparison. In Section 5, we evaluate both the proposed asynchronous wakeup protocol and power management schemes. Finally, we provide a brief review of related energy efficient protocols in the literature in Section 6, and conclude the paper in Section 7 with a list of future research tasks.

2. DESIGN OF ASYNCHRONOUS WAKEUP SCHEDULE

2.1 Problem statement

We represent a multi-hop ad hoc 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, therefore if $(v_i, v_j) \in E$, then $(v_j, v_i) \in E$. The neighboring set of a node v is denoted by $N(v)$. The major objective of asynchronous wakeup mechanisms is *to maintain network connectivity regardless of the power states nodes may be in.* 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.

In the absence of data transmission (i.e., when a node is in the idle state), a wakeup mechanism associates each node with a slot schedule of length T, termed as the *wakeup schedule function* (WSF). The WSF of a node v is represented by a polynomial of order $T - 1$ as $f_v(x) = \sum_{i=0}^{T-1} a_i x^i$, where T is the length of the schedule, $a_i = 0$ or 1, $\forall i \in [0, T - 1]$, and x is a place holder. If $a_i = 1$, the node should wake up in slot i. For two neighboring nodes to communicate, their wakeup schedules have to overlap regardless of the difference of their clocks. For ease of presentation, we assume for now the slot boundaries are aligned (i.e., two neighbors' schedules shift only by multiple of slots). We will relax this assumption in Section 3.

By definition, $k_v = f_v(1)$ is the total number of slots in which node v is scheduled to be awake every T slots. If the schedules of two nodes u and v overlap, the amount of time it takes for node u to reach node v is bounded by T, i.e., T is the worse-case delay to discover a neighbor due to power saving. Given a fixed value of T, we aim to make k_v/T as small as possible, subject to the constraint that the schedules of any two neighboring nodes overlap by m slots. The rationale behind seeking a small value of k_v is to enable nodes to stay in the low-power mode as much as possible.

Given the wakeup schedule $f_u(x)$ for node u, it is easy to show that $f_u^k(x) = f_u(x) \cdot x^k(\mod (x^T - 1))$ represents the cyclic shift of the original schedule by k slots. Let $\wedge$ represent the *and* operation between two polynomials, $|\cdot|$ the number of non-zero items of a polynomial. We have the following definition,

DEFINITION 1. *The degree of overlapping between two WSFs $f_u(x)$ and $f_v(x)$, denoted as $C(u, v)$, is defined to be the minimum number of common items of $f_u(x) \cdot x^l(\mod (x^T - 1))$ and $f_v(x) \cdot x^k(\mod (x^T - 1))$ for any integer $l, k \in [0, T-1]$, i.e., $C(u, v) = \min_{l,k\in[0,T-1]} |f_u^l(x) \wedge f_v^k(x)|$.*

The above definition mandates that the number of overlapping slots should be shift-invariant. Therefore, the problem of designing optimal WSFs can be stated as follows:

PROBLEM 1. *(Optimal WSF design problem) Given a fixed value of T,*

$$\begin{aligned} \min \quad & \overline{k}, \\ s.t. \quad & C(u, v) \geq m \quad \forall u \in V \text{ and } \forall v \in N(u), \end{aligned}$$

where $\overline{k}$ is the assemble average of the number of active slots in every T slots.

2.2 Theoretical bound for WSF design problem

One important question is how small the value of $\overline{k}$ can be in the optimal WSF design problem. In the following theorem we show the constraints that k_u, $\forall u \in V$ must satisfy.

THEOREM 1. *(Bound for the WSF design) Consider any two neighboring nodes u and v with WSFs of length T, $f_u(x)$ and $f_v(x)$. Given T and m, the necessary condition for $C(u, v) \geq m$ is $k_v \cdot k_u \geq m \cdot T$, where $k_u = f_u(1)$ and $k_v = f_v(1)$ are, respectively, the number of slots in which node u and node v are scheduled to be awake every T slots.*

PROOF. For any two WSF's $f_u(x)$ and $f_v(x)$, without loss of generality we fix $f_u(x)$ and cyclically shift $f_v(x)$ by ℓ, $\ell = 0, 1, ..., T-1$. For each non-zero a_i in $f_v(x) \cdot x^\ell (\mathrm{mod}(x^T - 1))$, it intersects with k slots in $f_u(x)$ as ℓ goes from 0 to $T - 1$ and any two slots in $f_v(x)$ cannot intersect with the same slot in $f_u(x)$ for the same ℓ. Therefore, the total number of intersections for all slots associated with non-zero a_i's in $f_v(x)$ is $k_u \cdot k_v$. On the other hand, since we require that $C(u, v) \geq m$ for any shift of $f_v(x)$, the maximum number of intersected slots (as ℓ goes from 0 to $T - 1$) is at least $m \cdot T$. By counting argument, $m \cdot T \leq k^2$ or equivalently, $k \geq \sqrt{m \cdot T}$. ☐

It follows immediately from Theorem 1 that when $k_v = k$, $\forall v \in V$, i.e., the duty cycles of all nodes are the same (called symmetric design), we have the following corollary:

COROLLARY 1. *(Bound for symmetric design) Given T and m and $k_v = k$, $\forall v \in V$, the feasible set of Problem 1 satisfies $k \geq \sqrt{m \cdot T}$.*

Corollary 1 gives the lower bound of the minimum duty cycle in the symmetric WSF design problem. Note that the result is similar to that for ensuring mutual exclusion in distributed systems [19]. In [11], it is shown that the problem of arbitrating mutual exclusion requests is equivalent to picking sets with pairwise non-null intersection. The condition for our optimal WSF design problem is stronger, since in addition to non-null intersection, the intersection must satisfy shift-invariant property.

