Cooperation Helps Power Saving

Guanfeng Liang and Nitin Vaidya
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Champaign, Illinois, USA
Email: {gliang2, nhv}@illinois.edu

(Technical Report, April 7, 2009)

***Abstract*—In wireless sensor networks (WSN), energy efficiency is crucial to achieving satisfactory network lifetime. The most commonly used and may be the only efficient method to reduce the energy consumption significantly is to turn off the radios most of the time, except when it has to participate in data communication. The key challenge is to operate the radio at a low duty cycle but still ensure the delay is relatively low. Various power-saving medium-access control (MAC) protocols have been proposed along this thread. However, most of such protocols focus on a point-to-point communication setting, in which a node will drop a overheard packet if it is not the destination. Moreover, a node may even try to avoid overhearing a transmission that is not destined to it because listening consumes energy. On the other hand, cooperative wireless communication has been drawing extensive attention in the past few years. Node cooperation has been exploited to reduce end-to-end delay, improve transmission reliability, etc. However, not much has been done in utilizing node cooperation to save energy. This idea may sound absurd since cooperation requires more nodes involved in a communication and would result in more energy being consumed. But is this true? In this paper, we will exploit the possibility of cooperative power saving in wireless ad-hoc networks. The trade-off between energy consumption and delay will be studied. Interestingly, our analytical and simulation results show that cooperation can indeed help achieve a better delay-power consumption trade-off.**

I. Introduction

The wireless sensor networks (WSN) has been the focus of many recent research and development efforts. Wireless sensor networks have application in military, commercial, and educational environments including environment monitoring, home networks of devices, and surveillance systems. Since nodes in a WSN are often equipped with batteries of limited capacity as the power source, energy conservation has long been a major interest in developing Medium Access Control (MAC) protocols for WSN systems.

Significant progress has been achieved in design of low-power hardware for mobile devices and the wireless network interface is usually a device's single largest consumer of power. In IEEE 802.11 Distributed Coordination Function (DCF), when a node is not transmitting, it persists in *idle mode* and continuously listens for incoming transmissions. Studies [1], [2] observe that the energy cost of idle listening is only slightly lower than the cost of transmitting and receiving. Therefore, it has been proposed to save power by turning the radio off (*sleep mode*) when it is not in use. Significant amount of energy can be saved by introducing such a sleep mode.

This research was supported in part by NSF grant 05-19817 and Army Research Office grant W-911-NF-0710287.

However, introducing a sleeping mode raises other problem: both the sender and receiver must be awake to communicate, but it is very difficult for a sleeping node to know when there is incoming transmission. The sleeping node may miss the communication opportunity which will result in long delivery delay and/or low energy efficiency. A solution is to use a Time Division Multiple Access (TDMA) scheme [3], [4] or synchronize nodes' wake-up schedules locally [5], [6], [7] but this requires nodes to synchronize with each other quite tightly, which can be quite a complex task in large networks with random node locations and imperfect (drifting) clocks. Letting nodes set their wake-up and sleeping times in a decentralized fashion reduces this complexity and this approach is adopted by many researchers due to its scalability and easiness for implementation. For example, in Quorum-based Power Saving (QPS) protocols [8], [9], [10], [11], [12], [13] each node follows its own awake/sleep cycle pattern. The cycle patterns are carefully designed so that the awake periods of two nodes are guaranteed to overlap regardless of the offset between their clocks. In STEM [14], nodes wake up periodically and listen the channel if they have no data to send. When packet arrives, a node stays awake for a long enough time until the destination wakes up and then they communicate.

Most of such power-saving MAC protocols have been focusing on solving the point-to-point or single route communication problem. In such systems, nodes will drop any overheard packet if it is not the destination (or on the route). In some sense, there is no/weak cooperation between nodes. Moreover, these protocols usually assume by intuition a symmetric power allocation: same duty cycle p (the fraction of time that a node is awake) for all nodes. Under such assumptions, the delay is about the order of $1/p^2$.

In this paper, we investigate the effect of both cooperation (relaying) and asymmetric power allocation. We identify circumstances when cooperation and/or asymmetry is beneficial. By studying a randomized wake-up scheme, we show that in a system of N nodes, cooperation with symmetric power allocation can reduce delay by a factor of $1/Np$. The delay can be further reduced to the order of $1/p$ with appropriate asymmetric power allocation. In a dense network (large N) with a low power consumption rate requirement (small p), the improvement is significant. To the best of our knowledge, this is the first paper trying to study the fundamental limits of the delay-power consumption trade-off MANET systems with cooperation and asymmetric power allocation.

The rest of this paper is organized as follows. Section II reviews current power-saving MAC protocols. Our system model is introduced in section III. In section IV and section V, the delay-power consumption trade-off of traffic independent and traffic dependent power-saving protocols is studied. Section VIII concludes the paper.

II. Related Work

The IEEE 802.11 standard has a power saving mode, for both PCF (Point Coordination Function) and DCF. In DCF, time is divided into the so-called *beacon intervals* by means of a distributed protocol for beacon transmission. At the start of each beacon interval, each node must stay awake for a fixed time interval, called ATIM window (ATIM stands for Adhoc Traffic Indication Message). During the ATIM window, any node has a packet destined for another node transmits an "ATIM frame" using the CSMA/CA (collision avoidance) mechanism specified in IEEE 802.11. A node that receives an ATIM frame replies by sending an ATIM-ACK. Such a node remains awake for the entire beacon interval, after transmitting the ATIM-ACK. The first node that receives an ATIM-ACK *wins* the whole ATIM window. Transmission of one or more data packets from the winning node to its destination can now take place during the beacon interval, after the end of the ATIM window. A node that has no outstanding packets to be transmitted can go into the doze state at the end of the ATIM window, if it does not receive an ATIM frame during the ATIM window.

In PSM specified in IEEE 802.11, all nodes use the same (fixed) ATIM window size, as well as identical beacon intervals [13]. Since the ATIM window size critically affects throughput and energy consumption, a fixed ATIM window does not perform well in all situations. [15] proposes a dynamic mechanism for choosing an ATIM window size.

