MEGA: An Energy Aware Algorithm for Self-Powered Wireless Sensor Networks in Sustainable Smart Infrastructure

Qi Dong[†], Yu Chen[†], Shahrzad Towfighian[‡]
[†]*Dept. of Electrical and Computing Engineering, Binghamton University, SUNY, Binghamton, NY 13902, USA*
[‡]*Dept. of Mechanical Engineering, Binghamton University, SUNY, Binghamton, NY 13902, USA*
E-mail: {qdong3, ychen, stowfigh}@binghamton.edu

***Abstract*—Smart infrastructure is attractive for possessing many desirable features, such as uninterrupted monitoring of health conditions, timely response to damages, and human-infrastructure interactions. Embedded sensors that collect information are critical for decision making. However, the lifetime of electronic sensors is a constraint to infrastructure lifetime if sensors are physically embedded in the infrastructure at construction time. In this paper, we studied a self-powered wireless sensor network that harvests energy from mechanical vibration in the environment. A dynamic, hierarchical algorithm called MEGA is proposed that constructs clusters and elects the cluster head based on residue energy and energy harvest rate. Taking a smart bridge as an application example, the simulation study has verified the effectiveness of the proposed protocol.**

***Keywords*-Smart Infrastructure, Self-Powered Wireless Sensor Networks (WSNs), Energy Harvesting Aware Protocol.**

I. Introduction

The rapid development and wide application of electronic sensors enable smart systems that are able to monitor, measure, analyze, communicate and act, based on information captured from sensors [13]. The capacity of information manipulation makes smart infrastructure able to respond intelligently to its environment, including natural environment changes, user demands, and interactivities among variant infrastructure. Not only it is expected to achieve an improved performance, but also enable the infrastructure to be more robust and safe by making appropriate decisions when facing disasters, attacks, or damages.

There are a plethora of reported efforts in developing smart infrastructure such as smart grid, smart water systems, smart transportation, and smart buildings [7], [22], [24]. However, the lifetime of electronic sensors could be a constraint to infrastructure lifetime if sensors are physically embedded into this infrastructure at construction time. For example, a bridge or building can be designed to last even a century. Therefore, smart infrastructure is expected to be able to efficiently harvest, manage, and use energy without depleting natural resources for a sustainable future.

Mechanical vibration is an attractive energy source due to its ubiquitous presence in bridges, airplanes, trains and other large-scale machinery [4]. Utilizing this abundant mechanical vibration energy to operate sensors offers a smart way to monitor the operational health of equipment and detect anomalies in performance. For example, the mechanical vibration of trains can be harvested power for temperature or vibration sensors to detect early signs of possible failure.

We consider a self-powered wireless sensor network (WSN) embedded in bridges for infrastructure health monitoring. When vehicles drive through the bridge, a resonator-based energy harvester converts the mechanical vibration energy into electricity. As illustrated in Figure 1, the self-powered sensor node consists of a nonlinear resonator based energy harvester. The stored energy then power the WSN for information collection. The nonlinear resonator has a broadband frequency range that suits harvesting from wideband environmental vibration [3], [11], [4]. The design of energy harvester and power conditioning will not be discussed in this paper due to the limited space.

In this paper, we focus on a novel WSN protocol that considers both residue energy and energy harvest rate. A simple finite state machine (FSM) is designed to denote the conditions of each individual sensor node. Coordinated by the FSM, an efficient maximum energy grade algorithm (MEGA) has been proposed to elect the cluster heads. Through extensive simulation, we have verified that MEGA scheme can tolerate the fluctuation of energy harvesting rate and statistically maintain a stable network topology.

The rest of the paper is structured as follows: Section

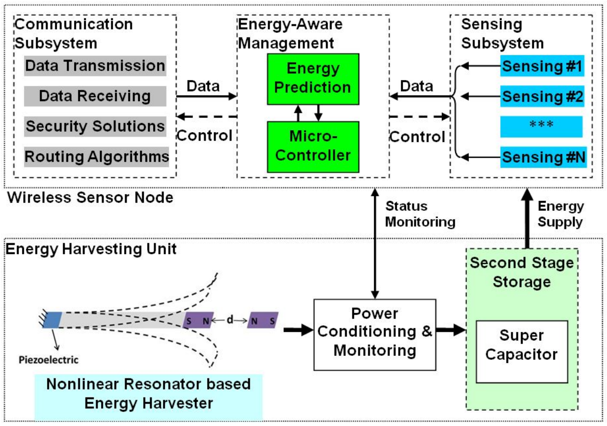

Figure 1. Block Diagram of the Self-Powered Sensor Node

II is the related work about energy-aware WSNs; Section III introduces the detailed design of our MEGA scheme; Simulation study is reported in Section IV; Section V wraps up this paper with discussions on our on-going efforts.

