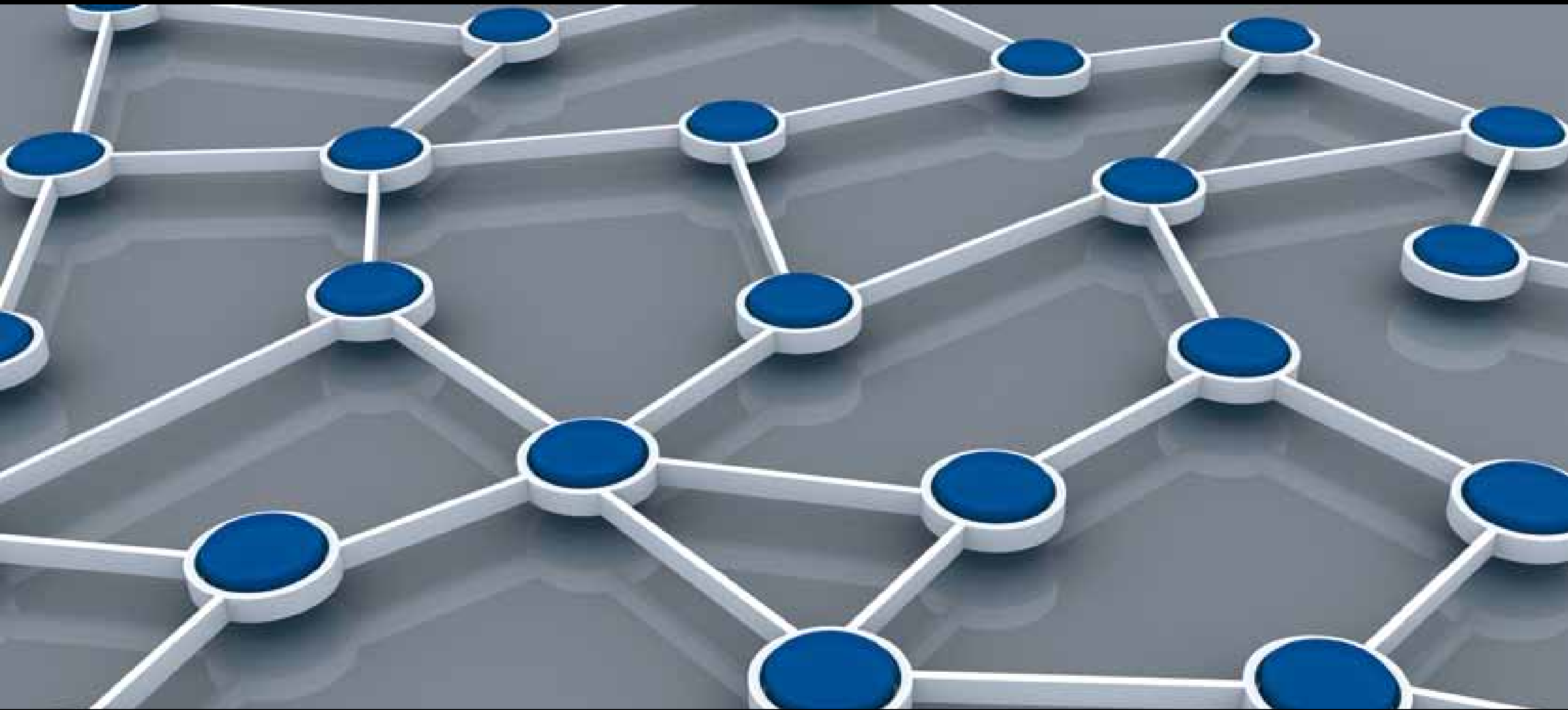
International Journal of Distributed Sensor Networks

Wireless Sensor Networks Based on Environmental Energy Harvesting

Guest Editors: Guangjie Han, Lei Shu, Al-Sakib Khan Pathan, Joel J. P. C. Rodrigues, and Abdelhamid Mellouk



Wireless Sensor Networks Based on Environmental Energy Harvesting

International Journal of Distributed Sensor Networks

Wireless Sensor Networks Based on Environmental Energy Harvesting

Guest Editors: Guangjie Han, Lei Shu, Al-Sakib Khan Pathan, Joel J. P. C. Rodrigues, and Abdelhamid Mellouk

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 816063, 2 pages
http://dx.doi.org/10.1155/2013/816063

Editorial

Wireless Sensor Networks Based on Environmental Energy Harvesting

Guangjie Han,[1] Lei Shu,[2] Al-Sakib Khan Pathan,[3] Joel J. P. C. Rodrigues,[4] and Abdelhamid Mellouk[5]

[1] *Department of Information & Communication Systems, Hohai University, Changzhou 213022, China*
[2] *Guangdong Petrochemical Equipment Fault Diagnosis Key Laboratory, Guangdong University of Petrochemical Technology, Maoming 525000, China*
[3] *Department of Computer Science, International Islamic University Malaysia (IIUM), 53100 Kuala Lumpur, Malaysia*
[4] *Instituto de Telecomunicações, University of Beira Interior, Rua Marques D'Avila e Bolama, 6201-001 Covilha, Portugal*
[5] *LiSSi Laboratory and Department of Networks and Telecommunications (IUT C/V), University of Paris-Est Créteil Val de Marne (UPEC), France*

Correspondence should be addressed to Guangjie Han; hanguangjie@gmail.com

Received 18 July 2013; Accepted 18 July 2013

With the development of wireless sensor networks (WSNs), more complicated and challenging applications now can be targeted by small and smart systems. Although most of these applications require little power, the gradual degradation of the batteries further decreases the useable lifetime of WSNs. Aside from relying on revolutionary development of battery technologies, one of the most important ways to improve the system lifetime is to "*harvest*" energy from ambient environmental energy sources. Energy harvesting devices efficiently and effectively capture, accumulate, store, condition, and manage this energy and supply it in a form that can be used to perform a helpful task.

There is in fact a constant need to design practical distributed and centralized algorithms and introduce novel theoretical models or evaluation methodologies to challenge various kinds of research problems originating from environmental energy harvesting for WSNs. Although there are a large number of developed applications using energy harvesting system power that are now practically available, there are still many problems that lead to considerable challenges in WSNs based on energy harvesting. To collate the recent advances and contributions from various perspectives on this issue, this special issue has been launched to publish a selection of high quality research papers. The objective is to push the theoretical and practical research forward for a deeper understanding in the fundamental algorithm, modeling, and analysis techniques of energy harvesting.

We had received a large number of manuscripts from all over the world, of which only few manuscripts could be selected for this special issue through a rigorous and thorough review process. These selected papers address different critical problems on different problems related to WSNs based on environmental energy harvesting. They are listed below.

(i) "*A Novel data classification and scheduling scheme in the virtualization of wireless sensor networks*" (E.-N. Huh and Md. M. Islam).

(ii) "*Battery-less 6LoWPAN-based wireless home automation by use of energy harvesting*" (A. M. Efendi et al.).

(iii) "*Thermoelectric energy harvesting for building energy management wireless sensor networks*" (W. Wang et al.).

(iv) "*A nonclairvoyant real-time scheduler for ambient energy harvesting sensors*" (H. E. Ghor et al.).

(v) "*On adaptive energy-efficient transmission in WSNs*" (N. Javaid et al.).

We are very hopeful that these papers will prove to be useful source of reference for both the researchers and the practitioners.

Acknowledgments

We are thankful to all those authors who considered submitting their works to this special issue, irrespective of whether their papers could be accepted or not. We are thankful to all the referees, who painstakingly reviewed the papers. Without their hard work and dedication, it would not have been possible to select these high quality papers within the given time limits of this special issue.

Guangjie Han
Lei Shu
Al-Sakib Khan Pathan
Joel J. P. C. Rodrigues
Abdelhamid Mellouk

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 546708, 14 pages
http://dx.doi.org/10.1155/2013/546708

Research Article

A Novel Data Classification and Scheduling Scheme in the Virtualization of Wireless Sensor Networks

Md. Motaharul Islam and Eui-Nam Huh

Department of Computer Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 446-701, Republic of Korea

Correspondence should be addressed to Eui-Nam Huh; johnhuh@khu.ac.kr

Received 25 February 2013; Revised 27 June 2013; Accepted 1 July 2013

Academic Editor: Al-Sakib Khan Pathan

Most of the nodes in a wireless sensor network (WSN) remain idle for the maximum period of their lifetime resulting in underutilization of their resources. There are many ongoing research studies to utilize the resources of sensor nodes in an efficient way. Virtualization of sensor network (VSN) is one of the novel approaches to utilize the physical infrastructure of a WSN. VSN can be simply defined as the virtual version of a WSN over the physical sensor infrastructure. By allowing sensor nodes to coexist on a shared physical substrate, VSN may provide flexibility, cost effectiveness, and manageability. This paper proposes a QoS-aware data classification and scheduling framework for VSN in the health care sector. We develop a tiny virtual machine called VSNware for health care applications, which facilitates QoS-aware forwarding of data packets, maintaining the reliability, delay guarantee, and speed. The simulation results also show that the proposed scheme outperforms the conventional WSN approaches.

1. Introduction

Recent advances in electronics have enabled the development of multifunctional smart sensor nodes that are small in size and communicate in an untethered manner over short distances. A sensor network consists of a large number of tiny sensor nodes that are densely deployed over a specific target area [1–4]. There is a robust deployment of WSNs in the health care sector because of their small size. Today's smart sensor node can efficiently monitor different vital signs such as the cardiac data, temperature, blood pressure, pulse rate, and saturation of peripheral oxygen (SPO_2) of a patient. In the health care scenario, applications demand different types of QoS requirements such as reliability, end-to-end delay, speed, and timeliness. There are many ongoing efforts to enhance the QoS issues of WSNs in the existing literature [5–7]. In this age of recession, providing QoS affordably in the WSN-based health care system is a big challenge for the increasing worldwide elderly population, which is the largest demographic group in the developed countries. For this very reason, researchers are searching for cost-effective ways to support QoS in WSNs for the health care sector.

Very recently, network virtualization has created a resonance among the network research community. The concept of sensor virtualization has also attracted a great deal of attention from industry and academia [8–10]. Virtualization of sensor network (VSN) can be defined as the separation of functions for the traditional wireless sensor network (WSN) service provider into two parts: the sensor infrastructure provider (SInP), which manages the physical sensor infrastructure, and the VSN service provider (VSNSP), which develops VSN by aggregating the resources from multiple SInPs and offers services to the application-level users (ALUs).

The WSN virtualization renaissance has originated mainly from the realization that most of the sensor nodes remain idle for most of the time in a WSN. Virtualization is one of the best ways to utilize the physical sensor infrastructure. VSN can provide a platform upon which novel sensor network architectures can be built, experimentally tested, and evaluated. In addition, virtualization in WSNs is expected to provide a clean separation of services and infrastructure and to facilitate new ways of doing business with sensor network resources among multiple service providers and application-level users [8].

In this paper, we propose QoS-aware data classification and a scheduling framework for the health care system in VSN. The sensor node senses parallel data and forwards it to a nearby node or gateway node. The gateway node classifies the data as urgent, suspicious, moderate, or normal. The classified data are passed through the decoding module and are queued up in the VSN queue. Finally, the scheduling module sends data to a specific path based on the priority, reliability, and delay requirements of the data packets.

The main contributions of this paper are as follows.

(a) A business model of the virtualization of sensor network is proposed.

(b) A tiny virtual machine called VSNware for health care applications has been developed for QoS-aware data packet forwarding.

(c) Packet classification and scheduling mechanisms are suggested.

(d) A detailed probabilistic analytical model of the reliability and delay for different traffics is proposed.

(e) Finally, the simulation results of the proposed scheme are presented with respect to other approaches.

The remainder of the paper is organized as follows. Section 2 reviews the background related to the conventional WSN, virtual sensor network, VSN, and related works. In Section 3, we discuss the detailed architecture of the sensor nodes and the sensor gateway router. Section 4 describes the VSN network model. Section 5 states the classification and scheduling of data packets. Sections 6 and 7 discuss a detailed mathematical model of the delay and reliability for different data traffics. Section 8 presents the performance evaluations and simulation results. Finally Section 9 concludes the paper.

2. Backgrounds

VSN is a brand-new research approach in the field of WSN. Before proceeding further, we need to clarify few basic concepts and the difference between traditional WSN, virtual sensor network, and VSN. In this paper, VSN means virtualization of a WSN as defined in the Introduction and in Section 2.3. The term VSN in this paper is synonymously used for the process of virtualization of a sensor network and for the network that supports virtualization.

2.1. Traditional WSN Approach. Traditional wireless sensor network consists of a large number of sensor nodes that are densely deployed either inside the phenomenon of interest or very close to it [1]. A sensor node senses its surrounding environment, performs necessary computation and processing, and sends the sensory data through multihop or directly to the coordinator node. The coordinator node may be a fixed node or a mobile node capable of connecting the sensor network to an existing communication infrastructure where a user can access the reported data. By integrating sensing, signal processing, and communication functions, a traditional sensor network provides a natural platform for hierarchical information processing [2–4]. The traditional WSN is dedicated for the monitoring of a particular event. But in VSN environment, the same infrastructure can be used by multiple stack holders.

2.2. Virtual Sensor Network. The virtual sensor network consists of a collaborative wireless sensor network. It is formed by a subset of sensor nodes of a wireless sensor network, with the subset being dedicated to a certain task or an application at a given time [11, 12]. In contrast, the subset of nodes belonging to the virtual sensor network collaborates to carry out a given application at a specific time. A virtual sensor network can be formed by providing logical connectivity among collaborative sensor nodes. Nodes can be grouped into different virtual sensor networks based on the phenomenon they track or the task they perform. The virtual sensor network protocol should provide the functionality for network formation, usage, adaptation, and maintenance of a subset of sensors collaborating on a specific task [13].

2.3. VSN and Its Business Model. Unlike the conventional WSN, the VSN environment has a collection of multiple heterogeneous sensor nodes that coexist in the same physical space. VSN is a type of network that creates a virtual topology on top of the physical topology of a traditional WSN. SInP in Figure 1 deploys different sensor nodes. In the traditional WSN infrastructure, the provider and the service provider are the same entity. VSN differentiates between the infrastructure provider and the service provider, thus providing a business model of true virtualization.

SInP deploys sensor network resources. It offers resources through programmable interfaces to different VSNSPs. Different interest groups can deploy sensor nodes and can make individual infrastructures, which can be used by the VSNSP to run individual applications. The VSNSP gets resources from multiple SInPs to deploy VSNs by sharing the allocated virtualized network resources to offer end-to-end application user services. The VSNSP can obtain resources from multiple SInPs. ALUs in the VSN model are similar to those of the existing WSN, except that the existence of multiple VSNSPs from competing SInPs provides a wide range of choices. Any end user can connect to multiple VSNSPs from different SInPs to use multiple applications.

2.4. Related Works. Currently there are few approaches in the WSN [5, 11–16] that focus on the virtual and overlay sensor network rather than the purist view of the VSN approach introduced in this paper. Table 1 summarizes a few of the research projects that act as the background of the proposed research approach. It demonstrates the contemporary research direction in the field of virtualization of sensor networks in general.

Recently, the Federated Secure Sensor Network Laboratory (FRESnel) has aimed to build a large-scale sensor framework. The goal of this project is to offer an environment that can support multiple applications running on each sensor node [14–16]. It provides an execution environment that hides the system details from the running applications. The system operates in a shared environment. The key characteristics of this approach are a virtualization layer that is running on each sensor node and provides abstracts access to sensor resources, which allows the management of these resources through policies expressed by the infrastructure

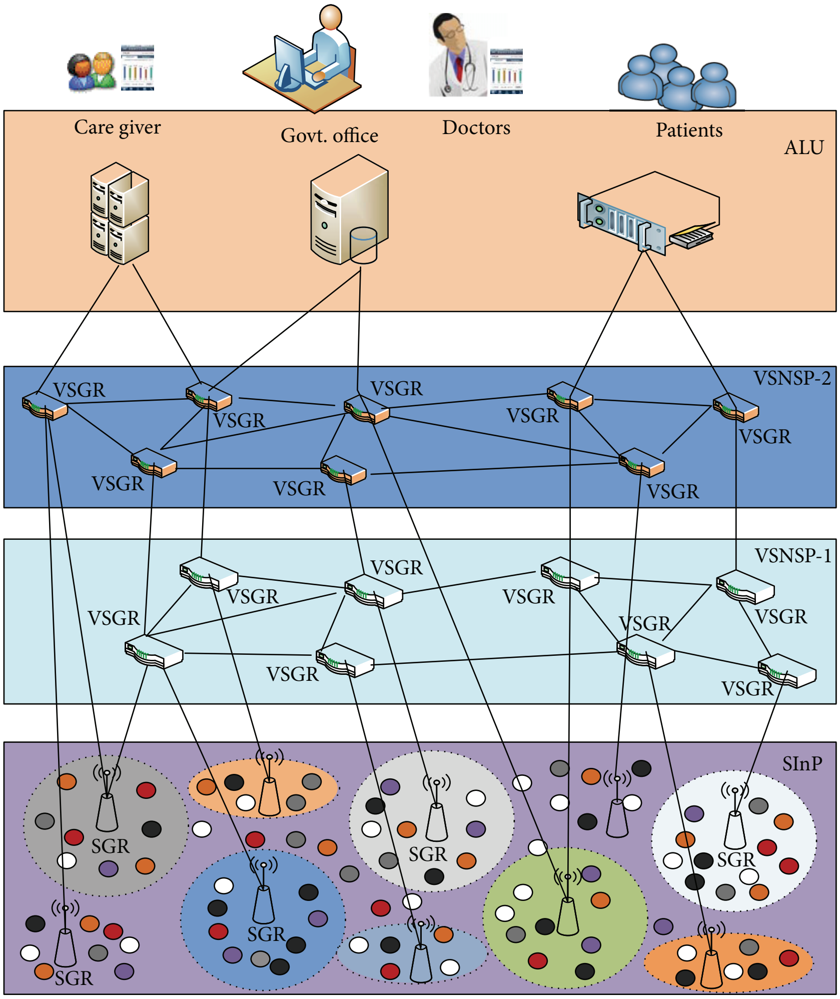


Figure 1: Business model for VSN.

Table 1: VSN research-related projects.

Projects	Research area	URL
FRESnel	To build a large-scale federated sensor network framework with multiple applications sharing the same resources	http://www.cl.cam.ac.uk/research/srg/netos/fresnel/index.html
VSNs	Random routing, virtual coordinates, and VSN support functions	http://www.cnrl.colostate.edu/Projects/VSNs/vsns.html
Sensor Planet	Nokia-initiated cooperation, a global research framework, on mobile device-centric large-scale wireless sensor networks	http://research.nokia.com/page/232
ViSE	Virtualization of sensor/actuator system, creating customized virtual sensor network test beds	http://groups.geni.net/geni/wiki/ViSE
DVM	To build a system that supports software reconfiguration in embedded sensor networks at multiple levels	http://nesl.ee.ucla.edu/project/show/51
SensEye	Multitier multimodal sensor networks	http://sensors.cs.umass.edu/projects/senseye/
SenQ	Complex virtual sensors and user-created streams can be dynamically discovered and shared	http://www.cs.virginia.edu/wsn/sensornets.html
WebDust	Multiple, heterogeneous, wireless sensor networks can be controlled as a single, unified, virtual sensor network	http://ru1.cti.gr/projects/webdust/

owner. A runtime environment on each node allows multiple applications to run inside the sensor node. It also provides policy-based application deployment that enables multiple applications to be deployed over the shared infrastructure. In MMSPEED [5], a novel packet delivery mechanism for QoS provisioning was proposed. It provides QoS differentiation in terms of two qualities, such as timeliness and reliability. This approach is based on multiple logical speed layers over a physical sensor network that is based on the conventional virtual sensor network. Based on the speed, it considers different virtual overlays. For virtual layering, it employs virtual isolation among the speed layers. This is accomplished by classifying the incoming packets according to their speed classes and then placing them into the appropriate priority queues. SenShare [15] is another platform that attempts to address the technical challenge of supporting multiple corunning applications in the sensor node. Here each application operates in an isolated environment consisting of an in-node hardware abstraction layer and a dedicated overlay sensor network. Instead of using a virtual machine, SenShare uses a hardware abstraction layer. It is a set of routines in software that emulates some platform-specific details, thereby giving programs direct access to the hardware. The Mate [17] and Melete [18] systems are based on the virtual machine approach that provides reliable storage and enables the execution of concurrent applications on a single sensor node. The VSN approach proposed in this paper is based on the Mate and Melete systems. This modified version of virtual machine is called VSNware. VSNware provides an environment to support different applications for health care systems such as cardiac data, blood pressure, blood sugar, and temperature sensing. VSNware helps to provide the purist view of the virtualization concept. It does so by separating the SInP and VSNSP as discussed in the previous sections. By applying the purist view of virtualization in VSN, this scheme can be efficiently used in the health care system, which is the main contribution of this paper. To the best of our knowledge, no previous research article has explored the VSN approach to design a ubiquitous health care system for the QoS-based vital data classifications and scheduling scheme.

3. Architecture of VSN

In this section we are going to introduce the detail architectural design of the proposed VSN approach and the description of its individual components elaborately.

3.1. System Architecture. Here we briefly describe the detailed system architecture and the software architecture of the sensor node and sensor gateway router (SGR). In the following sections we will explain the architecture in detail. The system architecture consists of three layers: the SInP, VSNSP, and ALU. The software architecture describes the virtualization of the individual sensor nodes and gateway router.

3.1.1. SInP. SInP consists of different sensor nodes. These sensor nodes sense different vital signs of the patient, such as the temperature, heart rate, blood pressure, and blood sugar. To sense the patient body in the VSN environment, we consider two types of sensor node: the fully functional device (FFD) and the reduced functional device (RFD) sensor nodes. SInP deploys sensor nodes in the hospital in a distributed manner. Each group of sensor nodes is divided into different logical areas, which are identified by circles to indicate the SGR domain. Each SGR domain may consist of one or more SGRs, which is an FFD sensor node. Each SGR supports sensor virtualization. In each domain there are many RFD sensor nodes that sense vital signs. The RFD is more resource-constrained than the SGR/FFD.

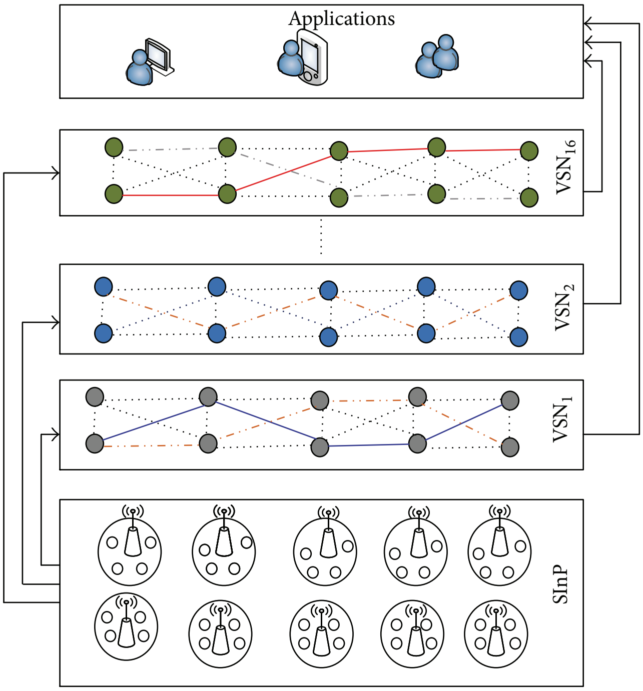


Figure 2: VSN architecture.

3.1.2. VSNSP. The VSNSP consists of many virtual SGRs (VSGRs), which are the virtual representations of the processing, storage, and other resources of the SGRs. The links between the VSGRs are the dynamically allocated channels between the SGRs. Since the VSN scheme is based on the IEEE 802.15.4 radio specification, it has 16 channels. Each channel is considered to be an individual path that consists of multiple links between SGRs. Each VSN provides a specific application service to the users. In Figure 2, we depict up to a maximum of 16 VSNs provided by a specific VSNSP as the underlying SInPs can support.

3.1.3. ALU. This layer consists of different application level users such as doctors, nurses, patients, or any other specialized users. Based on the application requirements, the ALU sends a request to the VSNSP. The VSNSP then maps the particular VSN according to the request of the specific application. Individual applications may use multiple VSNSP

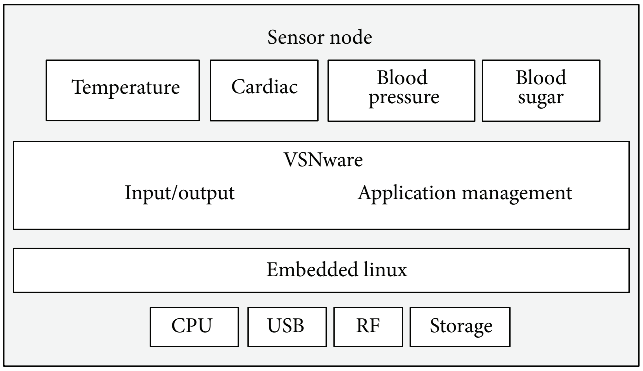


Figure 3: Architecture of sensor node.

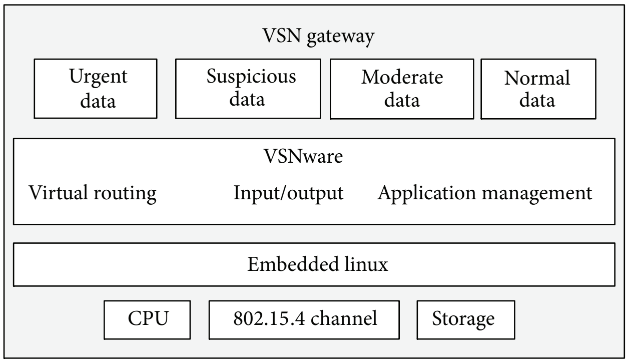


Figure 4: Architecture of SGR.

resources. The user may be a machine in the case of machine-to-machine communication, which can involve individual computers and any other smart device.

3.2. Software Architecture for Sensor Node and SGR. Figure 3 depicts the software architecture of a sensor node. It senses vital signs from patients in a health care system. A single sensor node performs multiple sensing tasks by using physical infrastructure virtualization as a service. The typical sensor node architecture consists of a physical layer, an operating system (OS) layer, a virtualization layer, and multiple sensing service layers. The lower layer consists of the physical sensor resources such as a central processing unit (CPU), USB module, RF module, and storage module. Layer 2 consists of a typical multitasking sensor network operating system. We use Embedded Linux in this case. Embedded Linux provides the environment to host the virtualization layer. The virtualization layer supports concurrent service execution. The virtualization layer includes the input/output and application management modules. Finally, the application layer runs multiple services over the virtualization layer, such as the sensing of temperature, cardiac data, blood pressure, and blood sugar.

Figure 4 depicts the detailed architecture of the SGR. The SGR is one of the key components in the overall VSN architecture for the health care system. An SGR is a fully functional sensor node that supports the concurrent processing of multiple applications. It also consists of a physical layer, sensor network operating system layer, VSNware layer, and application layer. The lower layer consists of the physical sensor resources, such as the central processing unit (CPU), RF module, and storage module. The sensor operating system layer consists of a typical multitasking sensor network operating system. In this model we use Embedded Linux as it is used in the individual sensor nodes. It provides the environment to run the VSNware. VSNware supports concurrent execution of the applications. VSNware consists of different modules such as forwarding/routing in VSN, input/output, and application management. Finally, the application layer provides the classified data, such as urgent, suspicious, moderate, and normal, over different VSNs based on the reliability and delay requirements.

4. Network Model of VSN

In this section we propose the network model of VSN based on graph theory. We consider a densely deployed large-scale and heterogeneous wireless sensor network in which N sensor nodes and M sensor gateway routers are uniformly distributed. The nodes and SGRs may determine their geographical locations. In fact, networking in such a WSN is very dynamic and differs from a traditional wired network. A node in a WSN is very tiny and consists of a small processing and storage unit. Since VSN is based on the existing SInP, it inherits most of the properties of a WSN. A link in VSN is the different channels used in a WSN. We use IEEE 802.15.4, which has 16 channels. We describe the network model of a WSN by using the graph theory that follows the procedure discussed in [19, 20]. We also discuss the VSN node and VSN link embedding. Virtual node embedding in VSN is like the conventional network embedding, but link embedding is quite different from the conventional approach. In this case, link embedding is done by dynamically using different channels that consist of multiple links.

4.1. SInP. We model the S network as a weighted undirected graph and denote it by $G^{\mathrm{SInP}} = (N^{\mathrm{SInP}}, L^{\mathrm{SInP}})$, where N^{SInP} is the set of physical sensor nodes and L^{SInP} is the associated links. SInP sensor nodes are divided into two functionalities based on their processing capability and storage space, that is, common widely deployed sensor nodes and a sensor gateway router. Each sensor gateway router in the SInP is associated with the CPU capacity weight value $C(N^{\mathrm{SInP}})$ and its GPS location $\mathrm{loc}(N^{\mathrm{SInP}})$ on a globally understood coordinate system. Each substrate link $l^{\mathrm{SInP}}(i, j) \in L^{\mathrm{SInP}}$ between two substrate gateway router nodes i and j is associated with the bandwidth capacity weight value $b(l^{\mathrm{SInP}})$ denoting the total amount of bandwidth. We denote the set of all substrate paths by P^s and the set of substrate paths from the source node s to the destination node d by $P^s(s, d)$. Figure 1 shows the substrate SInP network, where the sensing node is indicated by small circles of different colors and the sensor gateway routers are indicated by a node with a wireless antenna.

4.2. VSN Request by ALU. As we discuss the graph-based description of SInP, we also model the VSN request as weighted undirected graphs and denote a VN request in terms of the service request as $G^{\text{vsn}}(N^{\text{vsn}}, L^{\text{vsn}})$. We mention the requirement on virtual nodes and links of the substrate physical sensor network. Each VN request has an associated nonnegative value D^{v} expressing how far a virtual node $n^{\text{vsn}} \in N^{\text{vsn}}$ can be embedded from its preferred location $\text{loc}(n^{\text{vsn}})$. D^{v} is expressed naturally as a link delay or round-trip time from the $\text{loc}(n^{\text{vsn}})$.