2.3 Optimal Design and Considerations

In this section, we prove that the lower bound on the asynchronous wakeup is achievable using a constructive method. In particular, we investigate symmetric and asymmetric WSF design separately. (Recall that in the symmetric design all nodes have the same duty cycles, whereas in asymmetric design the duty cycles of nodes may be different.)

Symmetric WSF design. The problem of symmetric WSF design can be mapped to the symmetric block design problem in combinatorics.

DEFINITION 2. *(v, b, r, k, λ)-design is a family of b blocks of size k from a set V of size v, such that each element of V appears in r blocks and every two elements of V appear in λ common blocks. In particular, if $b = v$ (or equivalently $r = \lambda$), a (v, b, r, k, λ)-design is called a (v, k, λ)-design or a symmetric-design.*

It can be easily shown that under symmetric design every two blocks have λ common elements. As an example, consider the following design (124, 235, 346, 457, 561, 672, 713), where the number gives the positions of active slots. As shown in Figure 1, this is a (7, 3, 1)-design, i.e., there are 7 blocks of size 3 and any two blocks have one common elements. In the context of the WSF design problem, $T = b$, $m = \lambda$, i.e. the length of each schedule (block) is T and the number of overlapping (common elements) of any two schedules (blocks), is m. Instead of exploring all possible symmetric designs, we consider a subset of symmetric designs in which blocks are the cyclic translates of a single block.

DEFINITION 3. *A set $D = \{a_1, a_2, ..., a_k\}$ is a (T, k, m)-difference set, if for every $d \in \{0, 1, 2, 3, ..., T-1\}$, there are exactly m ordered pairs (a_i, a_j) such that $d = a_j - a_i \mod T$.*

It can be easily proved that, the definition of the difference set implies that $m(T - 1) = k(k - 1)$. Therefore, if there exists a (T, k, m)-difference set, the bound in Theorem 1 can be achieved asymptotically as T goes large. The multiplier Theorem given in [2] can be used to design a

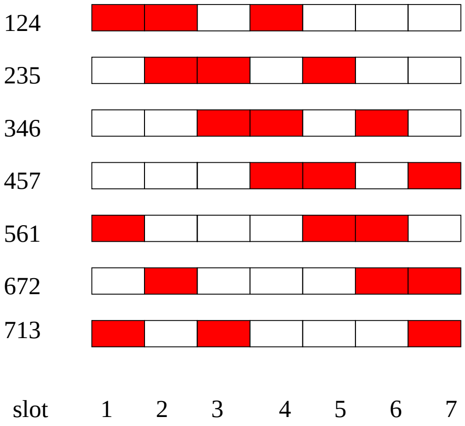


Figure 1: The slot assignment under the $(7, 3, 1)$-design. Each solid rectangle corresponds to an "on" slot and each hollow rectangle an "off" slot. The rectangles are numbered by their positions. Note that (i) each schedule repeats every 7 slots and has 3 active slots; and (ii) any two schedules overlaps for at least 1 slot.

(T, k, m)-difference set, although it does not guarantee the existence of such a set. Given T, k and m, we apply the multiplier theorem and enumerate among all the possible multipliers to find a valid design or prove the non-existence of such a design. Under the special case of $m = 1$, it has been proved that if k is a power of a prime, there exists a $(k^2+k+1, k+1, 1)$-design. The interested reader is referred to [2] for a detailed account of the multiplier theorem. Since all the schedules in the family of different sets are cyclic shift of one another, we can use a single WSF to represent them. An example of a feasible design is the $(7, 3, 1)$-design (Figure 1) with the corresponding WSF of $f(x) = 1+x+x^3$, T = 7. Another feasible design of longer blocks is the $(73, 9, 1)$-design in which every frame has 73 slots and 9 of them are "on" slots. The design can be represented by the WSF of $f(x) = 1+x+x^3+x^7+x^{15}+x^{31}+x^{36}+x^{54}+x^{63}, T = 73$.

Asymmetric design. In the case of asymmetric design, the duty cycles of nodes can be different. Theorem 1 gives a necessary condition for ensuring connectivity among neighboring nodes. In the case $m = 1$, this necessary condition can be trivially satisfied by assigning $k_u = T$ and $k_v = 1$ for $v \in N(u)$.

The asymmetric WSF design problem is related to the vertex covering problem. To illustrate this, consider a simplified problem in which only two different duty cycles, K and k, are used, where $K > k$ and $K \cdot k \geq m \cdot T$. Nodes with a duty cycle of K are colored black and the others are colored white. By Theorem 1, to satisfy the necessary condition, every edge in $G(V, E)$ should at least have one black node. In essence the set of black-colored nodes form a vertex cover set for $G(V, E)$. Let the ratio of black and white nodes be r. It is easy to construct graphs with $r > 1$, for example, odd cycles. For these types of graphs, asymmetric designs are less energy-efficient compared to symmetric designs. For graphs with small $r < 1$, the average duty cycle is given by

$$\overline{k} = \frac{rK + k}{r + 1} \geq \frac{2\sqrt{rKk}}{r + 1} \geq \frac{2\sqrt{rmT}}{r + 1}, \tag{1}$$

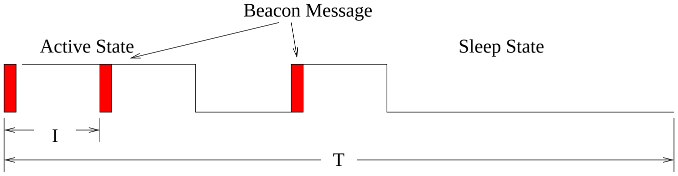


Figure 2: The frame structure used in asynchronous wakeup mechanisms.

where the last inequality results from Theorem 1. Equality holds when $K = \sqrt{\frac{mT}{r}}$ and $k = \sqrt{rmT}$. Since $r \leq 1$, we have $K \geq k$.

