This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

Stochastic Sleep Scheduling for Large Scale Wireless Sensor Networks

Yaxiong Zhao and Jie Wu
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122
{yaxiong.zhao, jiewu}@temple.edu

***Abstract*—This paper studies the sleep scheduling problem for large scale wireless sensor networks (WSNs), which have hundreds to thousands of sensors. Sensors are extremely sensitive to energy consumption because they are powered by batteries. In this paper, we propose stochastic sleep scheduling, a generic duty-cycling scheduling method based on stochastic theory. It lets each sensor fall asleep based on certain stochastic processes. This strategy requires no clock synchronization, and a little coordination amongst sensors. In order to characterize this design, we analyze its end-to-end communication delay and energy consumption using stochastic methods. According to the analysis, this method actually has a reasonably small delay due to the redundancy in WSNs. Simulation studies are used to verify the analytical results. We also apply stochastic sleep scheduling to S-MAC, a state-of-the-art duty-cycled sensor MAC protocol. Simulation results verify the energy consumption of the new MAC protocol.**

***Index Terms*—Sleep scheduling, medium access control (MAC), stochastic theory.**

I. Introduction

Wireless sensor networks (WSNs) [1] are deemed the next-generation information processing platform. In this paper, we consider large-scale WSNs, which consist of hundreds to thousands of sensors. Duty-cycling [2] is employed in order to save energy in WSNs, i.e. sensors fall asleep for a long time and wake up periodically to perform application tasks.

A sensor generally has three modules: *sensing*, *computing*, and *communication*. We only consider the scheduling of the communication module in this paper. Other modules of the sensors may employ their own policies to trade-off between energy-saving and efficiency. For example: detection delay and energy consumption should be considered in the scheduling of the sensing module.

Network-wide operations in large-scale WSNs need long convergence times, thus, incur considerable overhead. Even simple ones may become infeasible in large networks. Sensors need to turn on their radio for a long time to complete the operations. Although the operations themselves may incur a limited amount of overhead, they induce idle listening, which consumes considerable energy [2]. For example, synchronous scheduling requires clock-synchronization amongst all the sensors in the network, which may render it impractical in large networks because of the unacceptable overheads.

In this paper, we propose *stochastic sleep scheduling* (SSS)—a simple method based on stochastic theory. Instead of finding fixed schedules for each sensor, SSS lets each sensor draw schedules according to stochastic processes. Fig. I shows the fixed duty-cycling (a) and the stochastic scheduling (b). For example: sensors can follow a schedule with uniform distribution of range $[0, P_w]$ and $[0, P_s]$, where P_w and P_s are fixed values chosen by users. Sensors draw a random working period from $[0, P_w]$, and draw a sleeping period uniformly from $[0, P_s]$. This process continues sequentially. Because sensors schedule independently, they require no network-wide operations. On the other hand, sensors only need to exchange a small amount of messages to perform coordinations.

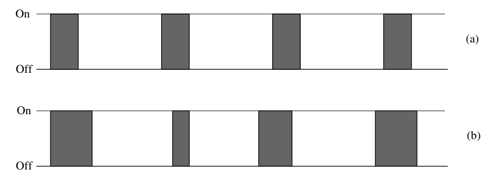


Fig. 1. Illustrations of fixed scheduling and stochastic scheduling. Shaded rectangles represent the working period. Fig. (a) shows the fixed scheduling. Fig. (b) shows the stochastic scheduling.

A crucial problem introduced by this design is how to control the excessive delay introduced by random scheduling. This paper studies the end-to-end delay distribution as the first step to address this problem. We assume a Possion traffic on all sensors in the network. We analyze the delay distribution of different stochastic processes, including uniform, exponential, and power law distributions. It turns out that, due to the redundancy in WSNs, stochastic duty-cycling provides a reasonably small delay compared with fixed scheduling. We then apply this design to the S-MAC [2] protocol to show its applicability. Another useful property of SSS is that it reduces the chance of sensors being attacked by adversaries [3], [4]. Adversaries need to scan the environment to find the location of the sensors, and then physically destroy them. Due to the limitation of the resources, attackers are only able to scan the network for a limited fraction of time or area. Therefore, by randomly choosing the working time, the probability of sensors being discovered is reduced.