4.3. SInP Network Resources Measurement. To measure the different types of resource usage of the SInP, we use the notion of utility. The substrate SInP node utility $U_{n}^{\text{SInP}}(n^{\text{SInP}})$ is defined as the total amount of processing power allocated to different virtual sensor nodes hosted on the substrate SInP node $n^{\text{SInP}} \in N^{\text{SInP}}$:

$$U_{n^{\text{SInP}}}\left(n^{\text{SInP}}\right) = \sum c\left(n^{\text{vsn}}\right). \tag{1}$$

The substrate SInP link utility $U_{l}^{\text{SInP}}(l^{\text{SInP}})$ is defined as the total amount of link usage by different virtual sensor nodes hosted on the substrate SInP node $n^{\text{SInP}} \in N^{\text{SInP}}$. It is actually the dedicated channel utilization to a specific virtual sensor node:

$$U_{l^{\text{SInP}}} = \sum b\left(l^{\text{vsn}}\right). \tag{2}$$

The substrate SInP storage or memory utility $U_{m}^{\text{SInP}}(m^{\text{SInP}})$ is defined as the total amount of storage usage by different virtual sensor nodes hosted on the substrate SInP node $n^{\text{SInP}} \in N^{\text{SInP}}$. It is actually the memory utilization of different virtual sensor nodes:

$$U_{m^{\text{SInP}}} = \sum s\left(m^{\text{vsn}}\right). \tag{3}$$

The total utility of the processing power, link, and storage can be calculated by summing up (1), (2), and (3). Here α, β, and γ are the weighted values to express the node, link, and storage capacity, respectively, by a single utility:

$$U_{T^{\text{SInP}}} = \alpha \sum c\left(n^{\text{vsn}}\right) + \beta \sum b\left(l^{\text{vsn}}\right) + \gamma \sum s\left(m^{\text{vsn}}\right). \tag{4}$$

4.4. Residual Resource Measurement. Residual resource management is performed by measuring the available resources remaining after utilization. In this section we have given the mathematical formulation of the remaining resources of the SInP sensor node, corresponding link, and storage only. The residual capacity of the SInP sensor nodes is defined as the total processing capacity of the sensor nodes, which is explained in (5):

$$R_{n^{\text{SInP}}}\left(n^{\text{SInP}}\right) = \sum_{n \in N} c\left(n^{\text{SInP}}\right) - U_{n^{\text{SInP}}}\left(n^{\text{SInP}}\right). \tag{5}$$

In a wireless sensor network, the communication is performed by wireless links. By a link, we mean a wireless link between different SGR nodes. We allocate different channels of a particular wireless link to particular applications in the virtualization of the sensor network. Equation (6) represents the residual channel capacity in the underlying SInP:

$$R_{l^{\text{SInP}}}\left(l^{\text{SInP}}\right) = \sum_{l \in L} b\left(l^{\text{SInP}}\right) - U_{l^{\text{SInP}}}\left(l^{\text{SInP}}\right). \tag{6}$$

There are two types of storage in the underlying SInP node: flash memory and SDRAM. In this mathematical model we only consider the SDRAM, which is only physical memory shared by different applications in the VSN applications. Equation (7) shows the total remaining residual storage for further applications:

$$R_{s^{\text{SInP}}}\left(s^{\text{SInP}}\right) = \sum_{m \in M} s\left(m^{\text{vsn}}\right) - U_{m^{\text{SInP}}}\left(m^{\text{SInP}}\right). \tag{7}$$

4.5. VSN Node and Link Embedding. In this work, the VSN node and link embedding are very much restricted to the SGR and the wireless link between different SGRs. Different VSNSP nodes share the same or different SGRs of the SInP. The typical sharing depends on the storage limit of the SGR. For wireless link embedding, we consider the efficient channel utilization. Individual VSNSPs provide particular services. For example, VSN-1 may provide urgent data services and use channel-1, while VSN-2 may provide suspicious data services and use channel-2 of the specific VSNSP. In this way, the same SInP can be used by different VSNSPs.

5. Packets Classification and Scheduling in VSN

This section discusses the packet classification and scheduling in VSN. Figure 5 depicts the packet classification and scheduling module of the SGR in a VSN environment. It consists of different components such as the traffic classifier, scheduling, channel allocation, and link estimator. Brief descriptions of the mechanisms are given below.

(i) The data packet is received by the IEEE 802.15.4 interface. Then the data is sent to the MAC reception module.

(ii) All of the data from the MAC reception passes through the traffic classifier module. Based on the information provided in the data packet, such as reliability, delay deadline, and priority information from the application layer, the data are classified as urgent, suspicious, moderate, and normal.

(iii) The classified data are passed through the 4-to-16 decoder module. Based on the availability of the VSN queues, the data are queued up.

(iv) The scheduling module sends the data to a specific path based on the priority and other requirements of the data packet and VSN queue.

(v) The channel allocation is based on the link estimator information and scheduling requirements.

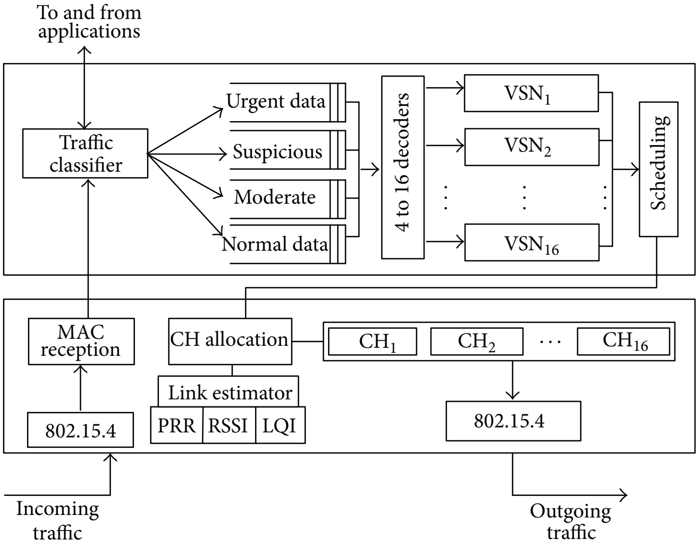


Figure 5: Protocol architecture of SGR.

(vi) The link estimator module depends on the PRR, RSSI, and LQI for link quality measurements.

(vii) Finally, the data packets are transmitted to the next VSN gateway.

In the following sections we describe the individual components in detail.

5.1. Traffic Classifier. The traffic classifier receives different types of data packets from the MAC reception module. A specific VSN may carry particular types of data packets or a combination of different types of data based on the application requirement. The data packets consist of different vital signs of the patient, such as the cardiac data, glucose level, blood pressure, pulse rate, and temperature. The packet received from the sensor nodes includes the data type, deadline, delay, and reliability requirements. Based on the information in the packet, the data are classified as urgent, suspicious, moderate, or normal. The traffic classification is context dependent.

Urgent Data. This includes emergency traffic or other data types specified by the applications. These types of traffic require approximately 100% reliability and a hard delay guarantee. It is usually event-triggered traffic and is generated whenever a life-threatening situation appears. For instance, when the heart rate and blood pressure of a patient exceed the danger limit, an emergency action is needed, which requires urgent transmission with the highest reliability and lowest delay.

Suspicious Data. This type of data requires a strict reliability requirement (>90%) but can tolerate a delay up to a certain limit, such as a medical image like an X-ray or ultrasonography. On the other hand, some data, such as a telemedicine video transmission, must meet a strict delay deadline but can tolerate some packet loss.

Moderate Data. Both delay and reliability guarantees are required. However, moderate data requires soft QoS rather than hard QoS. In this case, the reliability requirement is more than 80%. Different types of medical applications, such as heart rate and SPO$_2$ continuously generate data that must be delivered with moderate reliability and delay requirements.

Normal Data. This type of traffic does not require any strict delay or reliability constraints. It consists of the regular data for patients such as the temperature, blood pressure, and glucose level. For normal data, less than 70% percent reliability is maintained during transmission.

The classified data are transmitted over different VSNs according to their priority, reliability, and delay requirements. Data over different VSNs are forwarded to different users and applications through the dynamically allocated channels.

5.2. VSNs. From a technical point of view, all of the VSNs are the logical combination of the CPU resources, storage, and the link of the SInP. A VSN is formed dynamically based on the requirement of the application level requests provided by different users. However, a typical VSNSP consists of 16 VSNs, due to the dedicated and available channels in the physical layer. These 16 channels of the particular VSNSP can be allocated based on the priority, reliability, and end-to-end delay requirements of the traffic. A particular application may use more than one VSN to ensure guaranteed service. In a specific VSN, there are multiple communication paths by which the data may be transmitted.

5.3. Scheduling. The data scheduling is performed based on the information provided by different components, such as the traffic classifier, VSN priority, and channel allocation module. The goal of this module is to ensure application-specific reliability and end-to-end delay. Different applications have different reliability and end-to-end delay requirements.

5.4. Link Estimator. Link estimation is based on three parameters: the packet reception rate (PRR), received signal strength indicator (RSSI), and link quality indicator (LQI). Based on these parameters, the link estimator provides quality information regarding the particular link. The detailed mathematical derivations of these parameters are given below.

We estimate the current path state by using the link quality information, including the link quality indicator (LQI) and the received signal strength indicator (RSSI) [20, 21]. The RSSI is a function of the distance between two nodes and can be computed as follows:

$$\mathrm{RSSI}(d) = \mathrm{RSSI}(d_0) - 10n\log\left(\frac{d}{d_0}\right). \tag{8}$$

In (8), RSSI(d) is the received signal strength in *db* at a distance d from the source node. RSSI(d_0) is the received signal strength at a distance d_0 from the source and n is the attenuation exponent.

The IEEE 802.15.4 specification ensures that each incoming frame contains a link quality indicator (LQI) value. The LQI indicates the quality of the link at the time of the frame reception. According to the standard, the LQI value must be an integer that is uniformly distributed between 0 and 255, with 255 indicating the highest signal quality. The LQI is measured as follows:

$$\mathrm{LQI} = 255 + 3 \times P_{r\times\mathrm{dBm}}. \tag{9}$$

Here, $P_{r\times\mathrm{dBm}}$ is the power of a received frame expressed in decibel-milliwatts. If the computed value is a fraction, then a rounding operation is performed to obtain an integer. R_{ab} represents the link quality and received signal strength between two nodes. RSSI from node a to node b is represented by RSSI_{ab}, and RSSI from node b to node a is represented by RSSI_{ba}. In this case, we consider symmetric transmission. The LQI values of nodes a and b are represented as LQI_a and LQI_b. Thus R_{ab} can be calculated as follows:

$$R_{ab} = \mathrm{RSSI}_{ab} \times \mathrm{RSSI}_{ba} \times \mathrm{LQI}_a \times \mathrm{LQI}_b. \tag{10}$$

To calculate the packet reception rate (PRR), each node estimates the link loss rate for every outgoing link using the weighted average loss interval method discussed in [6]. It uses the interval between loss events to estimate the loss rate of a link. We denote the interval between the mth and (m + 1)th loss for the outgoing link of the ith path as $l_{i,1}(m)$. Then for the recent $1 \le m \le n$ losses, the average loss interval, $l_{i,1}$, is

$$l_{i,1}(i,n) = \frac{\sum_{m=1}^{n} l_{i,1}(m)\, w_m}{\sum_{m=1}^{n} w_m}, \tag{11a}$$

$$l_{i,1}(0,n-1) = \frac{\sum_{m=0}^{n-1} l_{i,1}(m)\, w_m}{\sum_{m=1}^{n} w_m}, \tag{11b}$$

$$l_{i,1} = \max\left(l_{i,1_{(i,n)}}, l_{i(0,n-1)}\right), \tag{11c}$$

where $l_{i,1}(0)$ is the interval since the most recent loss and w_m is the weight given to each loss interval. We compute the average PRR of the first hop of the ith path, $p_{i,1}$, using the average loss rate, $p_{i,1}^{c} = 1/l_{i,1}$, as

$$p_{i,1} = 1 - p_{i,1}^{c}. \tag{11d}$$

The communication nature of VSN enables the measurement of the success rate of a path by using passive information exchange. When a node forwards a packet in a path, it includes the success rate of the path in the packet. The success rate of the ith path of a node, $P_i(h_i)$, is given by

$$P_i(h_i) = \prod_{J=1}^{h_i} p_{i,j} = p_{i,1}\prod_{J=2}^{h_i} p_{i,j}, \tag{12}$$

where $p_{i,j}$ is the success rate of the jth hop and $p_{i,1}$ is the success rate of the first hop of the node. $\prod_{J=2}^{h_i} p_{i,j}$ is the success rate of the path from the downstream node, and the node overhears this from the forwarded packets of the downstream node.

5.5. Channel Allocation. Channel allocation is a dynamic process by which the system allocates a particular channel to the specific VSN. There are 16 channels in the 802.15.4 PHY layer specification, starting at 2.4 GHz. Based on the link status from the link estimator, this module allocates the channels. The channel quality is computed by the following equation:

$$R_{\text{Channel}} = \frac{\sum_{i=1}^{n} R_{ab}^{i}}{n}, \tag{13}$$

where R_{ab} is the quality of the link and n is the number of links on the channel.

With the help of the channel quality computation, the channel allocator selects the channel as follows.

Step 1. Periodically measure the channel quality with the R_{Channel} and PRR values.

Step 2. Define the scheduling probability $P(a)$, which is the probability that traffic is assigned to channel I, representing the normalized value of R_{Channel} relative to other channel values as shown:

$$P_i(a) = \frac{R_{\text{Channel}}}{\sum_{k}^{i} R_{\text{Channel}}}. \tag{14}$$

We measure the probability ranges for the channels using the scheduling probability, $P_i(a)$. Thus, for each channel i, the probability range is defined as follows:

$$\left(\sum_{k}^{i-1} p_i(a), \sum_{k}^{i} p_i(a)\right); \qquad i = 1, 2, \ldots, N, \quad \text{where } p(a) = 0, \quad \sum_{k=0}^{i} p_i(a) = 1. \tag{15}$$

Step 3. We use the probability ranges to follow the data to the channel in each interval of time. A priority is assigned to the channel, referring to its R_{Channel}. A channel with a high R_{Channel} has higher priority. This is dynamic and changes with time.

The channel allocation module selects a particular channel according to a generated random number between 0 and 1. The value of the random number falls into the range defined in (15). Thus the channel with index i related to the selected range is chosen to send the data packet. The probability range is used as the priority. Lower-priority channels have a smaller chance than higher-priority channels to follow data.

6. QoS Model in Delay Domain

This section provides a mathematical model for delay analysis for different types of packets and VSN. Different types of traffic require a certain delay guarantee. Here, we introduce the QoS differentiation model for urgent, suspicious, moderate, and normal data in the delay domain. λ_α, λ_β, λ_χ, and λ_δ are the arrival rates, and μ_α, μ_β, μ_χ, and μ_δ are the service rates of urgent, suspicious, moderate, and normal data, respectively. λ is the total arrival rate that indicates the number of incoming packets per second at the SGR, and $\lambda = \lambda_\alpha + \lambda_\beta + \lambda_\chi + \lambda_\delta$. $\mu = \mu_\alpha + \mu_\beta + \mu_\chi + \mu_\delta$ denotes the number of packets that depart per second. The packet arrival rate to each SGR is a Poisson process. For a given SGR, the end-to-end path delay is composed of the transmission delay and queuing delay. The transmission delay is avoided due to its negligence. We consider the $M/M/1$ queue model with nonpreemptive priority. We consider the individual priorities, p_α, p_β, p_χ, and p_δ, for different classes of traffic. The traffic intensity at the SGR is computed as follows:

$$p = \frac{\lambda}{\mu} = p_\alpha + p_\beta + p_\chi + p_\delta. \tag{16}$$

The probability that an urgent packet finds other packets in service is equal to the ratio of the time spent by the SGR on the suspicious, moderate, and normal packets. These are calculated as $\lambda_\beta/\mu = p_\beta$, $\lambda_\chi/\mu = p_\chi$, and $\lambda_\delta/\mu = p_\delta$.

Little's law [22, 23] gives us a good approximation regarding the queue behavior and a basis for predicting the performance of the individual queues:

$$E(n) = \lambda E(t). \tag{17}$$

Here, $E(n)$ is the average number of packets in the queue, λ is the packet arrival rate, and $E(t)$ is the average delay time per packet in the system.

Urgent Packet Processing. Urgent packets have the highest priority. For their processing, there are dedicated VSNs. This ensures almost negligible delay, which includes the short queuing delay.

Suspicious Packet Processing. For delay-guaranteed suspicious packets, the processing is done using the same method used for urgent packets. The packets that are not delay guaranteed face a longer queuing delay. The mean delay of such a packet depends on $E(n)$ and the packets in service. This can be formulated as follows:

$$\begin{aligned} E(t_\beta) &= \frac{E(n_\beta)}{\mu} + \frac{1}{\mu} + \frac{1}{\mu}(p_\chi + p_\delta), \\ E(t_\beta) &= \frac{1 + (p_\chi + p_\delta)}{(1 - p_\beta)\mu}, \\ E(t_\beta) &= \frac{1}{\mu} + \frac{p_\beta/\mu}{1 - p_\beta}. \end{aligned} \tag{18}$$

Moderate Packet Processing. This type of packet has to wait for any suspicious packets in service and the moderate packets in the ready queue. The delay can be calculated as follows:

$$E(t_\chi) = \frac{E(n_\beta)}{\mu} + \frac{E(n_\chi)}{\mu} + \frac{1}{\mu} + \frac{1}{\mu}p_\delta. \tag{19}$$

Since $p = p_\alpha + p_\beta + p_\chi + p_\delta$, (14) can be represented as follows:

$$E(t_\chi) = \frac{E(n_\beta)}{\mu} + \frac{E(n_\chi)}{\mu} + \frac{1}{\mu} + \frac{1}{\mu}(p - p_\alpha - p_\beta - p_\chi). \tag{20}$$

Normal Packet Processing. A normal packet has to wait for the suspicious and moderate packets in service and also for the normal packet in the queue. The delay can be calculated as follows:

$$\begin{aligned} E(t_\delta) &= \frac{E(n_\beta)}{\mu} + \frac{E(n_\chi)}{\mu} + \frac{E(n_\delta)}{\mu} + \frac{1}{\mu} + \frac{1}{\mu}p_\delta, \\ E(t_\delta) &= \frac{E(n_\beta)}{\mu} + \frac{E(n_\chi)}{\mu} + \frac{E(n_\delta)}{\mu} + \frac{1}{\mu} \\ &\quad + \frac{1}{\mu}(p - p_\alpha - p_\beta - p_\chi - p_\delta), \\ E(t_\delta) &= \frac{E(n_\beta)}{\mu} + \frac{E(n_\chi)}{\mu} + \frac{E(n_\delta)}{\mu} + \frac{1}{\mu}. \end{aligned} \tag{21}$$

The average delay of the arrived packet depends on the packets that are already buffered in the queue plus the packets in service.

7. QoS Model in Reliability Domain

This section provides a mathematical model for reliability analysis for different types of packets and VSN. Reliability is a unitless quantity that can be defined as the ratio of the number of unique packets received by the gateway node to the number of unique packets sent by the source node. In this paper, we consider both the link layer reliability and the network layer reliability. In a wireless network, MAC layer retransmission is used to improve the reliability. However, retransmission increases the delay at each hop. Moreover, MAC layer retransmission does not efficiently increase the reliability in a densely deployed WSN due to its increased medium contention. Here we consider a VSN-based approach to achieve the required reliability. For urgent traffic, the VSN approach provides around 100% reliability by using dedicated paths over multiple VSNs. Suspicious data with a reliability requirement of more than 90% are transmitted over multiple paths of multiple VSNs. Moderate data with a reliability requirement of more than 80% are transmitted over multiple paths of a specific VSN. Finally, normal traffic with reliability of less than 70% is transmitted over the available paths. This reliability-differentiated traffic is transmitted with the help of the scheduling module of the network layer and the channel allocation module of the link layer.

Let us assume that the number of unique data traffic sent by the source node is X_s and the number of unique data traffic received by the gateway router is X_r. Thus the reliability is $R = X_r/X_s$. The reliability calculation follows the multiplication law of probability:

$$p(k) = \prod_{i=1}^{k}\left(1 - p_i^d\right). \tag{22}$$

Now we will address the probabilistic model of reliability [5–7, 23] for different cases, such as multiple paths over multiple VSNs, multiple paths over a single VSN, and a single path over a specific VSN.

(A) Multipath over multi-VSN: in multipath over multi-VSN packet forwarding, if there are m paths in n VSNs, the probability that at least one copy of a packet is successfully received by the SGR is

$$\begin{aligned} p(m,n) &= 1 - \left[\prod_{i=1}^{16}\left[1 - p_i\left(k_i\right)\right]\right]\left[\prod_{j=1}^{m}\left[1 - p_j\left(k_j\right)\right]\right], \\ p(m,n) &= 1 - \left[\prod_{i=1}^{16}\left[1 - \prod_{i=1}^{k}\left(1 - p_i^d\right)\right]\right] \\ &\quad \times \left[\prod_{j=1}^{m}\left[1 - \prod_{j=1}^{m}\left(1 - p_j^d\right)\right]\right], \end{aligned} \tag{23}$$

where $p(m,n)$ is the probability of success for multipath and multiVSN packet forwarding with m paths and n VSNs. $p_i(k_i)$ and $p_j(k_j)$ are the probabilities of success for the ith VSN and jth path, respectively, and p_i^d and p_j^d are the probabilities that a packet is dropped by the ith VSN and jth path, respectively.

If X_s packets are sent and X_r packets are received by the gateway node with probability $p(m,n)$, then the number of total packets received by the SGR has a binomial distribution, and the probability mass function (pmf) is given by:

$$p\left[X_r = t\right] = \binom{X_s}{t}\left[p(m,n)\right]^t\left[1 - p(m,n)\right]^{X_s - t}. \tag{24}$$

The required reliability, R_{req}, is achieved when the number of unique packets received by the SGR is X_r. This implies that $X_r = R_{\text{req}}X_s$. To fulfill the requirement, X_r should be greater than or equal to R_{req}, so the probability that the required reliability is met in multipath over multi-VSN packet forwarding, $p_{m\text{Path}}^{m\text{VSN}}$, is

$$p_{m\text{Path}}^{m\text{VSN}} = \sum_{t=X_r}^{X_s}\binom{X_s}{t}\left[p(m,n)\right]^t\left[1 - p(m,n)\right]^{X_s - t}. \tag{25}$$

(B) Multipath over single VSN: in multipath over single VSN packet forwarding, if there are m paths over a VSN, the probability that at least one copy of a packet is successfully received by the SGR is

$$\begin{aligned} p(m) &= 1 - \left[\prod_{j=1}^{m}\left[1 - p_j\left(k_j\right)\right]\right], \\ p(m) &= 1 - \left[\prod_{j=1}^{m}\left[1 - \prod_{j=1}^{m}\left(1 - p_j^d\right)\right]\right], \end{aligned} \tag{26}$$

where $p(m)$ is the probability of success for multipath for a specific VSN packet forwarding with m paths. $p_i(k_i)$ and $p_j(k_j)$ are the probabilities of success for the ith VSN and jth path, respectively, and p_i^d and p_j^d are the probabilities that a packet is dropped by the ith VSN and jth path, respectively. Since X_s packets are sent and X_r packets are received by the gateway node with probability $p(m)$, then the number of total packets received by the SGR has a binomial distribution, and the pmf is given by

$$p\left[X_r = t\right] = \binom{X_s}{t}\left[p(m)\right]^t\left[1 - p(m)\right]^{X_s - t}. \tag{27}$$

The required reliability, R_{req}, is achieved when the number of unique packets received by the SGR is X_r. This implies that $X_r = R_{\text{req}}X_s$. To fulfill the requirement, X_r should be greater than or equal to R_{req}, so the probability that the required reliability is met in multi-path over single VSN packet forwarding, $p_{m\text{Path}}^{s\text{VSN}}$, is

$$p_{m\text{Path}}^{s\text{VSN}} = \sum_{t=X_r}^{X_s}\binom{X_s}{t}\left[p(m)\right]^t\left[1 - p(m)\right]^{X_s - t}. \tag{28}$$

(C) Single path over single VSN: in a single path over single VSN packet forwarding, to get the required reliability level, packets must be retransmitted since the failure probability is high. The probability denoted as $p_j(r)$ indicates that the jth hop successfully forwards a packet within r retransmission attempts. In such a scenario, if there is a path in a VSN, the probability that at least one copy of a packet is successfully received by the SGR is

$$p_j(r) = 1 - (p_j)^r. \tag{29}$$

The probability that a data packet is successfully received by the SGR in a single path over a specific VSN with a hop count s is

$$p(s) = \prod_{j=1}^{S} [p_j(r)] = \prod_{j=1}^{S} [1 - (p_j)^r]. \tag{30}$$

Here, $p(s)$ is the probability of success for a single path for single VSN packet forwarding. The required reliability, R_{req}, is achieved when the number of unique packets received by the SGR is X_r. This implies that $X_r = R_{\text{req}} X_s$. To fulfill the requirement, X_r should be greater than or equal to R_{req}, so the probability that the required reliability is met with single path over single VSN packet forwarding, $p_{\text{sPath}}^{s\text{VSN}}$, is

$$p_{\text{sPath}}^{s\text{VSN}} = \sum_{t=X_r}^{X_s} \binom{X_s}{t} [p(s)]^t [1 - p(s)]^{X_s - t}. \tag{31}$$

8. Performance Evaluations

In this section, we discuss the simulation environment and evaluation results. We have implemented and evaluated the VSNware on the Imote2 sensor node. The Imote2 sensor node has a Marvel PXA27x ARM processor with 400 MHz clock speed, 32 MB Flash, and 32 MB SDRAM. We have selected Imote2 as the sensor node for its advanced features such as its memory size and CPU speed. In this evaluation, the sensor node runs Embedded Linux as its operating system. The detailed system specifications are given in Table 2.

We develop a virtual machine for wireless sensor network called "VSNware." It is based on Embedded Linux. The VSNware environment restricts access to all of the physical resources on the node, thus ensuring that applications are only allowed to access the hardware through the VSNware. VSNware is available in all SGR nodes. VSNware supports concurrent application execution and dynamic application deployment. The VSNware supports applications implemented in high-level language, thereby enabling different applications from health care scenarios to be executed and run in the VSN environment. We compare the proposed VSN approach with MMSPEED [5] and the traditional WSN approach. MMSPEED provides a virtual network of multiple speed layers for a network-wide speed guarantee in terms of the reliability and timeliness. The traditional WSN approach in this performance evaluation process is used to emulate the exact scenario in the conventional method that is provided by the proposed VSN approach. In the following scenarios, utilization of VSNware is the technical point of a VSN-based system evaluation. Here, we focus on different issues such as the memory utilization, CPU utilization, and execution times of individual applications. Efficient memory and CPU utilization are the main concern of the VSN approach. The execution time and CPU utilization are related to each other. We have compared the memory and CPU utilization of our proposed VSN scheme to those of the MMSPEED and traditional approaches.

Table 2: System specifications.

Type	Specifications
Sensor node	Imote2
CPU	Marvel PXA27x ARM
CPU speed	400 MHz
Operating system	Embedded Linux
OS version	2.6.29
VM	VSNware
Flash size	32 MB
SDRAM size	32 MB
Interface	USB
Bandwidth	250 Kbps
Radio	IEEE 802.15.4

In Figure 6, we plot the memory usage of the traditional, MMSPEED, and VSN approaches. The sensor virtualization version includes the overhead of the applications due to the additional memory usage of a single sensor node. However, the overhead is linear and increases slowly based on the number of applications being deployed. In comparison to the MMSPEED and traditional approaches, the proposed VSN approach provides better performance. The performance evaluation shows that the proposed VSN approach reduces the average memory utilization by 53% and 56% as compared to the MMSPEED and traditional approaches, respectively.

In Figure 7, we plot the execution times of different applications for different numbers of virtualized sensor nodes. The figure shows the execution times of 3, 5, and 7 vital sign-sensing applications based on the virtualization of sensor network methodology. The execution time increases linearly based on the number of applications in a sensor node.

In Figure 8, we plot the CPU utilization versus the number of applications in the traditional approach, MMSPEED approach, and in the virtualization of sensor network scenario. The CPU utilization increases linearly in all of the cases. In this scenario, the VSN approach uses CPU resources efficiently, since it executes different applications on the same sensor node. The performance evaluation result shows that the proposed VSN approach has average CPU utilization that is 56% and 60% lower than that of the MMSPEED and traditional approaches, respectively.

In Figure 9, we depict the memory usage of different typical applications, such as medical imaging, cardiac, blood

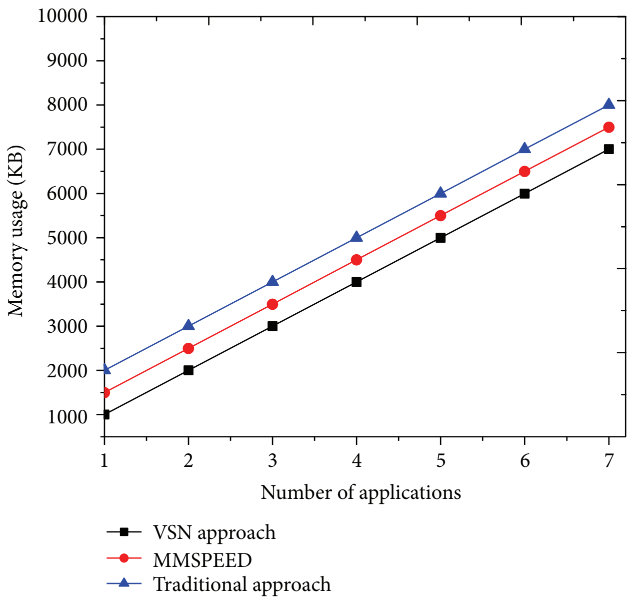


FIGURE 6: Comparative memory usage in VSN approach.