Finding the minimum vertex cover is an NP-complete problem. Its dependence on the topology contradicts one of the most important requirements of generating asynchronous wakeup schedules, i.e., adaptiveness to network dynamics. Therefore, in what follows, we only consider symmetric wakeup schedules. Asymmetric design is applicable in heterogeneous networks, where there exists "powerful" nodes with abundant power supply. These nodes can be assigned WSFs with high duty cycle and serve as relay nodes for power-constrained nodes that use WSFs with low duty cycles.

3. IMPLEMENTATION OF ASYNCHRONOUS WAKEUP SCHEDULES

A fundamental assumption in the above analysis is that slots are aligned to their boundaries. Although the assumption facilitates theoretical derivation, it may not be realistic, as slot alignment is as hard a problem as clock synchronization. Since nodes may initiate communication at different times, the discrepancy in schedules can be of the following two forms: (i) the slot boundaries are not aligned and (ii) even the slot boundaries are aligned, the schedules usually have phase shift relative to each other. The result of optimal symmetric block design (Section 2.3) ensures that there exist sufficient, overlapping active slots even if two schedules are phase shifts of each other. In this section, we propose, based on the optimal symmetric block design, an asynchronous wakeup protocol that operates without aligned slot boundary.

The protocol consists of two component procedures: (i) neighbor discovery that detects neighboring nodes in the power saving states and (ii) bookkeeping of neighbor schedules that is used to keep track of a neighbor's wakeup schedule to facilitate data communication.

3.1 Neighbor Discovery

Protocol description. The neighbor discovery procedure operates as follows. Each node divides its time axis into fixed-length frames of T slots. Each slot is in turn of length I. Every node chooses the same WSF to schedule its own active and sleeping slots, where the WSF is derived from the optimal block design in Section 2. Due to the randomness in the starting time and the clock shift, schedules are not synchronized. In an active slot, a node can both transmit/receive packets or listen to the channel, while in a sleeping slot, a node operates in the low-power mode. At the beginning of an active slot, a node transmits a beacon

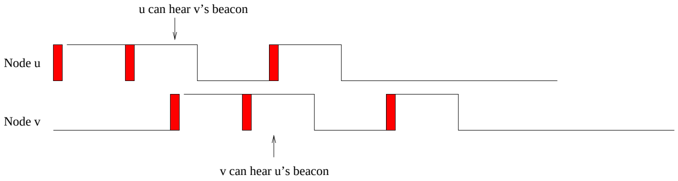


Figure 3: An example that shows that two neighboring nodes can hear each others beacon in the presence of clock shifts.

message piggybacking its own id and other information subject to the channel contention/resolution rule. The frame structure is illustrated in Figure 2.

Theoretical base. The following theorem establishes the correctness of the protocol without aligned slot boundary:

Theorem 2. *If the WSF schedule ensures a minimum of one overlapping slot, then with the use of the neighbor discovery protocol aforementioned, two neighboring nodes can hear each others beacons with probability one.*

Proof. We consider two cases:

- Case 1: Perfect alignment of slot boundaries. If the slot boundaries of any two nodes are aligned, with the use of the result of optimal symmetric block design, two nodes can hear each other' beacon messages.[1]
- Case 2: No alignment of slot boundaries. Suppose the clocks of nodes u and v differ by $l \cdot T + n \cdot I + \delta$, where $\delta \leq I$. If $\delta = 0$ (i.e., the difference is $l \cdot T + n \cdot I$), by the property of difference sets, there exists an pair (i, j) such that the i-th active slot of node u and the j-th active slot of node v overlaps (in other words $(i - j) \bmod T = I$). Similarly, if $\delta = I$ (i.e., the difference is $l \cdot T + (n+1) \cdot I$), there exists an pair (i', j') such that i'-th active slot of node u and the j'-th active slot of node v overlaps. Note that i and j cannot be the same as i' and j' simultaneously. Without loss of generality, assume $j \neq j'$. When the difference is $l \cdot T + n \cdot I + \delta$, node u can hear node v's beacon in slot i and node v can hear node u's beacon in slot j'. An example using (7,3,1)-design is given in Figure 3, in which node u and v schedule shift by $2 \cdot I + \delta$. Node u can hear v's beacon at slot 2 and node v can hear node u's beacon at slot 2.

Note that due to contention, a node may not be able to transmit its beacon message in an active slot that overlaps with those of other nodes. However, the probability of not being able to hear a neighbor's beacon message in nT slots diminishes geometrically as n gets large. ☐

The above theorem ensures that two neighboring nodes can always discover each other in bounded time if all beacon messages are transmitted successfully. In the presence of channel contention and collision, neighbor discovery can be done with probability one. This property holds true even in the case that two originally disconnected subsets of nodes join together. As long as they use the same WSF, they can form a larger connected network in finite time.

Implementation. To implement the protocol, each node keeps a neighbor list in which each entry has the following fields:

node id	clock	schedule	last_fresh	lifespan

We will discuss the use of various fields below. Whenever a new neighbor is discovered, a new entry is added. Each entry of the list is associated with a timer (called refresh timer), that is created and refreshed when any ongoing transmission or beacon message is overheard from the corresponding neighbor. When the refresh timer of an entry expires, the corresponding entry is removed and the corresponding neighbor is considered either dead or moved away.

3.2 Bookkeeping of Neighbor Schedules

If node u hears node v's beacon message, it knows that node v will stay awake for a duration of length I. When node u has buffered data for node v, it can start transmission after node v's beacon. However, since beacon messages use broadcast, they are subject to collision. Hence, it is inefficient to merely use beacons to solicit/announce transmission of buffered packets. In mobile scenarios with relatively low mobility, it is beneficial for a node to keep track of neighbor schedules and clock shifts in order to infer its neighbors' wakeup schedules.