II. Related work

Duty cycling for WSNs is an effective approach to save energy. Two aspects of the sleep scheduling have been under research in the past: the first is sensing module scheduling

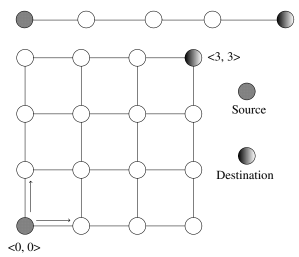


Fig. 2. Examples of regular networks: chain of nodes and grid.

for the detection of rare events. We refer to it as *sensing scheduling*. And the second is duty-cycled MAC protocols. The authors of [5], [6] designed distributed algorithms to minimize the detection delay of rare events. A more comprehensive scheduling approach is proposed in [7], which can achieve stochastically optimal detection under a temporally correlated sequence of event occurrences. In [8], the trade-off between energy consumption and event detection rate, is analyzed. A guaranteed delay wakeup scheduling algorithm is proposed in [9]. These works all use fixed scheduling, which incurs considerable control overhead. Synchronous MAC protocols, like S-MAC [2], [10], allow nodes to share schedules so that direct neighbors can communicate in a common time interval. On the contrary, asynchronous MAC has no restriction on the coordination between nodes, so that nodes need to keep idle listening, or continuously probing the channel.

III. The delay of stochastically scheduled WSNs

In this section, we analyze the end-to-end delay in regular networks using SSS. The regular networks are chains and grids of sensors. Fig. 2 depicts the examples. Note that we consider only the scheduling of the communication module of the nodes in WSNs. A node periodically changes its state to "working" and to "sleeping". The lengths of the working and sleeping periods of each node are determined by stochastic processes, whose parameters are specified by human administrators. We consider three distributions in this paper: uniform, exponential, and power law distribution.

A. Uniform distribution

1) Chain of nodes: First, we consider chains of nodes. The route from source to destination is a chain. Each node turns on and off independently. Thus, the expectation of the delay from Source to Destination in a n-hop chain is given by:

$$E(n) = \sum_{i=1}^{n} E_{i,d} \tag{1}$$

$E(n)$ is the expectation of the delay of a n-hop chain. Let us first consider the one-hop delay. As depicted in Fig. 2, two nodes are independently scheduled. Traffic is also independent to sleep scheduling. The expectation of delay of the i_{th}-hop is given by:

$$E_{i,d} = P_s \times E_{i,s} \tag{2}$$

where P_s is the probability that the next-hop node is asleep, and $E_{i,s}$ is the delay when the next-hop node is asleep. Recall

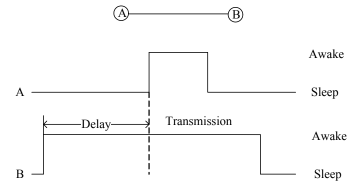
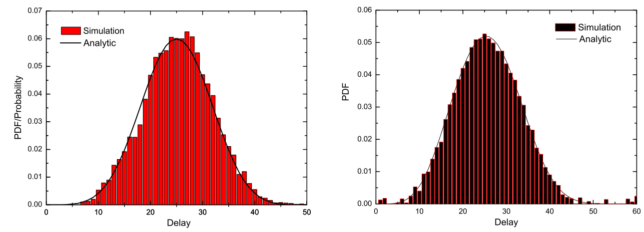


Fig. 3. One hop delay between two nodes in stochastic scheduling. A, the sender, needs to wait a period of time for B to awake.



