Using Network Traffic to Infer Power Levels in Wireless Sensor Nodes

Lanier Watkins, Johns Hopkins University Information Security Institute
Garth V. Crosby, College of Engineering, Southern Illinois University Carbondale
Afsana Sharmin, College of Engineering, Southern Illinois University Carbondale

Abstract—**In this paper we leverage the concept of information leakage to demonstrate the correlation between network traffic and available power levels in wireless sensor nodes brought about as a result of dynamic duty cycling. We show that this correlation can be used to remotely infer sensor node power levels. Essentially, our premise is that by determining the send rate of a wireless sensor node the current duty cycle mode and thus available power level of the node can be inferred.**

Our scheme, namely Power Efficient Path Selection (PEPS), is motivated by the fact that dynamic duty cycling attempts to streamline power usage in wireless sensor nodes by decreasing radio usage, which directly affects the node's network traffic send rate. PEPS is an enhancement to the shortest path algorithm that allows us to 1) reduce the volume of periodic messages since the energy level of neighboring nodes and their statuses are inferred rather than communicated via control packets, and 2) extend the lifetime of a wireless sensor network (WSN) through the selection of energy-aware communication paths. We demonstrate the performance and feasibility of PEPS through simulation and comparative study with the traditional Shortest Path algorithm. The results indicate significant energy savings and the extension of the lifetime of the wireless sensor network when PEPS is employed.

Index Terms—**wireless sensor nodes, resource characterization, power efficient routing, PEPS, energy aware routing**

I. Introduction

In previous work, we have demonstrated a correlation between network traffic and the state of a node's hardware. Previously, we used network traffic-based information leakage in [1-3] to infer CPU utilization, in [4] to infer memory utilization, and in [5] to infer CPU speed. In this paper, we proposed an energy aware protocol for dynamic duty cycled WSN where network traffic-based information leakage is used to infer battery power levels of neighboring nodes. Dynamic duty cycling [6-8] is not a new technology. Essentially, dynamic duty cycling saves power in a node by allowing nodes with different duty cycle levels to communicate. In our implementation, we allow nodes to change their own duty cycle level based on their internal power level; as a consequence, nodes with the fastest send rate have the most battery power level, and as their power level dissipates, the send rate of the node is lowered to preserve available power in the node. This process of lowering the network traffic send rate of nodes as their power dissipates continues until the battery power level is below operating levels. Therefore, by seeking out the nodes with the highest network traffic send rate (and the shortest path) and using them as multi-hop routes to the gateway, power-efficient path selection (PEPS) can be achieved. The benefits here are 1) increased path-level lifetimes, and 2) an elimination of the need for some periodic messages (i.e., health and status, energy level). PEPS enhances *shortest path routing* by taking into account the available power-level of a node before using it as a hop in a potential path. This is similar to *energy-aware routing* [9-11]; however, instead of passing messages to acquire energy/power level, PEPS requires nodes to infer the power-level of their neighbors by using existing network traffic sent during discovery.

Our contributions are i) exploring the correlation between network traffic and nodes' duty cycle, and ii) a novel approach to energy-aware routing in duty cycled WSN. The rest of our paper is organized as follows; in Section II we discuss the motivation for this work and in Section III we discuss related works. In Section IV we discuss our WSN model and our path level analysis approach. In Section V we discuss our experimental evaluation, and in Section VI we discuss the results. Finally, in Section VII we mention future work and conclude the paper.

II. Motivation

We believe that one way of achieving an energy (or power) efficient WSN is by allowing each node to dynamically duty cycle based on its internal power level, and to send all messages via the shortest and most energy efficient path. Doing this allow each node to optimize its radio usage such that it only communicates when it is absolutely necessary and does so via the most energy efficient route. This implies that a control framework must be in place to allow nodes with different duty cycles to communicate, and a control framework must also be in place to continuously propagate sensor node energy levels throughout the WSN. We are motivated by the effect that dynamic duty cycling has on a WSN. Unintentionally, dynamic duty cycling creates a correlation between network traffic and hardware state due to its native behavior of lowering power usage by decreasing packet send rate as a node's power level decreases. PEPS is possible because of this exploitable behavior.