We augment the beacon message with a time stamp that records the the clock and the current schedule of a sender node. Upon receiving a beacon message, a node keeps in the corresponding neighbor list entry (i) the relative clock in the clock field; and (ii) the schedule differences between the node and its neighbor in the schedule field. When a node is awake (either mandated by its own wakeup schedule or other higher-level power management schemes), it checks its neighbors' schedules and transmits pending packets (if any) to neighbors who are awake. When packets destined to a neighbor who is in the sleeping state arrive, they are buffered for later transmission.

The processing and propagation delay on the interface card and the wireless medium may result in inaccuracy in estimating neighbors' schedules. To mitigate the negative impact of such inaccuracy, one may introduce a small amount of slack time at the beginning and end of a neighbor's estimated active slot and disable packet transmission in the slack time.

Note that unlike IEEE 802.11 and the three schemes proposed in [20], we do not use the notion of ATIM windows. In IEEE 802.11, a node can optionally enters the sleep mode if it receives no ATIM frame in an ATIM window. In our scheme, in an active slot, a node has to transmit its own beacon and listen to beacons from other nodes for which it may have buffered packets. To ensure the correctness of the protocol, a node has to remain awake throughout the entire active slot.

4. POWER MANAGEMENT USING ASYNCHRONOUS WAKEUP

In Sections 2-3, we derive the optimal asynchronous wakeup schedule and propose a wakeup protocol that operates with-

---

[1] We assume that transmission of beacons is deferred by a random backoff period when the channel is idle and the beacon transmission time is at least one order of magnitude smaller than I.

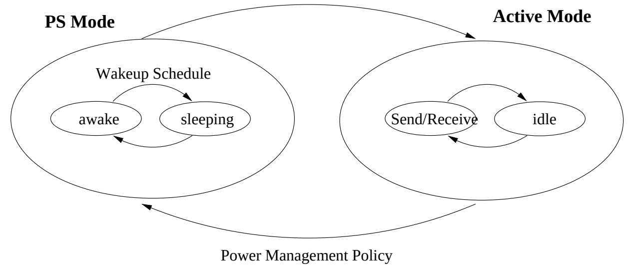


Figure 4: Relationship between the wakeup schedule and the communication schedule devised by a power management policy.

out requiring alignment of slot boundaries. A wakeup schedule is needed to maintain network connectivity when nodes are in power saving mode, and a power management policy is needed to further facilitate effective communication while saving as much as energy as possible. If nodes can only communicate during the active slots of the wakeup schedule, the capacity of the network will be greatly reduced and the delay experienced by packets may be prohibitively long, i.e., in the worst case nT for communication over an n-hop path. A power management policy overlays a desirable communication schedule over the wakeup schedule to decide when a node should go to sleep and wake up. The relationship between the wakeup schedule and the communication schedule devised by a power management policy is illustrated in Figure 4. In this section, we present two design choices of power management that can be laid on top of the proposed asynchronous wakeup protocol.

4.1 Slot-based power management

We propose in the MAC layer a signaling and reservation mechanism, with the objective of adapting the power management state to the actual traffic load in the network.

In the signaling protocol, if the number of buffered packets for an intended receiver exceeds a threshold L, the sender signals the receiver to remain on for the next slot. (In our implementation, L is set to the total number of packet transmissions possible in an active slot divided by the number of neighbors.) A node requested to stay awake sends an acknowledgment to the sender, indicating its willingness to remain awake in the next slot. The lifespan field in the neighbor list entry keeps the expected wakeup duration of a neighbor, and is set when the corresponding neighbor acknowledges the reservation request. If a node is not requested to stay awake by any of its neighbors, it follows its own wakeup schedule. The request is renewed on a slot-by-slot basis, i.e., the reservation only span one slot. If a neighbor is awake according to the schedule kept track of in the wakeup protocol, or it has previously acknowledged its willingness to stay awake in the current slot, then a node may deliver packets to it.

To implement this signaling mechanism, we may employ an IEEE 802.11 like medium access and contention resolution scheme. A signaling/reservation flag is piggybacked in data/ack messages and RTS/CTS messages using the has-more-data field in the MAC header.[2] No out-band signaling is required. The control granularity of the has-more-data field is slot-by-slot in our scheme. As compared to the ATIM announcement in IEEE 802.11, we do not reserve dedicated time slots for sending reservation request and acknowledgment. As the reservation information is piggybacked in data/ack/RTS/CTS messages, the mechanism is more resilient to packet losses. As will be shown in the simulation study, this leads to significant performance improvement at little cost.

4.2 On-demand power management

In [26], the authors propose an on-demand power management framework for ad hoc networks. In this framework, power management decisions are driven by data transmission in the network, and connectivity is only maintained between pairs of senders and receivers along the routes of data communication. This reduces energy consumption while maintaining effective communication. Specifically, in the on-demand power management, transitions from the power saving mode to the active mode are triggered by communication events in the network. Transitions from active mode to power saving mode is determined by a soft state timer, called the *keep-alive timer*. Initially, all the nodes are in power saving mode. Upon receipt of a packet, a node sets the keep-alive timer and switches to the active mode. Timer values depend on the type of packets received. Upon expiration of the keep-alive timer, a node switches from the active mode to the power saving mode. The keep-alive timer is maintained at a per-node basis and is aggregatible over multiple communication events. The interested reader is referred to [26] for a detailed account of the on-demand power management protocol.

The on-demand power management policy can be directly laid over the asynchronous wakeup schedule. A node piggybacks its power management information in the beacon messages. If the power management policy mandates a neighbor to remain awake, a node can communication with it directly. When a neighbor is asleep, the wakeup protocol can be used to trigger the setup of the *keep-alive timer*.