(a) 5-hop chain (b) 6-hop grid

Fig. 4. PDFs of end-to-end delay in a Chain of 5-hop, and a Grid with 6-hop, with Uniform sleep time distribution.

that the sleeping period is much larger than the working period, thus, $P_s \approx 1$. Substitute 2 into 1, we get the expectation of the n-hop delay as:

$$E(n) = \sum E_{i,s} \tag{3}$$

We omit the expectations of the analytic and simulation results in the interest of saving space, and the results perfectly match. Now, we are ready to derive the distribution of delay from S to D. The Probability density function (PDF) of the n-hop delay is given by the convolution of all PDFs in each hop [11]:

$$F(t) = \prod_{i=1}^{n} f_i(t) = f_1 \otimes f_2 \otimes \cdots \otimes f_n(t) \tag{4}$$

$f_i(t)$ is the PDF of the delay of the i_{th}-hop on the chain. If each node on the chains has the same schedule, i.e. all the nodes follow uniform distributions with the same expectations of sleeping and working periods: $E(s)$ and $E(w)$, the result is a n-time self-convolution. We plot the results for this distribution. According to stochastic theory, Eq. 4 quickly converges to normal distribution. This is a nice property, which means that we can tightly bind the delay with a high probability.

2) Grid network: First, we need to introduce several definitions. Each node in the grid is identified by a $2D$ coordinate. The node in the left-bottom corner of the grid has coordinate $\langle 0, 0\rangle$. The distance between two nodes is the sum of the absolute differences of two components of their coordinates. The routing protocol in the grid is a 2D greedy routing, i.e. packets will be forwarded to the first node that is available, and have a shorter distance to the destination. In a grid network, the statistic multiplexing of the nodes' schedules make the one-hop delay smaller.

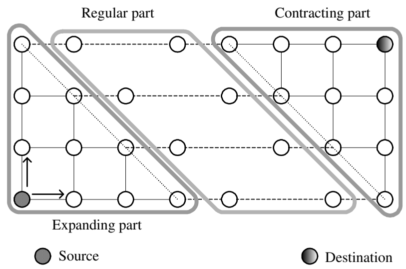


Fig. 5. Different groups in a grid network.

To extend the analysis to the grid, we need to specify the routing protocol. Greedy geometric routing is used in our analysis. In each node, the next-hop node is selected to reduce the distance to the destination, which is defined as the sum of the absolute difference on two axes. In this protocol, the distance from the current node to the destination decreases by 1 in each forwarding. Let us consider the one-hop case. A node has n neighbors as its next-hop to the destination. The delay of the message from source to next-hop (one of the n nodes) is $min\{D(A,B),...,D(A,C)\}$. We consider the case of only two next-hop neighbors. Assuming that all the neighbors schedule using the uniform distribution with the same parameters, the expectation of the one-hop delay in this case is $\frac{2\times E}{3}$, where E is the expectation of the sleep period of one of the next-hop nodes. The PDF is given by Eq. 5. In the general case where n next-hop nodes are present, the expectation of the one-hop delay is $\frac{2\times E}{n+1}$, and the PDF is given by Eq. 6. In Eq. 6, $F(t)$ is the cumulative distribution function (CDF) of the sleep time of all the next-hop nodes.

On the other hand, if all the next-hop neighbors have evenly-distributed schedules, the expectation of the delay for n such neighbors is $\frac{E}{n}$, which is half of that in the stochastic scheduling. However, it is proved in [12] that the optimal scheduling problem is, in general, NP-Hard. This means we are unable, in practice, to achieve such performance.

$$f(t) = \frac{2}{C} - \frac{2}{C^2}t \tag{5}$$

$$f(t) = (1-(1-F(t))^n)' \tag{6}$$

In Fig. 5, we show the arrangement of nodes on the potential routing paths, from the source to the destination. Nodes can be classified into three groups: expanding part, regular part, and contraction part. We denote the *level* as the set of nodes that have the same distance to the source (destination). We use a vector to represent the probabilities of nodes being selected as forwarder in a certain level. According to this definition, each part of the nodes has a unique pattern of spreading probability distribution between successive levels, which is expressed as transition matrices. Starting from the source, we can compute the probabilities of nodes being selected as forwarders in each level. In the expanding part, starting from the source, the i_{th} level has i nodes. The probability transition matrix from the i_{th} level to the $(i+1)_{th}$ level is a $i\times(i+1)$ matrix:

$$\mathbf{M_{i,E}} = \begin{pmatrix} 0.5 & 0 & \cdots & \cdots \\ 0.5 & 0.5 & 0 & \cdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0.5 & 0.5 \\ 0 & \cdots & \cdots & 0.5 \end{pmatrix} \tag{7}$$