SPAN [16], a power saving technique, elects a group of coordinators which are changed periodically. The coordinators stay awake and forward traffic for active connections. Non-coordinators follow the power saving mechanism in IEEE 802.11. SPAN introduces a new advertised traffic window following an ATIM window. During this advertised traffic window, the announced packets and the packets for the coordinators can be transmitted. After this window, only the packets for the coordinators can be transmitted, and non-coordinators can go to doze state if they do not have traffic to send or receive.

IEEE 802.11-based power saving protocols requires strict synchronization across nodes, which is quite a complex task for most random large networks. It is appealing if synchronization is not necessary and nodes can set their wake-up and sleep times in a decentralized/asynchronous fashion while guaranteeing overlapping of their wake-up periods for communication. In S-MAC [5] and its variants [6], [7], neighboring nodes form virtual clusters so as to set up a common sleep schedule. If two neighboring nodes reside in two different virtual clusters, they wake up at the listen periods of both clusters. A drawback of the S-MAC algorithm is this possibility of following two different schedules, which results in more energy consumption via idle listening and overhearing.

STEM [14] is a power-saving MAC protocol that does not require any synchronization. In STEM, each node periodically turns on its radio for a short time to listen if someone wants to communicate with it. Every node has the same wake-up period but the off-set is random. In order to transmit, when a packet arrives, the source node starts polling the destination node continuously. As soon as the destination node hears the poll, the link between the two nodes is activated. In the original STEM, two radios working on two separate channels are used: one radio is turned on only for data transmission, and the other radio on the "paging" channel and wakes up periodically. The aforementioned polling is performed on the paging channel.

Another example of asynchronous power-saving MAC protocols is the quorum-based protocols [8], [9], [10], [11], [12]. In a quorum-based protocol, the time axis on each station is divided evenly into beacon intervals. A station may stay awake or sleep during each beacon interval. Given an integer n, a quorum system defines a cycle pattern, which specifies the awake/sleep schedule during n continuous beacon intervals, for each station. The merit of QPS protocols is that a station is required to remain awake only $O(\sqrt{n})$ beacon intervals every cycle, and that at least one of these awake beacon intervals is guaranteed to overlap with that of another station.

Finally, we look at some work closely related to ours. The first example is [13]. Observing that in clustered environments there is no need to insist in all-pair neighbor discovery, the authors proposed an Asymmetric Cyclic Quorum (ACQ) system, which guarantees the neighbor discovery between each member node and the clusterhead in a cluster, and between clusterheads in the network. The ACQ system is the first asymmetric quorum system. A construction scheme is presented in this work, which assembles the ACQ system in O(1) time. It is shown that by taxing slightly more energy consumption on the clusterhead, the average energy consumption of stations in a cluster can reduce substantially than can be achieved by traditional QPS protocols. However, they only considered how to generate quorums given certain asymmetric power allocation and justified their claim by simulations. Neither the conditions under which asymmetry is beneficial nor the optimal allocation of power is discussed.

Another closely related work is [17]. This is an analytical paper on the end-to-end delay of multi-hop wireless networks with power-saving MAC. The authors considered the extreme case of decentralization: nodes go to sleep independently from each other, which is also the solution we adopt in the present paper. The authors obtained analytical bounds on the latency of a node with random i.i.d. (independent and identically distributed) active and sleeping periods using dynamic percolation theory. It is proved that any message generated by a node will reach the destination in a time proportional to the distance between the source and the destination, even though node's switching on/off schedules are not coordinated at all, their positions are random, and the durations of on/off period are such that the number of active nodes at any particular time is so low that the network is always disconnected. However, the relationship between power consumption and delay is not study and only symmetric power allocation is considered.

III. System Model

We consider a static sensor network in which time is slotted. The slot duration is the length of a packet transmission. In every slot, nodes decide to sleep or wake up independently from each other. For simplicity, we assume that a node consumes one unit of energy per slot when it wakes up, no matter it transmits, receives, or stays idle; and there is no energy consumption when a node is sleeping. This assumption is commonly used and usually valid since the power consumption of the transmit, receive, and idle mode are roughly the same and much higher than that of the off mode. Please notice that this assumption can easily be relaxed if different modes have different power consumption rates, and our analysis will still hold with minor modification. Further, since we want to focus on the effect of cooperation on delay and energy consumption and do not want it to be overshadowed by the effect of collisions, we will assume there is no collision. Collisions is unlikely in sensor networks where the total traffic load is typically low.

We are interested in the trade-off between the average delay of packet delivery and the total power consumption of the network (or average power consumption per node), instead of the power consumption of each individual node which is usually considered in the literature. The reason behind this choice is that in many cases, after the sensor network is set up, the total amount of energy consumed is the cost to pay. In many wireless network systems, such as industry monitoring, office temperature monitoring, mesh networks, etc., it is easy to support sensor nodes by the existing infrastructure. For example, imagine an industry monitoring system in a factory. In such a system, sensors can usually get power supply directly from the power line. And the factory is charged for the total electricity usage every month. In these systems, the total power consumption is more important than the individual power consumption or the life of the networks, from the system administrator/designer's point of view.

As the starting point, we will consider the case of single-hop networks. In a single-hop network, every node is within the transmission range of any other node. Also, we assume an uniform traffic pattern: the destination of packets generated by a node is chosen uniformly randomly from all the other nodes. In this section, we will first study the case when traffic load is not considered. We will analyze the delay-power consumption trade-off of symmetric randomized wake-up with and without cooperation, and asymmetric randomized wake-up with cooperation. After that, we will study the case in which power consumption is dependent on the traffic load. A deterministic periodic wake-up without cooperation (STEM [14]) and a modified asymmetric randomized wake-up with cooperation will be compared. We decide to use randomized schemes because of their tractability. And we claim that the results we derive from randomized schemes can be extended to deterministic schemes, within a constant factor.