We propose a solution that we believe make the aforementioned energy efficient WSN more feasible by solving the control message overhead problem created by the need for continuous synchronicity. Our scheme involves using the network traffic transmitted by dynamically duty cycled sensor nodes to infer the power level in each node instead of requiring control messages to propagate this information. Nodes can initially infer the power level of their neighbors during network discovery, and continuously during normal on-going data communications. Essentially, our method proposes message-less in-band control communication as opposed to the normal out-of-band message based communication used by most methods. This allows us to eliminate the need for a control framework to propagate power level information.

III. Related Works

A. Correlating Network Traffic and Hardware

In [1-4] the authors detailed a phenomenon they refer to as a delay signature. They observe this delay signature to exist in the network traffic of highly utilized nodes. Exhaustive experiments were developed and executed to amass an array of empirical data to characterize this delay signature. Ultimately, the authors used this delay signature to establish a correlation between the network traffic and the state of nodes' hardware. Also, the authors trace the cause of this delay signature to excessive context switching [1-3] and paging [4] in over utilized nodes. The authors exploit this phenomenon to develop a passive resource discovery algorithm for cluster computers. This body of work serves as our main motivation for investigating a correlation between network traffic and the duty cycle of wireless sensor nodes.

B. Shortest Path Routing in Wireless Sensor Networks

There have been several proposed improvements to the classical shortest path routing algorithm in wireless sensor networks. In [12], Banerjee *et al.* proposed a shortest path based geographical routing algorithm. K.S. Shrivaprakasha *et al.* [13] proposed an energy efficient shortest path algorithm. Their proposed algorithm, Energy Efficient Shortest Path (EESP), discovers an optimal path of minimum hops with maximum average node energy. The main difference with our protocol is that unlike [13] we infer node power levels based on duty cycling, hence we do not need any additional control packets to communicate energy levels among nodes. S. Lai *et al.* [14] proposed Fast Time-Dependent Shortest Path algorithm (FTSP), a shortest path algorithm for distributed asynchronous duty-cycled wireless sensor networks. However, no consideration was given to the remaining energy level of nodes' batteries, hence frequent node death (as a result of drained battery) may lead to network partitioning and ultimately shorter path/ network lifetime.

C. Energy-Aware Routing in Wireless Sensor Networks

The operation of wireless sensor networks depends on the cooperation of participating nodes. Since nodes have limited resources, one of the most important goals in designing efficient networks is minimizing the energy consumption in the network. There are several protocols and algorithms, which have been proposed for energy or power efficient routing in wireless sensor networks.

Curt Schurgers *et al.*[9] proposed a practical guideline based on the energy histogram and developed a spectrum of new techniques to enhance the routing in sensor networks. The Local Target Protocol (LTP) was introduced by Chatzigiannakis *et al.* [10]. LTP performs a local optimization to minimize the number of data transmissions. It is a hop by hop data propagation model. However, in faulty networks this protocol may behave poorly because of many backtracks due to frequent failure to find a next hop particle [11].

One of the earliest proposed energy efficient protocol is the Low-Energy Adaptive Clustering Hierarchy(LEACH) [15]. This is a clustering-based protocol that utilizes randomized rotation of local cluster base stations(cluster-heads) to evenly distribute the energy load among the sensors in the network. The use of clusters for transmitting data to the base station leverages the advantages of small transmit distances for most nodes, requiring only a few nodes to transmit the far distance to the base station.

In order to save energy different approaches such as multi-hop transmission technique [10] as well as clustering techniques [15] have been proposed. All such techniques do not address the possible overuse of certain sensors in the network. In multi-hop transmission toward the sink, the sensor nodes lying closer to the sink tend to be utilized exhaustively. Thus these sensor nodes may die out very early, resulting in network collapse, although there may still be significant amount of energy in other sensor nodes of the network. Similarly, in clustering techniques the cluster heads that are located far away with respect to the sink tend to utilize a lot of energy. Therefore, we investigate an alternate approach, based on the correlation of node traffic and power level in dynamic duty cycled WSNs, to develop a shortest path, energy-aware path selection protocol that preserves the lifetime of the network.

IV. Wireless Sensor Node Model and Path-Level Analysis

We attempt to model realistic attributes of a WSN such as energy dissipation due to transmitting over distances, receiving over distances, and dynamic duty cycling. This allows us to use a MATLAB computer model instead of hardware to test the feasibility of our method. We demonstrate our method by analyzing multi-hop paths chosen by our PEPS, and discussing how PEPS enhances shortest path routing.