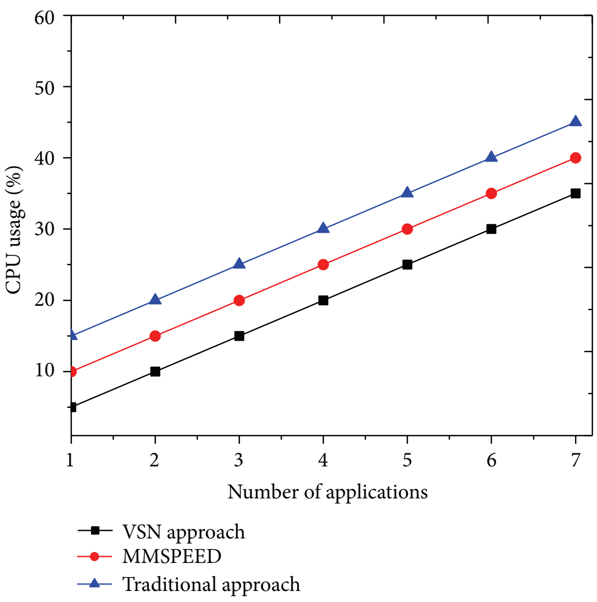


FIGURE 8: Comparative CPU usage in VSN approach.

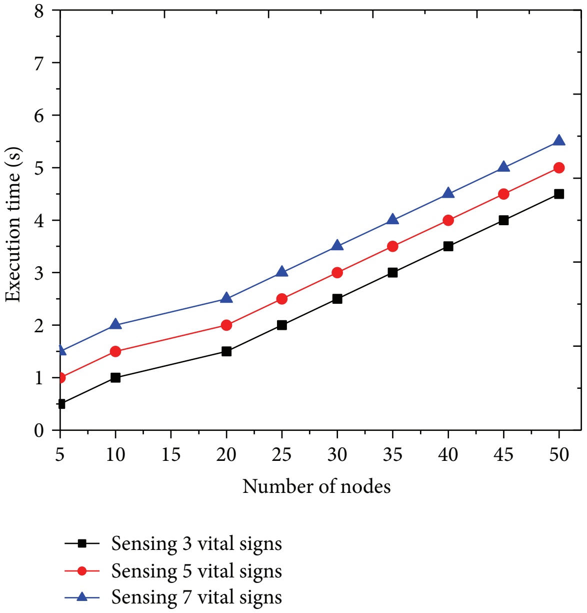


FIGURE 7: Execution time versus number of nodes.

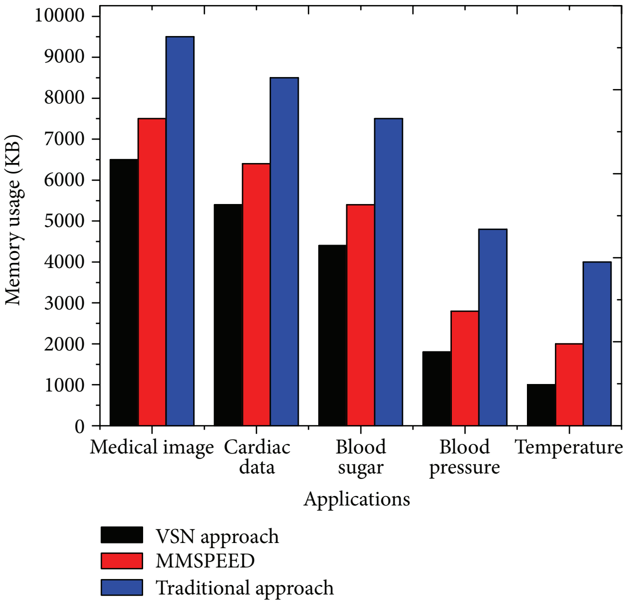


FIGURE 9: Comparative memory usage by applications.

sugar, blood pressure, and temperature. Medical image applications use more memory than other applications. Since the total memory in the Imote2 sensor node is 32 MB, it can provide the environment for the execution and running of the typical applications. The figure also demonstrates the memory usage of different applications in terms of the MMSPEED scheme and the traditional WSN and proposed VSN approaches.

In Figure 10, we have presented the end-to-end delays of different data flows. It shows that the average delay for the four classes of data packets increases with the increasing number of sensor nodes. It clearly shows that the average end-to-end delay of an urgent packet is significantly lower than that of the suspicious, moderate, and normal data packets.

In Figure 11, we have presented the end-to-end delays of different data flows with respect to the data rate. The data rate varies according to the priorities assigned to different data flows. Since the highest priority is assigned to the urgent class, it experiences the lowest delay with respect to the other classes of data packets. Suspicious and moderate data also experience minimum delay due to their priority and the VSN approach used in this scheme.

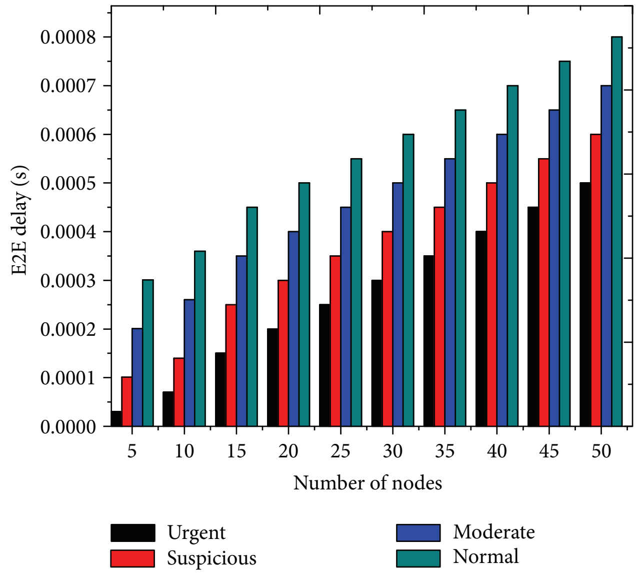


Figure 10: End-to-end delay versus number of nodes.

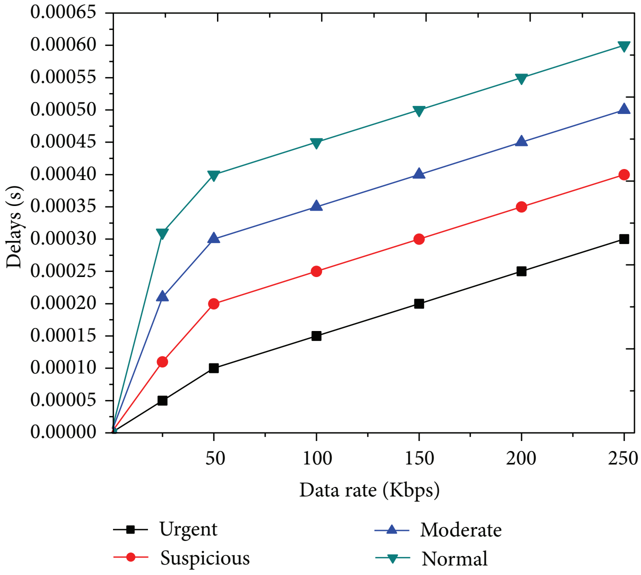


Figure 11: Delay versus data rate.

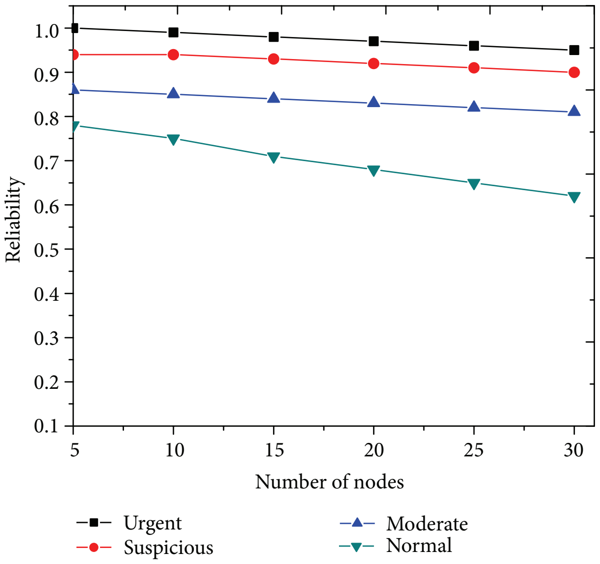


Figure 12: Reliability versus number of nodes.

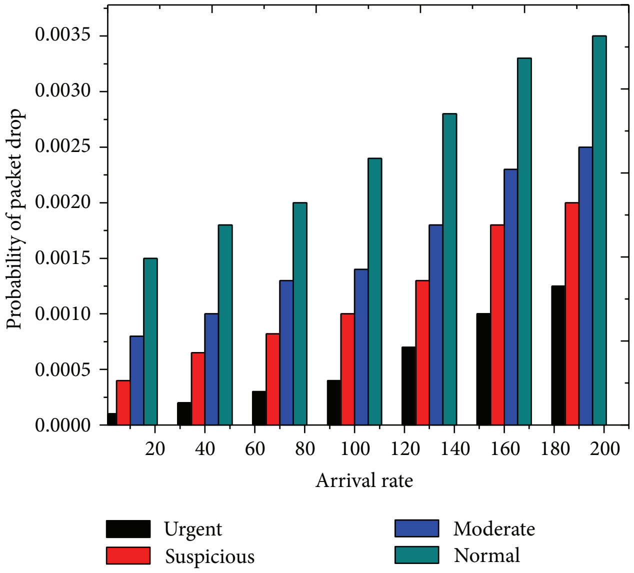


Figure 13: Packet dropping probability versus arrival rate.

Figure 12 shows the reliability of different data packets with respect to the number of nodes. We focused on designing our scheme to ensure 100% percent reliability, but, practically, it ensures an average of approximately 98% reliability. The reliability levels of the suspicious and moderate data are also significantly higher than that of the normal data.

In Figure 13, we demonstrate the packet dropping probability versus the arrival rate. Packet dropping depends on the packet arrival rate at the queues. The figure shows that the urgent data has the lowest loss, followed by the suspicious, moderate, and normal traffic.

9. Conclusions

In this paper, we propose a novel approach of a VSN-based packet delivery mechanism to provide service differentiation and probabilistic QoS assurance in the delay and reliability domains. It also explores QoS-based data classification and scheduling schemes for health care applications in a VSN environment. For the delay domain, we use multiple layers based on VSN so that different data packets can dynamically choose the appropriate layers according to the delay requirements of individual data traffic. For the reliability domain, we use multiple virtual layers as well as different paths within the corresponding VSN. The performance evaluations show that the VSN scheme provides low end-to-end delay and high reliability for the urgent data. It also significantly increases the performance for the other data classifications. Our future interest is to emphasize the large-scale and federated sensor network platform with multiple applications sharing the same physical resources that will facilitate the rapid deployment of the ubiquitous health care system.

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Republic of Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2013-(H0301-13-2001)).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[2] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[3] M. M. Islam and E.-N. Huh, "Sensor proxy mobile IPv6 (SPMIPv6)-a novel scheme for mobility supported IP-WSNs," *Sensors*, vol. 11, no. 2, pp. 1865–1887, 2011.

[4] M. M. Islam and E.-N. Huh, "A novel addressing scheme for PMIPv6 based global IP-WSNs," *Sensors*, vol. 11, no. 9, pp. 8430–8455, 2011.

[5] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–753, 2006.

[6] M. M. Alam, M. A. Razzaque, M. Mamun-Or-rashid, and C. S. Hong, "Energy-aware QoS provisioning for wireless sensor networks: analysis and protocol," *Journal of Communications and Networks*, vol. 11, no. 4, pp. 390–405, 2009.

[7] M. A. Razzaque, M. M. Alam, M. Mamun-Or-rashid, and C. S. Hong, "Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks," *IEICE Transactions on Communications*, vol. E91-B, no. 8, pp. 2589–2601, 2008.

[8] M. M. Islam, M. M. Hassan, G.-W. Lee, and E.-N. Huh, "A survey on virtualization of wireless sensor networks," *Sensors*, vol. 12, no. 2, pp. 2175–2207, 2012.

[9] M. M. Islam, J. H. Lee, and E. N. Huh, "An efficient model for smart home by the virtualization of wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 168735, 10 pages, 2013.

[10] M. M. Islam, M. M. Hassan, and E.-N. Huh, "Virtualization in wireless sensor network: challenges and opportunities," in *Proceedings of the 13th International Conference on Computer and Information Technology (ICCIT '10)*, December 2010.

[11] S. Kabadayi, A. Pridgen, and C. Julien, "Virtual sensors: abstracting data from physical sensors," in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '06)*, pp. 587–592, Buffalo-Niagara Falls, NY, USA, June 2006.

[12] J.-H. Shin and D. Park, "A virtual infrastructure for large-scale wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2853–2866, 2007.

[13] A. P. Jayasumana, H. Qi, and T. H. Illangasekare, "Virtual sensor networks—a resource efficient approach for concurrent applications," in *Proceedings of the 4th International Conference on Information Technology (ITNG '07)*, pp. 111–115, April 2007.

[14] S. Waharte, J. Xiao, and R. Boutaba, "Overlay wireless sensor networks for application-adaptive scheduling in WLAN," in *Proceedings of the 7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC '04)*, vol. 3079 of *Lecture Notes in Computer Science*, pp. 676–684, Toulouse, France, 2004.

[15] I. Leontiadis, C. Efstratiou, C. Mascolo, and J. Crowcroft, "Senshare: transforming sensor networks into multi-application sensing infrastructures," in *Proceedings of the 9th European Conference on Wireless Sensor Networks*, February 2012.

[16] C. Efstratiou, I. Leontiadis, C. Mascolo, and J. Crowcroft, "Demo abstract: a shared sensor network infrastructure," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 367–368, Zurich, Switzerland, November 2010.

[17] P. Levis and D. Culler, "Maté: a tiny virtual machine for sensor networks," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '02)*, pp. 85–95, New York, NY, USA, October 2002.

[18] Y. Yu, L. J. Rittle, V. Bhandari, and J. B. LeBrun, "Supporting concurrent applications in wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 139–152, November 2006.

[19] N. M. Mosharaf, K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proceedings of the IEEE 28th Conference on Computer Communications (INFOCOM '09)*, pp. 783–791, April 2009.

[20] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, 2012.

[21] A. H. Shuaib and A. H. Aghvami, "A routing scheme for the IEEE-802.15.4-enabled wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5135–5151, 2009.

[22] J. D. C. Little and S. C. Graves, "'Little's Law' , Massachusetts Institute of Technology," in *Building Intuition: Insights From Basic Operations Management Models and Principles*, D. Chhajed and T. J. Lowe, Eds., Springer Science, Business Media, LLC, 2008.

[23] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 4th edition, 2002.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 924576, 8 pages
http://dx.doi.org/10.1155/2013/924576

Research Article

Battery-Less 6LoWPAN-Based Wireless Home Automation by Use of Energy Harvesting

Ardiansyah Musa Efendi, Seungkyo Oh, Ali Fahmi Perwira Negara, and Deokjai Choi

Advanced Network Laboratory, School of Electronics and Computer Engineering, Chonnam National University, Gwangju 500757, Republic of Korea

Correspondence should be addressed to Deokjai Choi; dchoi@jnu.ac.kr

Received 7 April 2013; Accepted 25 June 2013

Academic Editor: Guangjie Han

Being one of the current main technologies, wireless sensor network (WSN) is absolutely critical and plays an increasingly important role in the development of a wide range of data acquisition, processing, and control applications. One interesting challenge application of the WSNs is Internet-based home automation system. The rapid expansion of IPv6, moreover, after IETF defined 6LoWPAN as a technique to apply IPv6 into WSNs standard, added potential for Internet paradigm communication, control, and monitoring of home automation devices from anywhere on the globe. The use of 6LoWPAN communications technology also helps lower expense of the system and decreases complexity of the home automation architecture. When implementing 6LoWPAN-based home automation system, the important feature is long periods of life to support full automation without the need for periodically replacing its batteries. To have this, 6LoWPAN home devices with total energy independence are needed. By using additional components for power management and the implementation of 3 V, 70 mA small polycrystalline silicon solar cell for solar energy harvesting, we can have home automation node for an indefinite amount of life time.

1. Introduction

Wireless sensor networks (WSNs) are employed in a wide range of data acquisition, processing, and control applications. Their advantages over traditional wired sensor networks include mobility, increased reliability, easier installation, and lower deployment cost. One interesting challenge application of the WSNs is home automation system. There are so many definitions of home automation available in the literature. [1] Bromley et al. describe home automation as the introduction of technology within the home to enhance the quality of life of its occupants, through the provision of automation services such as multimedia entertainment, telehealth, and energy conservation. The home environment has seen a rapid introduction of Internet protocol enabled technology. This technology offers new and exciting opportunities to increase the connectivity of devices within the home for the purpose of home monitoring and automation. Recently, home automation systems have been challenged with the two outstanding needs: the need for the high interoperability between home devices and the need for user interfacing and accessing to the system from different end points.

There has been significant research in the field of home automation. The authors of [2] developed an embedded board that physically connected all the home automation devices through integration with a personal computer based web server and then provided remote access to the system. However, the system requires an intrusive and expensive wired installation and the use of a high end computer. The authors of [3] introduced a Bluetooth based home automation system, by connecting each home device to a local Bluetooth subcontroller. This system reduces the amount of physical wiring required and the intrusiveness of the installation through the use of wireless technology. However, due to the sharing of a single Bluetooth module between numerous devices has the disadvantage of incurring an access delay. The authors of [4] defined a ZigBee-based home automation networks, a flexible home automation architecture, and trough adoption and evaluated the potential of ZigBee. However, this system still has problem with evolvability, scalability, and internet integration. End-to-end paradigm where only the end-to-end points participate in the application protocol exchanges which cannot be implemented with this solution. ZigBee needs intermediate local proxy server to enable communication between embedded home devices and Internet.

A possible strategy to solve the problem listed above could be adopting 6LoWPAN. The introduction of 6LoWPAN protocol enables home automation device based on 802.15.4 wireless sensor network standard to be compatible with IPv6 while maintaining low-power consumption and taking the nature of wireless networks into account [5]. The 6LoWPAN standard also promises the fulfillment of the emerging trend of embedded Internet technology in all aspects of everyday life [6], mainly because of its low costs, low-power, scalability, possibility to adapt to existing technologies. The authors of [7, 8] tried to implement 6LoWPAN for home automation network, but between the features of any 6LoWPAN-based home automation system are long periods of life without the need for periodically replacing its batteries. To have this, we need to design the 6LoWPAN home device with total energy independence. In this case, additional components for energy harvesting are needed, and the power management of the 6LoWPAN home devices is an important topic.

In this paper, we presented and analyzed implementation of energy harvesting in the 6LoWPAN-based home automation infrastructure. The rest of this paper is divided into five sections. Section 2 presents our 6LoWPAN-based home automation system. Section 3 discusses setting environment of our system design for 6LoWPAN-based home automation energy harvesting. Finally, Section 4 will evaluate and Section 5 will conclude our research in this system.

2. Overview of Our 6LoWPAN-Based Home Automation System

This section describes our conceptual design of a flexible home automation network using 6LoWPAN (see Figure 1). Our goal is to develop a home automation system that is robust, flexible, easy to use and has a wide range of capabilities. The system allows home owners to monitor and control connected devices in the home, through any Wi-Fi enabled device. Additionally, users may remotely monitor and control their home devices using any Internet enabled device. A home gateway is implemented to facilitate interoperability between heterogeneous 6LoWPAN and ordinary IPv6 network based on Ethernet and Wi-Fi. It also facilitates local and remote control and monitoring of the home devices and provides a consistent interface, regardless of the accessing device. Remote user communications traverse the Internet until they reach the home gateway. They are then wirelessly transmitted to the home devices using the 6LoWPAN.

2.1. 6LoWPAN Home Automation Network. As discussed, the proposed system architecture implements 6LoWPAN home automation network. The use of 6LoWPAN offers certain advantages and provides a comprehensive home automation solution. This technology is designed to be used on applications that required low data rate, low cost, low-power consumptions, and two-way wireless communications [9].

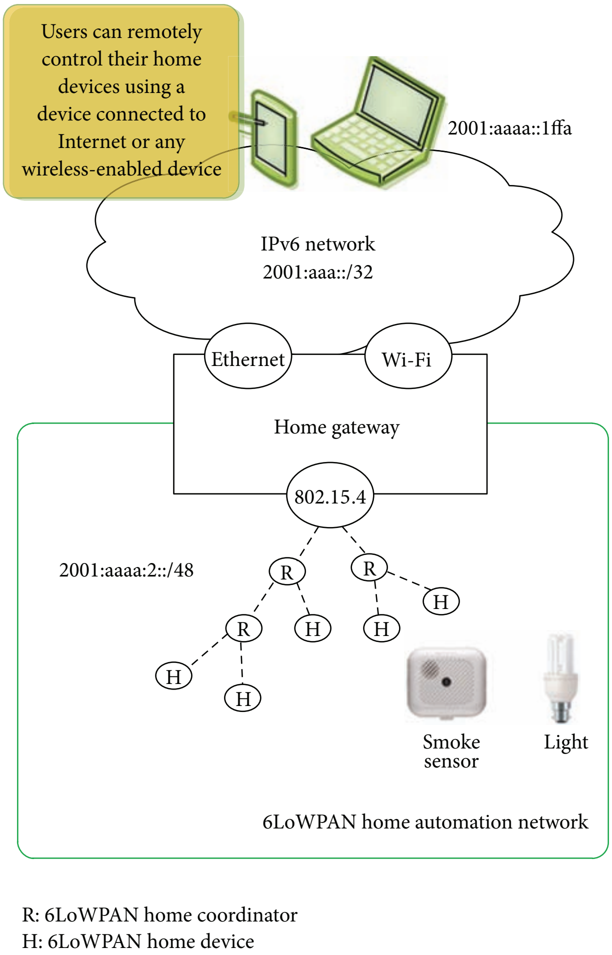

Users can remotely control their home devices using a device connected to Internet or any wireless-enabled device
2001:aaaa::1ffa
IPv6 network
2001:aaa::/32
Ethernet
Wi-Fi
Home gateway
802.15.4
2001:aaaa:2::/48
R
R
H
R
H
H
H
H
Smoke sensor
Light
6LoWPAN home automation network
R: 6LoWPAN home coordinator
H: 6LoWPAN home device

Figure 1: 6LoWPAN home automation system architecture.

The wireless nature of 6LoWPAN helps overcome the intrusive installation problem with the existing home automation systems identified earlier. The 6LoWPAN devices based on IEEE 802.15.4 standard theoretically provides 250 kbps data rate and 127 bytes frame size; this would need a considerable amount of compression for controlling most home automation devices using IPv6. The automatic installation and IPv6 addressing of 6LoWPAN provide novel solution to end-to-end connectivity and ubiquitous Internet-based home automation system and help tackle the expensive and complex architecture problems with existing home automation systems, as identified earlier.

2.2. Home Gateway. Home gateway, as depicted in Figure 2, is our edge route [10], and it is charged by providing interoperability between different connecting networks. The home gateway provides data translation services between Internet based on Ethernet/Wi-Fi and 6LoWPAN networks based on 802.15.4. One way to integrate 6LoWPAN into home gateway is to provide basic layer 1–3 functionality using a 6LoWPAN network processor, which used 802.15.4 as low-power wireless interface. In order to use 6LoWPAN wireless interface with a standard IPv6 protocol stack, our home gateway functionality

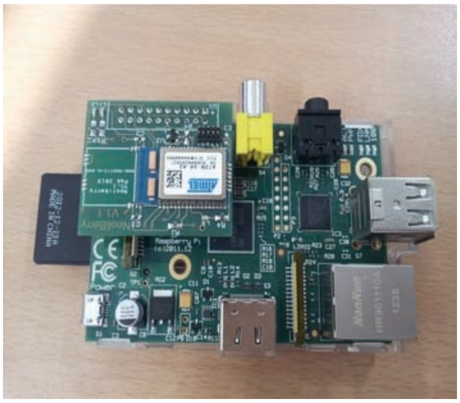


Figure 2: Home gateway.

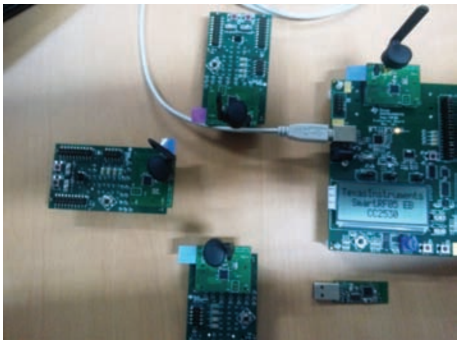


Figure 3: 6LoWPAN home nodes based on TI CC2530.

implemented 6LoWPAN adaption layer, 6LoWPAN-ND, and IPv6 RPL routing, IPv6 interconnection.

In order to interconnect home automation system, based on 802.15.4 and 6LoWPAN, with existing IPv6 network, based on Ethernet/Wi-Fi, the home gateway can act as a bridge or as a router. In router mode, this home gateway acts as a full-fledged IPv6 router, interconnecting two IPv6 subnets. The home automation subnet is managed by the RPL protocol, and the Ethernet subnet is managed by IPv6 NDP. In this mode, home gateway provides a virtual second interface to filter the packet. The router mode allows us to isolate 6LoWPAN mesh into its own subnet, therefore clearly identifying the home automation nodes. In bridge mode, this home gateway provides switching capabilities, allowing to interconnect a standard IPv6 based network with an RPL based 6LoWPAN mesh in one subnet. All incoming packets targeting an 802.15.4 interface or incoming multicast packets on the Ethernet interface are forwarded to the home automation segment. Conversely, all incoming packets targeting an Ethernet interface or incoming multicast packets on the LoWPAN interface are forwarded to the Ethernet segment. Home gateway is acting as an NDP proxy on the Ethernet side and is using NDP parameters to configure the 6LoWPAN mesh. Source and destination MAC addresses are translated, and addresses in ICMPv6 packets are also translated. This mode allow us to seamlessly integrate a 6LoWPAN mesh into an existing NDP based IPv6 network and aggregate several 6LoWPAN meshes into one virtual IPv6 subnet.

2.3. 6LoWPAN-Based Home Node. The 6LoWPAN node for this test bed is based on TI CC2530 application board [11]. The CC2530, depicted in Figure 3, is true system-on-chip (SoC) solution for 802.15.4 application based on SmartRF05 Evaluation Board. It combines the 2.4 GHz RF transceiver with 8051 MCU, in system 256 KB programmable flash memory, 8 KB RAM, batteries, and ambient/environment power source. In this environment, the application boards run Contiki, an open source operating system for memory efficient networked embedded system and wireless sensor networks. Contiki provides IP communication, both for IPv4 and IPv6, thanks to the embedded uIPv6 subsystem. The latter is an implementation of an IPv6/6LoWPAN stack, able to transmit IPv6 packets using the IEEE 802.15.4 radio of CC2530 chip. In our home automation system, this node has connections for digital ambient light sensors ISL 29023, an integrated ambient and infrared light-to-digital converter I^2C interface. In normal operation, typical current consumption of this sensor is 70 μA, and the power consumption can be reduced to less than 0.3 μA when powered down.

6LoWPAN-based home nodes have specific hardware characteristics and limitations. Most of these nodes have limited available energy. In our case, although AA batteries that provide the power to the 6LoWPAN-based home nodes are rechargeable, but, to save long periods of life without the need for periodically replacing its batteries, we need to design the 6LoWPAN home device with total energy independence. To solve this, additional components for power management and energy harvesting are needed. Thus, our 6LoWPAN-based home automation devices arepresented in the diagram as depicted in Figure 4. The voltage input from the energy harvester is used to charge the AA battery packs by the first stage DC-DC converter. Then battery voltage is supplied at a stable level to the 6LoWPAN home device main circuit. For power management purposes, the node also needs to continuously monitor the voltage and the current drawn from the battery pack, which is achieved by the energy measurement module.

3. Setting of Test Environment

6LoWPAN approach for home automation system is designed for control and monitoring of household devices. We are setting up a home automation scenario test environment to experiment interconnection between home automation devices, based on 6LoWPAN over IEEE 802.15.4, with an existing IPv6 network, based on Ethernet/Wi-Fi. To test interconnection between 6LoWPAN node and outside IPv6 network, we develop a light sensor remote and mobile control based on Android application. The remote user's communications transverse the internet until they reach our home gateway. After that, the communications are wirelessly transmitted to the 6LoWPAN-based home nodes. The application of this test bed has implemented IPv6 using Android API Inet6Address. The captures of our application are seen in Figure 5. The figure on the left showed the first screen of our application and the menu to select sensor nodes that will be monitored. The figure on the right showed

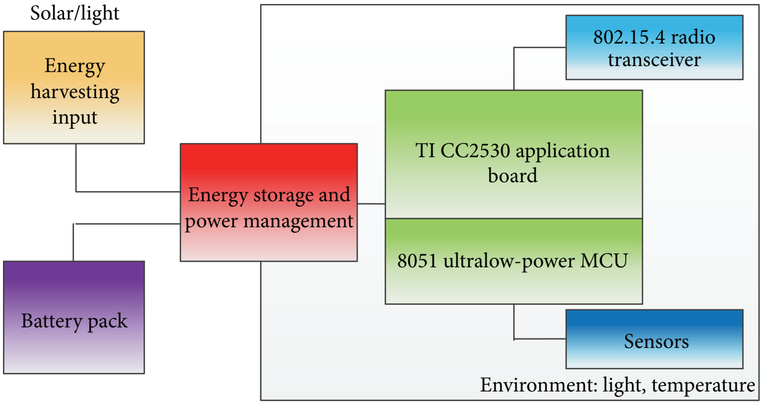
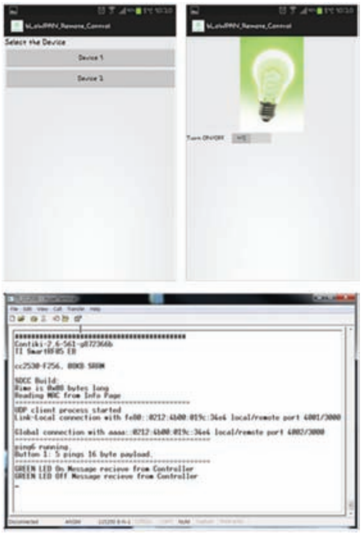

Solar/light
Energy harvesting input
Energy storage and power management
Battery pack
802.15.4 radio transceiver
TI CC2530 application board
8051 ultralow-power MCU
Sensors
Environment: light, temperature

Figure 4: 6LoWPAN-based home automation devices with energy harvesting.



```
##########################################
Contiki-2.6-561-g872366b
TI SmartRF05 EB

cc2530-F256, 8KB SRAM

SDCC Build:
Bime is 0x00 bytes long
Reading MAC from Info Page
==========================================
UDP client process started
Link-Local connection with fe80::0212:4b00:019c:34e4 local/remote port 4001/3000

Global connection with aaaa::0212:4b00:019c:34e4 local/remote port 4002/3000
================================================
ping6 running.
Button 1: 5 pings 16 byte payload.
================================================
GREEN LED On Message recieve from Controller
GREEN LED Off Message recieve from Controller
_
```

Figure 5: Application view.

on/off commands that sended a message to each 6LoWPAN home node.