4.3 Discussion

As compared to on-demand power management, the slot-based power management scheme operates at a finer time granularity. It establishes per-link state between a pair of neighbors using in-band signaling. On-demand power management, on the other hand, establishes per-node state whose transitions are triggered by active communication. Note that nodal states are more resilient to mobility and other network dynamics as compared to link states. However, the pitfall of the latter scheme is it may keep a node awake longer than necessary due to the fixed length of the keep-alive timers.

5. PERFORMANCE EVALUATION

To validate and evaluate the proposed design, we have implemented the proposed asynchronous wakeup mechanisms and the corresponding power management protocols in *ns-2* [21] with the CMU wireless extension, and conducted a simulation study using different traffic models in the multihop wireless networks. As a baseline, we also evaluate the performance in the absence of power management. The

---

[2] The has-more-data field is introduced in IEEE 802.11 MAC for power management in single-hop networks with APs. If this bit is set on, the receiver will remain awake.

Table 1: The power consumption model.

Transmit	Receive	Idle	Sleep
1400mW	1000 mW	830mW	130mW

performance metrics of interest are (i) the amount of power consumed, (ii) the packet delivery ratio, and (iii) the capability of adapting to network dynamics.

In the simulation study, all nodes communicate using half-duplex wireless radios that conform to 802.11-based WaveLAN wireless radios with a bandwidth of 2Mbps and a nominal transmission radius of 250m. To isolate the effect of routing overhead, we implement a greedy geographical routing protocol that forwards packet to a neighbor that is closest to the destination at each hop. We assume that the location information can be obtained via GPS or other location services, and the overhead of obtaining the location information is not simulated. Geographical location of a neighbor is piggy-backed in beacon messages.

We use the same energy model as in [4] (Table 1). The energy consumption for switching between awake and sleeping states is negligible and thus not considered. Two different schedules are used: $(7, 3, 1)$-design and $(73, 9, 1)$-design (Section 2). To ensure that frame lengths are approximately the same in the two designs for a fair comparison, we set the slot size I to be 0.1s and the frame length T to be 0.7s in the $(7, 3, 1)$-design, and the slot size I to be 0.01s and the frame length T to be 0.73s in the $(73, 9, 1)$-design. Note that the rule-of-thumb used to set the frame length is as follows: the frame length is set base on the desirable one-hop delay and the slot length is set to be the frame length divided by the number of slots in each frame. In addition, in both design, the slot size is an order of magnitude larger than the transmission and propagation delay to ensure the inaccuracy incurred in calculating neighbors' schedules is negligible.

Note that in the (T, k, m)-design, $T = 1$, $k = 1$, $m = 1$ (i.e., all intervals are active) corresponds to a network without power management. Therefore, in what follows, we use $T = 1$, $T = 7$ and $T = 73$, respectively, to denote an "always-on" network, a power-managed network with the wakeup schedule $(7, 3, 1)$, and one with the wakeup schedule $(73, 9, 1)$. Other simulation parameters are summarized in Table 2. All the results presented are averages of ten simulation runs.

Table 2: Parameters used in the simulations

Packet size	1024 bytes
Keep-alive timer	5s
HELLO interval	1s
Refresh timeout	5s

5.1 Study of Neighbor Discovery

In this section, we evaluate the neighbor discovery component of the asynchronous wakeup protocol and compare it against the scheduled rendezvous wakeup approach. The performance metric used is the neighbor discovery time defined as the average time to discover a new neighbor. The simulation is conducted in a static network with 50 nodes randomly placed in a two-dimensional 1500x300 region. All nodes are in the power saving modes following their wakeup schedules. Figure 5 depicts the average neighbor discovery time versus the schedule length T (in seconds). As expected, as T gets larger, both the (73, 7, 1) and (7, 3, 1)-designs incur longer neighbor discovery time. This result is consistent with Theorem 2 in Section 3. The reasons that there is slight difference between average neighbor discovery time for different schedules in the simulation run are, i) the probabilistic distribution of neighbor discovery time is dependent on the individual block design although the worse case bound under no beacon collision is only decided by the length of the schedule and ii) beacons may be lost due to collision.

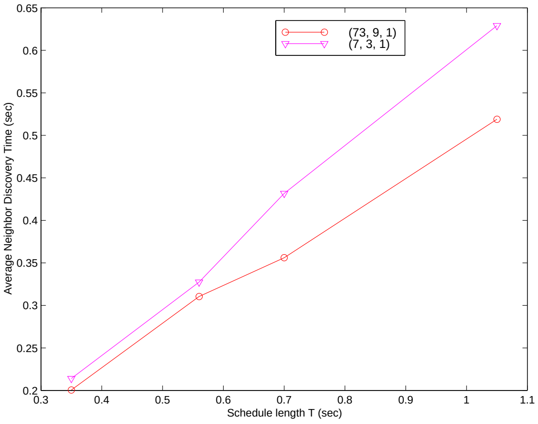


Figure 5: Impact of schedule length on the neighbor discovery time for the asynchronous wakeup mechanisms with the (73, 9, 1)-design and the (7, 3, 1)-design, respectively.

Figure 6 depicts the number of links discovered over time. Both the (73, 7, 1) and (7, 3, 1) designs use a schedule of total length 0.7s. The scheduled rendezvous wakeup mechanism is a hypothetical one. It is similar to IEEE802.11 PSM [1] with the beacon interval 0.7s and the ATIM window size 0.035s,[3] with the exception that the wakeup times are not synchronized. There are altogether 604 unidirectional links among 50 nodes. As shown in Figure 6, with the use of asynchronous wakeup mechanisms, all the links are detected in a small amount of time. In contrast, with the use of the scheduled rendezvous wakeup mechanism, only a subset of links can be detected among nodes that have overlapping wakeup intervals, due to the lack of time synchronization.