A. Wireless Sensor Network Model

Our wireless sensor node model is patterned after the *First Order Radio Model* as proposed by [15]. In this model, a wireless sensor node is described numerically in terms of its k-bit amplified transmitter, and its k-bit receiver. We also include another attribute which was not considered in the *First Order Radio Model*. We emulate some operating system and/or Media Access Control (MAC) functionality by including dynamic duty cycling as motivated by [6-8]. Our final wireless sensor network model dissipates energy per round based on equation 2 for transmitting only, equation 1 for receiving and transmitting, and Table 1 (as motived by [16]) for dynamic duty cycling. In equations 1 and 2, P_s is the node's send rate, E_{TX} is the energy required by the transmitter to send or receive 1 bit, E_{fs} is the amplifier's free space dissipation energy per bit, *l* is the packet length in bits, and *d* is the distance between the two communicating nodes. In Table 1, E_{av} is the available energy in the node and E_o is the initial energy. Other parameters used in the model are listed in Table 2.

$$E_{transmit/receive} = l*P_S(2E_{TX} + E_{fs}*d^2) \quad (1)$$

$$E_{transmit} = l*P_S(E_{TX} + E_{fs}*d^2) \quad (2)$$

Table 1: Duty cycle modes and packet send rates.

Duty Cycle (%)	Effective Data Send Rate (kbps)	Energy Range
100	then $P_s = E_o(n)*10k$	if $E_{av}(n) >= 0.84* E_o(n)$
35.5	$P_s*0.355$	$E_{av} <0.84*E_o(n)$ and $>=0.68*E_o(n)$
11.5	$P_s*0.115$	$E_{av} <0.68*E_o(n)$ and $>=0.52* E_o(n)$
7.53	$P_s*0.0753$	$E_{av} <0.52*E_o(n)$ and $>=0.36* E_o(n)$
5.61	$P_s*0.0561$	$E_{av} <0.36*E_o(n)$ and $>=0.20* E_o(n)$
2.22	$P_s*0.0222$	$E_{av} <0.20*E_o(n)$ and $>=0.04* E_o(n)$

Table 2. Model parameters

Sensor Deploy area	100 x 100 m^2
n, # of nodes	100
Range, R	20m
E_o, initial energy	Random
E_{TX}, transmit/receive energy	50nj/bit
E_{fs}, free space energy	10 pJ/bit/m^2
P_s, packet send rate	Based on node energy
l, packet length	1000 bits
d, distance	Varies

Our model has aspects of wireless sensor node hardware and software (i.e., operating system, MAC). We utilize this model in the evaluation of PEPS.

B. Path-Level Analysis

We do not claim that the concept of using the available power within a node to make path selections is novel. This is evident in many energy-aware algorithms [9-11]. However, to the best of our knowledge, our approach (i.e., inferring the power level of neighboring nodes instead of sending control messages, and allowing nodes to use this inferred information to choose power-rich paths in the network) is novel. We propose an approach that: 1) allows the nodes to infer the power-level of their neighbors by sensing their network traffic send rate, 2) does not require additional overhead to send power level data, and 3) compute the most power-efficient path based on distance and the available power in the sensor nodes.

For the remainder of this section we explain the function of PEPS by analyzing an example. We assume an eleven node network (feasible for a BAN) setup which is randomly situated on a 30 x 30 grid where each node has a random initial power-level and uses dynamic duty cycling except for the gateway node (i.e., Node 8) which has unlimited power (since it may be connected directly to a power source). Again, PEPS is an enhancement to the shortest-path algorithm, which we will refer to as shortest path selection (SPS) from this point on. PEPS chooses paths that have a longer lifetime (in some cases significantly longer) than SPS and does not require nodes to periodically send out health and status or energy level messages. We compare and contrast the multi-hop paths chosen from each source node to the gateway by these two methods.