II. Related Work

Energy consumption is a critical concern in WSNs powered by embedded batteries. Considerable effort has been reported to extend network lifetime. The communication subsystem consumes approximately the same energy whether in the reception, transmission, or idle states [2]. It has been shown that transmitting a single bit requires as much energy as executing thousands of instructions [19]. Turning off the radio whenever possible is a widely accepted practice [26].

There are three major approaches to reduce energy consumption from the perspectives of data processing, transmission, and network management: duty cycling [6], data-driven [9], and mobility [15]. Duty cycling allows sensor nodes to switch off the radio transceiver as soon as it has no data to send and resume as soon as there is data ready. There are three types of sleep/wakeup scheduling algorithms [17]: 1) on-demand protocols; 2) scheduled rendezvous schemes; and 3) asynchronous schemes. Ideally, the on-demand protocol can maximize energy saving and incur minimum delay.

Data-driven approaches can improve the energy efficiency further by reducing the transmitted data or modifying the data acquisition. Data reduction methods save power by transmitting only useful data [9], such as in-network processing, data compression, and data prediction. Energy-efficient data acquisition techniques reduce power by only generating data points during periods of activity, rather than simply oversampling all of the time, such as adaptive and model-driven sampling [8],[1].

Balanced energy distribution is more critical to improve the overall network performance. Many energy-efficient protocols and algorithms have been proposed to address network setup, robust routing and security issues. There are plenty of energy-aware routing solutions, for example, multipath and data-centric routing [18], [23].

These energy conservation schemes can effectively extend the lifetime of WSNs, but this lifetime is still bounded and finite. The limited battery power supply is a major roadblock that prevents the use of WSNs for large-scale applications where long term, uninterrupted data collection is required. These challenges necessitate self-powered WSNs, which have the capability of harvesting energy from the environment. Energy-harvesting WSNs have been reported [14], [21], but most focus only on the energy- harvesting schemes [10] or power management algorithms [16], [25] instead of the whole WSN system. The proposed MEGA scheme is unique in that it optimizes energy efficiency while it considers energy harvesting to networking.

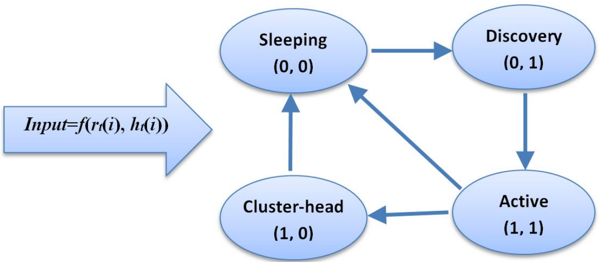

Figure 2. Finite State Transition Diagram

III. MEGA: Maximum Energy Grade Algorithm

This section presents the details of our MEGA scheme. A finite state machine (FSM) is introduced first, followed by design of our energy-aware cluster head election algorithm.

A. Nodes States Transferring

A FSM is proposed to denote conditions and tasks of each sensor node in different stages. Each node will decide to change the state based on its own energy level and energy-harvesting rate. Four states are defined using two bits: sleeping state (00), discovery state (01), active state (11), and cluster-head state (10). With periodically self-checking, nodes are aware of their own condition and environment, which enable them to switch states accordingly. The transition scheme is shown in Figure 2.

Initially, nodes are initialized with a certain amount of energy, and all of them are able to sense environment, build and maintain routing table, send or relay packets. Several nodes are randomly selected as cluster heads, which are in charge of data fusion and packets transmission at cluster head level. Due to the unbalanced energy consumption and energy harvesting efficiencies, nodes will transfer into different conditions. Those that have relatively higher energy harvesting rate or lower work load will maintain a high energy level. As time goes by, the energy level of some nodes will drop continuously. For instance, nodes that stay around cluster head are responsible for more packets relay, the nodes that stay in a versatile environment are required to generate and initiate more data, or nodes that stand at edge of the field harvest much less energy.