In the regular part, each level has the same number of nodes, denoted as n. The transition matrix is an $n\times n$ matrix, and is the same for each level:

$$\mathbf{M_{i,R}} = \begin{pmatrix} 1 & 0.5 & \cdots & \cdots \\ 0 & 0.5 & 0.5 & \cdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0.5 & 0.5 \\ 0 & \cdots & \cdots & 0.5 \end{pmatrix} \tag{8}$$

The contraction part is similar to the expanding part, with the number of nodes in each level continuously decreasing. To simplify the notation, we denote the level of nodes as the distance to the destination. The transition matrix from the i_{th} to the $(i-1)_{th}$ level is an $i\times(i-1)$ matrix.

$$\mathbf{M_{i,C}} = \begin{pmatrix} 1 & 0.5 & \cdots & \cdots \\ 0 & 0.5 & 0.5 & \cdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0.5 & 0.5 \\ 0 & \cdots & 0.5 & 1 \end{pmatrix} \tag{9}$$

We need to sum the delays between levels of nodes to get the accumulative delay distribution from source to destination. In the expanding part, the delays between nodes in different levels are the same, i.e. the accumulative sum of two next-hop delay can be computed, just as that in the chain of nodes, where the delay of each hop is changed to Eq. 6, and the length of the chain is changed to the length of the expanding part.

$$D_{expanding} = \sum_{i=1}^{len(expanding)} M_{i-1,E} \times D(L_{i-1}, L_i) \tag{10}$$

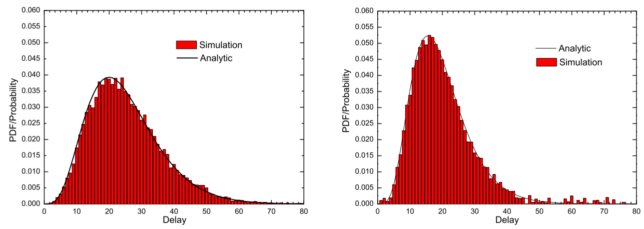
The next step is to compute the delay in the regular part. In the first level in the regular part, the probability distribution is computed by the analysis in the expanding part. From upper to lower, the vector to represent the probability in the vertical line is given by:

$$P = \begin{bmatrix} P_1 & P_2 & \cdots & P_n \end{bmatrix} \tag{11}$$

The initial probability of each node being selected is calculated by multiplying the matrix in the Eq. 7 several times, depending on the length of the expending part. The length of the expanding part and contract part are the same, which is $min\{D_x, D_y\}$. The length of the regular part is $D_x + D_y - min\{D_x, D_y\}$, or $max\{D_x, D_y\}$. Each time we need to calculate the distribution of the next level nodes:

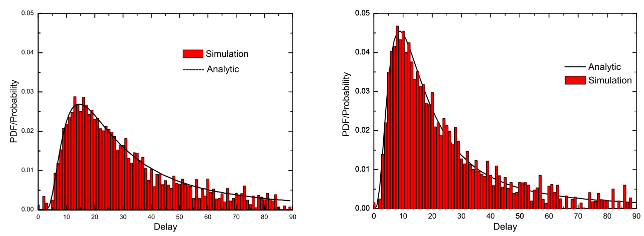
$$D_{regular} = \sum_{i=1}^{len(regular)} M_{i-1,R} \times P \times D(L_{i-1}, L_i) \tag{12}$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings



(a) 5-hop chain (b) 6-hop grid

Fig. 6. PDFs of the end-to-end delay in a 5-hop chain, and a 6-hop grid, with Exponential sleep time distribution.



(a) 5-hop chain (b) 6-hop grid

Fig. 7. PDF of end-to-end delay in a 5-hop chain, and a 6-hop grid with Power law sleep time distribution.

The analysis is similar for the contraction part. Two nodes on the border have one next-hop. It is computed using transition matrices:

$$D_{contraction} = \sum_{i=1}^{len(contraction)} M_{i,C} \times P \times D(L_{i-1}, L_i) \quad (13)$$

Summing all three parts, we can get the expectation of the delay and the corresponding PDF in Fig. 4. As we can see, because there are 2 next-hop nodes in the grid networks, the achieved delay is much better than that of the chain networks. This property is called *stochastic multiplexing* (SM). SM is the most prominent benefit of SSS in dense networks, which makes it practical compared to the synchronous scheduling methods.