As we know, to have home automation systems which have long periods of life, the power management is important. In our implementation, the home gateway is always connected to USB port, and no batteries are needed, but as it has been discussed, our 6LoWPAN home devices based on TI CC2530 need batteries as power source. To measure the current consumption of our devices the voltage of a resistor 10 Ω placed in series with the node was measured. It is determined as long as I is below 30 mA. However, the current consumption of our 6LoWPAN device is almost independent of the input voltage. Once the current is determined, the average current consumption can be found using the general formula

$$I_{\text{avg}} = \sum_{i=0}^{n} \left(\frac{T_i}{P_i} * I_i \right) + \left(1 - \sum_{i=0}^{n} \left(\frac{T_i}{P_i} \right) \right) * I_{\text{sleep}}, \qquad (1)$$

where T_i is time for which device consumes average current I_i, P_i is total time period for which average consumption is measured, I_{sleep} is current consumption while in sleep mode, and I_{avg} is average current consumption over period P_i.

Knowing I_i, I_{sleep}, and T_i we can find I_{avg} based on the period of active sequences. As a final step, calculate the total life time of our 6LoWPAN home device, and know that

$$\frac{\text{Battery capacity [mAh]}}{\text{Average current [mA]}} = \text{lifetime [h]}. \qquad (2)$$

The battery capacity will differ from one battery type to another. In this research, two AA sized Duracell Deluxe batteries are used, and the characteristics of this battery are shown in Table 1. The energy consumption and the power input of 6LoWPAN-based home automation node depend largely on the application and the sensor used. When the nodes are up and are running in the small home automation network, the average current consumption during the 292.5 ms is 34.6 mA. Using an ampere meter, the sleep current of the system was measured to be 4.8 μA. Now proceed to find the total average current consumption, based on (1), for the 5-second (= 5,000 ms) packet interval. Substituting in the formula values from Table 1 provide

$$\left(\frac{292.5}{5000} * 34.6 \right) + \left(1 - \frac{292.5}{5000} \right) * 0.0048\,\text{mA} = 2.212\,\text{mA}. \qquad (3)$$

Equation (2) can now be used to calculate the expected lifetime of the system:

$$\frac{2850\,\text{mAh}}{2.212\,\text{mA}} = 1288\,\text{hrs} = 53\ \text{days}. \qquad (4)$$

If the home device is configured to transmit one packet every 5 seconds, with small application acknowledgment and no data polling, the board can operate for 53 days with two AA Duracell Deluxe batteries. Hence, even whenour 6LoWPAN home devices have better possible working days with AA batteries than [12], energy harvesting technology is still needed to increases the robustness and availability of this system. Energy harvesting is the process by which energy is captured, derived, and stored for wireless autonomous devices from the surrounding environment such as solar

Table 1: Battery characteristics.

Max. charge voltage	1.5 V
Nominal voltage	1.2 V
Nominal capacity	2850 mAh
Standard charge	270 mA/16 h
Fast charge	2700 mA/1.1 h

Table 2: The most common energy harvesting sources.

Harvesting technology	Power density	Source of information
Photovoltaic cells	40 mW/cm^3	Borgoine, 2011 [16]
Piezoelectric	10–100 uW/cm^3	Chandrakasan et al., 2008 [17]
Thermoelectric ($\Delta t = 15$ deg K)	9.4 uW/cm^3	Yang et al., 2013 [18]
Acoustic noise (149 dB)	252 uW/cm^3	Kralov et al., 2011 [19]

power, thermal energy, and wind energy. In recent years this term has been applied mainly to sensor networks, where wireless sensor devices employ this process to replenish their energy resources [13, 14]. The implementation of energy harvesting in 6LoWPAN home automation has been introduced by [8, 13]. When they applied this technology to their 6LoWPAN-based temperature monitoring node, it makes their system became energy independent. However they did not give enough examination and detail energy consumption needed by 6LoWPAN-based home node while transmitting data application. We believe that this is very important due to the optimized system while, using the harvested energy, the power management design should achieve two fundamental requirements: energy-independent operation and node operation performance level [15].

We can approximate the sensor network with a closed energy system where each node has a total power output from the energy source $Ps(t)$ at time t, and a total energy being consumed at that time is $Pc(t)$. There are three cases which can be separated to model the energy behavior of a load and write the physical condition on energy conservation: harvesting system with no energy storage, harvesting system with ideal energy buffer, and harvesting system with nonideal energy buffer [15]. These conditions help us to derive requirements on $Ps(t)$ and $Pc(t)$ which allow energy-independent operation to be guaranteed.

The remaining energy E by the home automation node at any time t is given by sum of its total power output $Ps(t)$ minus its total energy being consumed $Pc(t)$ and can be estimated by the following formula:

$$E(t) = \int_0^t (Ps(t) - Pc(t))\, dt, \tag{5}$$

where $E(t)$ is remaining energy of the home automation node, $Ps(t)$ is total power output from energy harvesting source, and $Pc(t)$ is total energy consumption of the automation node.

A node is deemed energy independent if its remaining energy satisfies the following formula:

$$E(t) > 0, \quad \forall t > 0. \tag{6}$$

Depending on the application and location, a variety of sources for energy harvesting have been researched [13, 14, 20, 21] such as a solar power and thermal, vibration, and kinetic energy. Photovoltaic (solar) cell has the capability of converting light energy into electrical energy. Several research efforts have been conducted and so far have demonstrated that photovoltaic cells can produce sufficient power to maintain a microsystem. The vibrational harvesters use one of three methods: electromagnetic, electrostatic, or piezoelectric. Piezoelectric energy harvesting sources alter mechanical energy into electrical energy by straining a piezoelectric material to complete energy harvesting solution optimized for high-output impedance energy. This is well suited for low-power WSN nodes, since it accumulates energy over a long period of time to enable efficient use of short power bursts. Summary of the most common energy harvesting sources is depicted in Table 2.

All of the energy sources stated in the beginning have small energy density values compared to more classic energy sources, such as batteries. In the past, the use of radio transceivers often implied large amounts of power consumption. This is no longer the case today, as recent advances in the design of low-power electronics and energy storage have made wireless sensor networks a prime candidate for the successful integration of energy harvesting techniques. By analyzing the data from Table 2 we can see that solar cells offer the best efficiency while at the same time being an environmentally-friendly power source. In overall, photovoltaic energy conversion is a well-known integrated circuit compatible technology that offers the highest power output levels, when we compare it with the other energy harvesting mechanisms. However, its power output is strongly dependent on environmental conditions, that is, varying light intensity due to the placement (indoor/outdoor). This can be a suitable energy source for locations in which the availability of light to network nodes can be guaranteed to a sufficient degree and for which battery supply is impractical. For indoor lighting, the main source of energy is the lighting system itself. This may seem strange, but a solar energy harvesting system can collect the light and store the energy in a battery for use when it is dark, keeping a system running for year without available.

A 6LoWPAN home node that has energy harvesting as a supplement to the battery energy can maximize the life time and virtually run for an infinite amount of time without the need for periodically replacing its batteries. However, total power output needed from energy harvesting depends largely on energy consumption of the 6LoWPAN home node application and the sensor used. In order to maximize home node performance and satisfy energy independence, we used 3 V, 70 mA small polycrystalline silicon solar cell.

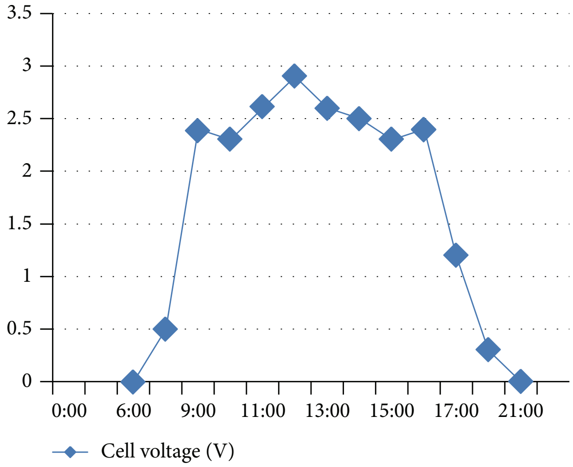


Figure 6: Photovoltaic cell voltage measurement.

Figure 7: Air packet capture view.

4. Evaluation

We established energy harvesting system that can harvest sufficient energy for the 6LoWPAN home device needs. To successfully power our nodes we did the following experiment: we measured the average output power on a fixed 1 kΩ load when the cell was in full sunlight. The result of our measurement is depicted in Figure 6. At noon, about 10 hour, the voltage condition reaches stable energy level at 2-3 V, so can generate electricity to give enough storage capacity for our energy harvesting system.

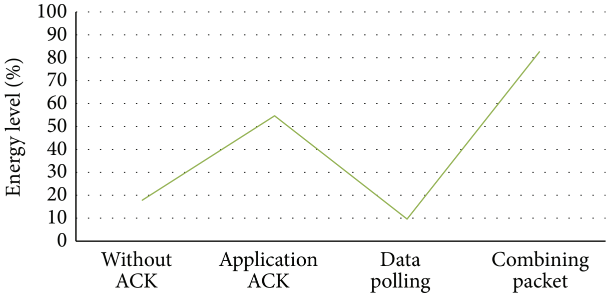
With an average power 87 mW, we can calculate the total energy harvested Es in one day by the solar cell:

$$Es = Ps \times ts = 87\,\text{mW} \times 10\,\text{h} = 3132\ \text{Joule}, \tag{7}$$

where Es is total energy harvested from source, Ps is total power output from harvesting source, and ts is total duration of energy harvested.

In addition to taking into account light sensors with an additional 1 mA at most and implying that no application acknowledgment is implemented, our 6LoWPAN home device consumed a maximum of 3 mA at 3 V from the power source. The total energy consumption Ec for the 6LoWPAN home node to run without pause for the duration of a single day will be

$$Ec = Vs \times Ic \times tc = 3\,\text{V} \times 4\,\text{mA} \times 24\,\text{h} = 1037\ \text{Joule}, \tag{8}$$



	Without ACK	Application ACK	Data polling	Combining packet
Current (mA)	2.212	6.774	1.195	10.232
Energy (J)	574	1756	310	2653

Figure 8: Energy consumption measurement.

where Ec is total energy consumed by 6LoWPAN home node, Vs is power source voltage, Ic is total average current consumption by 6LoWPAN node, and tc is total duration of energy energy consumed.

When the 6LoWPAN node is up and running in the home automation network, we use the packet sniffer to visualize the packet going over the air. Figure 7 shows air packet capture view of our 6LoWPAN home automation node transmission. The detail of various results and explanation of what is happening in the home automation device during a data transmission is depicted in Table 3.

At the same time, we also measure total average current consumption measurements of our 6LoWPAN home automation system during transmission of packets by using the oscilloscope. Summary results of total energy consumption for the data transmission without ACK, with application ACK, data polling, and combining data packets are shown in Figure 8. Our result proves that, given enough storage capacity and enough incident radiation, solar energy harvesting can power a 6LoWPAN home node for an indefinite amount of life time. Taking into account the fact that nodes employ power management in the software stack, alternating between long periods of sleep, and only short intervals when they are active, there is actually energy independence with excess energy produced. This additional energy is stored in the battery pack to be consumed during the night or on clouded days.

In Table 4, the detail comparison of our proposed solution with relevant works in wireless home automation system is shown. Our proposed system enables home users to check the home automation device status and control them remotely from the globe. In addition, the use of 6LoWPAN communications technology also helps lower expense of the system and decreases complexity of the home automation architecture. And then, by implementing solar energy harvesting we have 6LoWPAN-based home automation systems which have long periods of life without the need for periodically replacing their batteries.

5. Conclusion

We have presented in this paper our work to answer the issues mentioned at the beginning of this paper. We proposed the simple but reliable system that can control home

Table 3: Current consumption detail.

Event	Description	Duration [ms]	Current [mA]
(1)	Waking up	45	0.68
(2)	Processing data packet	25.6	30.2
(3)	Transmit packet and receive ACK	16.6	78.6
(4)	Request and receive ACK	21	98.6
(5)	Postprocessing packet	18	29.4
(6)	Request data (single poll)	29	94.3
(7)	Prepare to sleep	6	26.4
(8)	Set up radio	4.5	24.2
(9)	Start CSMA-CA	5.2	90.8
(10)	Switch from RX to TX	3	64.4
(11)	Switch from TX to RX	2.9	62.3
(12)	Prepare for deep sleep	21	24.2

Table 4: Feature in existing and proposed systems.

No.	System	Access		Complexity	Power management
		Direct control at home	Internet		
1	Bluetooth, Sriskanthan et al. 2002 [3]	√		Low	Home power
2	ZigBee, Gill et al. 2009 [4]	√		Low	Battery
3	6LoWPAN, Dörge and Scheffler 2011 [7]	√	√	Low	Battery
4	Our proposed system	√	√	Low	Energy harvesting

automation devices over 6LoWPAN. The usage of 6LoWPAN decreases complexity of architecture and low fiscal expense. Transparent Internet-based home device controlling also can be implemented. In our system, we proposed the use of TI CC2530, low energy consumption SoC solution for 802.15.4 application combined with Contiki, an open source OS for memory embedded systems which has connections with digital ambient light sensors ISL 29023. The use of 6LoWPAN home gateway provides smart platform while interconnecting the home automation network, based on IEEE 802.15.4 and 6LoWPAN, with existing IPv6 network, based on Ethernet.

The important features of our 6LoWPAN-based home automation system are long periods of life and energy efficient, without the need for periodically/every 53 days replacing its batteries. We designed the 6LoWPAN home automation device with total energy independence by using 3 V, 70 mA small polycrystalline silicon solar cell for energy harvesting. Given enough storage capacity and enough incident radiation, solar energy harvesting can power a 6LoWPAN home automation node for an indefinite amount of life time.

Acknowledgments

This research was financially supported by the Ministry of Education (MOE) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation.

References

[1] K. Bromley, M. Perry, and G. Webb, "Trends in smart home systems, connectivity, and services," 2003, http://www.nextwave.org.uk/.

[2] A. R. Al-Ali and M. Al-Rousan, "Java-based home automation system," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 498–504, 2004.

[3] N. Sriskanthan, F. Tan, and A. Karande, "Bluetooth based home automation system," *Microprocessors and Microsystems*, vol. 26, no. 6, pp. 281–289, 2002.

[4] K. Gill, S.-H. Yang, F. Yao, and X. Lu, "A ZigBee-based home automation system," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 422–430, 2009.

[5] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (LoWPANs): overview, assumptions, problem statemen, and goals," RFC 4919, 2007.

[6] M. Kovatsch, M. Weiss, and D. Guinard, "Embedding internet technology for home automation," in *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '10)*, September 2010.

[7] B. M. Dörge and T. Scheffler, "Using IPv6 and 6LoWPAN for home automation networks," in *Proceedings of the 1st IEEE International Conference on Consumer Electronics (ICCE '11)*, pp. 44–47, September 2011.

[8] D. Ş. Tudose, A. Voinescu, M.-T. Petrăreanu et al., "Home automation design using 6LoWPAN wireless sensor networks," in *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '11)*, pp. 1–6, June 2011.

[9] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*, Wiley, Chichester, UK, 2009.

[10] A. M. Efendi, A. F. P. Negara, S. Oh, and D. Choi, "A design of 6LoWPAN RPL border router with multi-uplink interface: ethernet and Wi-Fi," in *Proceedings of International Conference on Internet Services Technology and Information Engineering (ISTIE '13)*, 2013.

[11] "A True System-on-Chip Solution for 2.4-Ghz IEEE 802.15.4 and ZigBee Applications," CC2530, http://www.ti.com/lit/ds/swrs081b/swrs081b.pdf.

[12] A. P. Escola, *Development of a wireless sensor with 6LoWPAN support [M.S. thesis]*, Universitat Politecnica de Catalunya, 2009.

[13] V. C. Lee, *Energy harvesting for wireless sensor networks [M.S. thesis]*, Electrical Engineering and Computer Sciences, University of California, Berkeley, Calif, USA, 2009.

[14] R. Vullers, R. Schaijk, H. Visser, J. Penders, and C. Hoof, "Energy harvesting for autonomous wireless sensor networks," *IEEE Solid-State Circuits Magazine*, vol. 2, no. 2, pp. 29–38, 2010.

[15] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transaction on Embedded Computing Systems*, vol. 6, no. 4, article 32, 2007, special section LCTES'05.

[16] N. Borgoine, "Harvest energy from a single photovoltaic cell," *Journal of Analog Innovation*, vol. 21, no. 1, pp. 1–6, 2011.

[17] A. P. Chandrakasan, D. C. Daly, J. Kwong, and Y. K. Ramadass, "Next generation micro-power systems," in *Proceedings of Symposium on VLSI Circuits Digest of Technical Papers (VLSIC '08)*, pp. 2–5, June 2008.

[18] M.-Z. Yang, C.-C. Wu, C.-L. Dai, and W.-J. Sai, "Energy harvesting thermoelectric generators manufactured using the complementary metal oxide semiconductor process," *Sensors*, vol. 13, no. 2, pp. 2359–2367, 2013.

[19] I. Kralov, S. Terzieva, and I. Ignatov, "Analysis of methods and MEMS for acoustic energy harvesting with application in railway noise reduction," in *Proceedings of International Conference on Innovations, Recent Trends and Challenges in Mechatronics, Mechanical Engineering and New High-Tech Products Development (MECAHITECH '11)*, vol. 3, pp. 56–62, 2011.

[20] F. Yildiz, "Potential ambient energy-harvesting sources and techniques," *The Journal of Technologies Studies*, vol. 35, no. 1, pp. 40–48, 2009.

[21] R. J. M. Vullers, R. van Schaijk, I. Doms, C. Van Hoof, and R. Mertens, "Micropower energy harvesting," *Solid-State Electronics*, vol. 53, no. 7, pp. 684–693, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 232438, 14 pages
http://dx.doi.org/10.1155/2013/232438

Research Article

Thermoelectric Energy Harvesting for Building Energy Management Wireless Sensor Networks

Wensi Wang, Victor Cionca, Ningning Wang, Mike Hayes, Brendan O'Flynn, and Cian O'Mathuna

Microsystems Group, Tyndall National Institute, Cork, Ireland

Correspondence should be addressed to Wensi Wang; wensi.wang@tyndall.ie

Received 12 March 2013; Accepted 14 May 2013

Academic Editor: Al-Sakib Khan Pathan

A thermoelectric energy harvester powered wireless sensor networks (WSNs) module designed for building energy management (BEM) applications is built and tested in this work. An analytic thermoelectric generator (TEG) electrical model is built and verified based on parameters given in manufacturer data sheets of Bismuth Telluride TEGs. A charge pump/switching regulator two-stage ultra-low voltage step-up DC/DC converter design is presented in this work to boost the <0.5 V output voltage of TEG to usable voltage level for WSN (3.3 V). The design concept, device simulation, circuits schematic, and the measurement results are presented in detail. The prototype device test results show 25% end-to-end conversion efficiency in a wide range of input temperatures/voltages. Further tests demonstrate that the proposed thermoelectric generator design can effectively power WSN module which operates with a 1.7% duty cycle (5.8 seconds measurement time interval) when the prototype is placed on a typical wall-mount heater (60°C surface temperature). The thermoelectric energy harvesting powered WSN demonstrates duty cycles significantly higher than the required duty cycle for BEM WSN applications.

1. Introduction

Ubiquitous computing and short range wireless communication have been developing rapidly since 1990s. Evolving from basic point-to-point radio frequency communication, wireless communication technology now provides reliable data links within ad hoc networks. In the similar period of time, the developments of advanced and low cost sensors significantly increase the applications of modern sensing systems. The combination of wireless communication and sensor technologies has made it possible to transform the conventional large scale monitoring equipment into smart wireless sensor node within ad hoc networks.

Wireless sensor networks (WSNs) technologies have become a research focus in recent years for wide range of applications. One important application of WSN is within the building energy management (BEM) area. Energy consumption of commercial and residential buildings is responsible for 40% of energy usage in the USA since mid-2000s [1]. Many have suggested that the utilization of building energy management systems potentially leads to 25–40% energy savings from smart control of heating, ventilation, air conditioning (HVAC), and lighting [2–4]. Applications of WSN for building energy management system have been addressed in [5–8].

In these studies, one issue that has been frequently discussed for BEM WSN development is the limited lifetime of wireless sensor modules (also known as "mote") due to battery energy capacity. With increased number of motes in the WSN, the mandatory requirement to replace battery for several hundreds even thousands of WSN motes significantly increases the maintenance cost and reduces system reliability.

Energy harvesting methods have been proposed to scavenge ambient energy in order to "self-power" WSN systems. Energy harvesting technologies present a solution to supply WSN mote with infinite ambient energy instead of locally stored battery energy. Various types of ambient energy sources have been proposed for energy harvesting, such as indoor and outdoor photovoltaic energy [9, 10], mechanical vibration energy [11, 12], and thermoelectric energy [13–16].

In BEM applications, a large number of heating devices in the HVAC units, boiler, water heaters, and hot water pipes exist in most commercial/residential buildings. These potential heat sources with 50–100°C temperature provide ideal energy sources for building energy monitoring systems.

Utilization of these thermoelectric energy sources may lead to the highly desired "deploy-and-forget" WSN; that is, once the energy harvesting powered mote is deployed onto the heat source, the WSN becomes self-powered and achieves power autonomy. However, the implementation of thermoelectrical energy harvesting in order to achieve the continuous and maintenance-free WSN mote operation is significantly constrained by the ultra-low voltage (less than 0.5 V), small power (1 mW or sub-1 mW), and device size limitation of thermoelectric generator (TEG).

Despite these challenges, thermoelectric energy harvesting technologies are developing at fast speed in recent years. Many works addressed the high ZT figure of merit thermoelectric generator using MEMS or nanotechnology fabrication processes [17–19]. Ultra-low voltage DC/DC converter for thermoelectric generator applications has also been proposed in [14, 16]. However, thermoelectric energy harvester system level design for wireless sensor module is an area less addressed. Although the thermoelectric energy harvester systems presented in [20–22] give a well-defined general review and analysis, these works lack the important details on power management circuit design and components selection.

The purpose of this paper is to demonstrate a practical methodology concerning the thermoelectric energy harvester design for WSN mote in real-world BEM applications. This method firstly characterizes the power consumption profile of WSN mote. The thermoelectric energy harvester model is then used to estimate the TEG device size and configurations for the WSN power consumption. Associated power management circuit is then designed towards high conversion efficiency with TEG configuration. This methodology builds and optimizes energy harvester based on real-world WSN mote power consumption. This application-oriented design concept incorporates more realistic design considerations when the device is deployed in real-world conditions.

This paper introduces a thermoelectric generator electrical characteristics model for low temperature applications (hot side temperature <100°C, cold side temperature is room temperature with passive heat sink cooling) and its results verification based on a custom manufactured Bismuth Telluride (Bi_2Te_3) TEG module.

The main contribution of this paper lays in the power management circuit design for ultra-low voltage DC/DC conversion: a two-stage DC/DC converter circuit with ultra-low start-up voltage charge pump and switched-mode boost converter is proposed to step up the minimal 250 mV TEG voltage output. The detailed circuit design and components selection are presented. Energy storage unit and associated output power regulator circuit are introduced to complete the thermoelectric energy harvester design. The power consumption of Tyndall Zigbee WSN mote is also presented and compared to the power generated from thermoelectric energy harvester. Finally, the thermoelectric energy harvesting powered WSN prototype device and its evaluation results are presented. The energy flow and conversion efficiency in each stage of power conversion are presented based on measurements made on the prototype.

The rest of the paper is organized as follows. In Section 2, background and related works in the area of thermoelectric energy harvesting are introduced and compared. Section 3 shows the proposed system architecture of thermoelectric energy harvester. Section 4 addresses the detailed design issues of thermoelectric energy harvesting powered WSN mote system. Section 4.1 presents the energy consumption of WSN mote designed in Tyndall and its typical power profile. Section 4.2 shows the proposed TEG simulation model for room temperature applications and its verifications. Section 4.3 discusses the power management circuits design and implementation. Section 4.3 also introduces the energy storage unit and the output power regulator circuits. Section 5 presents the TEG prototype device and the experimental results. Section 6 summarizes the conclusions of this paper.

2. Background and Related Works

Thermoelectric energy harvesting is based on the Seebeck effect which directly converts temperature difference into electricity. The structure of a typical vertical TEG module is illustrated in Figure 1. When a temperature difference is applied cross the P-N types of TEG couples, voltage potential is generated on the the TEG couple. The thermoelectric material performance is usually measured in thermoelectric figure of merit ZT [13].

Currently, a typical "high" energy conversion efficiency commercially off-the-shelf TEG module features a ZT around 0.6-0.7 in room temperature. This was achieved by $Bi_xSb_{2-x}Te_3$ type material in the late 1960s [24]. MEMS based TEGs have been proposed in recent years in [17, 19]. The MEMS based TEG although it adopts the similar material used in conventional machined TEG, due to the large numbers of the thermo-couples, the power output of MEMS based TEG increases significantly.

Designated for powering miniaturized wireless sensor nodes, the TEG module and its heat sink are inherently small in form factor. The temperature difference available on a conventional heat sink in such condition is also small (<10°C from a <100°C heat source). With relatively low temperature difference and limited thermocouples size, further power conditioning is essential to the energy harvesting system design.

Dalola et al. presented a TEG powered sensor and transmitter circuit in their paper [25]. Charge pump converter is used as the only DC/DC conversion circuit. Ramadass and Chandrakasan proposed an advanced thermoelectric power management solution [26]. A DC/DC converter with minimum start-up voltage at 35 mV is designed and implemented with a 0.13 μm CMOS process. A manual controlled switch is required to start up the DC/DC converter. Carlson et al. demonstrates a 20 mV start-up voltage DC/DC converter in [14]. Although this DC/DC converter achieves a 75% maximum converter efficiency, it requires an external voltage source to initialize the start-up circuit.

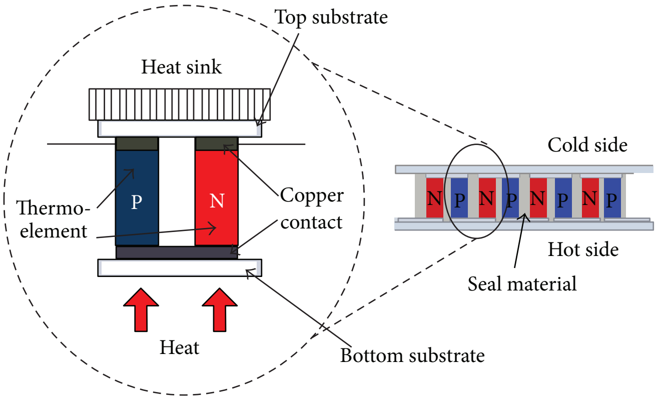


FIGURE 1: Seebeck effect in thermoelectric energy harvesting.

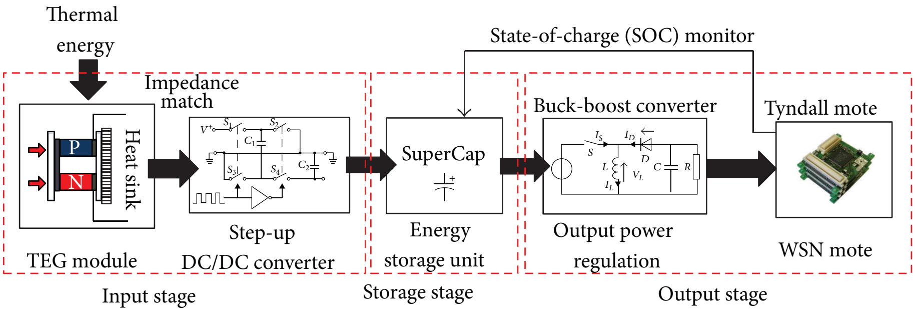


FIGURE 2: Block diagram of thermoelectric energy harvesting powered wireless sensor network.

Linear Technology released an energy harvesting power management chip LTC3108 [27]. It achieves top efficiency at 40% using a 1:100 transformer setup. However, for input voltage at 0.2 V or higher, the conversion efficiency decreases to less than 10% when output voltage is set to 4.5 V (1:100 transformer setup). Texas Instruments also released an energy harvesting power management chip BQ25504 for thermoelectric energy harvesting [28]. It features a minimal start-up voltage of 40 mV. BQ25504 requires a start-up current of several mA in order to obtain 1.8 V power supply voltage on the storage capacitor. During this start-up phase, this converter operates in cold start mode with low conversion efficiency. For many 1 mW or sub-1 mW energy harvesting applications, it is possible that the harvested power in the cold start phase is not sufficient to charge the storage capacitor to its required voltage threshold, which potentially leads to unsuccessful start-up, and the system cannot enter the normal operation mode.

3. Thermoelectric Energy Harvesting Powered WSN System Architecture

A complete thermoelectric energy harvesting powered WSN system consists of five subsystems as shown in Figure 2.

(1) Thermoelectric generator (TEG): main design issues are type of material, number, size, and connection configuration of thermocouples.

(2) Ultra-low voltage step-up DC/DC converter: the output voltage of most of TEGs is less than 500 mV. Conventional boost converter and charge pump cannot provide the low start-up voltage required by the TEGs. DC/DC converter with lower minimal start-up voltage is essential for thermoelectric energy harvesting.

(3) Energy storage unit: the conventional rechargeable battery, which requires relatively complex charging circuit and has a limited lifetime and charge cycles, is not suitable for most low power energy harvesting applications. Electrical double layer capacitors (also known as supercapacitors or SuperCaps), which only require a simple charging circuitry and have a long operation lifetime (>10 years), are considered to be the preferred energy storage unit (ESU) in energy harvesting.