Once the energy level decreases below a preset threshold and the energy harvesting efficiency is not sufficiently high, a node will follow the inner state machine to switch to sleeping state. In most situations, cluster heads tend to consume much more energy than cluster members, because cluster heads are required to process packets and spend large amounts of energy to transmit over a long distance. When a cluster head fails to provide persistent service, it will send a cluster head campaign request packet before going to sleep. Each cluster member, which received the cluster head campaign request packet and originally belongs to this cluster, will start cluster head election process. One node will finalize this process by broadcasting its new cluster head status.

Sleeping state node will do nothing but harvest energy and self-condition check with its perception and transmission components turned off. When a sleeping node has accumulated enough energy and has a relatively steady energy harvesting rate, it will transit to discovery state in order to perceive the environment and build its routing table. Note that discovery state is only a transitional state. Once the node acquires knowledge about its surrounding and its energy level is not restricted, it enters active state. Active nodes are able to sense the environment, transmit data, and are qualified to campaign for a cluster head.

B. MEGA: Cluster Head Election Algorithm

In this section, we introduce maximum energy grade algorithm (MEGA), a simple but efficient method to elect cluster head. When nodes are capable of harvesting energy, residue energy is no longer the only factor to determine the cluster head. We also need to weigh the harvest ability and environment energy supply of each node.

At any given time, only nodes in active state are eligible for a cluster head campaign. A campaign process is triggered only when the current cluster head can no longer afford to continue its tenure as its residue energy has reached the lowest threshold and the harvest rate is too low to support cluster head working load. A new cluster head is needed before the current cluster head transits to sleeping state. A cluster head will broadcast an election request packet before it decides to sleep. This packet includes information of election request and a global timer T_{global} to begin the campaign. Each cluster member in this cluster should make a decision on whether or not to join this campaign. Unless it has adequate residue energy, a node will not join the cluster head campaign process. This process will continue until a new cluster head is elected.

When initiating a new cluster head campaign, several parameters are critical, such as residue energy, energy harvest rate, number of neighbors. Once cluster members receive a cluster head campaign request packet, they calculate a broadcast time $T(i)$ based on its current condition records. After $T(i)$ from the initiated global time T_{global}, node *K* will send a cluster head campaign response packet to the cluster head. When receiving a response packet, the cluster head will set notified state from 0 to 1, and transfer the cluster head position to this node by sending an authorized packet to it. Once the notified state is 1, the previous cluster head won't respond to any cluster with head campaign response packet, and will automatically go to sleep state. The node that received the authorized packet will set itself up as cluster head, and broadcast its cluster head status to the entire cluster. Meanwhile, other cluster members will terminate the cluster head campaign on receiving the cluster head broadcast packet, which includes information of previous cluster head and current cluster head. A fresh cluster cycle is initiated properly. Obviously, in this scheme, the node with smaller $T(i)$ is more likely to become the cluster head. Here we define $T(i)$ in MEGA as:

$$T(i) = \frac{1}{\alpha * r_t(i) + \beta * h_t(i) + \gamma * neighbor_t(i)} \qquad (1)$$

where $r_t(i)$ stands for current residue energy, and $h_t(i)$ stands for mean harvest rate of energy in a given period of time T_{range}, and $neighbor_n(i)$ is the number of direct neighbor of node *i*. α, β, γ are constant coefficients, which indicate the importance level of $r_t(i)$, $h_t(i)$ and $neighbor_n(i)$ respectively. Intuitively speaking, in different environment and network layout, these coefficients are quite different to reflect the different importance level of each variable.

$r_t(i)$, $h_t(i)$, and $neighbor_n(i)$ are key factors to determine $T(i)$. $r_t(k)$ and $h_t(k)$ determine how long the node serves as cluster head. Sufficient amount of residue energy guarantees the topology of this cluster will not be changed in a short time, and high energy harvest efficiency sustains the cluster nodes for a potentially longer time. $neighbor_n(k)$ is important because the relationships among nodes within the same cluster have great effect on energy consumption distribution of the whole cluster. A node positioned in a densely populated area is more likely to serve as a cluster head for a longer period. In contrast, in a sparsely populated region, the cluster head consumes more energy for inter-cluster communication. Additionally, around the cluster head is a many-to-one traffic pattern [20]. Nodes around cluster head are responsible for mostly packets relay among the cluster. More neighbor nodes means each neighbor has less burden on relaying packets and prolongs its awake period to achieve a more balanced energy distribution of the cluster. In addition, more nodes close to cluster head implies less average hop-count of each packet.