5.2 Static Networks with On-off Traffic Sources

In this set of experiments, we study the performance of asynchronous wakeup mechanisms under on-off traffic sources. As compared to long-lived traffic sources, on-off traffic is more disruptive to power management due to the lack of persistent loads in the network. The simulation is conducted in a 1500x300 static network with 50 nodes. Altogether there are 30 connections between randomly selected sender and destination pairs. Each sender is an on-off CBR traffic source with interleaved "on" and "off" periods of length 10s and 50s, respectively. The simulation lasts for 900s, and the sending rate of each source is varied from low to medium load.

---

[3] In IEEE802.11 PSM, a node stays awake during the entire ATIM window in the power saving mode.

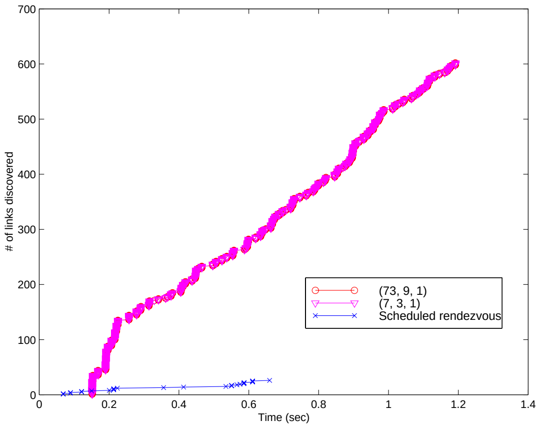


Figure 6: Number of links detected over time for the scheduled rendezvous wakeup mechanism and the asynchronous wakeup mechanisms with the (73, 9, 1)-design and the (7, 3, 1)-design. Note that the curves for the asynchronous wakeup mechanisms are very close to each other.

Figure 7 depicts the packet delivery ratio and the energy consumed for the asynchronous wakeup mechanism equipped with on-demand power management, slot-based power management with different WSF functions, and without power management. In the case of no power management, neighbors are discovered, similar to ADOV [14], using local HELLO messages. The average exchange interval of HELLO messages is set to once per second. Also shown in Fig. 7 are the levels of confidence of the simulation results.

We observe significant energy saving under the proposed asynchronous wakeup protocol, while the packet delivery ratio is comparable to that in the case without power management. The packet delivery ratio under on-demand power management is higher than that under slot-based power management (note, however, the smallest packet delivery ratio is at least 94%). This is due to the fact that on-demand power management maintains per-node states as opposed to per-link states as in slot-based power management. Packets destined for a neighbor that is known to be awake can be delivered. However, energy consumption under on-demand power management is higher due to the use of fixed-length keep-alive timers. Slot-based power management, on the other hand, manages the power at a finer time granularity.

Use of different wakeup schedules affects energy consumption and packet delivery ratio under both power management schemes. A shorter schedule saves less energy but also incurs less packet losses because the duty cycle of a node is higher. One counter-intuitive result is that with power management the packet loss rate drops slightly when the traffic load increases. This is because even if each source transmits as the highest rate of 45kbps, the network is still under utilized (as compared to the 2Mbps raw bandwidth) and the decrease in the packet loss rate may be attributed to the fact that at higher data rates, power management enables nodes to stay active longer and thus new arrivals do not have to wake up neighbors. According to the simulation results, on-demand power management in conjunction with the (73, 9, 1) wakeup schedule can achieve a good balance between energy consumption and packet delivery ratio.

5.3 Mobile Networks

In this set of experiments, we simulate a mobile network with 10 long-lived CBR connections transmitting at 1kBps in a 50-node 1500mx300m network. All nodes move at 20m/s with pause times varying from 15s to 75s.

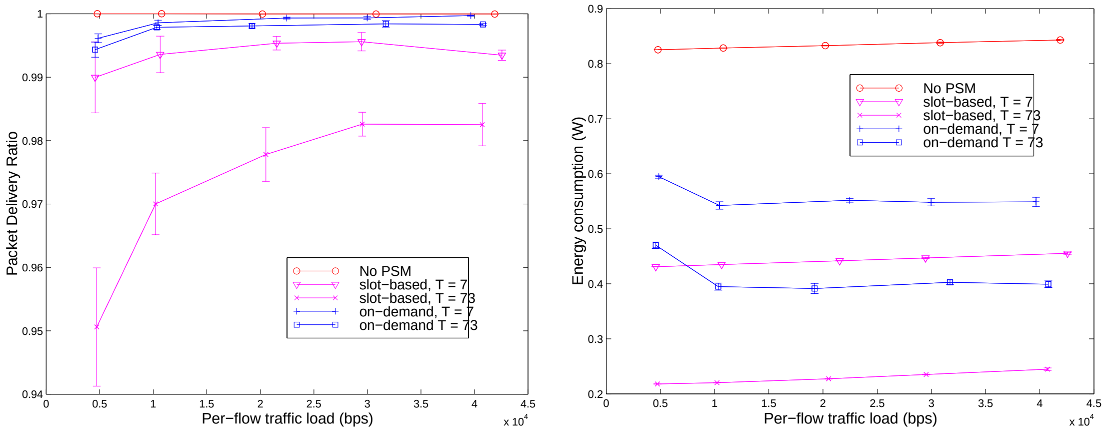
To demonstrate the effectiveness of neighbor discovery, Figure 8 depicts the energy throughput and the packet loss rate under different protocols. Due to power saving, it takes some time to discover a new neighbor which is closest to the destination. The network can be temporarily partitioned when some neighbors move out of the range and new neighbors are not yet discovered. Similar to the static scenarios, the level of power saving as a result of the asynchronous wakeup mechanism is quite significant. However, the packet delivery ratio under slot-based power management deteriorates as the mobility becomes high. The greedy geographical forwarding protocol we have implemented does not perform ring zero search and does not salvage packets destined for a neighbor that has moved away. However, we believe this impacts all the protocols in a similar fashion. One interesting result is that the loss rate under the asynchronous wakeup mechanism with on-demand power management is less than that of an always-on network. This is because the beacon interval used in an always-on network allows routing-layer HELLO messages to be sent at the frequency of one per second. while in the asynchronous wakeup mechanism, beacons are sent at a higher rate (e.g. around 0.7s). As a result, with the use of the asynchronous wakeup mechanism the neighbor's information is more likely to be updated. Another factor that accounts for this phenomenon is that the asynchronous wakeup schedule also serves to randomize the wakeup time among neighbors, which in turns reduces resource contention.