(4) Output power regulation: output voltage regulator is used to extract "usable" energy from energy storage unit. For SuperCap type ESU with a changeable output voltage, the output power regulation is important to obtain a constant voltage to the WSN mote. In addition, with a buck-boost converter topology, the output voltage regulator also increases the amount of stored energy that can be used for WSN mote.

(5) WSN mote: the energy harvester design is largely based on the power consumption and other electrical

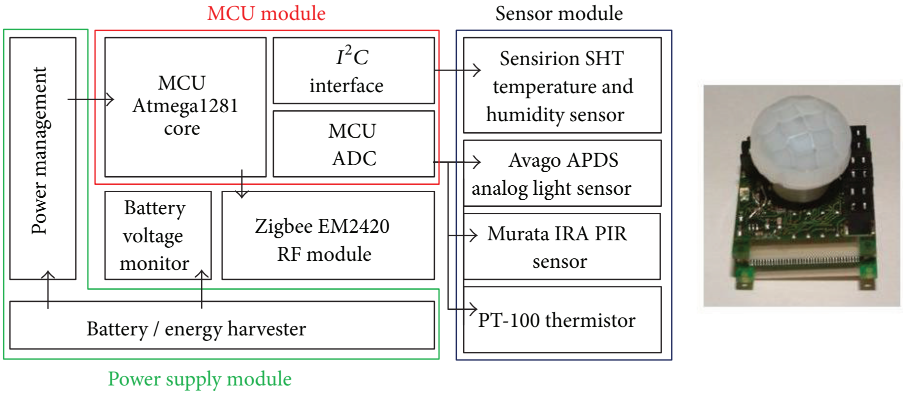


Figure 3: Wireless sensor node for building energy management application [23].

characteristics of WSN mote. The power consumption analysis of WSN mote is an essential step before energy harvester design.

In addition, these building blocks of thermoelectric energy harvester are not isolated components but interrelated subsystems. Several design considerations and challenges need to be addressed in the practical design and implementation of thermoelectric energy harvesting powered mote.

(1) The energy equilibrium between mote power consumption and thermoelectric energy harvester power generation must be achieved to ensure a continuously operating WSN mote.

(2) The impedance matching between TEG and DC/DC conversion has direct impact on the system power conversion efficiency.

(3) The energy storage unit and power consumption of WSN mote will determine the lifetime of mote when thermal energy source is temporarily unavailable.

Detailed design and analysis of each subsystem and the relationships between the aforementioned subsystems are presented in the following sections.

4. Design and Optimization of Thermoelectric Energy Harvesting Powered WSN Mote

4.1. Power Consumption of Wireless Sensor Module. A typical wireless sensor network for BEM applications consists of a number of wireless sensor nodes (motes). For each mote, it normally features (1) microcontroller unit (MCU); (2) wireless communication unit; (3) sensors with digital/analog interfaces; and (4) power supply.

The Tyndall mote [29] layout and implementation for BEM application is shown in Figure 3. The sensor interfaces are I^2C bus for digital sensors and 10-bit analog-to-digital converter (ADC) for analog sensors. The microcontroller adopted in the design is an Atmel1281, and the RF module is CC2420 radio chip.

WSN mote features high power consumption (10–100 mW) in active mode and low power consumption in sleep mode (10–50 μW). Thus, WSN mote often operates in active/sleep duty cycles in order to reduce average power consumption. During operating cycles, each mode of operation has an intrinsic power consumption value. The main operating modes are

(i) initialization and clear channel assessment (CCA) mode,

(ii) sensing mode (sensors are active and being sampled through the ADC),

(iii) preamble and payload Tx (RF transmit) mode,

(iv) acknowledgement (ACK) Rx (receiving) mode, and

(v) sleep mode.

In addition to the power consumption profile, the average power consumption of the WSN mote is also determined by the active mode duty cycle D. The average power consumption P_{Avg} can be expressed as

$$P_{\text{Avg}} = P_{\text{act}} \cdot D + P_{\text{slp}} \cdot (1 - D) . \qquad (1)$$

The power consumption values for the Tyndall mote shown in Figure 4 are listed in Table 1.

In order for a WSN mote to operate indefinitely, the average power consumption needs to be lower than the average harvested power, and this can be controlled by adjusting the duty cycle(s), D, of the application. Furthermore, the capacity of the power source (battery or supercapacitor) needs to be large enough to support peak power consumption, for example, the power consumption during RF transmission.

Based on the power consumption and duty cycle characteristics of WSN mote, thermoelectric energy harvester is designed for Tyndall mote to achieve power autonomy with large duty cycle.

4.2. Thermoelectric Generator Module Design and Electrical Model. The basic thermoelectric effect, Seebeck effect, describes a phenomenon that generates voltage difference when temperature gradient is applied across two series connected dissimilar materials as shown in Figure 1. The equivalent circuit for the thermoelectric voltage generation is illustrated in Figure 5.

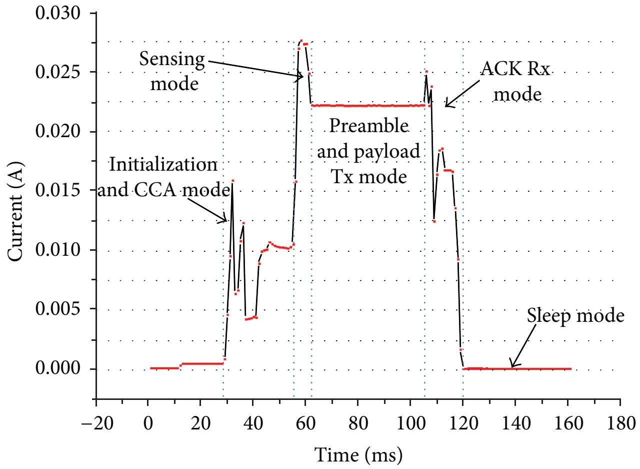


Figure 4: Measured current consumption of main operating modes (Tyndall WSN mote, Tx transmitting power = 0 dBm).

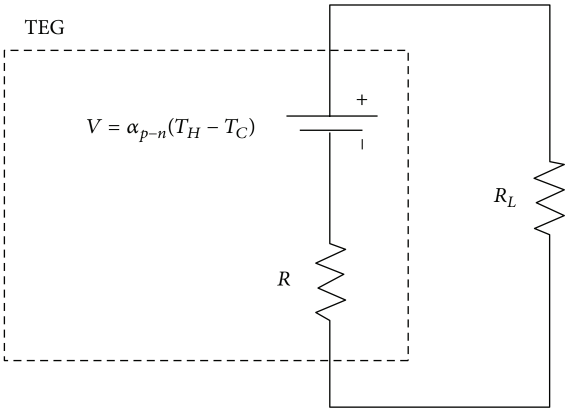


Figure 5: Basic equivalent circuit of thermoelectric generator.

The voltage generated from the single TEG pair is

$$V_L = \alpha_{p-n}\Delta T \times \frac{R_L}{R_L + R}, \tag{2}$$

where V_L is the load voltage and α_{p-n} is the Seebeck coefficient; it is the difference between positive and negative Seebeck coefficient in p-type and n-type materials. R and R_L are the internal resistance of TEG and load resistance, respectively. ΔT is the temperature difference between the two sides of TEGs. For this single pair TEG, the output power P_L can be expressed as

$$P_L = R_L \cdot \left(\frac{\alpha_{p-n}\Delta T}{R_L + R}\right)^2. \tag{3}$$

Single pair of thermocouples can only generate very limited voltage and power. In most cases, thermoelectric module consists of large number of thermocouples to increase the voltage and power output. With increased number (normally several hundreds) of thermo-couples, the interconnections are mostly facilitated using copper contact. In addition, ceramic substrates are used to physically support the TEGs. The thermoelectric generator layout and key parameters is shown in Figure 6.

Table 1: The power consumption of a Tyndall BEM mote at 3.3 V (sleep mode is 300 seconds, that is, 5-minute measurement interval).

Mode	Symbol	Power (mW)	Time (mSec)	Energy (mJ)
Init & CCA	P_{IC}	27.1	27	0.73
Sensing	P_S	82.5	5	0.41
RF Tx	P_{Tx}	73.2	42	3.07
RF Rx	P_{Rx}	58.5	13	0.76
Active mode total	P_{Act}	57.1	87	4.97
Sleep	P_{slp}	0.033	300,000	9.9

A typical TEG module [30] with precision machined thermo-couples is shown in Figure 7.

The Bi_2Te_3 based thermocouple has a thermal conductivity λ at approximately 1.5 W $\cdot$ $m^{-1}K^{-1}$. The ceramic substrate thermal conductivity λ_S is 120–150 times larger at 180 W $\cdot$ $m^{-1}K^{-1}$. The thickness of substrate is around 0.3–0.5 mm. With both upper and lower substrates, the total thickness is 0.6–1 mm. The impact on heat transfer from the thick substrates is no longer negligible. Considering the substrate thermal conductivity, the actual temperature difference $\Delta T'$ on the thermo-couple is less than the measured temperature difference ΔT.

The temperature difference across the thermo-couples is illustrated in Figure 8. The actual temperature difference on the thermo-couples is expressed as

$$\frac{\Delta T'}{\Delta T} = \frac{1}{1 + 2(\lambda/\lambda_S) \cdot (L_S/L)}, \tag{4}$$

where L and L_S are the thermo-couple length and substrate height, respectively. It is worth noting that (4) only considers the heat transfer within thermoelectric materials, but the heat transfers at material interfaces are not considered in this equation.

Based on this heat transfer model and temperature difference $\Delta T'$, the voltage output of N pairs of thermo-couples module can be expressed as

$$V = \frac{N \cdot \alpha_{p-n} \cdot \Delta T'}{R + R_L} \cdot R_L. \tag{5}$$

With increased number of thermo-couples, the internal resistance R also increases due to series connection of large number of thermo-couples and copper contacts. The resistance of the TEG module with N pairs of thermo-couples is shown as

$$R = N \cdot \left(\frac{\rho \cdot L}{A} + \frac{2\rho_C L_C}{A_C}\right), \tag{6}$$

where ρ is thermo-couple electrical resistivity, ρ_C is the (copper) contact electrical resistivity, A is the thermo-couples cross-section area, and A_C is the contact (vertical) cross-section area. Derived from (5) and (6), the analytic model

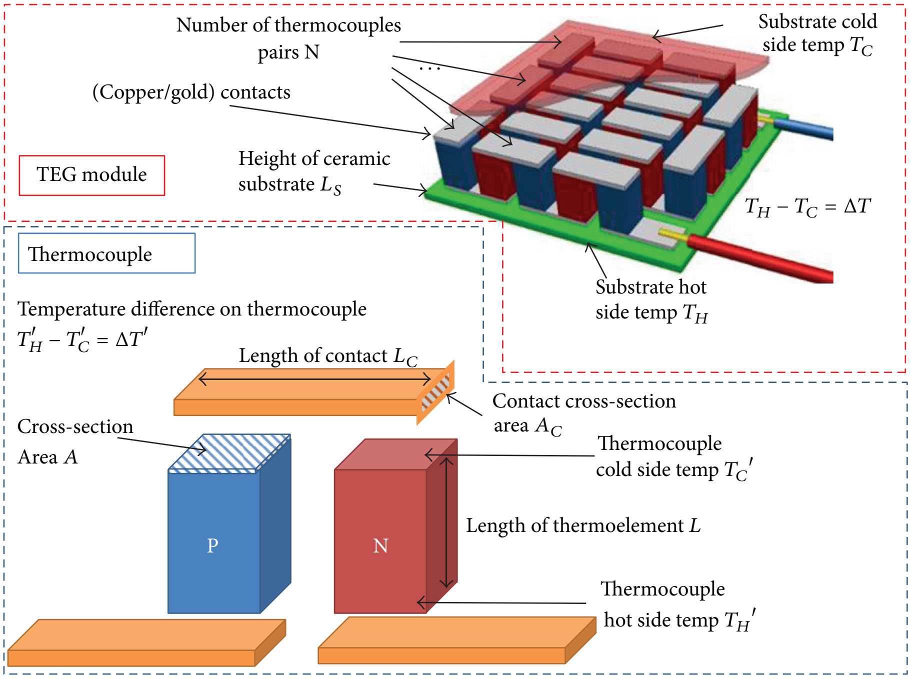

Number of thermocouples pairs N
Substrate cold side temp T_C
(Copper/gold) contacts
TEG module
Height of ceramic substrate L_S
$T_H - T_C = \Delta T$
Substrate hot side temp T_H
Thermocouple
Temperature difference on thermocouple $T'_H - T'_C = \Delta T'$
Length of contact L_C
Contact cross-section area A_C
Cross-section Area A
Thermocouple cold side temp $T_C{}'$
P
N
Length of thermoelement L
Thermocouple hot side temp $T_H{}'$

FIGURE 6: Thermoelectric generator (TEG) module and thermo-couple structures.

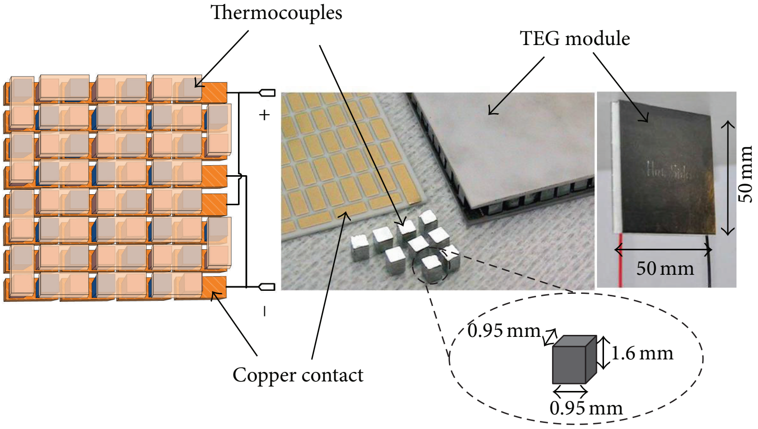


FIGURE 7: Thermoelectric generator layout, thermo-couple and thermoelectric module pictures.

for maximum output power simulation at matched load condition ($R = R_L$) is

$$P_{\max} = \frac{N \cdot \left(\alpha_{p-n} \Delta T\right)^2}{\left(\rho L/A + 2\left(\rho_C L_C / A_C\right)\right) \cdot \left(1 + 2\left(\lambda L_S / \lambda_S L\right)\right)^2}. \quad (7)$$

The thermoelectric module used to verify this model is provided by thermonamic [30]. This TEG module is a custom-designed module for low power generation. It adopts Bi_2Te_3 thermo-couples with ZT figure of merit around 0.7 at room temperature. For each thermo-couple, the cross-section area is 0.91 mm^2. A thin layer of ceramic substrate and a layer of heat-conductive foamed carbon thermal pad are applied on each side of the module. The carbon thermal pad is used to increase the heat transfer from the heat source to the module. The total thickness of the module is 3.4 mm, while the thickness of ceramic layer and copper contacts is 0.9 mm on each side. The height of the thermo-couple is 1.6 mm. In this basic unit, the total number of thermo-couple pairs is 127 (16 × 8 array with 1 thermo-couple pair removed to accommodate contact leads). This custom-designed module can be manufactured into configurations with 16 pairs × N_{col}, where N_{col} is the number of column and N_{col} is a multiple of 2. The power factor for both N type and P type thermo-couple is approximately 36 μW/cmK^2. The main parameters of this module are summarized in Table 2.

A series of tests were conducted in order to verify this analytic electrical simulation model. The test setup is illustrated in Figure 9. The heat source is a temperature controlled hot plate. The TEG module is cooled by a passive "fin" type heat sink. A PicoTech ADC-11/12 data acquisition system is set up to monitor the temperature on the outer

Table 2: Conventional machined thermoelectric module parameters summary.

Symbol	Definition	Value
A	Thermocouple cross-section area	$0.91\ \text{mm}^2$
N	Number of thermocouples pairs	127
L	Length of thermo-couples	1.6 mm
L_S	Height of ceramic substrate	0.9 mm
L_C	Length of contacts (between thermo-couples)	1.7 mm × 2 (upper and lower substrate)
A_C	Contacts cross-section area	$0.1\ \text{mm}^2$
λ	Thermocouple thermal conductivity	$1.5\ \text{Wm}^{-1}\text{K}^{-1}$
λ_S	Substrate thermal conductivity	$180\ \text{Wm}^{-1}\text{K}^{-1}$
ρ_C	Contacts electrical resistivity	$1.6 \times 10^{-8}\ \Omega{\cdot}\text{m}$
α_N^2/ρ_N	Power factor—N type	$36\ \mu\text{W/cmK}^2$
α_P^2/ρ_P	Power factor—P type	$36\ \mu\text{W/cmK}^2$

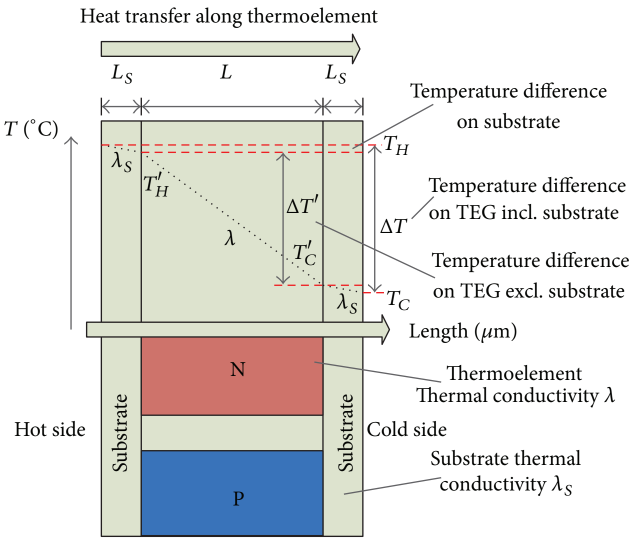

Heat transfer along thermoelement
L_S
L
L_S
T (°C)
Temperature difference on substrate
T_H
λ_S
T'_H
$\Delta T'$
λ
T'_C
ΔT
Temperature difference on TEG incl. substrate
Temperature difference on TEG excl. substrate
λ_S
T_C
Length (μm)
N
Thermoelement Thermal conductivity λ
Hot side
Substrate
Substrate
Cold side
Substrate thermal conductivity λ_S
P

Figure 8: Heat transfer within thermoelectric generator.

side of upper/lower substrates by using PT-100 temperature sensors. The I-V characteristics of the TEG module are measured by oscilloscopes and multimeters.

The room temperature during the experiments is approximately 20°C. Four hot side temperatures are applied, 50°C, 60°C, 70°C, and 80°C. The measured I-V characteristics of thermonamic TEG module are shown in Figure 10. The measured power-voltage characteristics are illustrated in Figure 11.

The load resistance tested in the characterization is between 1 Ω and 1 KΩ. The maximum power is obtained when the load resistance is 8.5 Ω. The analytic model and the measured results are compared in Table 3. The simulation in the thermonamic TEG shows a high level of consistency with the measured TEG electrical characteristics. The power simulation errors are less than 5% of the measurement values.

This analytic model is then realized in MatLab and used to simulate the output voltage and power in this work. It is also used to calculate the internal resistance of TEG.

4.3. Thermoelectric Energy Harvesting Power Management Circuit Design and Implementation. From the device characterization, one main issue discovered for thermoelectric energy harvesting is that the voltage of the TEG output is one order of magnitude lower than the WSN operating voltage. A voltage step-up circuit is required to boost the 100–500 mV input voltage to 2.5–4.5 V output voltage. This problem leads to two types of proposed power management methods, the first one uses ultra-low voltage boost converter with a large conversion ratio transformer; the second method uses low voltage charge pump and boost converter, a two-stage step-up design.

As introduced in [14], ultra-low voltage boost converter with a transformer conversion ratio 1 : 100 can step up input voltage as low as 20 mV to 2.0–4.5 V. The main concern in this type of design is the inherent low conversion efficiency for high ratio voltage step up.

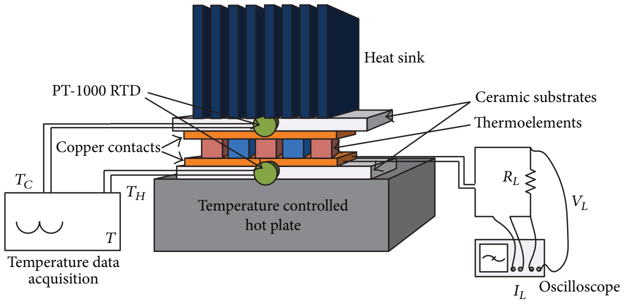


Figure 9: Thermoelectric generator electrical characterizations test configuration.

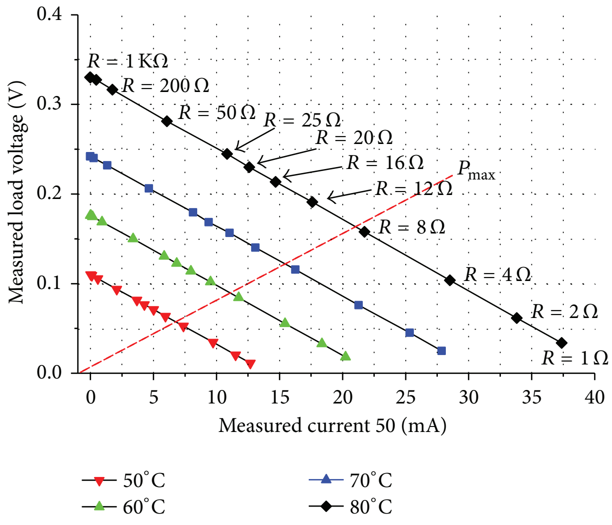


Figure 10: Measured I-V (current-voltage) characteristics of thermonamic TEG module.

Table 3: Thermonamic TEG characterizations results and simulation results at matched load.

Heat source temperature (°C)	50	60	70	80
Module temperature difference (°C)	2.5	4.0	5.5	7.5
Measured voltage (V)	0.053	0.084	0.116	0.158
Simulated voltage (V)	0.055	0.088	0.121	0.167
Measured power (mW)	0.346	0.886	1.676	3.117
Simulated power (mW)	0.353	0.920	1.722	3.265
Power simulation error (%)	**+1.8%**	**+3.7%**	**+2.7%**	**+4.6%**

This work adopts the second type of DC/DC conversion, the low voltage charge pump and boost converter two-stage step-up design. The start-up DC/DC converter is a Seiko Instruments S882Z-18 ultra-low voltage charge pump with a minimal start-up voltage at 0.25 V–0.3 V. The main DC/DC converter is Texas Instruments TPS61020 with a 0.9 V minimal start-up voltage.

In addition to the multiple stage power conversion, impedance matching is also considered in this design. Previous study confirmed that Bi_2Te_3 materials have small temperature coefficient of electrical conductivity within 50–100°C temperature range [31]. By revisiting the characterization of TEG power-voltage characteristics illustrated in Figure 11, it can be clearly seen that when temperature changes from 50°C to 80°C, the matched load resistance (equals to internal resistance) only changes less than 3%. The internal resistance of TEG is mainly determined by the thermocouples configuration. When the configuration is finalized, the TEG will have a near constant source resistance.

Changes on the duty cycle of the boost converter can effectively adjust the input resistance of the power management module. By matching the input resistance of power management circuit R_{IN} and the TEG source resistance T_{TEG} as shown in Figure 12, the energy transfer from TEG module to power management circuit is at maximum efficiency. As introduced in the last section, TEG source resistance can be accurately simulated based on the aforementioned TEG analytic electrical model.

Another design issue related to the thermoelectric energy harvesting power management is the energy storage unit and its output power regulation. In this work, supercapacitor is used as the energy storage unit. The porous structure of electrode material in supercapacitor effectively separated by electrochemical property of the electrolyte instead of thick physical dielectric layer ensures a large capacitance of several Farads. However, one issues that has not been fully addressed in the previous literature is the leakage current of supercapacitor. Due to the small power consumption of WSN mote, the 10–100 μA level leakage current is no longer negligible.

To investigate the leakage current characteristics, self-discharge tests were conducted on 4 different supercapacitors (Table 4). All the super-capacitors were precharged to the same voltage level. They were then isolated, and the voltage drops were monitored periodically by using a Pico Technologies ADC-11/12 data acquisition device. The data acquisition device has an output end impedance of 1 MΩ during measurements and a 10 MΩ impedance in idle mode, which effectively eliminated the current flow through the probe. The voltage drop is therefore only related to the self-discharge of the super-capacitors. The 24-hour results of the self-discharge tests are presented in Figure 13.

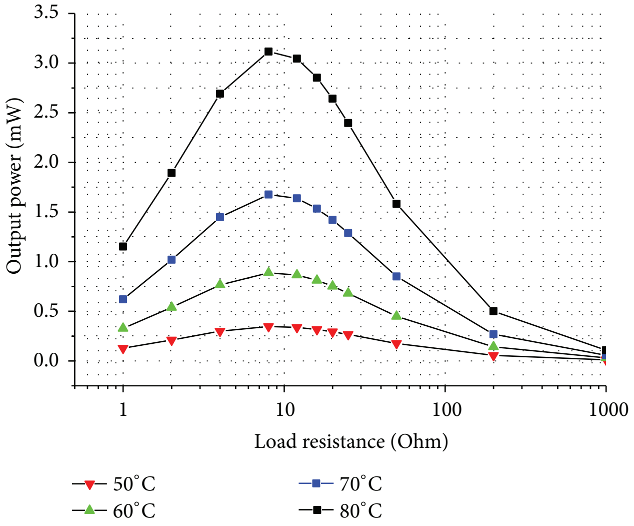


Figure 11: Measured power-voltage characteristics of thermonamic TEG module.

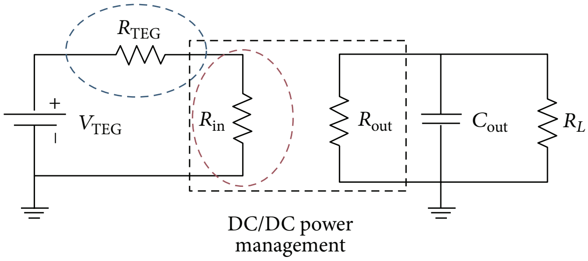


Figure 12: TEG and power management circuit impedance matching.

Table 4: Supercapacitor average leakage current and leakage correlation ρ_{leak}.

Mfr.	Capacity C_s (F)	Average leakage current I_{leak} (μA)	Leakage correlation $\rho_{leak} = I_{leak}/C_sV$ (μA/V·F)
Maxwell	5.00	35.6	3.54
EPCOS	4.10	19.7	2.41
Panasonic GoldCap	0.22	2.02	4.50
AVX BestCap	0.10	0.38	1.91

These results confirm that the self-discharge rates (SDR) of the super-capacitors are considerably higher than those of rechargeable batteries (5–10% monthly self-discharge rate). The super-capacitors have SDR ranging from 45% to 15% every 24 hours. Therefore, without an intermittently available

Table 5: Component selection for thermoelectric energy harvester.

Component name	Value
C_{IN}	47 μF
C_{CP}	1 μF
C_{SC} (SuperCap)	2.5 F
C_{OUT}	47 μF
L_2	4.7 μH
R_2	390 kΩ
R_4	270 kΩ
R_6	180 kΩ
C_{SU}	10 μF
C_1	47 μF
C_2	10 μF
L_1	22 μH
R_1	510 kΩ
R_3	1.6 MΩ
R_5	1 MΩ
$R_7 = R_8$	1 MΩ

energy source every few days, using the super-capacitors alone as a long-term storage solution is not feasible.

I-V characterizations were conducted to investigate the input resistance R_{IN} of power management module.

Based on these measurement results, a near linear correlation is found between leakage current $I_{\text{leak}}(t)$ and the product of $C_s \times V(t)$ at time t during the first 24 hours of self-discharge. The self-discharge mechanism in the early phase of this experiment is dominated by Faradic redox reactions generated ionic species concentration near the carbon surfaces [32]. The self-discharge due to this phenomenon significantly decreases after first 8–24 hours [33]. The leakage correlation ρ_{leak} is between 1.91 to 4.5 in various capacitors. For large supercapacitor (>5 F), the leakage current is of the same order as the current consumption of WSN modules. Clearly, the leakage characteristics of the super-capacitor have a significant impact on the operation time of the WSN mote.

Based on these design considerations, the complete schematic of the thermoelectric energy harvester is shown in Figure 14.

In order to extract most of the energy from the super-capacitor, a buck-boost converter Texas Instruments TPS61220 is used for the output voltage regulator. The input voltage range is between 0.7 V and 5.5 V. The output voltage is programmed to 3.3 V for Tyndall WSN mote. The detailed component selection and value are given in Table 5.

In this work a super-capacitor array C_{SC} with total capacitance of 2.5 F is used. The fully charged capacitor has an energy capacity of 31.25 Joules or 8.68 mWh. Becouse design requires larger capacity, super-capacitor with higher capacity or thin film battery [34] can also be used. The super-capacitor voltage is monitored by Atmega1281 micro-controller ADC through R7/R8 voltage divider.

The results of the characterization are shown in Figure 15.

For input voltage (TEG output voltage) between 0.25 V and 0.7 V, the input resistance R_{IN} is within a small range between 8 Ω and 12.2 Ω. By applying the TEG analytic electrical characteristics model, the output voltage and internal resistance of TEG module are simulated. In the BEM applications, the target size of the TEG is approximately 50 mm × 50 mm. A 36 × 32 thermo-element (one of the two piles in a thermo-couple) array with 54 mm × 48 mm dimension meets the form factor requirement. In terms of thermo-couples, the proposed TEG has 576 thermo-couples. These 576 thermo-couples can be arranged in several different configurations. The simulated TEG internal resistance R_{TEG} in different configurations is compared to input resistance R_{IN} of the thermoelectric energy harvester power management module in Figure 15.