The paths chosen by PEPS for our eleven nodes example network are listed in Table 3 and the paths chosen by SPS are listed in Table 4. The connections between the nodes of this network are illustrated in Figure 1, and the paths chosen by both methods are illustrated in Figure 2. We only consider the paths of interest (i.e., multi-hop paths). We begin with the multi-hop path from node 1 to the gateway (i.e., Node 8), PEPS chose path 1-11-8 and SPS chose the same path. Next, PEPS chose path 3-11-8 while SPS chose 3-2-8, both paths produced the same lifetime but SPS' choice had a shorter path (i.e. in terms of actual distance and not just number of hops) . This behavior repeats with the choice of 9-11-8 by PEPs and the choice of 9-2-8 by SPS. Next, PEPS chose paths 5-10-8, 6-11-8, and 7-11-8 while SPS chose paths 5-11-8, 6-2-8, and 7-2-8 respectively. Each of the paths chosen by PEPS had a longer lifetime, but the nodes chosen by SPS had shorter path lengths. The two interesting points here are that: 1) PEPS chooses a longer lifetime path or the path with the same lifetime as SPS, and 2) PEPS either chooses the same path or a longer path than SPS. Note, the lifetime in this example is dependent on the randomly chosen initial power-level (i.e., E_o) per node and the threshold we chose for node death, which was 0.01. The important point is that due to the nature of PEPS, it chose each hop based on shortest distance to the gateway and most available power; whereas SPS simply chose the closest distance to the gateway. If there were any dead nodes in this example, it is possible that SPS may have chosen them. In such cases PEPS will require all nodes that drop below a certain power-level to send out a *dead node* message that notifies neighbors to route around the node if possible. Other energy aware algorithms may periodically send a *health and status* message, which would inform neighboring nodes of this scenario. As can be seen in Figure 2, for PEPS, node 11 seemed to form a *backbone* that connected the majority of the nodes to the gateway. This backbone (i.e., Node 11) is the closest to the gateway from the nodes in question and has the highest power-level. Node 2 seemed to play a similar role for SPS; however, no consideration of power level was taken into account. Another key point is that due to the nature of PEPS, it basically avoided using Node 2 as a backbone, mainly because it had a lower power level than Node 11.

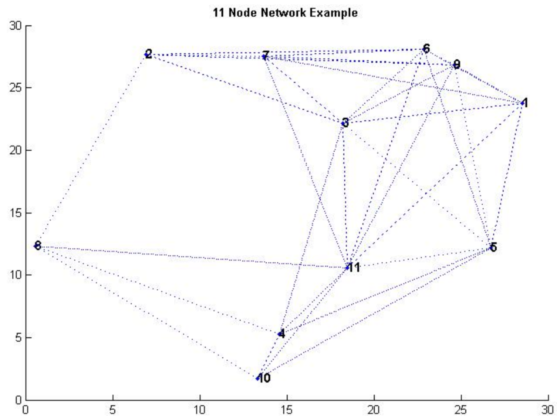
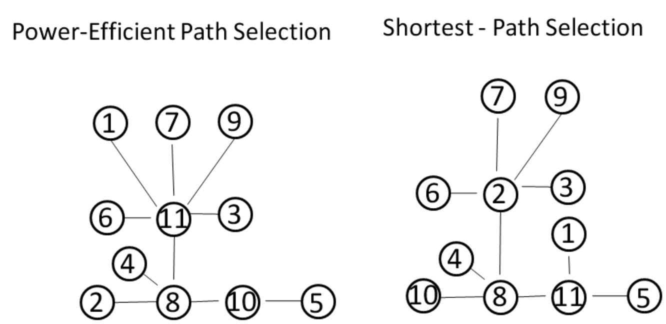


Figure 1. Example 11 Node Network



Node	1	2	3	4	5	6	7	9	10	11
Power Level	0.03	0.02	0.012	0.013	0.016	0.025	0.018	0.0011	0.026	0.022

Figure 2. Routing Example

Table 3. Power-Efficient Routing Example

Path	Path Length	Dead Node	Initial Power of Dead Node	Lifetime (Rounds)
1-11-8	34.5532	11	0.022	54
2-8				
3-11-8	29.5535	3	0.012	34
4-8				
5-10-8	33.572	10	0.026	61
6-11-8	36.0138	11	0.022	54
7-11-8	35.5707	11	0.022	54
9-11-8	35.3509	9	0.011	19
10-8				
11-8				

Table 4. Shortest-Path Routing Example

Path	Path Length	Dead Node	Initial Power of Dead Node	Lifetime (Rounds)
1-11-8	34.5532	11	0.022	54
2-8				
3-2-8	29.1643	3	0.012	34
4-8				
5-11-8	26.4169	5	0.0199	54
6-2-8	32.5533	2	0.02	50
7-2-8	23.3808	2	0.00098613	50
9-2-8	34.3475	9	0.011	19
10-8				
11-8				

V. Experimental Evaluation

In this section we describe the network model and the simulation of a dynamically duty cycled WSN coupled with energy aware routing. Additionally, we introduce the energy aware routing algorithm. Our method allows for the inference of the power level of sensor nodes, and uses this information to choose the shortest and most energy efficient path to the gateway.