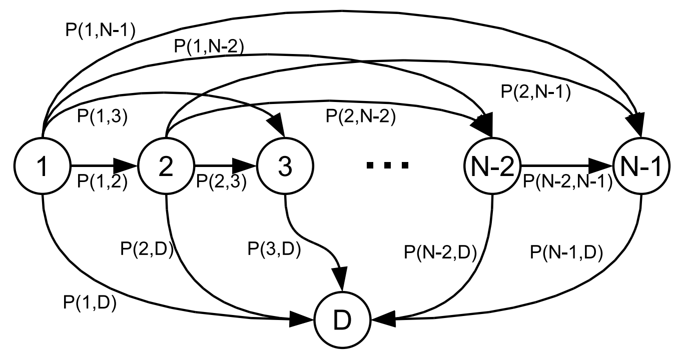

Fig. 1. Flooding

IV. Analysis: Traffic Independent Schemes

In this section, we will study the delay-power trade-off of traffic independent schemes. A traffic independent scheme is a scheme that schedules a node to wake up or sleep regardless of whether this node has a packet to send or not. As a result, the delay-power trade-off of such schemes does not depend on the packet arrival process.

A. Symmetric Randomized Wake-Up

By symmetric randomized wake-up, we mean that in every time slot, each node in the network wakes up independently with the same probability p. So in a network with N nodes, the total power consumption rate is Np units of energy per time slot.

If there is no cooperation, after a packet arrives, it can not be transmitted until both the transmitter and the receiver nodes wake up. It is easy to see the delay is $D_{sym} = 1/p^2$. We have to point out that a similar relationship can be observed in the deterministic quorum-based protocols: in symmetric quorum systems, in order to guarantee a *meeting* between two nodes once every n slots, each node has to stay awake for $\sqrt{n}$ slots in every cycle, which results in an average delay of $n/2$ and a power consumption rate of $1/\sqrt{n}$ units of energy per slot.

For the case in which nodes cooperate, we need to first define how they cooperate. In this paper, we will only consider cooperation in the form of relaying. By relaying, the transmitter can first send the data packet to some nodes in the network other than the destined receiver. A node that receives a relayed packet may keep on relaying the packet to some other nodes repeatedly until the receiver gets the packet from one of these them (we will term this strategy as *flooding*). Or, the node will only transmit the packet when it meets the destined receiver. In the latter case, a packet will be transmitted at most twice. This means the source node will only transmit once to nodes other than the destination and it will once transmit for a second time if it meets the destination. Meanwhile, the relaying nodes will only transmit the packet to the destination node. In this paper, we will call this scheme as *SYM2*. The SYM2 scheme is a more practical scheme compared with flooding. The reason is: when N gets large, the packet may be transmitted for many times with flooding and the negligible extra power consumption of transmit/receive mode will be cumulated and becomes not negligible any more; but SYM2 keeps the extra power consumption negligible by upper bounding the number of transmissions per packet by a small number (2).

1) Flooding: In flooding, the number of nodes that have a copy of a particular data packet can be modeled by the following Markov chain as shown in Fig.1. With N nodes, the Markov chain has N states, namely state 1, 2, ..., N-1, and state D. State i represents the number of nodes besides the destination node that have a copy of a packet. And state D represents that the packet has been received by its destination. When a new packet arrives at a node, it always starts the Markov chain at state 1, and the chain is terminated at state D. Following the symmetric independent wake-up schedule, the transmission probabilities of this Markov chain are given by:

$$P(i,j) = (1-(1-p)^i)(1-p)\binom{N-i-1}{j-i} p^{j-i}(1-p)^{N-j-1}, \ i<j<N \tag{1}$$
$$P(i,D) = (1-(1-p)^i)p \tag{2}$$
$$P(i,i) = 1 - \sum_{j=i+1}^{N} P(i,j), \ i<N \tag{3}$$
$$P(D,D) = 1. \tag{4}$$

Denote D_i as the expected delay until packet reception at the destination given the current state is i. It can be written in the following way

$$D_i = 1 + D_i P(i,i) + \sum_{j=i+1}^{N-1} D_j P(i,j) \tag{5}$$

This can be rewritten in a recursive way as

$$D_i = \frac{1 + \sum_{j=i+1}^{N-1} D_j P(i,j)}{1-P(i,i)}. \tag{6}$$

Noticing that in state N-1, the delay until delivery is a Geometrically distribute r.v. (random variable) with success probability

$$P(N-1,N) = (1-(1-p)^{N-1})p. \tag{7}$$

So the expected delay given in state N-1 is

$$D_{N-1} = 1/P(N-1,D). \tag{8}$$

Then the total delay after a packet arrives with the flooding scheme is $D_{flood} = D_1$ can be computed recursively according to Eq.6.

2) SYM2: The analysis of SYM2 is similar. The number of copies of a packet in the network follows a Markov chain as shown in Fig.2, which is defined by the following transition probabilities:

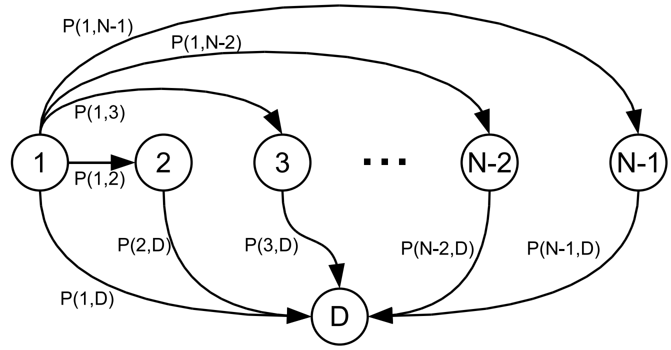

Fig. 2. SYM2

$$P(1,i) = p(1-p)\binom{N-2}{i-1} p^{i-1}(1-p)^{N-i-1}, \ 1<i<N \tag{9}$$
$$P(i,D) = (1-(1-p)^i)p \tag{10}$$
$$P(1,D) = p^2 \tag{11}$$
$$P(1,1) = (1-p) + p(1-p)^{N-1} \tag{12}$$
$$P(i,i) = (1-p) + p(1-p)^i, \ 1<i<N \tag{13}$$
$$P(D,D) = 1. \tag{14}$$