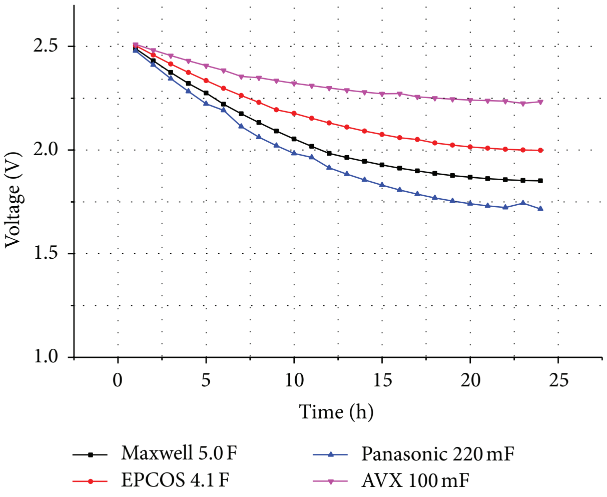


Figure 13: Supercapacitors self-discharge test over 24 hours. (Conditions: room temperature: 10°C; all samples have been fully charged and discharged for 100 times before the test.)

For 2 parallel-connected 288 thermo-couple pairs, R_{TEG} is 9.9 Ω. R_{TEG} in this configuration matches with power management module input resistance R_{IN} with a less than 20% error in the 0.25 V–0.75 V input voltage range. The impedance match error is less than 10% when the input voltage is between 0.25 V and 0.38 V. The verification of this matched impedance is presented in the implementation and experimental results section.

Figure 16 shows the TEG module output voltage simulation based on the analytic model. The simulation results show that for 2 parallel-connected 288 thermo-couple pairs configuration, the TEG output voltage is between 280 mV and 450 mV when the temperature difference on the substrates is 3–5°C. This output voltage is higher than the minimal start-up voltage of the charge pump and can start up the charge pump and the other circuits in the power management module.

5. Thermoelectric Energy Harvester Implementation and Experimental Results

Based on the proposed TEG module and power management circuit design, the device manufacturing and assembling

Table 6: TEG power management energy transfer.

Stage	Voltage (V)	Current (mA)	Power (mW)	Efficiency (%)
TEG output	0.25	17.0	4.08	
Charge pump Seiko S-882Z	1.48	0.797	1.18	28.9%
Boost converter TI TPS61020	3.25	0.338	1.097	92.9%
Buck/boost converter TI TPS61220	3.30	0.312	1.029	93.8%
End-to-end conversion	0.25 → 3.3 V	17.0 → 0.312 mA	4.08 → 1.029 mW	25.2%

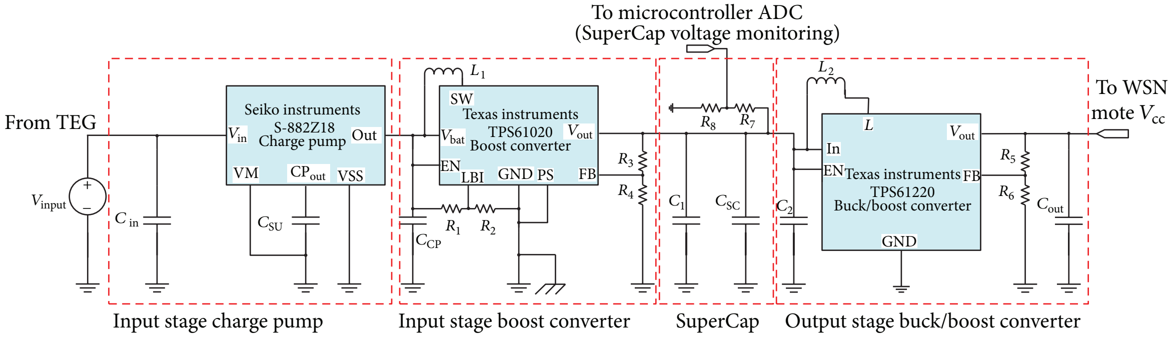


Figure 14: Schematic of thermoelectric energy harvester.

were conducted. Four thermonamic TEG modules are used to assemble the TEG module into 288 × 2 thermo-couple configuration. The complete prototype is shown in Figure 17. The form factor of the thermoelectric energy harvester powered WSN mote is 6 cm (L) × 5 cm (W) × 7 cm (H).

The prototype design also took the thermal dissipation on the PCB layer into consideration. The heat sink of the TEG is fastened to the PCB through a set of four long screws. This configuration allows typical indoor airflow to further cooling of the heat sink and generates higher temperature difference.

The viability of the TEG design and the application on WSN were tested through a set of experiments. The prototype was placed on hotplate in various temperatures to test the start-up performance, continuous operation efficiency, and energy storage charge time. The experiment result of the TEG start up at 60°C is shown in Figure 18.

The charge pump S882-Z starts at 0.25–0.3 V, and the voltage on storage capacitor of charge pump C_{SU} and the voltage on output capacitor of the charge pump C_{CP} start to increase. The charge pump is moving towards the target voltage 1.8 V. When C_{CP} voltage reaches 0.95 V (0.9 V minimal start up voltage + 50 mV hysteresis), the main boost converter TI TPS61020 starts to operate and steps up the 0.95 V input voltage to charge the super-capacitor. The output voltage regulator TI TPS61220 starts to operate when super-capacitor voltage reaches 0.7 V and steps up the input voltage to 3.3 V output voltage.

At 60°C hot side temperature, when the energy harvester reaches thermal static state, the output voltage of TEG module is measured at 0.25 V as shown in Figure 18. The open circuit voltage of TEG before power regulation is at 0.47 V. The output voltage 0.25 V is close to the theoretical maximum power point at half of open circuit voltage 0.235 V (0.47 V × 0.5). This proves the concept of impedance matching between the TEG source resistance and the input resistance of power management proposed in this work.

Further tests were conducted to investigate the conversion efficiency of this proposed thermoelectric energy harvester. The I-V characterization of the TEG module output power and the step-up DC/DC converter output power are presented in Figure 19.

Figure 20 shows the power-voltage characterizations of TEG output power and the step-up DC/DC converter output power. The matched load TEG maximum output power is measured at 4.08 mW. The output power of step up DC/DC converter (charge pump and switching regulator) thermoelectric energy harvester is measured at 1.1 mW.

Table 6 summarizes the energy transfer in the power management circuit when the hot side temperature is 60°C. With a 28.9% conversion efficiency, most of the power loss is due to the conversion energy loss in the ultra-low voltage charge pump. Both the boost converter (second stage of the step-up DC/DC conversion) and buck/boost converter (output voltage regulator) achieved conversion efficiency higher than 90%. The system end-to-end conversion efficiency is 25.2%.

It is worth noting that the super-capacitor(s) are not considered in this conversion efficiency summary. The thermoelectric energy harvester performance with super-capacitors is characterized separately. The super-capacitor charging results from 3 difference hot side temperatures are presented in Figure 21.

When charging the capacitive load, the average charging power P_{avg} during the complete charging phase (charged

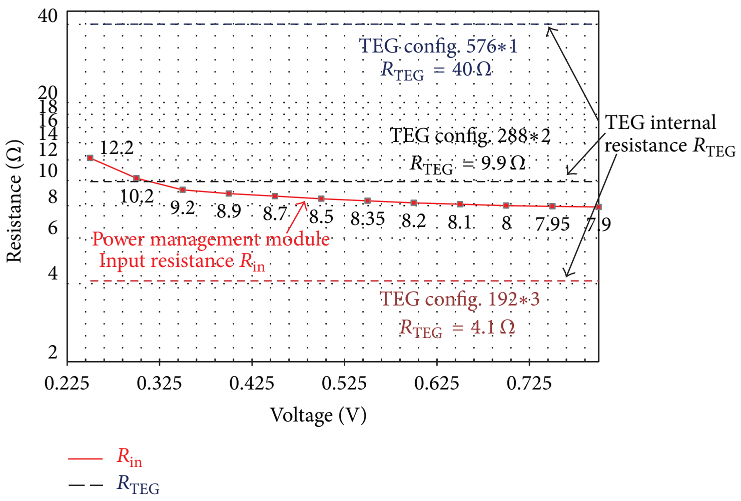


FIGURE 15: Thermoelectric energy harvester power management module input resistance and TEG internal resistance impedance matching.

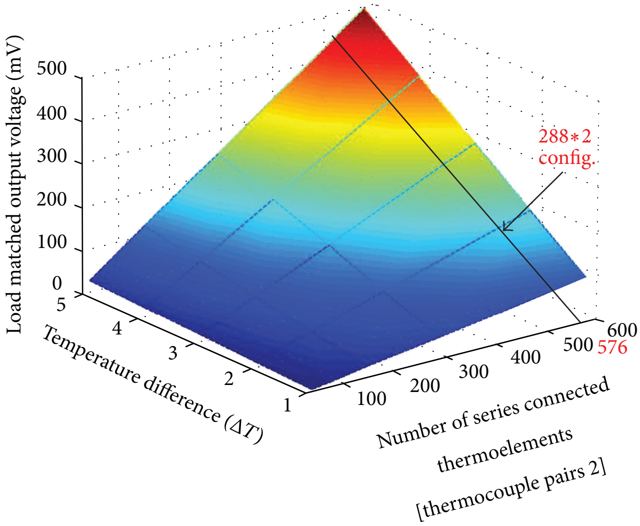


FIGURE 16: TEG module output voltage simulation.

from 0 V to the target voltage V_{target} 3.3 V) can be calculated as

$$P_{\text{avg}} = \frac{C_{\text{SC}} \cdot V_{\text{target}}^2}{2 \cdot T_{\text{chrg}}}, \tag{8}$$

where C_{SC} is the super-capacitor capacitance, T_{chrg} is the total charging time. Based on the measured result shown in Figure 21, the average charging power is 0.95 mW, 2.1 mW and 4.35 mW on 60°C, 70°C, and 80°C hot surfaces, respectively.

The thermoelectric energy harvester performance with WSN mote is then evaluated based on WSN power consumption and energy storage leakage power consumption. WSN mote can be programmed with different time intervals τ between two measurements (active mode). Since the active mode time T_{act} is generally constant for certain application (90 mSec in this application), the measurement time interval τ will determine the active mode duty cycle, D,

$$D = \frac{T_{\text{act}}}{T_{\text{act}} + \tau} \times 100\%. \tag{9}$$

Once the duty cycle D is programmed, the average power consumption can be calculated using (1). The WSN average power consumption and the energy storage leakage power are measured and illustrated in Figure 22. In order to continuously operate WSN mote from harvested power, the average power consumption must be higher than the thermoelectric energy harvester output power. For each thermoelectric energy harvester operating temperature, the harvested power has a minimal WSN mote measurements time interval and a maximum active mode duty cycle. The thermoelectric energy harvester powered WSN operation limits are shown in Figure 22.

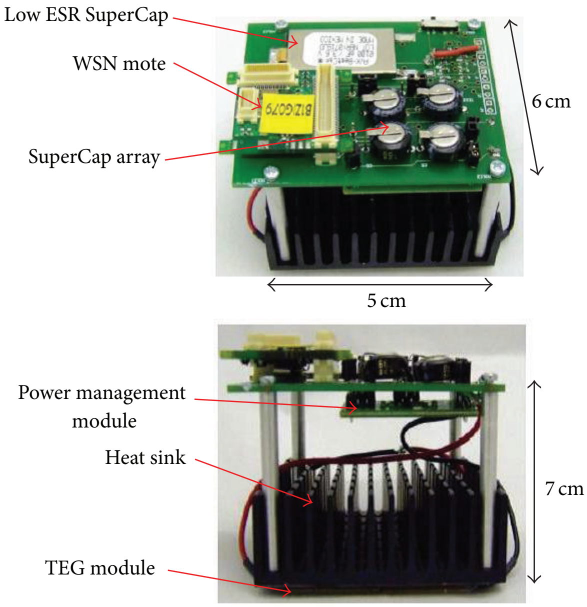


Figure 17: Thermoelectric energy harverting and power management module prototype for BEM applications.

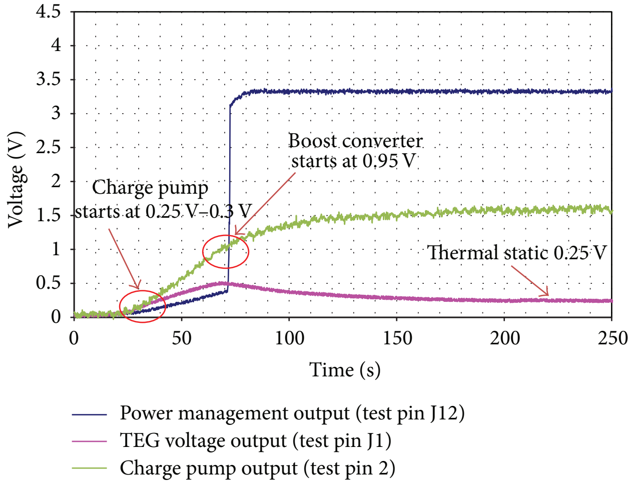


Figure 18: Thermoelectric energy harvester start-up test results.

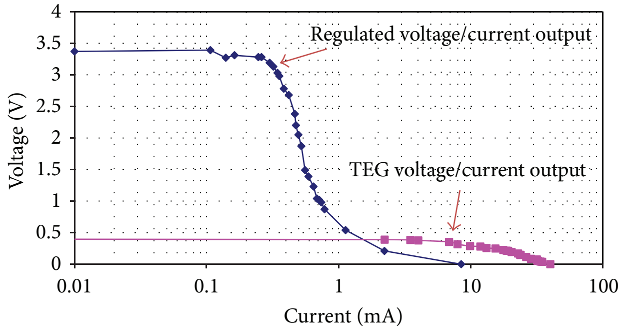


Figure 19: *I-V* characterization of the TEG module output power and the step-up DC/DC converter output power (hot side temperature 60°C).

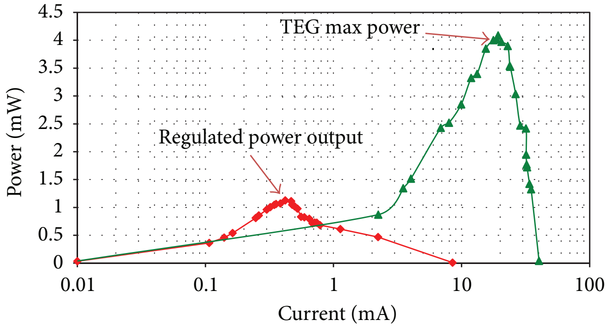


Figure 20: Power-voltage characterizations of the TEG module output power and step-up DC/DC converter output power (hot side temperature 60°C).

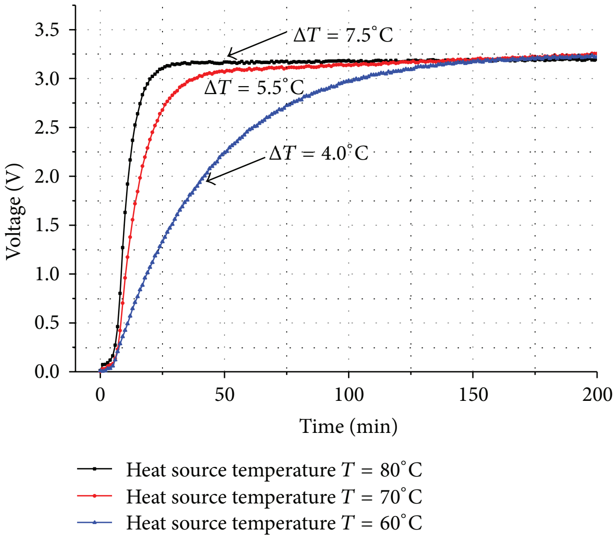


Figure 21: Thermoelectric energy harvester prototype charging.

For 60°C hot side temperature, the minimal measurement time interval is 5.8 seconds; that is, when placing the thermoelectric energy harvester on a 60°C heat source, the generated power allows WSN mote to make a BEM measurement and transmit the data every 5.8 seconds. For higher hot side temperatures, the minimal measurement time intervals are shorter. Most BEM applications (light intensity, temperature, relative humidity, etc.) require measurements time interval between 1 and 10 minutes. The thermoelectric energy harvester proposed in this work can effectively provide a "power-autonomous" power supply for BEM WSN motes when thermal energy is available.

When compared with recent work in the thermoelectric energy harvesting, this bulk TEG based thermoelectric energy harvester proposed in this work achieved the self-start capability suggested in [35] and can be deployed for long-term operation. The TEG generated power is measured at 0.95 mW with a 4°C temperature difference in this work, whilst in [35], 0.2 mW is generated with a 3.5°C temperature difference. In addition to the higher output power, the bulk

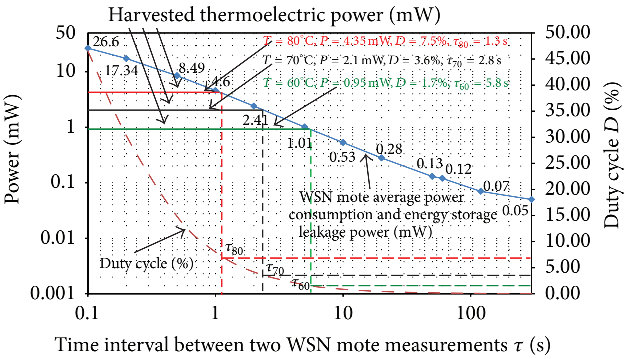


Figure 22: Thermoelectric energy harvester prototype charging supercapacitor.

TEG can be manufactured at lower cost than the MEMS TEG utilized in [35].

6. Conclusion

In this paper, low temperature thermoelectric energy harvester is investigated for the use of powering wireless sensor network. The main focus is on the energy conversion efficiency improvement from the prospective of power management and TEG design. This study characterized the Bi_2Te_3 based thermoelectric generator with hot side temperature between 50°C and 100°C. A TEG output power/voltage/resistance analytic model is created to simulate the device performance in low temperature energy harvesting. This simulation model shows high level of consistency with the measured results with maximum error less than 5% in the experiments.

Based on the characterizations and the simulation of the TEGs, power management circuits with emphasis on low voltage step up are investigated in the second half of this paper. In order to obtain a regulated output voltage from the less than 0.5 V low input voltage, multiple-stages voltage regulation is considered in this work. The power management adopted the charge pump/switching regulator two stages design to obtain lower conversion ratio on each stage. The charge pump starts up the voltage regulation when the TEG voltage is higher than 250 mV, whilst the boost converter starts up at 0.95 V.

It has been noticed that by adjusting the configurations of TEG, the source resistance and output voltage can be modified to match the power management circuit input impedance. Based on this method, the TEG was reconfigured based on the analytic simulation model with a 9.9 Ω source resistance, whilst the power management circuit input resistance is matched with the source resistance with less than 10% error (0.25 V–0.38 V input voltage). The device implementation consists of two 160 thermo-couple pairs TEG modules configured in the parallel-connected layout. At 60°C, the output voltage of TEG (input voltage of power management) is measured at 0.25 V, close to the 0.235 V maximum power voltage (half of the open circuit voltage 0.47 V).

Although the charge pump runs in a relatively low conversion efficiency of 28–30%, the high conversion efficiency (approximately 93%) of the second stage voltage regulator TPS61020 (from 1.0 V to 3.3 V) and output voltage regulator TPS61220 (approximately 94%) allows the entire power management circuit to operate at an overall efficiency close to 25%.

Several experiments were conducted to measure the performance of the TEG in different temperatures. For tests with heat source temperature higher than 60°C in room temperature environment, the power generated, regulated, and supplied from the prototype device is sufficient to operate WSN mote running in low duty cycles and reached the target of power autonomous operation with thermoelectric energy harvesting.

Acknowledgment

This work was supported by ENIAC/JU project 2010 (270722-2) ERG "Energy for a Green Society: from sustainable harvesting to smart distribution. Equipments, materials, design solution and their applications."

References

[1] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[2] Y.-J. Wen and A. M. Agogino, "Personalized dynamic design of networked lighting for energy-efficiency in open-plan offices," *Energy and Buildings*, vol. 43, no. 8, pp. 1919–1924, 2011.

[3] B. Roisin, M. Bodart, A. Deneyer, and P. D'Herdt, "Lighting energy savings in offices using different control systems and their real consumption," *Energy and Buildings*, vol. 40, no. 4, pp. 514–523, 2008.

[4] E. Mills, "Why were here: the 230 billion global lighting energy bill," in *Proceedings from the 5th European Conference on Energy Efficient Lighting*, Citeseer, 2002.

[5] Y.-J. Wen and A. M. Agogino, "Control of wireless-networked lighting in open-plan offices," *Lighting Research and Technology*, vol. 43, no. 2, pp. 235–248, 2011.

[6] G. G. Mueller, I. A. Lys, K. J. Dowling et al., "Wireless lighting control methods and apparatus," US Patent 7659674, 2010.

[7] H. Park, J. Burke, and M. B. Srivastava, "Intelligent lighting control using wireless sensor networks for media production," *KSII Transactions on Internet and Information Systems*, vol. 3, no. 5, pp. 423–443, 2009.

[8] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building-environment control with wireless sensor and actuator networks: centralized versus distributed," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3596–3605, 2010.

[9] W. Wang, T. O'Donnell, N. Wang, M. Hayes, B. O'Flynn, and C. O'Mathuna, "Design considerations of sub-mw indoor light energy harvesting for wireless sensor systems," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 6, no. 2, article 6, 2010.

[10] W. Wang, N. Wang, M. Hayes, B. O'Flynn, and C. O'Mathuna, "Power management for sub-mw energy harvester with adaptive hybrid energy storage," *Journal of Intelligent Material Systems and Structures*, 2012.

[11] R. H. Bhuiyan, R. A. Dougal, and M. Ali, "A miniature energy harvesting device for wireless sensors in electric power system," *IEEE Sensors Journal*, vol. 10, no. 7, pp. 1249–1258, 2010.

[12] S. P. Beeby, M. J. Tudor, and N. M. White, "Energy harvesting vibration sources for microsystems applications," *Measurement Science and Technology*, vol. 17, no. 12, article R175, 2006.

[13] G. J. Snyder, "Thermoelectric energy harvesting," in *Energy Harvesting Technologies*, pp. 325–336, 2009.

[14] E. J. Carlson, K. Strunz, and B. P. Otis, "A 20 mV input boost converter with efficient digital control for thermoelectric energy harvesting," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 741–750, 2010.

[15] G. Sebald, D. Guyomar, and A. Agbossou, "On thermoelectric and pyroelectric energy harvesting," *Smart Materials and Structures*, vol. 18, no. 12, Article ID 125006, 2009.

[16] J. P. Carmo, L. M. Goncalves, and J. H. Correia, "Thermoelectric microconverter for energy harvesting systems," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 3, pp. 861–867, 2010.

[17] H. Böttner, D. G. Ebling, A. Jacquot, J. König, L. Kirste, and J. Schmidt, "Structural and mechanical properties of Spark Plasma sintered n- and p-type bismuth telluride alloys," *Physica Status Solidi*, vol. 1, no. 6, pp. 235–237, 2007.

[18] A. I. Hochbaum, R. Chen, R. D. Delgado et al., "Enhanced thermoelectric performance of rough silicon nanowires," *Nature*, vol. 451, no. 7175, pp. 163–167, 2008.

[19] M. Strasser, R. Aigner, C. Lauterbach, T. F. Sturm, M. Franosch, and G. K. M. Wachutka, "Micromachined CMOS thermoelectric generators as on-chip power supply," *Sensors and Actuators*, vol. 114, no. 2-3, pp. 362–370, 2004.

[20] D. Samson, M. Kluge, T. Becker, and U. Schmid, "Wireless sensor node powered by aircraft specific thermoelectric energy harvesting," *Sensors and Actuators*, vol. 172, no. 1, pp. 240–244, 2011.

[21] C. Lu, S. P. Park, V. Raghunathan, and K. Roy, "Analysis and design of ultra low power thermoelectric energy harvesting systems," in *Proceedings from the 16th ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED '10)*, pp. 183–188, IEEE, August 2010.

[22] V. Leonov, "Human machine and thermoelectric energy scavenging for wearable devices," *ISRN Renewable Energy*, vol. 2011, Article ID 785380, 11 pages, 2011.

[23] W. Wang, R. O'Keeffe, N. Wang, M. Hayes, B. O'Flynn, and C. O'Mathuna, "Practical wireless sensor networks power consumption metrics for building energy management applications," in *Proceedings of the 23rd European Conference Forum Bauinformatik 2011, Construction Informatics*, Cork, Ireland, September 2011.

[24] G. Nolas, J. Sharp, and H. Goldsmid, *Thermoelectrics: Basic Principles and New Materials Developments*, vol. 45, Springer, 2001.

[25] S. Dalola, M. Ferrari, V. Ferrari, M. Guizzetti, D. Marioli, and A. Taroni, "Characterization of thermoelectric modules for powering autonomous sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 1, pp. 99–107, 2009.

[26] Y. K. Ramadass and A. P. Chandrakasan, "A battery-less thermoelectric energy harvesting interface circuit with 35 mV startup voltage," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 333–341, 2011.

[27] L. Datasheet, Linear Technology Corporation, Milpitas, Calif, USA, 2010.

[28] M. Pinuela, D. Yates, S. Lucyszyn, and P. Mitcheson, "Current state of research at imperial college london in rf harvesting and inductive power transfer," in *Proceedings of PowerMEMS*, pp. 41–44, 2010.

[29] K. Menzel, D. Pesch, B. O'Flynn, M. Keane, and C. O'Mathuna, "Towards a wireless sensor platform for energy efficient building operation," *Tsinghua Science and Technology*, vol. 13, no. 1, pp. 381–386, 2008.

[30] Thermonamic, "Modules for power generation," 2009, http://www.thermonamic.com/.

[31] H. J. Goldsmid, "The electrical conductivity and thermoelectric power of bismuth telluride," *Proceedings of the Physical Society*, vol. 71, no. 4, article 312, pp. 633–646, 1958.

[32] B. W. Ricketts and C. Ton-That, "Self-discharge of carbon-based supercapacitors with organic electrolytes," *Journal of Power Sources*, vol. 89, no. 1, pp. 64–69, 2000.

[33] M. J. Guan and W. H. Liao, "Characteristics of energy storage devices in piezoelectric energy harvesting systems," *Journal of Intelligent Material Systems and Structures*, vol. 19, no. 6, pp. 671–680, 2008.

[34] J. B. Bates, N. J. Dudney, B. Neudecker, A. Ueda, and C. D. Evans, "Thin-film lithium and lithium-ion batteries," *Solid State Ionics*, vol. 135, no. 1–4, pp. 33–45, 2000.

[35] Q. Huang, C. Lu, M. Shaurette, and R. Cox, "Environmental thermal energy scavenging powered wireless sensor network for building monitoring," in *Proceedings of the 28th International Symposium on Automation and Robotics in Construction (ISARC '11)*, pp. 1376–1380, Seoul, Korea, June-July 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 732652, 11 pages
http://dx.doi.org/10.1155/2013/732652

Research Article

A Nonclairvoyant Real-Time Scheduler for Ambient Energy Harvesting Sensors

Hussein El Ghor,[1,2] Maryline Chetto,[1] and Rafic Hage Chehade[2]

[1] *IRCCyN Lab, University of Nantes, 1 rue de la Noë, 44321 Nantes, France*
[2] *Lebanese University, IUT Saida, Saida 813, Lebanon*

Correspondence should be addressed to Hussein El Ghor; hussein.ghor@ul.edu.lb

Received 3 March 2013; Revised 22 April 2013; Accepted 25 April 2013

Academic Editor: Guangjie Han

Ambient energy harvesting also known as energy scavenging is the process where energy is obtained from the environment, converted, and stored to power small devices such as wireless sensors. We present a variant of EDF scheduling algorithm called EH-EDF (Energy Harvesting-Earliest Deadline First). Decisions are taken at run-time without having prior knowledge about the future energy production and task characteristics. We gauge the performance of EH-EDF by means of simulations in order to show its benefits. We evaluate and compare several variants of EH-EDF in terms of percentage of feasible task sets. Metrics such as average length of the idle times are also considered. Simulations tend to demonstrate that no online scheduler can reach optimality in a real-time energy harvesting environment.

1. Introduction

An algorithm is said to be nonclairvoyant if its scheduling decisions are taken at run-time with no prior knowledge about the characteristics of the future tasks [1]. Consequently, a nonclairvoyant scheduling algorithm is necessarily online. The problem of online scheduling in real-time systems has been a fertile ground for theoretical research for many years.

There are many real-time applications concerned with nonpredictability and consequently with nonclairvoyant scheduling. In that system, periodic and aperiodic tasks coexist. Periodic tasks typically arise from sensor data or control loops at regular intervals. In contrast, aperiodic tasks generally arise from arbitrary events (external interrupts).

When considering real-time systems that take time as the only limiting factor, it is important to differentiate between underloaded and overloaded real-time systems. A real-time system is said to be underloaded if there exists a feasible schedule for the workload; that is, the deadlines of all tasks are met under timing constraints. On the contrary, overloaded real-time systems do not have a feasible schedule where all tasks meet their deadlines. Thus, the objective will be to optimize some criteria such as the ratio of deadline success.

In addition, real-time systems can be classified into three categories: hard, soft, and weakly hard. In hard real-time systems, all tasks must be guaranteed to complete within their deadlines. For soft real-time systems, it is acceptable to miss some of the deadlines occasionally with additional value for the system to finish the task, even if it is late. In weakly hard real-time systems, tasks are allowed to miss some of their deadlines but there is no associated value if they finish after the deadline.

Real-time task scheduling determines the order in which tasks have to be executed. The well-known scheduling algorithm is the Earliest Deadline First (EDF) algorithm [2]. EDF schedules at each instant of time t the ready task whose deadline is closest to t. EDF algorithm is optimal in underloaded settings; that is, EDF is guaranteed to meet all the task deadlines for any feasible task set.

Nowadays, energy management is becoming the central topic of research in real-time systems. In today's applications, most real-time embedded systems are powered by batteries. Therefore, great interest has risen in powering these systems by renewable energy sources. Many energy harvesting methods can be used to harvest energy from a controlled or ambient environment either to power devices directly or to store the energy in capacitors or batteries for later use. Radio-frequency- (RF-) powered systems, solar-powered systems, wind-powered systems, motional energy harvesting systems, thermoelectric-powered systems, and piezoelectric conversion systems are examples of such methods. These harvesting methods support a wide range of applications such as Heliomote [3] and Prometheus [4] and can also be used to increase the lifetime of preexisting devices.