A. Network Model

Let us denote a single sensor node as n_i and the set of all nodes by N, i.e. N=$\{n_1, n_2, n_3, \ldots n_k\}$. We assume the network to be an arbitrary connected graph, *G= (V, E),* of sensor nodes, where each vertex corresponds to a sensor node in the network. Edge (n_i, n_j) is in *E* if, and only if, n_i and n_j are one hop neighbors, that is, within each other transmission range. In our model, *G* is strongly connected, that is, any two non-neighboring sensor node $n_i, n_j \in N$ can communicate via multi-hop routing. This means that packets from the source to the destination are forwarded by intermediate nodes. Also, we assume that each sensor node, $n_i \in N$ stores a power-efficient shortest path table, which consists of the available power level in each node and the distance to each node. Our problem can be stated as follows. Find the shortest path to the gateway node, $n_g \in N$, corresponding to the maximum achievable path lifetime, $\forall n_i$.

We have made the following assumptions for our wireless sensor network.

1. Nodes are deployed in a two dimensional space. The area is 100 m x 100 m.
2. All nodes remain stationary after deployment.
3. Except for the gateway, all nodes are homogenous in terms of communication, and processing capabilities.
4. The energy dissipated by a node in transmitting and receiving a l-bit message over a distance *d* is given by equation 1 and the energy dissipated by a node in just transmitting a l-bit message a distance *d* is given by equation 2.

5. Each node is dynamically duty cycled, so nodes with the fastest packet send rates have the highest power-level. As nodes dissipate power, their network traffic send rate is reduced as illustrated in Table 1.
6. Each node is continuously sending data
7. Initially a random power-level is assigned to each node.
8. The lifetime is considered at the path-level and is defined as the number of rounds it takes for the first node along the path to die.

B. Algorithm: High level Description of Energy Aware Routing

//Routing Discovery
Compute Distance between $n_{i,j}$, $\forall\ i \neq j$
Determine n_i duty cycle, $\forall\ i$
//Create Routing Table
Sort duty cycle levels for $n_{i,}$, $\forall\ i$
Sort shortest path for $n_{i,j}$, $\forall\ i \neq j$
Select $n_{i,j}$ if $\forall\ i \neq j$, n_j *has highest duty cycle and shortest path*
Else select $n_{i,j}$ if $\forall\ i \neq j$, n_j *has highest duty cycle*
Else select shortest path
Break

C. Experiment Setup

We use MATLAB to simulate a scenario where a dynamically duty cycled 101 nodes WSN is randomly deployed on a 100 x 100 grid. The initial energy and initial duty cycle level are randomly chosen. The WSN has only one gateway (i.e., Node 100), which acts only as a sink with unlimited energy.

We assume that once these nodes are deployed, they immediately send out control messages to discover their neighbors. During discovery, each node infers the distance to its neighbors using the received signal strength and their power levels from the packet send rate of the received discovery messages using a wireless sensor node sniffer [17] or some other method. At the end of discovery, each node knows the distance to its neighbors and their available power levels (See Table 1.). This information allows each node to compute the most power-efficient and shortest path through the network to the gateway. Also, this information is assumed to be propagated throughout the network only once. Since nodes are dynamically duty cycled, their duty cycle mode changes with their available power level and thus their send rate changes as well. Any changes in the power levels of neighbors can be continuously inferred by determining the packet send rate from previous data sent by the neighbors. Also, once nodes die they are assumed to inform their neighbors by sending a *dead node* message and the neighbors can inform the rest of the network. Further, if the gateway has not heard from any node that has not announced its death within a specified timeframe it will re-initiate discovery and the updated set of live nodes will be re-discovered.