When there is not new cluster head elected successfully in one round of campaign, a campaign failure happens. If the requested cluster head receives no response packet within a round, for example, it happens when transmission channel is jammed, cluster head should start a new campaign request after a period of time and increase the round counter by one. A campaign failure may occur when $T(i1)$ and $T(i2)$ are too close to win the campaign for either of them. One solution to this problem is to increase the time interval between two broadcast times T. Intuitively, an initiated broadcast time $T(i)'$ is calculated as

$$T(i)' = T(i) * 2 - T_{const} \qquad (2)$$

where T_{const} is a constant that allows cluster members to avoid a long time to start broadcasting campaign packet.

C. Routing Algorithm

We consider *one knowledge* approach as the core feature of our design. As a sensor node is always equipped with very limited computation power and storage, we aim to reduce

Table I
ROUTING TABLE

Neighbor ID	One-Hop Cost	Overall Cost
ID1		
ID2		
...		

memory space utility to reduce the burden and prolong life cycle of nodes.

1) Routing Table Structure: As discussed above, cluster head election is partially based on the number of neighbors. More immediate neighbors leads to a greater opportunity of being elected as the cluster head. In this paper, a simple routing protocol is proposed, which takes advantage of the knowledge of neighbors.

All nodes maintain the same structure of routing table, despite different roles of cluster members and head nodes. As shown by Table I, the proposed routing table consists of only three data entries: ID of direct neighbors, the one-hop cost from the node to a neighbor, and the overall cost through this neighbor node to data sink. Consequently, no matter what the state a node stays, a routing table with the same format is used, only with different records in it.

To cluster members, there should be at least one path from source node to cluster head. If the cluster head is not located within the direct neighbor scope, the worst situation is that a node cannot find any available neighbor. It has to send packets directly to cluster head with possibly high energy costs. Cluster heads maintain the same routing table, but in a higher level. Their neighbors are peer cluster heads and the cost of the path to the base station.

2) Routing Table Generation and Maintenance: Initially, all nodes have empty routing tables by default and start routing discovery process. Each node broadcasts a routing discovery packet with certain transmitting power $P_{discovery}$. Each node decides whether or not to accept a neighbor by evaluating the signal strength power of the received discovery packet. Then the nodes calculate the one-hop cost according to $f(r_t(i), E_C)$, where $r_t(i)$ represents the current residue energy of its neighbor i, and E_C represents energy consumption on transmitting a fixed-length packet from neighbor to the node. ρ is a positive constance to reflect the importance of neighbers residue energy level, which is also dependent on current network conditions. Initially, the overall cost is set as infinity. In fact, the calculated one-hop cost is not the real cost to transmit a packet, but a relative cost used to construct the most efficient transmission path. The one-hop cost is calculated as

$$f(r_t(i), E_C) = E_C - \rho \cdot r_t(i) \tag{3}$$

Once clusters are formed, cluster members assess the overall cost by spreading this information from nodes near cluster head to the edge of the cluster. A cluster member that knows its own overall cost will broadcast an *acknowledge packet* to its neighbors, and nodes shall update their routing table after receiving the acknowledge packet. The overall topology is gradually built up in this manner. To maintain the routing table, nodes broadcast their routing information periodically. In each period, each node will receive acknowledge packets from its active neighbors. Packets carry information of the least costly path from neighbors to cluster head. The least costly path is defined as the path that has the lowest summation of one-hop cost and overall cost among all entries in routing table. This information will be recorded as the overall cost value in this node's routing table along with the corresponding neighbor ID and one-hop cost. When sending or forwarding packets, nodes will choose the neighbor in its least costly path as the relay to cluster head. Eventually, each routing path from cluster members to cluster head remains the lowest cost.

When node changes its state, routing table of this node and all cluster members should change accordingly. If a sleeping node resumes active state, it first transits to the Discovery state, such that the node constructs its routing table by monitoring acknowledge packet from its neighbors. After one or two periods, this node should have the complete picture of this cluster and is willing to transit from discovery state to active state and broadcasts its own acknowledge packet in the next period. Similarly, an active node will affect the whole cluster if it transits to sleeping state. A going-to-sleep node is required to broadcast a *sleepy packet*. Neighbors will delete it from their routing table on receiving this sleepy packet. It is possible that a sleepy node is on a least costly path. It will cause changes to the entire cluster. As a result, residual energy is a critical parameter on calculating cost because it is desired to keep these critical nodes alive as long as possible.