Recently, we addressed the scheduling problem for a uniprocessor platform that is powered by a renewable energy storage unit and uses an harvester such as photovoltaic cells. We presented a scheduler called EDeg (Earliest Deadline with energy guarantee) [5]. The set of tasks is perfectly known offline as in applications where all the tasks run periodically. EDeg is clairvoyant since it must know in advance both the energy source profile and the characteristics of the tasks (arrival time).

To extend the applicability of EDeg scheduling framework, we need to adapt it to situations where the scheduler has to take decisions without a priori knowledge of the future. For real-time energy harvesting applications, a scheduling algorithm will be nonclairvoyant if, in addition, it ignores the incoming environmental energy in the future. We may imagine an application where either the set of tasks or the future energy profile is known but not both.

We focus here on on-line nonclairvoyant scheduling in an underloaded real-time energy harvesting system that executes aperiodic tasks on a uniprocessor platform. We propose a scheduling algorithm named Energy Harvesting-Earliest Deadline First (EH-EDF) which extends the well-known EDF algorithm. We modify EDF so as to count for the limitation of energy. We benefit from a slack-based method to let the processor idle and thus to recharge the energy storage unit as much as possible without violating deadlines.

The remainder of the paper is organized as follows. In the next section, we summarize the related work. The system model and necessary terminology are introduced in Section 3. In Section 4, we present the fundamental concepts about the slack time. Section 5 describes our scheduling scheme, EH-EDF, with some indications about practical issues. Section 6 illustrates the simulation study, whereas the preliminary results are presented in Section 7. Section 8 concludes the paper and gives some new directions for future work.

2. Literature Review

Most of the previous research work around real-time scheduling disregards energy management or assumes that the energy is not a limiting factor for task execution.

Energy consideration is now added as a crucial issue because of the great advances in both hardware and software technology. This enables system designers to develop large, complex embedded systems. Such systems consume a large amount of power and rely mainly on a limited energy storage. Many technical challenges lie ahead in order to make an energy harvesting system work effectively. Among them is to either minimize the total energy consumption without violating deadlines or maximize the performance of hard energy constrained systems with a fixed energy budget.

With the goal to minimize the total energy consumption, Pillai and Shin [6] present several novel algorithms for real-time dynamic voltage scaling called real-time DVS (RT-DVS). They modify the OS's real-time scheduler and task management service in order to achieve significant energy savings without violating deadlines. Later, Aydin et al. [7] address the problem of power-aware scheduling for periodic tasks with the aim to reduce CPU energy consumption by the help of dynamic voltage scaling. The authors propose an offline algorithm to compute the optimal speed, assuming worst-case workload for each arrival. An online speed reduction mechanism is introduced to recompute energy based on the actual workload. The third component in this solution is to perform a speculative speed adjustment mechanism based on the expected workload. Unlike the work in [6], Aydin et al. [8] take into account the frequency-dependent and -independent power components as well as the power consumption of components other than the CPU when addressing the problem of minimizing overall energy consumption.

Many other studies address the ways to maximize the system performance of underloaded real-time systems that have to operate under a fixed energy budget.

Moser et al. [9] give an optimal scheduling algorithm called LSA for tasks with deadlines, periodic or not, that run on a monoprocessor device that is powered by a rechargeable storage unit. They consider that the source power is predictable but time varying. LSA can be considered as an idling variant of EDF. The system starts executing a task only if the task has the earliest deadline among all ready tasks, and the system can keep on running at the maximum power until the deadline of the task. In that work, the consumption power of the computing system is characterized by some maximum value which implies that for every task, its total energy consumption is directly connected to its execution time through the constant power of the processing device. The main disadvantage of this work lies in that the LSA algorithm executes tasks at full power. Moreover, in practice, the total energy consumed by a task is not necessarily proportional to its execution time.

In [5], we relax the restrictive hypothesis that links energy requirement and execution time of tasks. We present a scheduling algorithm called EDeg (Earliest Deadline with energy guarantee). Simply executing tasks according to the EDF rule either as soon as possible (EDS) or as late as possible (EDL) may lead to violate some deadlines. EDeg executes tasks according to the EDF rule with idling phases and relies on two fundamental concepts, namely, slack time and slack energy. Before authorizing a task to execute, we must ensure that the energy availability will permit to execute all future occurring tasks and the current highest priority one. When this condition is not verified, the processor has to stay idle so that the storage unit recharges as much as possible and as long as all the deadlines can still be met despite execution postponement. In [10], we prove the efficiency of this scheduler through a simulation study. EDeg is clearly clairvoyant since it needs both the characteristics of the future occurring tasks and prediction about the future incoming energy.

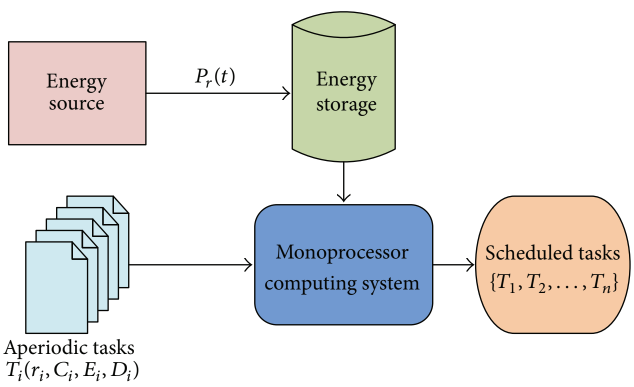

Energy source
$P_r(t)$
Energy storage
Monoprocessor computing system
Scheduled tasks $\{T_1, T_2, \ldots, T_n\}$
Aperiodic tasks $T_i(r_i, C_i, E_i, D_i)$

FIGURE 1: Real-time energy harvesting system.

3. System Model and Terminology

3.1. Task Set. We consider a set of aperiodic tasks that execute on a uniprocessor platform as depicted in Figure 1. Each task is known by the system at the time of its arrival. An aperiodic task set can be denoted as follows: $\Psi = \{\tau_i, i = 1, \ldots, n\}$. Every task τ_i is characterized by (r_i, C_i, D_i, E_i), where r_i represents the arrival time of task τ_i. In the worst case, the execution of τ_i requires a Worst Case Execution Time (WCET) of C_i time units. And it consumes a Worst Case Energy Consumption (WCEC) given by E_i. We assume that the WCEC of a task has no relation with its WCET. A deadline for τ_i occurs at time D_i by which the task should complete its execution. We assume that $0 \leq C_i \leq D_i - r_i$ for each $1 \leq i \leq n$.

Definition 1. The processor load L_p of a task set Ψ gives the processor utilization of Ψ:

$$L_p = \sum_{i=1}^{n} \frac{C_i}{D_{\max}}, \quad (1)$$

where $D_{\max}$ represents the longest deadline in Ψ.

Definition 2. The energy load L_e, measured in joules/s or energy unit/time unit, gives the average power consumed by Ψ:

$$L_e = \sum_{i=1}^{n} \frac{E_i}{D_{\max}}. \quad (2)$$

3.2. Energy Source. We assume that the ambient energy is harvested and converted into electrical power. We cannot control the energy source but we can predict the expected availability with a lower bound on the harvested source power output, namely, $P_r(t)$. Generally, the harvested power is time varying including solar energy which can be assumed constant on average in a long-term perspective. However, on a short-term perspective, the harvested power is highly unstable. This power is then the instantaneous charging rate that incorporates all losses caused by power conversion and charging process. Clearly, we make no assumption about the nature and dynamics of the energy source, making our approach more easily implemented in real systems where data about the energy source may not be available beforehand.

3.3. Energy Storage. We consider an ideal energy storage unit (supercapacitor or battery) of nominal capacity E, corresponding to a maximum energy (expressed in Joule or energy unit). The energy level has to remain between two boundaries $E_{\min}$ and $E_{\max}$ with $E = E_{\max} - E_{\min}$. The stored energy may be used at any time later and does not leak any energy over time. If the storage is fully charged and we continue to charge it, energy is wasted. In contrast, if the storage is fully discharged, no task can be executed.

At some time t, the stored energy is denoted as $E(t)$. At any time, the stored energy is no more than the storage capacity; that is,

$$E(t) < E \quad \forall t. \quad (3)$$

Considering a task set $\Gamma = \{\tau_i = (r_i, C_i, D_i, E_i) \mid i = 1 \ldots n\}$, we want to compute the remaining energy in the energy storage unit at time t. We assume that the energy storage capacity is equal to E energy units at $t = 0$. Let τ_i be the highest priority instance ready at time $t = 0$. As tasks are ordered according to their deadline under EDF, τ_i must be run first. The remaining energy in the energy storage unit at time $t = C_i$ is

$$E(t) = E + \int_0^{C_i} P_r(t)\, dt - E_i. \quad (4)$$

4. Fundamental Concepts

4.1. Slack Time. The slack time of a hard deadline task set at current time t is the length of the longest interval starting at t during which the processor can stay idle without leading to deadline violations.

Let us consider a task set Ψ as described previously. Let Ψ' be the set of tasks ready to be processed at current time t. And let us define the slack time of task τ_j as the maximum processor time that can be used after executing τ_j and higher priority tasks. Then the slack time of $\tau_j \epsilon \Psi'$ is computed as follows:

$$\mathit{slack} \cdot \mathit{time}\left(\tau_j, t\right) = \left(D_j - t\right) - \sum_{D_i \leq D_j} C_i. \quad (5)$$

It comes that the slack time of the system at time t is computed from the slack time of all the tasks as follows:

$$\mathit{slack} \cdot \mathit{time}(t) = \min\left(\mathit{slack} \cdot \mathit{time}\left(\tau_j, t\right)\right). \quad (6)$$

4.2. Illustrative Example 1. Consider a task set $\Psi = \{\tau_i, i = 1, \ldots, 4\}$ with $\tau_i = (r_i, C_i, D_i)$. Let $\tau_1 = (0, 3, 18)$, $\tau_2 = (4, 2, 12)$, $\tau_3 = (5, 3, 24)$, and $\tau_4 = (0, 4, 16)$. Let us compute the slack time at time 6 after executing the tasks according to EDS from 0 to 6.

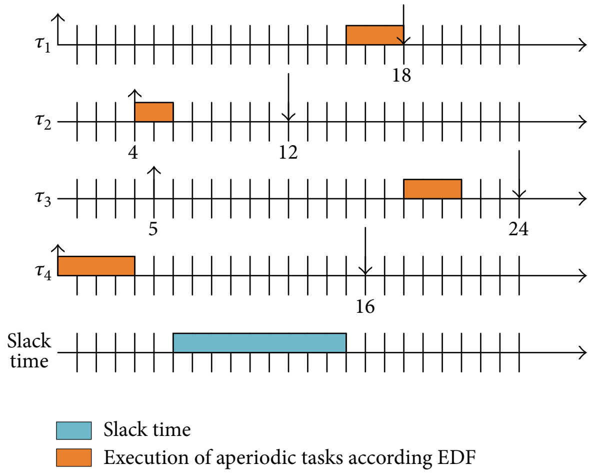


Figure 2: Computing the slack time at $t = 6$.

τ_4 is executed from time 0 to time 4 and τ_2 from time 4 to time 6. At time 6, tasks τ_1 and τ_3 are both ready for execution. Their slack time and the slack time of the system are computed according to (5) and (6):

$$slack\cdot time\left(\tau_1,6\right) = \left(D_1-6\right) - \sum_{D_i\le D_1} C_i = (18-6)-3 = 9,$$

$$slack\cdot time\left(\tau_3,6\right) = \left(D_3-6\right) - \sum_{D_i\le D_3} C_i = (24-6)-6 = 12,$$

$$slack\cdot time\left(6\right) = \min\left(slack\cdot time\left(\tau_1,6\right),\ slack\cdot time\left(\tau_3,6\right)\right) = 9. \tag{7}$$

Figure 2 describes the resulting schedule where the processor is let idle from time 6 during a time interval whose length equals the slack time. We note that τ_1 starts execution at the latest time while τ_3 has a slack equal to 3 time units before deadline. This corresponds to the slack time of τ_3 minus the slack time of the system.

4.3. Illustrative Example 2. Consider the same task set as in Section 4.2. Nevertheless, we add a task $\tau_5 = (8,5,20)$. Assume that we execute τ_4 from time 0 to time 4 and then τ_2 from time 4 to time 6. When computing the slack time at time 6, we have two ready tasks τ_1 and τ_3 with $slack\cdot time(\tau_1,6) = 9$ and $slack\cdot time(\tau_3,6) = 12$. Consequently $slack\cdot time(6) = 9$.

τ_5 is released at $t = 8$. This leads to the update of the slack time (Figure 3). First, we note that the slack time function linearly decreases with time when the processor is let idle. And the slack time of a task is only affected by tasks with a lower deadline. As the deadline of τ_3 is lower than the deadline of the new occurring task τ_5, we deduce that $slack\cdot time(\tau_3,8) = slack\cdot time(\tau_3,6) - 2 = 10$.

As the deadline of τ_1 is greater than the deadline of the new occurring task τ_5, the computation of the slack time of τ_1 must be achieved thanks to (5):

$$slack\cdot time\left(\tau_3,8\right) = (24-8)-(3+3+5) = 5. \tag{8}$$

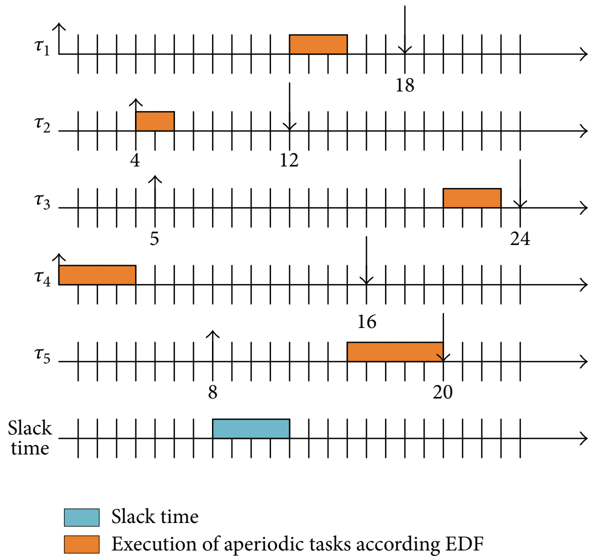


Figure 3: Updating the slack time at $t = 8$.

By the help of (5), we have $slack\cdot time(\tau_5,8) = 4$. Thus the slack time of the system will be changed to 4 in order to meet the deadline of the new occurring task (τ_5).

Now, we are prepared to introduce a new online scheduler specifically adapted to aperiodic tasks in an energy harvesting context.

5. The EH-EDF Scheduling Algorithm

In this section, the scheduler ignores the future energy production and the future arrival times of tasks.

5.1. Presentation of the Scheduler. The intuition behind EH-EDF algorithm is to schedule aperiodic tasks as soon as possible according to EDF. When a new task arrives, it is inserted in the ready task list. When the energy in the storage unit reveals to be insufficient for executing tasks, the only solution consists in postponing them as much as possible. We have to perform the computation of the slack time of the system from the ready task list. The scheduler lets the processor idle until the energy storage unit replenishes or the slack time becomes zero.

The slack time is updated whenever a new task arrives even in the recharging phase. The processor continues idling as long as the system has slack.

We propose the so-called *Energy Harvesting-Earliest Deadline First (EH-EDF) algorithm* following the idea described previously.

The major components of the EH-EDF algorithm are $E(t)$ and $slack\cdot time(t)$. $E(t)$ is the residual capacity of the storage unit at time t which is the energy that is currently stored and $slack\cdot time(t)$ is the slack time of the system at current time t. PENDING is a Boolean which equals true whenever there is at least one instance in the ready list queue. We use the function wait() to put the processor in sleep mode and

```
Input: A Set of aperiodic Tasks $\Psi = \{\tau_i \mid \tau_i = (r_i, C_i, E_i, D_i)\ i = 1, \ldots, n\}$
Scheduled according to EDF, current time $t$, battery with capacity ranging from $E_{\max}$ to $E_{\min}$,
energy level of the battery $E(t)$, source power $P_r(t)$.
Output: EH-EDF Schedule.
while "(1)" do
    while "PENDING=true" do
        while "($E(t) > E_{\min}$)" do
            execute()
        end "while
        while" ($E(t) < E_{\max}$ and $Slack \cdot time(t) > 0$) "do
            wait()
        end" while
    end "while
    while" PENDING=false "do
        wait()
    end" while
end "while
```

Algorithm 1: Energy Harvesting-Earliest Deadline First (EH-EDF).

function execute() to put the processor in active mode and schedule the tasks according to EDF.

The framework of the EH-EDF scheduling algorithm is as Algorithm 1.

From the EH-EDF framework, we notice that tasks do not run after $E_{\min}$. EH-EDF charges the energy storage to the maximum level, provided there is sufficient slack time and the storage unit is not fully replenished. Such condition can be easily detected through an interrupt mechanism and adequate circuitry between the storage unit and the processing device. The slack time is computed when entering the *wait* state and decremented at each time instant.

Therefore, we waste recharging power only when there are no pending tasks in the ready list and the storage unit is full.

5.2. Illustrative Example. Consider a task set Ψ with five aperiodic tasks as in the previous example such that $\Psi = \{\tau_i \mid 1 \le i \le 5\}$, where $\tau_i = (r_i, C_i, D_i, E_i)$. Let $\tau_1 = (0, 3, 18, 9)$, $\tau_2 = (4, 2, 12, 12)$, $\tau_3 = (5, 3, 24, 7)$, $\tau_4 = (0, 4, 16, 10)$, and $\tau_5 = (8, 3, 20, 10)$. The energy storage capacity is assumed to be equal to 10 energy units. For sake of simplicity, the rechargeable power, P_r, is constant along time and equals 2.

Ψ is temporally feasible; that is, all deadlines can be met when abstracting for energy. But Ψ reveals to be not feasible with energy limitations since the storage unit empties at time 6.

When applying EH-EDF (Figure 4) to Ψ, the energy storage capacity empties at $t = 6$. The energy storage recharges as much as possible. The recharging time is computed from the current *slack time* in order to still guarantee all the deadlines while avoiding energy overflow.

In details, the energy storage is full at time 0. The highest priority task T_4 executes until time 4 when the energy storage capacity is given by the following formula: $E(4) = E_{\max} - E_1 + P_r C_1 = 8$ energy units.

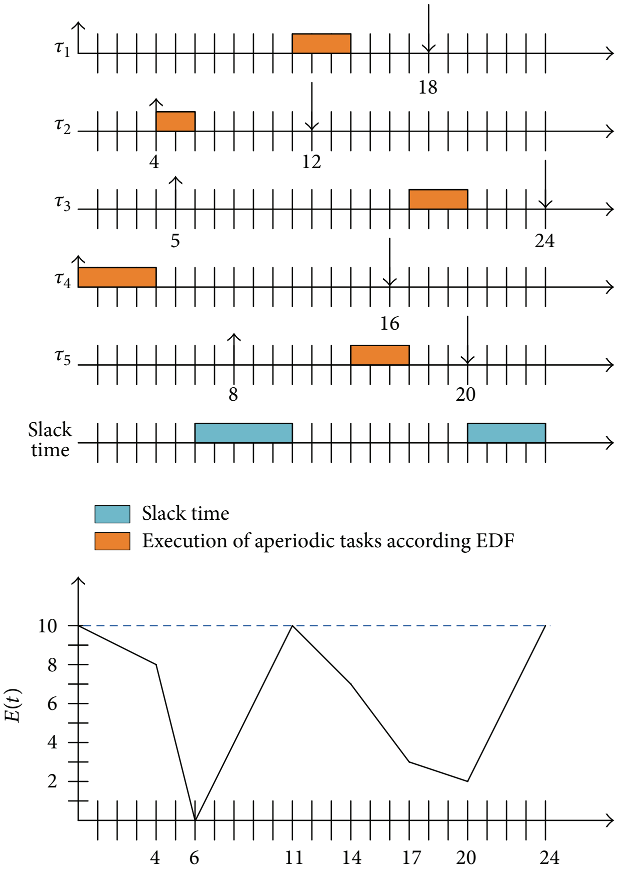


Figure 4: Example on EH-EDF scheduling.

T_2 is ready at time 4. As the highest priority task, it executes until time 6 when the energy storage empties. The processor has to remain idle as long as the storage has not fulfilled and the slack time is not zero. According to (5), the slack time of all released tasks and the slack time of the system are computed.

T_1 and T_3 are released at time 6. The slack time of T_1 and T_3 is equal to 9 and 15, respectively. As $slack \cdot time(6) = 9$, the processor has to stay idle until time 15 for recharging the energy storage unit.

T_5 is released at time 8. As the slack time for T_5 is 6, $slack \cdot time(8) = 6$. The battery is recharged until time 11 when it is full. Thus, we stop recharging at time 11 to avoid wasting energy.

At time 11, the energy storage is equal to 10 energy units, and T_1 has the highest priority. It executes until time 14 and the remaining energy $E(14) = 7$ energy units. T_5 is then the highest priority task and executes until time 17 when the energy level equals 3 energy units.

At time 17, T_3 executes until time 20 where the energy level equals 2 energy units. The processor has no task to execute and remains idle until time 24 where the energy storage is full again.

In contrast to EDF, EH-EDF feasibly schedules the task set Ψ given the characteristics of the storage unit and the power source profile.

6. Simulation Study

This section describes experiments that have been conducted to evaluate the Energy Harvesting-EDF (EH-EDF) algorithm. To measure the effectiveness of EH-EDF, we develop a discrete-event simulation in C/C++. We report a performance analysis which consists of five experiments.

The simulation environment consists of a simulation kernel (scheduler) with a number of components involved in the management and analysis of simulations. The main components are the task generator, the scheduler, and the CPU.

The generator of aperiodic tasks has been designed to accept the following input parameters: the number of desired tasks n, the processor load L_p, the energy load L_e, and the recharging power $P_r(t)$. The output is a task set $\Psi = \{\tau_i(r_i, C_i, D_i, E_i),\ i = 1\ \ to\ \ n\}$. The execution time of tasks is randomly generated such that $L_p = \sum_{i=1}^{n}(C_i/D_{\max}) \leq 1$. Moreover, the energy consumption of tasks is randomly generated from the energy load factor such that $L_e = \sum_{i=1}^{n}(E_i/D_{\max}) \leq P_r$. Deadlines are greater than or equal to the computation times.

Simulation results are then ordered to excel files to be stored and analyzed.

6.1. Formal Definition of Scheduling Strategies. For the sake of comparison, we implement five energy harvesting scheduling policies where aperiodic tasks execute as soon as possible according to *EDF*.

> EH-EDF: when the battery empties, the processor is put into sleep mode until the battery replenishes or the slack time becomes zero.
>
> EH-EDFx: when the battery empties, the processor is put into sleep mode for x units of time where x is an input of the scheduler.
>
> EH-EDF1: when the battery empties, the processor is put into sleep mode until the energy level reaches a threshold value, namely, E_{th}, given as an input of the scheduler.
>
> EH-EDF2: when the battery empties, the processor is put into sleep mode until the slack time becomes zero regardless of the energy level.
>
> EH-EDF3: there are two threshold parameters, namely, $E\text{th}_{\min}$ and $E\text{th}_{\max}$. when the energy level reaches $E\text{th}_{\min}$, the system is put into sleep mode until either the slack time be null or the energy level be $E\text{th}_{\max}$.

6.2. Measurement Support

6.2.1. Aperiodic Task Sets Generation. We use a simulator that generates 50 tasks with maximum deadline equal to 3360. The worst-case computation times are set according to the processor load L_p, where L_p can be 30%, 60%, or 90%. Results presented in this section are averages over groups of fifty task sets.

6.2.2. Energy Parameters Generation. The energy consumptions of tasks (WCEC) are randomly generated but constrained by the energy load L_e. All tasks are assumed to linearly consume their energy budget over time. In addition, all tasks are dischargeable. This means that E_i/C_i is greater than $P_r(t)$ for all t. The rechargeable power is constant along time during the execution of a task and varies from one task execution to another. A random generator enables us to produce for every quantum of time a power energy profile with minimum value 10 and a maximum value here 35.

6.2.3. Simulations Description. We start the simulation with a battery fully recharged ($E(0) = E_{\max}$). When a deadline is missed, we discard the task and update the slack time. The simulation is repeated for 50 task sets for a given processor and energy utilization ratio. For a fair comparison of the previous strategies, all simulations are performed under the same conditions. We report the performance analysis that consists of the following measures:

(i) percentage of feasible task sets;

(ii) the impact of the slack time and energy storage capacity on the performance of EH-EDF;

(iii) average idle time corresponding to recharging phases of the energy storage;

(iv) energy storage low level.

The above measurements are compared under different scenarios for the five energy harvesting scheduling policies stated previously. These policies cover all the possibilities of the EH-EDF algorithm. We measure the impact of the slack time and energy storage capacity on the performance of EH-EDF, EH-EDFx, and EH-EDF1.

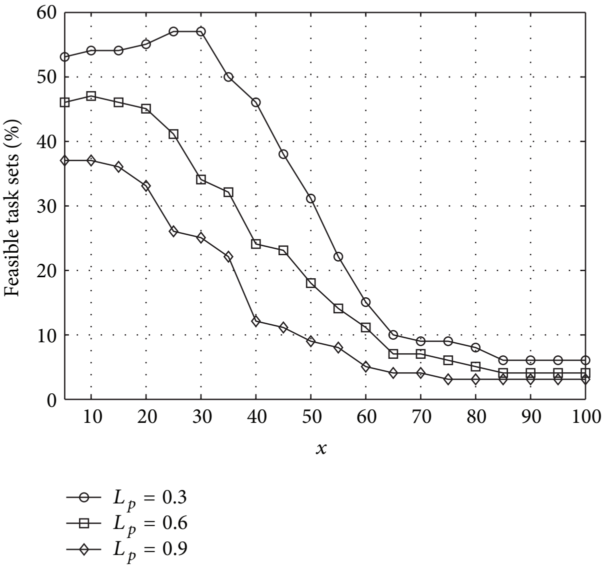


Figure 5: Effects of parameter x on EH-EDFx ($E_{\max} = 100$).

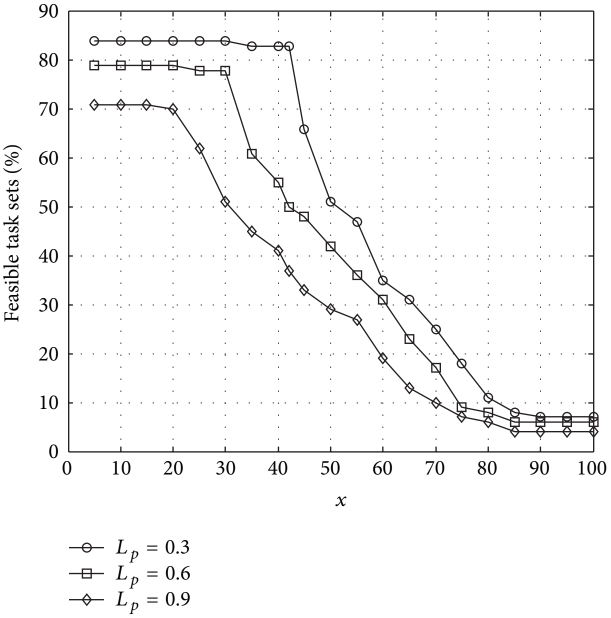


Figure 6: Effects of parameter x on EH-EDFx ($E_{\max} = 200$).

7. Preliminary Results

7.1. Impact of Parameter x on EH-EDFx. In this section, we experiment on the effect of the slack time (x) on EH-EDFx. We report the results of this simulation study where the processor load L_p is set to 0.3, 0.6, and 0.9, respectively. The rechargeable power is constant during execution of a task and varies between a task and another. We took a random function that randomly gives a number between 10 and 35. The maximum ambient power is 35. So all tasks are discharging tasks ($L_e \leq 35$). Simulations are performed first with $E_{\max} = 100$ (Figure 5) and second with $E_{\max} = 200$ (Figure 6).

When L_p is set to 0.3, EH-EDFx benefits from the high idle time to recharge the energy storage. Thus, any parameter x will be acceptable to recharge the battery without violating deadlines till $x = 30$. After this value, the percentage of feasible task sets begins to decrease, and a higher number of deadlines are missed.

When L_p is set to 0.6, the total idle time decreases. We observe that the performance of EH-EDFx is roughly constant until x reaches 20 where the number of violated deadlines begins to increase.

At higher values of processor load, the performance of EH-EDFx is approximately constant until x reaches a value of 15 where a higher number of deadlines are violated.

In the second experiment, we double the size of the energy storage unit ($E_{\max} = 200$) while keeping the other parameters unchanged.

When L_p is set to 0.3, any parameter of x will be acceptable to recharge the energy storage without violating deadlines till a high value of $x = 42$ where the percentage of feasible task sets begins to decrease. The percentage of feasible task sets is 84% when $x = 42$ which is approximately 58% more than in the case when $E_{\max} = 100$. This is because as the size of the energy storage increases, EH-EDFx will be able to execute more tasks, and consequently the percentage of feasible task sets will increase. As L_p increases, the percentage of feasible task sets is, respectively, 44% and 50% more for $L_p = 0.6$ and 0.9 than in the case when $E_{\max} = 100$.

We conclude that the slack time and the energy storage capacity have a great impact on the system performance. As we increase the energy storage size, the mean system life time increases, but without reaching optimality.

7.2. Impact of Parameter E_{th} on EH-EDF1. In this section, we experiment on the effect of parameter E_{th} on EH-EDF1 (Figures 7 and 8). We report the results of this simulation study where the processor load L_p is set to 0.3, 0.6, and 0.9, respectively. Simulations are performed first for $E_{\max} = 100$ (case a) and second for $E_{\max} = 200$ (case b).

For $E_{\max} = 100$, we observe that EH-EDF1 gives approximately an average constant performance until $E_{\text{th}} = 0.2E_{\max}$. In details, when L_p is set to 0.3, the percentage of feasible task sets for EH-EDF1 is constant until a critical value of E_{th} (15% for case (a) and 25% for case (b)). After this critical value, the performance increases without reaching optimality. When $L_p = 0.6$, the performance of EH-EDF1 is constant until a critical value of E_{th} (20% for case (a) and 30% for case (b)). As L_p increases, the percentage of feasible task sets decreases until it reaches a maximum (76% for case (a) and 89% for case (b)).

As a conclusion, we demonstrate through the previous simulations that the slack time and energy storage capacity have a great effect on the performance of EH-EDF algorithm. In addition, we note that EH-EDFx and EH-EDF1 give approximately the same performance levels in terms of deadline missings.