In this case, the number of copies of a packet in the network will not increase after the first transmission. And the delay till reception from state i is Geometrically distributed r.v. with success probability $P(i,N)$. Similar to Eq.6, the expected delay with SYM2 is

$$D_{SYM2} = \frac{1 + \sum_{j=2}^{N-1} D_j P(1,j)}{1-P(1,1)}, \tag{15}$$

where

$$D_i = \frac{1}{P(i,D)} \ ,2<i<N. \tag{16}$$

In general, Eq.15 cannot be further simplified. To better understand the delay of the SYM2 scheme, we will consider the asymptotic case when N gets large. With a large N, whenever a source node wakes up, it typically meets around $Np-1$ other nodes. With probability p, the destination is one of the $Np-1$ nodes. Otherwise, after the source node wakes up for the first time, there are about Np nodes, including the source, in the network holding a copy of the packet. The probability that one of these Np nodes meets the destination in a slot is $(1-(1-p)^{Np})p$. So the total delay can be estimated by

$$\widetilde{D}_{SYM2} = \frac{1}{p} + \frac{1-p}{(1-(1-p)^{Np})p} \tag{17}$$
$$\approx \frac{1}{p} + \frac{1-p}{Np^3} \tag{18}$$
$$\approx \frac{1}{Np^3}. \tag{19}$$

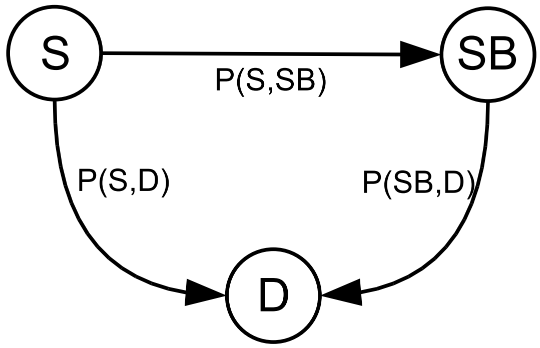

Fig. 3. ASYM

B. Asymmetric Randomized Wake-Up with Cooperation

Generally speaking, the wake up probability can be different from node to node so that some network metrics are optimized. But allowing each node to have a different wake up probability leads to a N-dimensional optimization problem. When N is large, which is the typical case, the complexity of solving the optimization problem usually become prohibitively high and the problem itself becomes intractable. In order to better understand the delay-power trade-off with cooperation, we will only consider a special case of the asymmetric scenario: N-1 nodes share the same wake-up probability p_1, and the one node left may use a different wake-up probability p_2. We will refer to this special node as the *beacon node* from now on. The transmission scheme is similar to SYM2: each packet will be transmitted for at most twice, i.e., it will be relayed for only once. Moreover, we will further constrain ourselves by only allowing relaying through the beacon node. We will term this scheme as *ASYM* from now on.

The ASYM scheme can be modeled by a 3-state Markov chain as shown in Fig.3 with transition probabilities as follows:

$$P(S,D) = p_1^2 \tag{20}$$
$$P(S,SB) = p_1(1-p_1)p_2 \tag{21}$$
$$P(SB,D) = p_1(1-(1-p_1)(1-p_2)) \tag{22}$$
$$P(D,D) = 1. \tag{23}$$

Here S, D, and B represent the location of copies of a packet: source, destination, and the beacon node, respectively. Then the total delay of a packet can be calculated in a way similar to Eq.6 and Eq.15

$$\begin{aligned} D_{ASYM} &= \frac{1 + D_{SB}P(S,SB)}{1 - P(S,S)} \\ &= \frac{1 + D_{SB}P(S,SB)}{P(S,SB) + P(S,D)} \\ &= \frac{2(p_1 + p_2 - p_1p_2) - p_1}{p_1(p_1 + p_2 - p_1p_2)^2}. \end{aligned} \tag{24}$$

Now, optimization of the delay over assignment of p_1 and p_2 can be done given the total power consumption constraint

$$(N-1)p_1 + p_2 = Np, \tag{25}$$

where p is the average wakeup probability per node. The closed-form of the optimal solution can be obtained by solving this optimization problem. However, it is still a bit too complicated to draw some meaningful observations. We will further constrain the relaying scheme: the source node always transmit packets to the beacon node, even if it meets the destination node first, and the destination always gets the packet from the beacon node. By making this constraint, we are giving up the chance of direct one-hop transmission from the source to the destination in the original ASYM scheme, and will give an upper bound of D_{ASYM}.

Notice that, in the constrained version of ASYM, the one-hop delay from source to beacon and from beacon to destination are two i.i.d. (identical independent distributed) Geometric r.v.'s, both with success probability p_1p_2. So the total expected delay is

$$\begin{aligned} \bar{D}_{ASYM} &= \frac{2}{p_1p_2} \\ &= \frac{2}{p_1(Np - (N-1)p_1)} \end{aligned} \tag{26}$$

It is easy to see that the optimal value of p_1 and p_2 that minimize the above equation is

$$\begin{aligned} p_1^* &= \frac{Np}{2(N-1)} \\ p_2^* &= \frac{Np}{2}. \end{aligned} \tag{27}$$

However, since both p_1 and p_2 are probabilities and cannot be greater than 1, we have the following:

- $Np < 2$

$$\begin{aligned} p_1^* &= \frac{Np}{2(N-1)} \\ p_2^* &= \frac{Np}{2} \\ \bar{D}_{ASYM} &= \frac{8(N-1)}{N^2p^2} \end{aligned} \tag{28}$$

- $Np \geq 2$

$$\begin{aligned} p_1^* &= \frac{Np-1}{N-1} \\ p_2^* &= 1 \\ \bar{D}_{ASYM} &= \frac{2(N-1)}{Np-1} \end{aligned} \tag{29}$$

V. Analysis: Traffic Dependent Case

In the previous section, we have studied the case when nodes wake up at a same frequency with or without packet to send. To better trade off delivery delay and energy consumption, it makes sense to wake nodes more frequently if it has a packet to send. And when a node has no packet to send, it only needs to wake up infrequently so that it can still receive packets from other nodes. In this case, the power consumption of a network depends on the traffic load. Here we assume a uniform traffic scenario: for every node in the network, its destination is chosen uniformly at random from the all the other nodes. Packets arrive at every node at the same rate λ.