Routing table is managed differently when an active node has been elected as new cluster head. When a cluster head election process starts, all cluster members evacuate the obsolete routing table and compete for cluster head. Once a new cluster head is elected, a new cluster routing table initialization process is launched. The new cluster head obtains the cluster head routing table from the previous cluster head. Only some modification in cost is needed because changing cluster head within a cluster has little effect on network topology at high level, i.e., the cluster head level. Then, the new cluster head will further update its routing table at the next period and broadcast an acknowledge packet to its peer cluster heads in neighborhood.

IV. SIMULATION

A. Simulation Set Up

Our protocol is implemented using Network Simulator 2 (NS2) version 2.34, and we simulated a real situation in which 500 nodes are deployed on a 1500 meters long and 50 meters wide bridge. Base station (data sink) is installed

Table II
DESIGN PARAMETERS

Node number	500
Simulation time	400 $seconds$
Maximum harvest rate	4 mW
Start energy for each node	4 J
Communication Energy Consumption	50 nJ/bit [12]
Threshold to sleep	0.5 J
Threshold to wake up	3 J

at the end of bridge to receive and process packets from sensor nodes. Nodes deployed in the middle of the bridge can harvest more energy than nodes deployed in rear due to larger vibration amplitude in the middle of the bridge. We assume the energy harvesting efficiency follows Gaussian distribution, with maximum harvesting rate of $4mW$ [5] in the middle of the bridge and minimum of $2mW$ at the end of bridge. Without a simplified traffic model, we assume vehicles come in a constant rate. Table II shows the setting of major parameters.

Cluster heads are selected randomly at the beginning of the simulation. All nodes maintain their neighbor list as they will receive periodic broadcast packets from their neighbors. Meanwhile, nodes will sense environment to acquire information and transmit data to their neighbor. All intermediate cluster heads relay the data to base station by following the lowest cost path. Random generation of cluster heads won't cause much problem in simulation results because nodes are able to change state according to energy level and environment. If a node hasn't heard any advertisement from member nodes for a given period of time, it will take the role of cluster head. On the contrary, if a cluster head has a very small amount of members, it will give up the cluster head state and try to join another cluster as a member node.

B. *Experimental Results*

Several pivotal parameters are considered for different scenarios to test feasibility of our protocol. Cluster scope and neighbor scope are two of the most significant factors. Cluster scope determines the maximum distance between cluster members and reachable cluster heads. Neighbor scope defines the maximum distance between two cluster members or two separate cluster heads. Two metrics are selected, number of alive nodes and amount of sensed data. The first parameter reflects sustainability of nodes and the topology stability of the WSN. The second parameter describes the network efficiency on transmitting data.

As shown in table III, 10 different scenarios are adopted to examine our protocol. Cluster range in the scenario #1 is relatively small, so nodes can only communicate with close neighbors. From scenario #2 to scenario #8, scope for each cluster and maximum distance between cluster heads on transmission were changed. In the scenarios #9 and #10, the communication scope is increased to very large, allowing nodes transmit packets with less restriction.

Table III
SETTING FOR DIFFERENT SCENARIOS

Scenarios	Member Scope	Cluster Scope	Head Scope
Scenario 1	20 m	70 m	100 m
Scenario 2	40 m	70 m	100 m
Scenario 3	40 m	70 m	150 m
Scenario 4	40 m	100 m	100 m
Scenario 5	40 m	100 m	150 m
Scenario 6	40 m	100 m	180 m
Scenario 7	40 m	150 m	200 m
Scenario 8	40 m	500 m	500 m
Scenario 9	1000 m	1000 m	1000 m
Scenario 10	1500 m	1500 m	1500 m

1) Number of Alive Nodes: Nodes change their state based on their current energy condition, while cluster configuration has great effects on node conditions. Intuitively, node consumes more energy to transmit packet for longer distance. Our configuration restricts each node to transmit packets within a limited range for energy conservation. However, restricting transmission in too small a range is not an optimal decision. In scenario #1, each cluster manages very few cluster members, which actually increased cost. In addition, nodes are likely to find very few neighbors or cannot even find any neighbor with small transmission range restriction, which leads to some nodes not being able to find the next hop and cannot function in the network within a period of time.