6. RELATED WORK

We categorize existing network solutions to energy saving in ad hoc networks into two categories: (i) power control and (ii) power management.

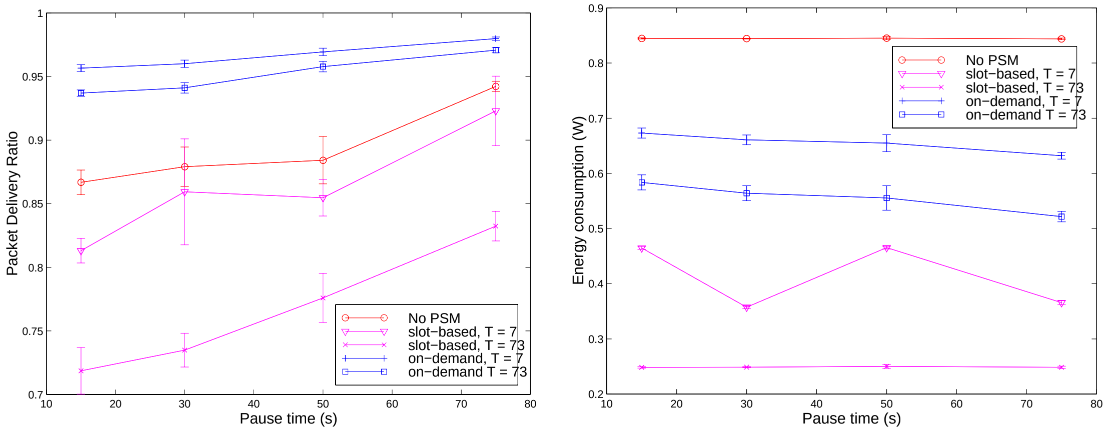
Power control [23, 12]. is concerned with determination of adequate transmission power to maintain network connectivity, reduce interference and increase spatial reuse in ad hoc networks. Several research efforts [15, 22] also consider the issue of energy-efficient unicast/multicast/broadcast routing from the perspective of power control. The key idea is that under certain radio propagation model, it is sometimes more energy-efficient (in terms of the total power consumed) to use smaller transmission power and relay messages via intermediate nodes as compared to using the maximum transmission power and reaching all nodes directly. Cagalj *et al.* [3] have proved that the problem of optimal energy-efficient broadcast is NP-complete. While power control has been recognized to be an effective means to increase network capacity [7, 8], its usefulness in reducing energy consumption is heavily dependent upon the radio propagation model and hardware specifications [9].

Power management. is concerned of which set of nodes should be turned on/off and when, for the purpose of energy saving and network longevity. It can utilize information available from all the layers in the protocol stack. At the MAC layer, power management decisions are made based



(a) Packet delivery ratio (b) Energy consumed

Figure 7: Packet delivery ratio and energy consumed in a 1500x300 static network with 50 nodes with on-off traffic sources.



(a) Packet delivery ratio (b) Energy consumed

Figure 8: packet delivery ratio and energy goodput vs. traffic load in a 50-node, 1500x300 mobile network with 10 long-lived CBR connections.

on local information. The PAMAS power-saving medium access protocol [18] turns off a node's radio when it overhears that a packet is not destined for it. This approach is well-suited for radios that incur more overhead in processing a received packet than in listening to an idle medium. In the case of an idle medium, the node must remain on all the time for incoming transmission. Thus, the effectiveness of PAMAS is limited to reducing power consumption incurred in processing unnecessary packets. Note that PAMAS can be combined with most high level power management schemes that aim to reduce energy consumption incurred in idle states.

Ye *et al.* [25] and Schurgers *et al.* [16] proposed, in the context of designing power-efficient MAC protocols for wireless sensor networks, to use asynchronous wakeup schedules. S-MAC [25] uses the dominating-awake-interval, while STEM [16] uses a wakeup schedule similar to periodically-full-awake-interval. To facilitate efficient data communication, S-MAC introduces a mechanism called *message passing* which modifies the network allocation vector (NAV) for virtual channel reservation in IEEE 802.11 type of MAC protocols. The major problem with message passing is that it is application-specific and suffers from poor packet-level fairness. In addition, the dominating-awake-interval wakeup is less optimal compared to our design thus consumes more power. STEM takes a data-driven approach for power management. The key idea is that a node remains awake until it has not receive any message destined for it for certain time. STEM uses separate control and data channels, and hence the contention among control and data messages is alleviated. The energy saving of STEM is dependent on that of the control channel. In comparison, the proposed wakeup protocol is based on optimal block design and targeted for single channel wireless networks. (although we claim that the proposed protocol can be readily extended to multi-channel wireless networks). The benefit of optimal asynchronous wakeup can be combined with other advanced power management techniques.