A. Deterministic Periodic Wake-Up without Cooperation (STEM)

STEM [14] is a well-known MAC protocol for power saving. In STEM, each node periodically turns on its radio for a short time to listen if someone wants to communicate with it. When a packet arrives, the source node starts polling the destination node continuously. As soon as the destination node hears the poll, the link between the two nodes is activated. In the original STEM, two radios working on two separate channels are used: one radio is turned on only for data transmission, and the other radio on the "paging" channel and wakes up periodically. The aforementioned polling is performed on the paging channel. For the purpose of fair comparison, we will consider a single channel single radio time-slotted version of STEM, described as follows: every node wakes up periodically once every T slots; at the beginning of every slot, a packet arrives with probability λ; once a packet arrives, the source node stays awake polling the destination; as soon as the source meets the destination, all buffered packets are transmitted.

Since packets arrive independent of the wake up schedule, on average a packet has to wait for $T/2$ slots before the destination wakes up and gets delivered. So the expected delay of STEM is:

$$D_{STEM} = \frac{T}{2}. \tag{30}$$

Let slot 0 be the first slot the destination node's sleeping period, the first packet arrives at the source node in slot i with probability $(1-\lambda)^i\lambda$, $i = 0, 1, ..., T-1$. And with probability $(1-\lambda)^T$ no packet arrives during the whole duty cycle. So the expected number of slots the source node can sleep within a duty cycle of the destination node is

$$\begin{aligned} F &= \sum_{i=0}^{T-1} i(1-\lambda)^i\lambda + T(1-\lambda)^T \\ &= \frac{(1-\lambda) - (1-\lambda)^{T+1}}{\lambda}. \end{aligned} \tag{31}$$

Then the power consumption rate (per slot) for STEM can be formulated as a function of the delay and traffic load:

$$\begin{aligned} P_{STEM}(D,\lambda) &= \frac{T-F}{T} + \frac{F}{T}\frac{1}{T} \\ &= 1 - \frac{(1-\lambda) - (1-\lambda)^{2D+1}}{2\lambda D} \\ &\quad + \frac{(1-\lambda) - (1-\lambda)^{2D+1}}{4\lambda D^2} \end{aligned} \tag{32}$$

The first term in the above equation is the fraction of time that the source node has to stay awake to transmit packets. The second term means that within the F/T fraction of time that the source can sleep, it still need to wake up $1/T$ for its own duty cycle.

Note that in the case when there is multiple destinations, Eq.32 is in fact a lower bound of the power consumption since in this case the source node has to stay up more due to the asynchronous duty cycles at the destination nodes.

B. Modified Asymmetric Randomized Wake-Up with Cooperation

In this subsection, we will introduce a minor modification to ASYM (ASYM2) so that the power consumption is traffic dependent: after a packet arrives, the source node switches into the *active mode* and wakes up with probability p_A until the packet is transmitted to the destination or the beacon node; in other case, it staying in a power save mode and wakes up with probability p_1. Assuming the destination has no packet to send and always wakes up with probability p_1, the worst case delay of ASYM2 can be obtained similar to Eq.29:

$$D_{ASYM2} = \frac{p_A p_1 + p_A p_2 + p_1 p_2 - 2p_A p_1 p_2}{p_A p_1 (p_A + p_2 - p_A p_2)(p_1 + p_2 - p_1 p_2)}. \tag{33}$$

It is not hard to see that a source node stays in the active mode for $\frac{\lambda}{\lambda + p_A(p_1+p_2-p_1p_2)}$ fraction of the time. So the power consumption of a node is

$$\frac{\lambda p_A + p_A(p_1 + p_2 - p_1 p_2)p_1}{\lambda + p_A(p_1 + p_2 - p_1 p_2)}. \tag{34}$$

And the average (per node) power consumption becomes

$$P_{ASYM2} = \frac{N-1}{N}\frac{\lambda p_A + p_A(p_1 + p_2 - p_1 p_2)p_1}{\lambda + p_A(p_1 + p_2 - p_1 p_2)} + \frac{p_2}{N}. \tag{35}$$

Given Eq.33 and Eq.35, the optimal trade-off between delay and power can be done over (p_A, p_1, p_2).

Since the expression for optimal delay-power trade-off is quite complicated, in order to get a better understanding, here we will consider the special case when $p_A = p_2 = 1$. This gives us a upper bound on the optimal delay (or lower bound on optimal power consumption) under constraint of total power consumption (or delay). In this case, the delay becomes

$$D_{ASYM2} = \frac{1}{p_1}, \tag{36}$$

and the power consumption can be expressed as a function of the delay and traffic load:

$$P_{ASYM2}(D,\lambda) = \frac{(N-1)(\lambda + \frac{1}{D})}{N(\lambda+1)} + \frac{1}{N}. \tag{37}$$

VI. Performance Evaluation

In this section, we verify our analysis results through simulation and compare different power saving schemes.