B. Exponential distribution

The methods to compute the expectation and the PDF are the same as for uniform distribution. We do not want a sensor to work for a very long time, thus we need to assign a cut-off bound for the working period in scheduling. Nevertheless, as long as the ratio of working period and sleep period is small, the analysis is still applicable:

$$\sum_{T_i \in T} E_{T_i} \leq E \quad (14)$$

C. Power law distribution

We need to determine the exponent α first. Then, the scaling factor C. C is chosen to ensure the accumulative probability converges to 1; and the expectation is a predefined value. According to stochastic property, larger exponents let the distributions act more like an exponential one, which has a narrower distribution range. Here, in our discussion, α is 2. The domain of the distribution is in [1, $e^{expectation}$], and the scaling factor is $e^{expectation}/(e^{expectation}-1)$. In this setting, the expectation is chosen beforehand.

$$f(x) = C \times x^{-\alpha} \quad (15)$$

D. Energy consumption

Energy consumption is determined by the power p_w when sensors are working. The lifetime of the sensors is, in turn, determined by the ratio of the working time. Suppose we want the sensors to have a lifetime of at least T, and the initial energy of the sensors is E, the ratio γ of working in the entire time should be:

$$\gamma \leq \frac{E}{p_w \times T} \quad (16)$$

Sensors need to wake up to work for enough time to handle all the traffic in the network. Suppose that the rate of the messages is M, and the bandwidth of the physical layer is B, and the ratio of the working time should be long enough to transmit M messages with rate B. So, the ratio γ should be:

$$\gamma \geq \frac{M}{B} \quad (17)$$

E. Extension of the analysis to random networks

In this section, we extend the results in the above analysis to random networks. In the random networks, sensors are uniformly placed in a rectangular planar field. All nodes have the same transmission range. We try to find the expectation, and the PDFs of the delay between two nodes, which have a certain Euclid distance.

The routing protocol is the same as that used in Grid networks, where multiple routes to the same destination are stored at each sensor, and packets are forwarded to the first available next-hop node. The routes are computed using any shortest path algorithms. Unlike Grid networks, where the number of next-hop nodes is enumerable, it is only available through counting in Random networks for known source-destination pairs.

To make the analysis tractable, we first assume that a group of nodes are along the possible routes between a pair of sensors. This group of nodes are called a *forwarding set* (FS). A FS is divided into multiple levels, like that in the Grid networks. We further assume that all the nodes in a FS have the same number of next-hop nodes. Then, the number of levels of the FS between a source-destination pair is approximated based on the FS' shape, and the density of the network. The problem that comes about is finding the delay in this modified network. Due to its complexity, we find the parameters using experiments. We then get the results using the similar methods as the above analysis. The results are not presented here due to lack of space.

IV. APPLICATION OF STOCHASTIC SLEEP SCHEDULING

In this section we present a sensor MAC protocol incorporating SSS, which is based on S-MAC [2].

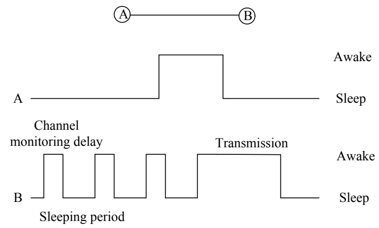


Fig. 8. Illustration of adaptive listening. Sender periodically wake up to check the availability of the receiver's channel.

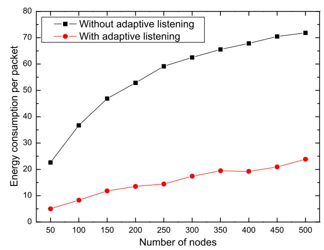


Fig. 9. The energy consumption in random networks with and without adaptive listening.

A. Stochastic sleep scheduling with adaptive listening

S-MAC [2] is an excellent MAC protocol for sensor networks with extremely-low duty cycling, adaptive listening, and synchronization functionality. We give a short introduction to Sensor MAC. Our sensor MAC protocol borrows the synchronization module of the original Sensor MAC protocol.