Figure 3 reflects the number of alive nodes in different scenarios. As the figure shows, nodes tend to transmit packets consistently within a small transmission scope. For nodes with no limitation on transmission packets, as in Scenario #10, they will die very soon because broadcasting self condition at the beginning of the simulation consumes a large amount of energy. However, in Scenario #9, about half of the nodes can survive from beginning broadcasting process so they become stable in the following time. In other scenarios, some nodes will transit to sleeping state for small amount of residue energy. But the number of alive nodes becomes stable as sleeping nodes are able to harvest enough energy within a period and go back to active state.

2) Packet Transmission Rate: The ability of transmitting data reflects the efficiency of WSNs. In the MSGA scheme, when a node cannot build a path to data sink through a cluster head, no data packet is delivered. Therefore, the number of received data packets indicates how well the awake nodes are connected. We examined the number of delivered packets in the 10 different scenarios as Figure 4 shows. In most scenarios, packet flow in the network is consistent and persistent. With appropriate scope setting, network can achieve the best performance in a sufficiently long time. In an ideal situation, the whole network stays in a stable condition with a relatively fixed number of clusters and each cluster has a relatively fixed number of members. The network is in a dynamic balanced state. In the last

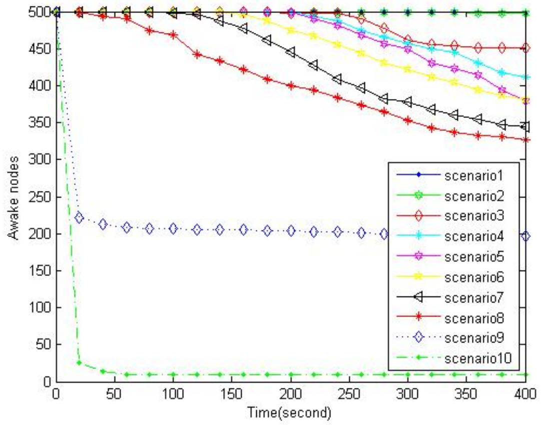

Figure 3. Nodes Awake State

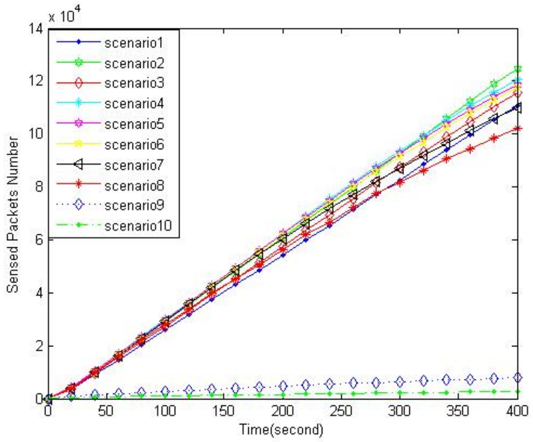

Figure 4. Data Transimission Ability

two scenarios, due to the limited number of awake nodes and limited transmission resources, the network can transmit only a small number of packets.

V. CONCLUSIONS

Self-powered wireless sensor networks can survive for a long time and relieve the constraints on the lifetime of smart infrastructure. In this paper, a novel routing protocol named MEGA has been studied, which is able to maintain a stable network topology and achieve good data throughput. Comparing to other existing energy sensitive wireless routing protocol, MEGA considers both the current energy level and the energy harvesting efficiency. Instead of time oriented (just as what LEACH did), MEGA takes an energy oriented approach, which enables MEGA to balance the energy distribution of the entire network, and reduce the overhead on forming clusters and choosing cluster heads.

While a smart bridge is considered as the case study, MEGA is also useful in monitoring buildings, machines, and wherever vibration exists. In addition, beyond mechanical vibration-based energy harvester, the MEGA protocol can be applied to any other self-powered WSNs that harvest solar, wind, or chemical energy [7], [24].

Actually, this paper merely deals with basic networking functions under a simplified energy harvesting situation. More questions are to be answered by our on-going efforts. A temporal opportunistic network model is being studied for a more thorough understanding of the network properties in variant energy harvesting environments. Also, we are extending our simulator to including factors such as reliability of the nodes, robustness facing malicious attacks, and simulate energy harvesting efficiency patterns.

REFERENCES

[1] C. Alippi, G. Anastasi, C. Galperti, F. Mancini, and M. Roveri, “Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications,” in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*. IEEE, 2007.