A. Traffic Independent Schemes

We first compare the traffic independent schemes discussed in the previous sections. As shown in Fig.4, our analysis results matches the simulation results very well. We can see that all schemes with cooperation have lower delay than the non-cooperative one at all different power consumption levels. Among these cooperative schemes, flooding achieves the lowest delay and SYM2 has the highest. This is mainly because in SYM2 every packet is transmitted for at most twice while in flooding a packet will be transmitted as many times as possible before it reaches the destination. So in reality, flooding consumes much more energy than SYM2. We have to point out that, although ASYM only allows at most two

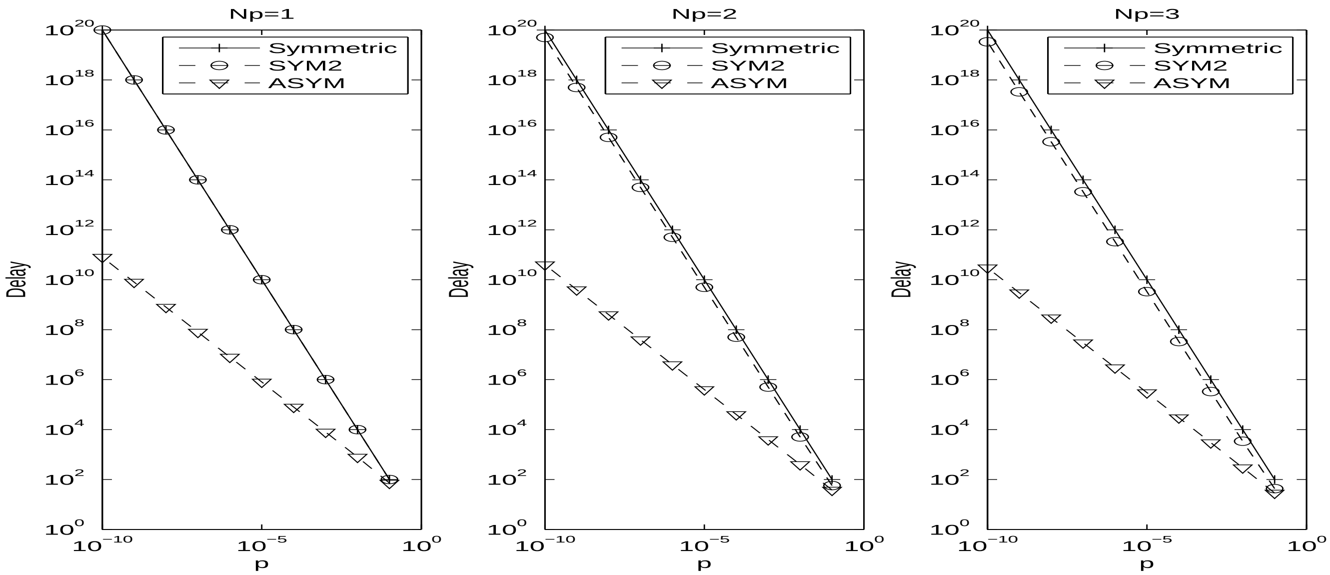

Fig. 5. Asymptotic Delay-Power trade-off with fixed Np

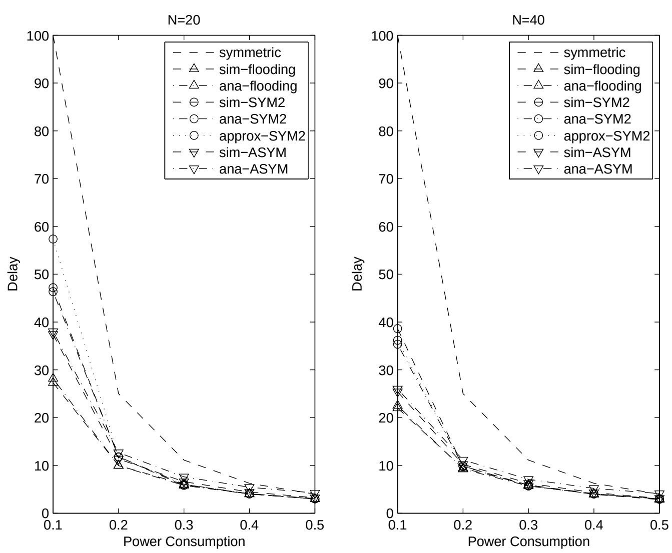

Fig. 4. Traffic Independent Schemes

transmissions per packet, it achieves a delay almost as low as flooding, especially when N is large.

We also observe that when $N = 20$, Eq.19 is not a good approximation of D_{SYM2}. The explanation for this is: when we introduce Eq.19, we first approximate the random number of relaying nodes with a constant Np. We are actually applying Law of Large Number, which inherently requires a large N for this approximation to be valid. We can see Fig.4, as N gets large (40), Eq.19 is a pretty close estimation of D_{SYM2}.

B. Asymptotic Performance

While designing a practical wireless sensor system, a major concern is the cost: one-time deployment cost and the long-term maintenance cost. The one-time deployment cost includes the cost to purchase/develop the sensor nodes, the labor work required to deploy the nodes, etc. This one-time cost is roughly proportional to the number of sensor nodes. And the long-term maintenance cost, in this paper's scope, is the energy cost, which is linear in the duration that the system runs. A system designer should be able to find a good balance among the number of nodes to deploy, the power consumption rate, and the delay. So it is helpful to study the delay-power relationship in the asymptotic case when N approaches infinity while Np remains constant.

As we can see in Fig.5, when Np is fixed and N increases, the delay of the SYM2 is at the same order of the delay of the non-cooperative scheme ($1/p^2$) within a constant factor, while ASYM achieves a delay at a lower order. This consists with our analysis. According to Eq.19, when $Np = C$

$$D_{SYM2} \approx \frac{1}{C}\frac{1}{p^2} = \Theta(\frac{1}{p^2}). \tag{38}$$

And from Eq.28 and Eq.29, we have

- $Np < 2$

$$\begin{aligned}\bar{D}_{ASYM} &= \frac{8(N-1)}{N^2p^2} \\ &\approx \frac{8}{Cp} = \Theta(\frac{1}{p})\end{aligned} \tag{39}$$

- $Np \geq 2$

$$\begin{aligned}\bar{D}_{ASYM} &= \frac{2(N-1)}{Np-1} \\ &\approx \frac{2}{p} = \Theta(\frac{1}{p}).\end{aligned} \tag{40}$$

Although by cooperation SYM2 improves the delay-power trade-off compared with the non-cooperative scheme, it is only by a constant factor under the total power budget constraint. On the other hand, given the same number of sensors and the same network-wide total power consumption rate, ASYM reduces the delivery delay by a factor of order of p (or $1/N$). This is a significant improvement in a dense network setting.