Each switch has a constant overhead. Given an overhead upper bound O, we can find the associated minimum cycle for the switching state. Then, we need to consider how to reduce the overhead in waiting for the available channel. When a node has packets to forward, it first checks the availability of the channel to the next-hop node enroute to the destination. Usually, the channel is unavailable due to a large sleep period. To avoid idle listening, we let the node fall asleep periodically with the sleep period computed from the sleep schedule of the next-hop node. Nodes only wake up for a very short time to send a beacon packet, then wait for the response, with a short timeout. To avoid missing the wake up time of the next-hop node, we need to find an adequate sleep period to make sure that the probability of detecting the awakening of the next-hop node is high. The channel assessment must persist for enough time to ensure accuracy. The time interval is set as the *one-hop round trip time* (RTT), between two neighboring sensors. RTT is updated during each transmission between two neighbors. Link-layer acknowledgment is used to ensure link-layer reliability. Therefore, whenever a new ACK is received, sensors will update their RTT values using a simple linear interpolation, as shown in Eq. 18. When the source detects that the channel is available, it will send a pilot signal to hold the channel to prevent the next-hop node from prematurely sleeping.

$$RTT_{new} = \alpha \times RTT_{old} + (1-\alpha) \times RTT_{current} \qquad (18)$$

B. Simulation results

Fig. 9 presents the energy consumption per packet of SSS with, and without, AL. AL essentially changes the idle waiting of senders to periodical wake-ups, thus, the energy consumption is just a fraction of the original design. Each time the nodes wake up to check the availability of the receiver's channel, they consume a small fraction of energy, which is determined by the one-hop RTT, and the energy consumption of the short pilot signal.

V. Conclusion and Future Work

In this paper, we studied the stochastic sleep scheduling for large scale WSNs. It guarantees the performance probabilistically. We analyzed the distributions of the delay of different distributions in regular and random networks. Simulation results verified our analysis. We also incorporated stochastic sleep scheduling into S-MAC, and introduced adaptive listening technique to further reduce the energy consumption. We believe that SSS brings flexibility in the design of duty-cycled protocols for WSNs, and may potentially provide efficient scheduling with low overhead. SSS may also be useful in improving the network's ability to counter-effect active attacks from the adversaries.

Acknowledgement

This work is supported in part by NSF grants CNS 0422762, CNS 0626240, CCF 0830289, and CNS 0948184.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, pp. 102–114, 2002.

[2] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. of IEEE Infocom'02*, April 2002, pp. 1567–1576.

[3] W. Gu, X. Wang, S. Chellappan, D. Xuan, and T. Lai, "Defending against search-based physical attacks in sensor networks," in *Proc. of IEEE MASS '05*, Nov. 2005, pp. 520–527.

[4] X. Wang, W. Gu, S. Chellappan, K. Schosek, and D. Xuan, "Lifetime optimization of sensor networks under physical attacks," in *Proc. of IEEE ICC'05*, May 2005, pp. 3295–3301.

[5] F. Wang and J. Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *Proc. of IEEE Infocom'09*, vol. 5, April 2009, pp. 3295–3301.

[6] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proc. of ACM IPSN'05*, April 2005, pp. 20–27.

[7] N. Jaggi and K. Kar, "Near-optimal sleep scheduling of a rechargeable sensor under temporal correlations," in *Proc. of Forty-fifth Annual Allerton Conference*, 2007, pp. 991–993.

[8] C. Sengul, M. J. Miller, and I. Gupta, "Adaptive probability-based broadcast forwarding in energy-saving sensor networks," *ACM Trans. on Sen. Netw.*, vol. 4, no. 2, pp. 1–32, 2008.

[9] R. Cohen and B. Kapchits, "An optimal wake-up scheduling algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 570–581, 2009.

[10] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *Proc. of ACM SenSys'06*. ACM Press, 2006, pp. 321–334.

[11] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Companies, 1991.

[12] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *Proc. of IEEE Infocom'05*, March 2005, pp. 2470–2481.