VI. RESULTS

In this section we discuss the results of applying PEPS to an emulation of a dynamically duty cycled WSN. We also demonstrate how PEPS enhances SPS via direct comparison. We also point out behavior of PEPS which demonstrates its feasibility in a real-world WSN. We begin this discussion by revealing that regardless of the size of the network chosen, there are only five possible outcomes in this comparison: 1) PEPS could chose a path with a long lifetime while SPS chooses a shorter lifetime path (Figures 1 and 2), 2) PEPS could chose a long lifetime path and SPS could choose a path with a node that dies the first round (Figures 3 and 4), 3) both PEPS and SPS could choose the same path with the same lifetime (Figures 5 and 6), 4) the user chosen start node could die the first round and no comparison can be done (Figures 7 and 8), or 5) the chosen node is directly connected to the gateway and no multi-hop path exists. Table 5 summarizes the chosen path examples and Figures 1-8 illustrate these path examples. The reason for the limited number of outcomes is because PEPS is an enhancement to the Dijkstra's shortest path algorithm. So regardless the size of the network, it should find the shortest path option (if one exists) per Dijkstra's algorithm coupled with our addition of the largest overall path lifetime; therefore, SPS should never choose a path with a longer lifetime than PEPS, unless the distance between nodes dominates the available power level in the nodes in question. In essence, PEPS provides energy aware path selection with about the same control message over-head as a shortest path method; however, more processing is needed by PEPS. It may be possible that other energy aware methods could choose paths with larger lifetimes; however, with the exception of PEPS all energy aware methods known to us require additional over-head control messages which lower the energy efficiency of the overall network, especially if the WSN is already burdened with control messages to manage dynamic duty cycling.

We are not aware of any research, which couples dynamic duty cycling with energy aware routing; therefore, there is no direct comparison that we can make. Therefore, we discuss PEPS in terms of its enhancements to SPS.

Our choice of a path-level analysis is in contrast to the traditional analysis approach, which treats the WSN like a black box, and its overall lifetime and other statistics are discussed. Our approach can be considered as taking a snapshot of one particular aspect of the WSN and analyzing it; therefore, to get an overall picture of the WSN one would simply take the aggregate of the perspective of all nodes in the network similar to the example, which was analyzed in Section IV-B. An overall picture of the 100 nodes WSN in our experimental section is illustrated in Table 5. Notice, the shortest and power efficient paths for all 100 nodes in the WSN fall into one of the five categories. The majority of the paths (i.e., Path 1, Path 2 and Path 3) reveal that PEPS adds energy awareness to SPS. The remaining 22 paths cannot be used for comparison, because they reflect either non-multi-hop paths or paths initiated with dead nodes.

Specifically, in Figure 3 notice for Node 99, PEPS routes around several of the nodes chosen by SPS in Figure 4, and by doing this, a much larger path lifetime is achieved. In the overall WSN, , there are several groups of paths that exhibit this same behavior just as there are several groups of paths that exhibit the same behavior as Paths 2 and 3. Examples are illustrated in Figures 5 -8. Since the nodes are static, this viewpoint (i.e., path) of Node 99 will not change unless one of the nodes in this path dies. Essentially, every node in the WSN will have a viewpoint relative to its location in the WSN, and nodes will simply forward their data to the neighbor in the direction of the shortest and most energy efficient path and each node will locally ensure that this goal is preserved. Very similar to the spanning tree in Figure 2a, PEPS will choose a spanning tree composed of the nodes with the highest power levels as the *backbone* and all other nodes will have a path to the gateway via this spanning tree across the *backbone*. It is primarily this behavior (i.e., ability to locate and utilize high power level paths within a dynamically duty cycled WSN) that demonstrates that PEPS is a feasible scheme for power efficiency in dynamically duty cycled WSNs.

Table 5. Summary of PEPS and SPS Comparison

		Power-Efficient		Shortest Path		
	Description	Lifetime	Length	Lifetime	Length	Number of Paths in Category
Path 1	PEPS > SPS	5068	81	25	78.8	16
Path 2	SPS Chooses Dead	1692	107.3	1	97.1	39
Path 3	PEPS = SPS	1851	41.1	1851	41.1	23
Path 4	User Chooses Dead	1	68.2	1	49.3	16
Path5	Direct Connection					6