At the network layer, power management schemes can take advantage of topological information. In the geographical adaptive fidelity (GAF) framework [24], each node is associated with a "virtual grid." All nodes in the same grid are considered "equivalent" from the perspective of forwarding packets and maintaining network connectivity, and coordinate with each other to determine who should sleep and for how long, with the objective of ensuring the presence of at least one active node in each grid. Data source or sink nodes remain on and intermediate nodes monitor and balance energy use. Span [4] is a distributed and randomized protocol in which nodes make local decisions on whether they should sleep or join a forwarding backbone. Nodes that choose to stay awake and maintain network connectivity/capacity are called *coordinators*. A non-coordinator node elects itself as a coordinator if any two of its neighbors cannot communicate with each other directly or indirectly through one or two existing coordinators. The information needed for coordinator election is exchanged among neighbors via HELLO messages. The coordinator announcement is broadcast locally, deferred by an interval reflecting the residual power of a node.

Asynchronous wakeup. As mentioned in Section 4, asynchronous wakeup mechanisms are orthogonal to power control/management protocols and can be combined with the latter to magnify the benefit. The closest work that we know of in generating wakeup schedule is the quorum-based scheme proposed by Tseng *et al.* [20]. In their scheme, a node arbitrarily picks one column and one row of entries in an $n \times n$ array to be awaken, and the duty cycle of a node is $(2n-1)/n^2 = (2\sqrt{T}-1)/T$, where $T = n^2$. As the theoretical bound for $m = 2$ is $\sqrt{2T}/T$, by Theorem 1 and Corollary 1, the optimal schedule we propose has lower duty cycle and thus more energy efficient.

7. CONCLUSION AND FUTURE WORK

In this paper, we take a systematic approach to addressing the protocol design issues of asynchronous wakeup mechanisms. We formulate the problem of generating wakeup schedules in the asynchronous wakeup mechanism as a block design problem and derive theoretical bounds under different communication models. Based on the optimal results obtained from the block design problem, we design an asynchronous wakeup protocol which can detect neighboring nodes in finite time without requiring slot alignment. The asynchronous wakeup protocol is also resilient to packet collision and network dynamics. We consider two power management schemes that can be laid on top of the asynchronous wakeup protocol: slot-based power management and and on-demand power management.

Simulation results validate the design of our proposed protocols. In particular, the energy consumed under the asynchronous wakeup mechanism with the $(7,3,1)$-design is approximately half of that without power management, while the energy consumed under the $(73,9,1)$-design is approximately 1/3 to 1/4. As compared to the asynchronous wakeup mechanism with the $(7,3,1)$-design, the mechanism with the $(73,9,1)$-design can achieve much lower reciprocal of power consumption per unit data delivery at the expensive of slightly higher packet loss rate. On-demand power management in conjunction with the asynchronous wakeup mechanism with the $(73,9,1)$-design can achieve a good balance between energy consumption and packet delivery ratio.

As part of our on-going work, we are investigating efficient broadcast communication protocols under the asynchronous wakeup mechanisms. Note that in the asynchronous wakeup mechanism, a single transmission is not sufficient to reach all neighbors since neighbors may wakeup at different time intervals. A straw-man approach is to replace the broadcast with several unicasts. We are exploring how to combine techniques to reduce the broadcast storm problem and neighbor reservation techniques to handle flooding more efficiently. Further, the impact on routing protocols based on this type of network-wide flooding (for example, routing discovery in DSR [10] or ADOV [14]) is one of our ongoing work.

Acknowledgment

The authors would like to thank Prof. Terry Todd for the information on RFID and Prof. Douglas West for his help with the proofs in Section 2.

8. REFERENCES

[1] Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. ANSI/IEEE Std 802.11, 1999 Edition.

[2] I. Anderson. *Combinatorial Designs and Tournaments*, chapter 2. Oxford University Press, 1998.

[3] M. Cagalj and J.-P. Hubaux. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In *Proc. of the 8th Int'l Conference on Mobile Computing and Networking (Mobicom)*, 2002.

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proc. of ACM/IEEE 7th Int'l Conf. on Mobile Computing and Networking (*MobiCom*)*, July 2001.

[5] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.

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

[7] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. In *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, 1998.

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

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the Hawaii International Conference on System Sciences*, 2000.

[10] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. draft-ietf-manet-dsr-04.txt, 2001.

[11] M. Maekawa. A $\sqrt{N}$ algorithm for mutual exclusion in

decentralized systems. In *ACM Transactions on Computer Systems*, volume 3, pages 145–159, 1985.

[12] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad hoc networks : Theory, architecture, algorithm and implementation of the COMPOW protocol. In *European Wireless Conference*, 2002.

[13] M. Nosovic and T. Todd. Low power rendezvous and RFID wakeup for embedded wireless networks. In *Annual IEEE Computer Communications Workshop*, 2000.

[14] C. E. Perkins and E. M. Royer. Ad hoc on demand distance vector (AODV) routing. Internet draft, IETF MANET working group, July 2000.

[15] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. In *Proceedings of the 1998 IEEE International Conference on Communications (ICC)*, 1998.

[16] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Topology management for sensor networks: Exploiting latency and density. In *Proceedings of The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.

[17] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven algorithm and protocol design for energy-efficient wireless sensor networks. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2001.

[18] S. Singh and C. S. Raghavendra. PAMAS: Power aware multi-access protocol with signalling for ad hoc network. *ACM Computer Communication Review*, pages 5–26, July 1998.

[19] P. K. Sinha. *Distributed Operating Sytsems*, chapter 6. IEEE Computer Society Press, 1996.

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

[21] UCB/LBNL/VINT. Network simulator - ns (version 2). http://www-mash.cs.berkeley.edu/ns/, January 2000.

[22] P. J. Wan, G. Calinescu, X.-Y. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *Proc. of INFOCOM*, 2001.

[23] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proc. of IEEE INFOCOM*, 2001.

[24] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. of ACM/IEEE 7th Int'l Conf. on Mobile Computing and Networking (*MobiCom*)*, July 2001.

[25] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. of INFOCOM*, 2002.

[26] R. Zheng and R. Kravets. On-demand power management for ad hoc networks. In *Proc. of INFOCOM*, 2003.