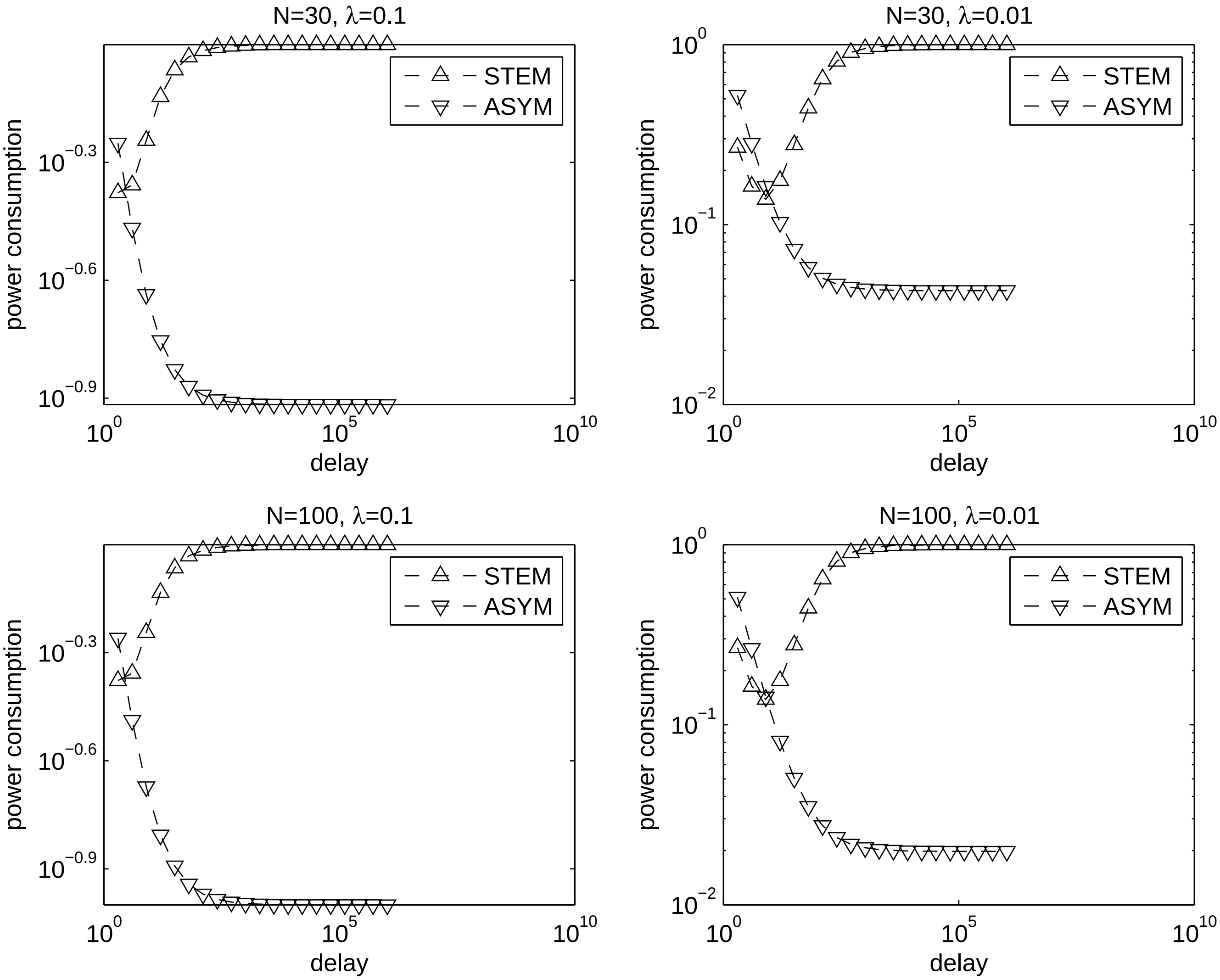

Fig. 6. STEM v.s. ASYM2

Moreover, although there is no formal proof, it is reasonable to believe that given a total power budget Np, the optimal delay should be within a constant factor of $1/p$ since each node wakes up roughly once every $1/p$ slots. So we claim that ASYM is an order optimal scheme.

C. *Traffic Dependent Schemes*

In this section, we investigate the performance of the traffic dependent schemes: STEM and ASYM2. Fig.6 compares STEM and ASYM2 at different network sizes with different traffic loads. We can see that if the latency requirement is stringent, ASYM2 will consume a little bit more energy than STEM. But as we relax the latency requirement, the power consumption of ASYM2 becomes less than STEM. It is interesting to notice that in STEM, as the delay requirement becomes looser, the power consumption decreases only to a certain optimal point. Beyond that, further relaxing the delay requirement cannot reduce power consumption any more. Instead, the power consumption goes up. This means that even if delay is not a concern, there is a minimum power consumption for STEM, given a certain amount of traffic. This optimal operating point can be achieved by tuning the duty-cycle T carefully. On the other hand, in ASYM2, the power consumption decreases monotonically as the delay requirement relaxes.

VII. Multi-Hop Networks

In the previous sections we have studied the optimal delay-power trade-off that can be achieved by cooperation and asymmetric power allocation in the single-hop networks. In practice, the area a wireless network is deployed over is usually larger than the communication range of any single node and data packets have to be transported in a multi-hop fashion. In this section, we will study the optimal power allocation in the multi-hop setting.