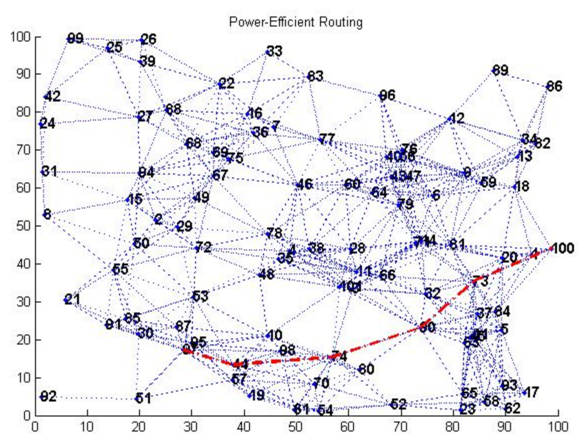


Figure 3. PEPS Example Path 1

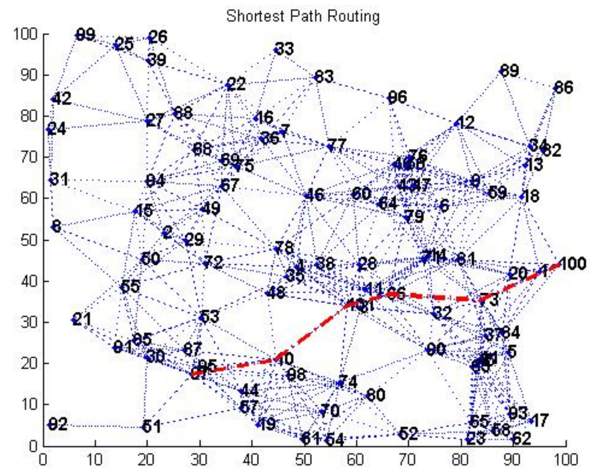


Figure 4 SPS Example Path 1

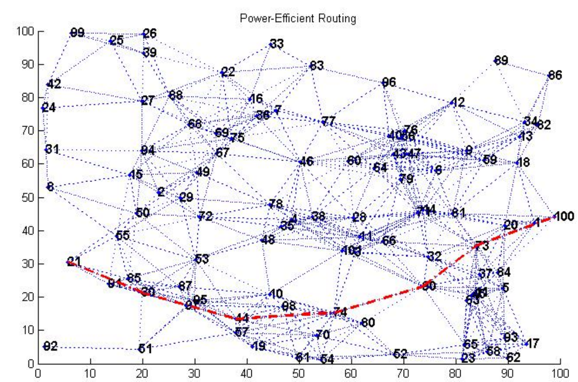


Figure 5. PEPS Example Path 2

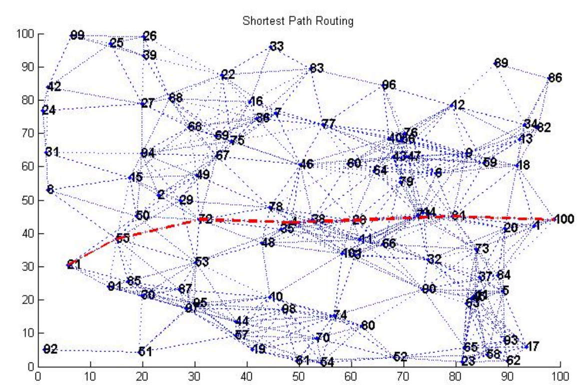


Figure 6. SPS Example Path 2

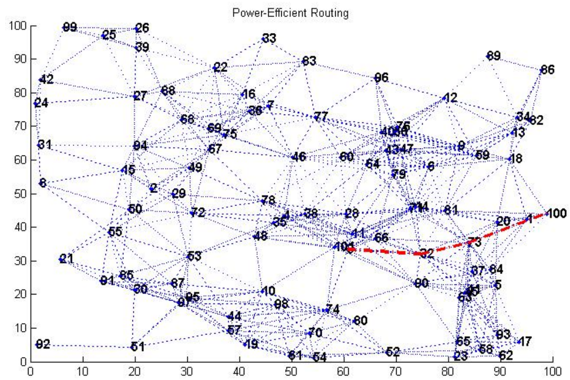


Figure 7. PEPS Example Path 3

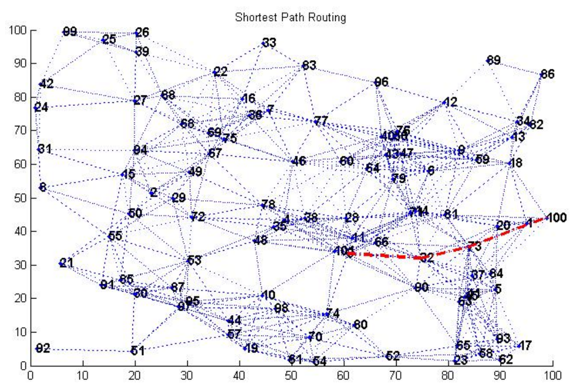


Figure 8. SPS Example Path 3

VII. CONCLUSION AND FUTURE WORK

The simulation results demonstrate the feasibility of developing a shortest path power efficient path selection scheme for dynamic duty cycled WSNs. The key feature in such a WSN is the fact that there is an exploitable relationship between the power level of the sensor nodes and their network traffic. In future work, we would like to study the effects of the delays caused by nodes dynamically duty cycling on the overall mission of the WSN. Also, we would like to implement PEPS using real wireless sensor nodes.