[2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[3] B. Ando, S. Baglio, C. Trigona, N. Dumas, L. Latorre, and P. Nouet, “Nonlinear mechanism in mems devices for energy harvesting applications,” *Journal of Micromechanics and Microengineering*, vol. 20, no. 12, p. 125020, 2010.

[4] A. Arrieta, T. Delpero, A. Bergamini, and P. Ermanni, “Broadband vibration energy harvesting based on cantilevered piezoelectric bistable composites,” *Applied Physics Letters*, vol. 102, no. 17, 2013.

[5] D. A. Barton, S. G. Burrow, and L. R. Clare, “Energy harvesting from vibrations with a nonlinear oscillator,” *Journal of Vibration and Acoustics*, vol. 132, no. 2, p. 021009, 2010.

[6] S. Bhattacharya, G. Xing, C. Lu, G.-C. Roman, O. Chipara, and B. Harris, “Dynamic wake-up and topology maintenance protocols with spatiotemporal guarantees,” in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 5.

[7] Y. Chen, C. M. Twigg, O. A. Sadik, and S. Tong, “A self-powered adaptive wireless sensor network for wastewater treatment plants,” in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*. IEEE, 2011.

[8] P. Clauwaert and W. Verstraete, “Methanogenesis in membraneless microbial electrolysis cells,” *Applied microbiology and biotechnology*, vol. 82, no. 5, pp. 829–836, 2009.

[9] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “In-network aggregation techniques for wireless sensor networks: a survey,” *Wireless Communications, IEEE*, vol. 14, no. 2, pp. 70–87, 2007.

[10] M. Gorlatova, A. Wallwater, and G. Zussman, “Networking low-power energy harvesting devices: Measurements and algorithms,” *Mobile Computing, IEEE Transactions on*, vol. 12, no. 9, 2013.

[11] A. Hajati and S.-G. Kim, “Ultra-wide bandwidth piezoelectric energy harvesting,” *Applied Physics Letters*, vol. 99, no. 8, p. 083105, 2011.

[12] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *Wireless Communications, IEEE Transactions on*, vol. 1, no. 4, pp. 660–670, 2002.

[13] N. Hoult, P. J. Bennett, I. Stoianov, P. Fidler, Č. Maksimović, C. Middleton, N. Graham, and K. Soga, “Wireless sensor networks: creating smart infrastructure,” in *Proceedings of the ICE-civil engineering*, vol. 162, no. 3. Thomas Telford, 2009, pp. 136–143.

[14] X. Jiang, J. Polastre, and D. Culler, “Perpetual environmentally powered sensor networks,” in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE, 2005.

[15] H. Jun, M. H. Ammar, and E. W. Zegura, “Power management in delay tolerant networks: a framework and knowledge-based mechanisms.” in *SECON*, vol. 5. Citeseer, 2005, pp. 418–429.

[16] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 6, no. 4, p. 32, 2007.

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

[18] S. Pattem, B. Krishnamachari, and R. Govindan, “The impact of spatial correlation on routing with compression in wireless sensor networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 4, p. 24, 2008.

[19] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000. [Online]. Available: http://doi.acm.org/10.1145/332833.332838

[20] S. K. Singh, M. Singh, and D. Singh, “Routing protocols in wireless sensor networks–a survey,” *International Journal of Computer Science & Engineering Survey (IJCSES) Vol*, vol. 1, pp. 63–83, 2010.

[21] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 3, pp. 443–461, 2011.

[22] M. Thomas, “Smart infrastructure: the future,” *www.raeng.org.uk/smartinfrastructure*, 2012.

[23] P. Thulasiraman, S. Ramasubramanian, and M. Krunz, “Disjoint multipath routing to two distinct drains in a multi-drain sensor network,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 643–651.

[24] S. Tong, M. Tong, C. M. Twigg, Y. Chen, and O. A. Sadik, “Energy efficient control for smart wastewater treatment plants using self-powered wireless sensor networks,” *Sensor Letters*, vol. 11, no. 9, pp. 1689–1694, 2013.

[25] C. M. Vigorito, D. Ganesan, and A. G. Barto, “Adaptive control of duty cycling in energy-harvesting wireless sensor networks,” in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON’07. 4th Annual IEEE Communications Society Conference on*. IEEE, 2007, pp. 21–30.

[26] A. Warrier, S. Park, J. Min, and I. Rhee, “How much energy saving does topology control offer for wireless sensor networks?–a practical study,” *Computer Communications*, vol. 30, no. 14, 2007.