To start with, we consider the simplest case when nodes are connected as a chain. In a chain with N nodes indexed as $1, 2, ..., N$ in the increasing order, a node i can only communicate with its immediate neighbors, i.e. node $i-1$ and $i+1$. There is only one data flow from node 1 to node N. Given a certain amount of total power budget C, we want to minimize the end-to-end delay by assigning each node's duty-cycle appropriately. For each hop in the chain from node i to $i+1$, the delay is $1/p_i p_{i+1}$, $i = 1, 2, ..., N-1$. And the problem can be formulated as the following convex optimization problem:

$$\min_{\mathbf{p}} \sum_{i=1}^{N-1} \frac{1}{p_i p_{i+1}} \qquad (41)$$
$$s.t. \sum_{i=1}^{N} p_i = C$$

Proposition 1. *There exist a unique power allocation that minimizes the end-to-end delay minimization problem.*

Proof 1. *It is easy to see that each term in the summation in the objective function is a strictly convex function in* $\mathbf{p}$. *By summing up these* $N-1$ *terms, the objective function is also a strictly convex function in* $\mathbf{p}$. *So the solution to the optimization problem is unique.* □

Then we have the following proposition

Proposition 2. *The optimal power allocation of the end-to-end delay minimization problem is distributed symmetrically around* $N/2$. *In other words,*

$$p_i^* = p_{N+1-i}^* \qquad (42)$$

for all $i = 1, 2, ..., N$.

Proof 2. *We can re-index the nodes in the reverse direction such that the source node is now indexed as node* N *and the destination node is now node* 1. *Then the minimization problem can be rewritten as*

$$\min_{\mathbf{q}} \sum_{i=1}^{N-1} \frac{1}{q_i q_{i+1}} \qquad (43)$$
$$s.t. \sum_{i=1}^{N} q_i = C,$$

where q_1 *is the power budget for the destination node and* q_N *is the one for the source node. This is exactly the same optimization problem as Eq.41. So if* $\mathbf{p}^* = \{p_1, p_2, ..., p_N\}$ *is a solution to the original problem,* $\mathbf{q}^* = \{p_1, p_2, ..., p_N\}$ *solves the problem in Eq.43, which means* $\bar{\mathbf{p}}^* = \{p_N, p_{N-1}, ..., p_1\}$ *also solves Eq.41. And from Proposition 1, we have*

$$p_i^* = p_{N+1-i}^*. \qquad (44)$$

□

By Proposition 2, the number of variables can be reduced to $\lceil N/2 \rceil$ in the optimization. We can find the closed form solution when N is small. For example:

- $N = 3$
$$p_1^* : p_2^* = 1 : 2 \qquad (45)$$
- $N = 4$
$$p_1^* : p_2^* = 1 : \frac{1+\sqrt{5}}{2} \qquad (46)$$
- $N = 5$
$$p_1^* : p_2^* : p_3^* = 1 : 1 + \frac{1}{\sqrt{2}} : \sqrt{2} \qquad (47)$$

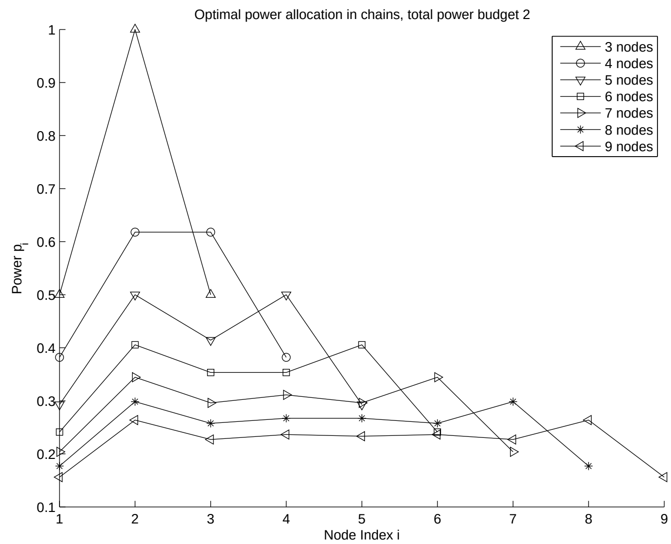

Fig. 7. Optimal power allocation in chains

Fig.7 shows the optimal power allocation in chains consist of 3 to 9 nodes obtained by solving the optimization problem Eq.41, with a total power budget equals to 2.

VIII. Conclusion

In this paper, we investigate the effect of both cooperation (relaying) and asymmetric power allocation in power saving MAC protocols. We identify circumstances when cooperation and/or asymmetry is beneficial. By studying a randomized wake-up scheme, we show that in a network with N nodes given a certain power budget, cooperation with symmetric power allocation can reduce delay by a constant factor. The delay can be further reduced by a factor of the of p with appropriate asymmetric power allocation. In a dense network (large N) with a low power consumption rate requirement (small p), the improvement is significant. We also believe the asymptotic power allocation scheme achieves the order optimal delay in the asymptotic case. To the best of our knowledge, this is the first paper trying to study the fundamental limits of the delay-power consumption trade-off of wireless sensor systems with cooperation and asymmetric power allocation.

References

[1] L. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1548–1557 vol.3, 2001.

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

[3] K. Sohrabi and G. Pottie, "Performance of a novel self-organization protocol for wireless ad-hoc sensor networks," *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th*, vol. 2, pp. 1222–1226 vol.2, 1999.

[4] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wirel. Netw.*, vol. 12, no. 1, pp. 63–78, 2006.

[5] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 3, pp. 493–506, June 2004.

[6] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 171–180.

[7] P. Lin, C. Qiao, and X. Wang, "Medium access control with a dynamic duty cycle for sensor networks," *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 3, pp. 1534–1539 Vol.3, March 2004.

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

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

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

[11] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks," *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, pp. 200–209 vol.1, 2002.

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

[13] S.-H. Wu, C.-M. Chen, and M.-S. Chen, "An asymmetric quorum-based power saving protocol for clustered ad hoc networks," *Distributed Computing Systems, 2007. ICDCS '07. 27th International Conference on*, pp. 1–8, June 2007.

[14] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *Mobile Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 70–80, Jan-Mar 2002.

[15] E.-S. Jung and N. Vaidya, "An energy efficient mac protocol for wireless lans," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002, pp. 1756–1764 vol.3.

[16] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2001, pp. 85–96.

[17] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of wireless sensor networks with uncoordinated power saving mechanisms," in *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2004, pp. 109–120.