REFERENCES

[1] Lanier Watkins, RaheemBeyah, and Cherita Corbett, "Passive Identification of Under Utilized CPUs in High Performance Cluster Grid Networks," in the proceedings of IEEE International Conference on Communications (ICC), May 2008

[2] Lanier Watkins, RaheemBeyah, and Cherita Corbett, "Using Network Traffic to Passively Detect Under Utilized Resources in High-Performance Cluster Grid Computing Environments," in the proceedings of ACM International Conference on Networks for Grid Applications (GRIDNETS), October 2007.

[3] L. Watkins, W. H. Robinson, and R. A. Beyah, "A passive solution to the CPU resource discovery problem in cluster grid networks," IEEE Transactions on Parallel and Distributed Systems, vol. 22, pp. 2000-2007, 2011.

[4] Lanier Watkins, William H. Robinson, RaheemBeyah, "A Passive Solution to the Memory Resource Discovery Problem in Grid Computing," in IEEE Transactions on Network and Service Management (TNSM), 2010.

[5] L. Watkins, C. Corbett, B. Salazar, K. Fairbanks and W. H. Robinson, "Using Network Traffic to Remotely Identify the Type of Applications Executing on Mobile Devices," in the proceedings of IEEE Mobile Security Technologies (MoST), San Francisco, CA, May 23, 2013.

[6] B. Murray, T. Baugé, R. Egan, C. Tan, and C. Yong, "Dynamic Duty Cycle Control with Path and Zone Management in Wireless Sensor Networks," in IEEE Wireless Communications and Mobile Computing Conference, 2008.

[7] P. Lin, C. Qiao and X. Wang, "Medium Access Control With A Dynamic Duty Cycle For Sensor Networks", in IEEE Wireless Communications and Networking Conference, 2004.

[8] T. Dam and K. Langendoen, "An Adaptive EnergyEfficient MAC Protocol for Wireless Sensor Networks", in the ACM Conference on Embedded Networked Sensor Systems, 2003.

[9] Curt Schurgers, Mani B. Srivastava, "Energy efficient routing in wireless sensor networks" MILCOM Proceedings Communications for NetworkCentric Operations Creating the Information Force IEEE, 2001.

[10] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. "Smart dust protocols for local detection and propagation," Distinguished paper, in proceedings of the second ACM workshop on Principles of Mobile Computing-POMC'2002, pp-44-51.

[11] Azzedine Boukerche, Algorithms and Protocols for Wireless Sensor Networks , JohnWiley & Sons INC, 2009 pp 437-451.

[12] I. Banerjee, I. Roy, A. Choudhury, B. Sharma and T. Samanta, "Shortest Path Based Geographical Routing Algorithm in Wireless Sensor Network," in the proceedings of the International Conference on Communication, Devices and Intelligent Systems, Kolkata, India, December 28-29, 2012, pp. 262-265

[13] K. Shivaprakasha, M. Kulkarni, "Energy Efficient Shortest Path Routing Protocol for Wireless Sensor Networks", in the proceedings of International Conference on Computational Intelligence and Communication Systems, Bali, Indonesia, July 26-28, 2011, pp. 333-337

[14] S. Lai and B. Ravindran, " On Distributed Time-Dependent Shortest Paths over Duty-Cycled Wireless Sensor Networks," in the proceedings of IEEE INFOCOM, San Diego, USA, March 15-19, 2010 , pp. 1-9

[15] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in the proceedings of the 33rd Hawaii International Conference on system sciences-HICSS2000.

[16] Tiny OS Website July 2013: http://www.tinyos.net/tinyos-1.x/doc/mica2radio/CC1000.html

[17] TinyOS Website Accessed July 2013: http://tinyos.cvs.sourceforge.net/viewvc/tinyos/tinyos-2.x-contrib/tub/apps/PacketSniffer_802_15_4/