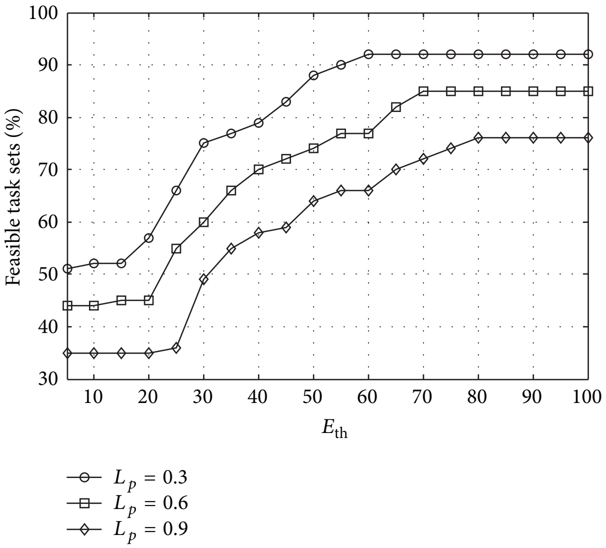


Figure 7: Effects of parameter E_{th} on EH-EDF1 (case a).

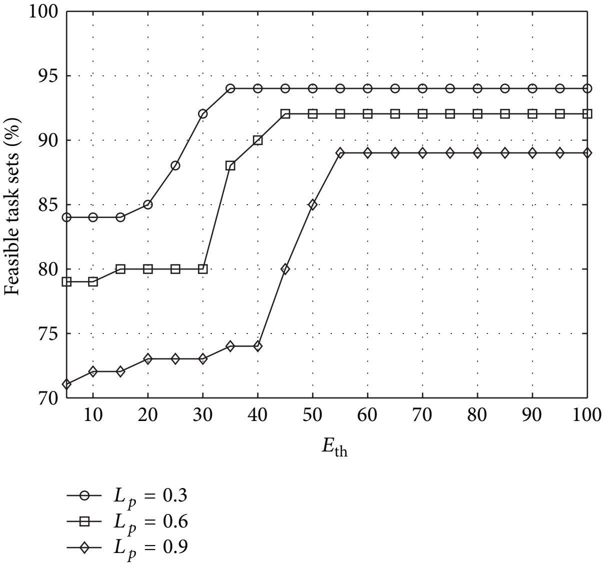


Figure 8: Effects of parameter E_{th} on EH-EDF1 (case b).

7.3. Percentage of Feasible Task Sets. In this section, we experiment on task sets which are feasible. Simulations are performed by varying E_{max}. Based on the previous simulations, we choose $x = 20$ for EH-EDFx and $E_{th} = 25\%$ for EH-EDF1 and EH-EDF3. We report the results of this simulation study where the processor load (L_p) is set to 0.3, 0.6, and 0.9, respectively. EH-EDF2 is eliminated from this section due to its poor performance. This proves that the maximum upper bound for the energy storage capacity has a great impact on the performance.

First, we consider that $E_{max} = 100$ (Figure 9). Our experiment demonstrates that EH-EDF outperforms the other policies. This is because EH-EDF will benefit from the idle time to recharge the energy storage capacity without violating deadlines.

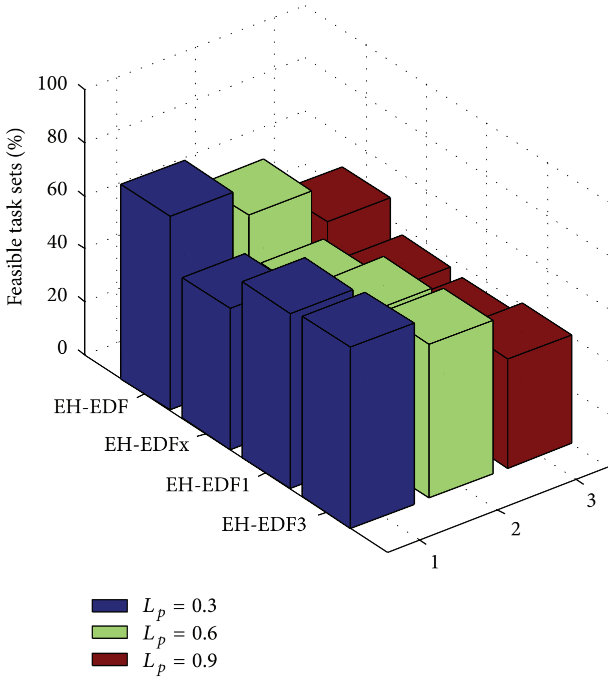


Figure 9: Percentage of feasible task sets (E_{max} = 100).

In details, when L_p is set to 0.3, EH-EDF proves to have the highest percentage of feasible task sets with 6.9%, 9.5%, and 27.4% more than EH-EDF1, EH-EDF3, and EH-EDFx, respectively. As L_p increases, EH-EDF outperforms the other policies but with a performance decrease of 15% from the first case. This is because as L_p increases, the total idle time decreases and consequently the relative performance of EH-EDF decreases. At higher values of processor load, the performance loss of EH-EDF is about 34% when compared with low processor load.

Secondly, we double the size of the capacity of the energy storage unit while keeping the other parameters unchanged (Figure 10). As previously, EH-EDF gives a percentage of feasible task sets 11%, 18%, and 24%, respectively, higher than with EH-EDF1, EH-EDF3, and EH-EDFx, respectively. When the size of the energy storage unit is doubled, the performance increases of about 21%.

As a summary, this experiment shows that it is highly probable that no online algorithm can achieve optimality. In other words, only clairvoyant algorithms that have a complete knowledge of the task properties and energy production can achieve a valid schedule whenever one exists.

7.4. Energy Storage Low Level. In this section, we measure the number of times the energy storage unit empties by varying the processor load. We consider the same values as depicted in Section 7.3. Simulations are performed for $E_{max} = 100$ (case a) and $E_{max} = 200$ (case b). When $E_{max} = 100$, we observe from Figure 11 that EH-EDF presents the best behavior relative to the other policies. This is because EH-EDF will benefit from the idle time to recharge the energy storage to its maximum value while respecting all deadlines. In details, the average number of times the energy storage is

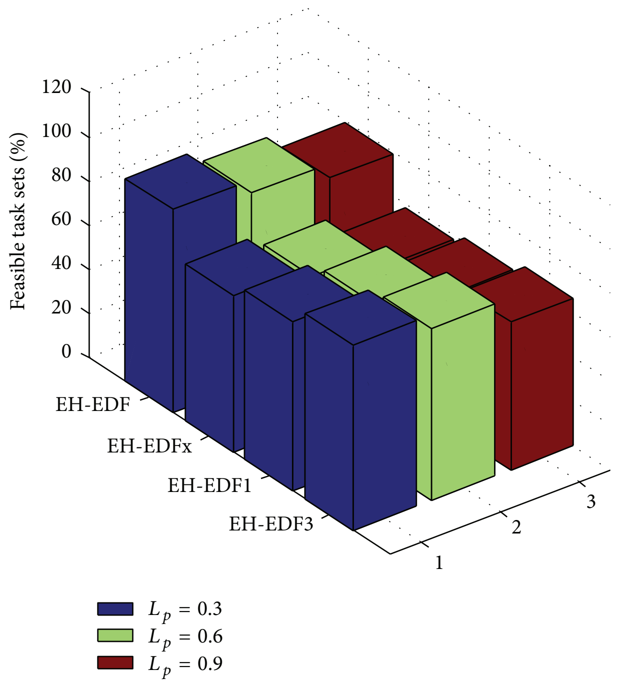


FIGURE 10: Percentage of feasible task sets ($E_{\max} = 200$).

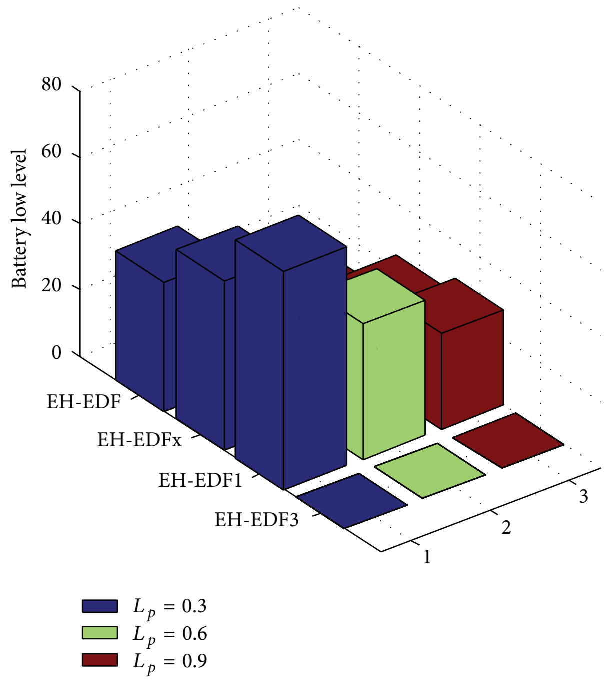


FIGURE 11: Battery low level ($E_{\max} = 100$).

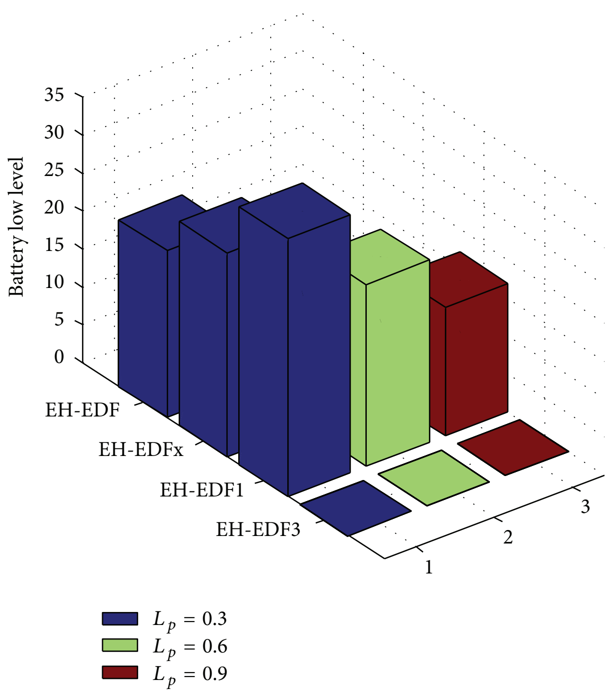


FIGURE 12: Battery low level ($E_{\max} = 200$).

empty under EH-EDF is, respectively, 38% and 49% less than under EH-EDF*x* and EH-EDF1.

Furthermore, we note that EH-EDF3 cannot reach the empty state $E_{\min}$ since the system is put in the empty state when the energy capacity reaches $E_{\text{th}\min}$ that is greater than $E_{\min}$ (by default equal to zero).

When the energy storage capacity is doubled, EH-EDF still has the lowest number of energy storage low level (Figure 12). The average number of times the energy storage empties under EH-EDF is, respectively, 32% and 45% less than under EH-EDF*x* and EH-EDF1. As the energy storage capacity increases, the number of energy storage low levels decreases since the energy storage has a higher ability to execute tasks.

7.5. Average Idle Time. The average idle time has a great impact when studying the efficiency of EH-EDF especially in systems that use the Dynamic Power Management mechanism (DPM). DPM provides efficiency only if the idle times are sufficiently long because of inherent time and energy overhead induced by state switching. Consequently, the longer is the average idle time, the lower is the impact of the energy and time overheads incurred by DPM on the overall performance.

Moreover, the length of the idle time has a great impact on the life time of the energy storage unit regardless of its type (battery or supercapacitor). Charging any storage unit is not linear and consequently the more it is paused, the more energy it recharges.

In this section, we compute the average idle time by taking two values for $E_{\max}$. When $E_{\max} = 100$ (Figure 13), we observe that EH-EDF maximizes the average idle time, respectively, by 70%, 58%, and 45% when compared with EH-EDF*x*, EH-EDF3, and EH-EDF1. The reason is that, in EH-EDF, the processor is put into sleep mode as long as the slack time is positive and the energy level is less than $E_{\max}$.

When $E_{\max} = 200$, we observe from Figure 14 that EH-EDF still maximizes the average idle time, respectively, by 77%, 60%, and 49% when compared with EH-EDF*x*, EH-EDF3, and EH-EDF1.

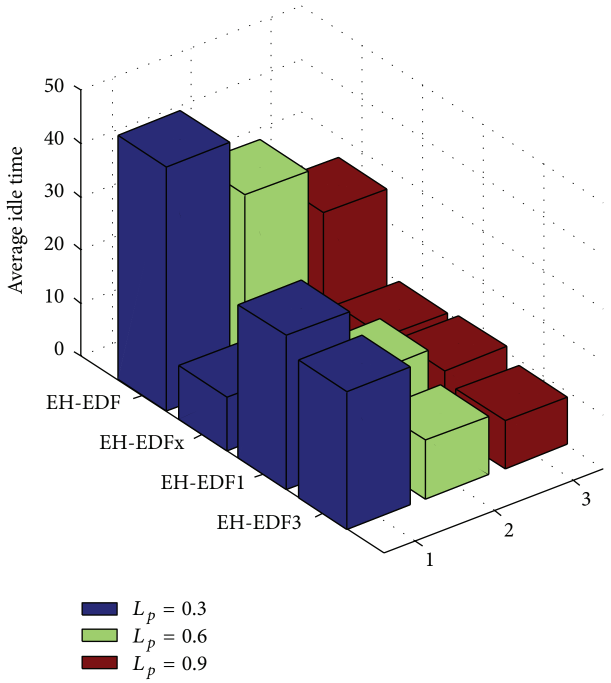


Figure 13: Average idle time ($E_{\max}$ = 100).

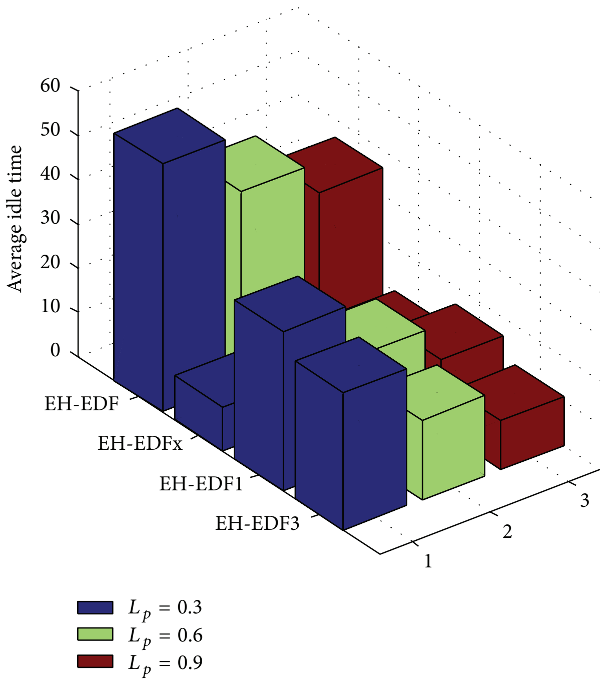


Figure 14: Average idle time ($E_{\max}$ = 200).

8. Conclusions

We studied an energy harvesting sensor node which supports a set of aperiodic tasks with real-time constraints. The arrival times, deadlines, and energy demands of the tasks are not known to the node in advance. We focussed on online scheduling with no lookahead including energy production. We presented and analyzed through an experiment an idling-EDF-based scheduling algorithm called EH-EDF.

Traditional online algorithms such as EDF behave poorly because they consume the energy greedily and not adaptively. We recently proved in [11] that EDF remains the best nonidling scheduler but has a zero competitive factor for the energy harvesting model. We consequently propose several variants of EDF to derive more efficient scheduling solutions.

The experiment demonstrates that EH-EDF offers an acceptable and even good performance in a wide range of situations. We study the impact of the slack time and the threshold energy level on the performance of EH-EDF in terms of percentage of feasible task sets. We show that EH-EDF outperforms EH-EDF1, EH-EDF3, and EH-EDF*x* by, respectively, 7%, 10%, and 27%. Furthermore, EH-EDF proves to be better than EH-EDF*x* and EH-EDF1, respectively, by 38% and 49% in terms of the number of times the energy storage empties.

Finally, the advantage of the EH-EDF algorithm lies in the average duration of the processor idle times which is higher compared with other heuristics. As a result, leakage and overhead incurred by the implementation of DPM mechanism are avoided under EH-EDF.

The next step of our work will be to extend EH-EDF to the Dynamic Voltage and Frequency Scaling (DVFS) technology.

References

[1] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, Cambridge, UK, 1998.

[2] M. L. Dertouzos, “Control robotics: the procedural control of physical processes,” in *Proceedings of the International Federation for Information Processing Congress*, pp. 807–813, 1974.

[3] k. Lin, J. Yu, J. Hsu et al., “Demo abstract: heliomote: enabling long-lived sensor networks through solar energy harvesting,” in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys ’05)*, November 2005.

[4] X. Jiang, J. Polastre, and D. Culler, “Perpetual environmentally powered sensor networks,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05)*, pp. 463–468, April 2005.

[5] H. El Ghor, M. Chetto, and R. H. Chehade, “A real-time scheduling framework for embedded systems with environmental energy harvesting,” *Computers and Electrical Engineering*, vol. 37, no. 4, pp. 498–510, 2011.

[6] P. Pillai and K. G. Shin, “Real-time dynamic voltage scaling for low-power embedded operating systems,” *Operating Systems Review*, vol. 35, no. 5, pp. 89–102, 2001.

[7] H. Aydin, R. Melhem, D. Mossé, and P. Mejia-Alvarez, “Power-aware scheduling for periodic real-time tasks,” *IEEE Transactions on Computers*, vol. 53, no. 5, pp. 584–600, 2004.

[8] H. Aydin, V. Devadas, and D. Zhu, “System-level energy management for periodic real-time tasks,” in *Proceedings of the 27th IEEE International Real-Time Systems Symposium (RTSS ’06)*, pp. 313–322, December 2006.

[9] C. Moser, D. Brunelli, L. Thiele, and L. Benini, “Real-time scheduling for energy harvesting sensor nodes,” *Real-Time Systems*, vol. 37, no. 3, pp. 233–260, 2007.

[10] M. Chetto, H. El Ghor, and R. Hage Chehade, “Real-time scheduling for energy harvesting sensors,” in *Proceedings of the 6th International Conference on Internet Technology and Secured Transactions*, pp. 396–402, 2011.

[11] M. Chetto and A. Queudet-Marchand, “A note on EDF scheduling for real-time energy harvesting systems,” *IEEE Transactions on Computers*, 2013.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 923714, 10 pages
http://dx.doi.org/10.1155/2013/923714

Research Article

On Adaptive Energy-Efficient Transmission in WSNs

M. Tahir,[1] N. Javaid,[1] A. Iqbal,[1] Z. A. Khan,[2] and N. Alrajeh[3]

[1] *COMSATS Institute of Information Technology, Islamabad 44000, Pakistan*
[2] *Faculty of Engineering, Dalhousie University, Halifax, NS, Canada B3J 1Y9*
[3] *BMT, CAMS, King Saud University, Riyadh 11633, Saudi Arabia*

Correspondence should be addressed to N. Javaid; nadeemjavaid@comsats.edu.pk

Received 1 March 2013; Revised 19 April 2013; Accepted 20 April 2013

Academic Editor: Joel Rodrigues

One of the major challenges in design of wireless sensor networks (WSNs) is to reduce energy consumption of sensor nodes to prolong lifetime of finite capacity batteries. In this paper, we propose energy-efficient adaptive scheme for transmission (EAST) in WSNs. EAST is an IEEE 802.15.4 standard compliant. In this scheme, open-looping feedback process is used for temperature-aware link quality estimation and compensation, wherea closed-loop feedback process helps to divide network into three logical regions to minimize overhead of control packets. Threshold on transmitter power loss ($\mathrm{RSSI}_{\mathrm{loss}}$) and current number of nodes ($n_c(t)$) in each region help to adapt transmit power level (P_{level}) according to link quality changes due to temperature variation. Evaluation of the proposed scheme is done by considering mobile sensor nodes and reference node both static and mobile. Simulation results show that the proposed scheme effectively adapts transmission P_{level} to changing link quality with less control packets overhead and energy consumption as compared to classical approach with single region in which maximum transmitter P_{level} assigned to compensate temperature variation.

1. Introduction

WSNs are currently being considered for many applications, including industrial, security surveillance, medical, environmental, and weather monitoring. Due to limited battery lifetime at each sensor node, minimizing transmitter P_{level} to increase energy efficiency and network lifetime is useful. Sensor nodes consist of three parts: sensing unit, processing unit, and transceiver [1]. Limited battery requires low power sensing, processing, and communication system. Energy efficiency is of paramount interest, and optimal WSN should consume minimum amount of power.

In WSNs, sensor nodes are widely deployed in different environments to collect data. As sensor nodes usually operate on limited battery, each sensor node communicates using a low power wireless link, and link quality varies significantly due to environmental dynamics like temperature and humidity. Therefore, while maintaining good link quality between sensor nodes, we need to reduce energy consumption for data transmission to extend network lifetime [2–4]. IEEE802.15.4 is a standard used for low energy, low data rate applications like WSN. This standard operates at frequency of 2.45 GHz with channels up to 16 and data rate of 250 kbps.

To efficiently compensate link quality changes due to temperature variations, we propose a new scheme for P_{level} control EAST that improves network lifetime while achieving required reliability between sensor nodes. This scheme is based on combination of open-loop and closed-loop feedback processes in which we divide network into three regions on basis of threshold on $\mathrm{RSSI}_{\mathrm{loss}}$ for each region. In open-loop process, each node estimates link quality using its temperature sensor. Estimated link quality degradation is then effectively compensated using closed-loop feedback process by applying the proposed scheme. In closed-loop feedback process, appropriate transmission P_{level} control is obtained which assigns substantially less power than that required in existing transmission power control schemes.

The rest of the paper is organized as follows. Section 2 briefs the related existing work and motivation for this work. In Section 3, we provide the readers with our proposed scheme. In Section 4, we model our proposed scheme. Experimental results have been given in Section 5.

2. Related Work and Motivation

To transmit data efficiently over wireless channels in WSNs, existing schemes set some minimum transmission P_{level} for maintaining reliability. These schemes either decrease interference among sensor nodes or increase unnecessary energy consumption. In order to adjust transmission P_{level}, a reference node periodically broadcasts a beacon message. When nodes hear a beacon message from a reference node, nodes transmit an ACK message. Through this interaction, the reference node estimates connectivity between nodes.

In local mean algorithm (LMA), a reference node broadcasts LifeMsg message. Nodes transmit LifeAckMsg after they receive LifeMsg. Reference nodes count the number of LifeAckMsgs and transmission P_{level} to maintain appropriate connectivity. For example, if the number of LifeAckMsgs is less than NodeMinThresh, transmission P_{level} is increased. In contrast, if the number of LifeAckMsgs is more than NodeMaxThreshold transmission, P_{level} is decreased. As a result, they provide improvement of network lifetime in a sufficiently connected network. However, LMA only guarantees connectivity between nodes and cannot estimate link quality [5].

Local Information No Topology/Local Information Link-state Topology (LINT/LILT), and Dynamic Transmission Power Control (DTPC) use $\text{RSSI}_{\text{loss}}$ to estimate transmitter P_{level}. Nodes exceeding threshold $\text{RSSI}_{\text{loss}}$ are regarded as neighbor nodes with reliable links. Transmission P_{level} is also controlled by packet reception ratio (PRR) metric. As for the neighbor selection method, three different methods have been used in the literature: connectivity based, PRR based, and $\text{RSSI}_{\text{loss}}$ based. In LINT/LILT, a node maintains a list of neighbors whose $\text{RSSI}_{\text{loss}}$ values are higher than the threshold $\text{RSSI}_{\text{loss}}$, and it adjusts the radio transmission P_{level} if the number of neighbors is outside the predetermined bound. In LMA/LMN, a node determines its range by counting how many other nodes acknowledged to the beacon message it has sent [6].

Adaptive transmission power control (ATPC) adjusts transmission P_{level} dynamically according to spatial and temporal effects. This scheme tries to adapt link quality that changes over time by using closed-loop feedback. However, in large-scale WSNs, it is difficult to support scalability due to serious overhead required to adjust transmission P_{level} of each link. The result of applying ATPC is that every node knows the proper transmission P_{level} to use for each of its neighbors, and every node maintains good link qualities with its neighbors by dynamically adjusting the transmission P_{level} through on-demand feedback packets. Uniquely, ATPC adopts a feedback-based and pairwise transmission P_{level} control. By collecting the link quality history, ATPC builds a model for each neighbor of the node. This model represents an *in situ* correlation between transmission P_{level} and link qualities. With such a model, ATPC tunes the transmission P_{level} according to monitored link quality changes. The changes of transmission P_{level} reflect changes in the surrounding environment [7].

Existing approaches estimate variety of link quality indicators by periodically broadcasting a beacon message. In addition, feedback process is repeated for adaptively controlling transmission P_{level}. In adapting link quality for environmental changes, where temperature variation occurs, packet overhead for transmission P_{level} control should be minimized. Reducing the number of control packets while maintaining reliability is an important technical issue [8].

Radio communication quality between low power sensor devices is affected by spatial and temporal factors. The spatial factors include the surrounding environment, such as terrain and the distance between the transmitter and the receiver. Temporal factors include surrounding environmental changes in general, such as weather conditions (temperature). To establish an effective transmission P_{level} control mechanism, we need to understand the dynamics between link quality and $\text{RSSI}_{\text{loss}}$ values. Wireless link quality refers to the radio channel communication performance between a pair of nodes. PRR is the most direct metric for link quality. However, the PRR value can only be obtained statistically over a long period of time. $\text{RSSI}_{\text{loss}}$ can be used effectively as binary link quality metrics for transmission P_{level} control [9].

Radio irregularity results in radio signal strength variation in different directions, but the signal strength at any point within the radio transmission range has a detectable correlation with transmission power in a short time period. There are three main reasons for the fluctuation in the $\text{RSSI}_{\text{loss}}$. First, fading causes signal strength variation at any specific distance. Second, the background noise impairs the channel quality seriously when the radio signal is not significantly stronger than the noise signal. Third, the radio hardware does not provide strictly stable functionality [10].

Since the variation is small, this relation can be approximated by a linear curve. The correlation between $\text{RSSI}_{\text{loss}}$ and transmission P_{level} is approximately linear. Correlation between transmission P_{level} and $\text{RSSI}_{\text{loss}}$ is largely influenced by environments, and this correlation changes over time. Both the shape and the degree of variation depend on the environment. This correlation also dynamically fluctuates when the surrounding environmental conditions change. The fluctuation is continuous, and the changing speed depends on many factors, among which the degree of environmental variation is one of the main factors [11].

Proposing energy-efficient transmission scheme EAST helps efficiently compensate link quality changes due to temperature variation. Estimated packet overhead for adaptive power control temperature measured by sensors is utilized to adjust transmission P_{level} for all three regions based on $\text{RSSI}_{\text{loss}}$. Compared to single region in which large overhead of control packets occurs even due to small change in link quality, multiple regions have reduced overhead. Closed-loop feedback process is executed to minimize control packets overhead and required transmitter P_{level}.

3. Proposed Energy-Efficient Transmission Scheme

In this section, we present-energy efficient transmission scheme that maintains link quality during temperature variation in wireless environment. It utilizes open-loop

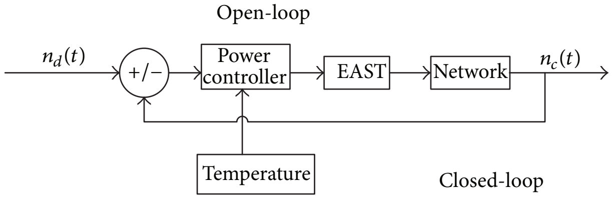


Figure 1: Block diagram.

process based on sensed temperature information according to temperature variation. Closed-loop feedback process based on control packets is further used to accurately adjust transmission P_{level}. By adopting both open-loop and closed-loop feedback processes, we divide network into three regions: A, B, and C for high, medium, and low $\text{RSSI}_{\text{loss}}$, respectively.

In order to assign minimum and reachable transmission P_{level} to each link, EAST is designed. EAST has two phases, that is, initial and run-time. In initial phase, reference node builds a model for nodes in network. In run time phase based on previous model, EAST adapts the link quality to dynamically maintain each link with respect to time. In a relatively stable network, control overhead occurs only in measuring link quality in initial phase. But in a relatively unstable network because link quality is continuously changing, initial phase is repeated and serious overhead occurs. Before we present block diagram for the proposed scheme, some variables are defined as follows: (1) current nodes in a region $n_c(t)$, (2) desired nodes in a region $n_d(t)$, and (3) error: $e(t) = n_d(t) - n_c(t)$, (4) P_{level}.

Figure 1 shows system block diagram of the proposed scheme. PRR, ACK, and $\text{RSSI}_{\text{loss}}$ are used to determine connectivity. ACK estimates connectivity, but it cannot determine link quality. PRR estimates connectivity accurately, but it causes significant overhead [8]. In our scheme, we use $\text{RSSI}_{\text{loss}}$ for connectivity estimation, which measures connectivity with relatively low overhead.

Power controller adjusts transmission P_{level} by utilizing both the number of current nodes and the temperature sensed at each node. Since power controller is operated not merely by comparing number of current nodes with desired nodes but by using temperature-compensated P_{level}, it can reach the desired P_{level} rapidly. If temperature is changing then temperature compensation is executed on basis of relationship between temperature and $\text{RSSI}_{\text{loss}}$. Network connectivity is maintained with low overhead by reducing feedback process between nodes which is achieved due to logical division of network.

Transmission power loss due to temperature variation was formulated using the relationship between $\text{RSSI}_{\text{loss}}$ and temperature experimented by Bannister et al. The mathematical expression for $\text{RSSI}_{\text{loss}}$ due to temperature variation is as follows [12]:

$$\text{RSSI}_{\text{loss}}\,[\text{dBm}] = 0.1996 * (T\,[^{\circ}\text{C}] - 25\,[^{\circ}\text{C}]). \qquad (1)$$

To compensate $\text{RSSI}_{\text{loss}}$ estimated from (1), we have to control output P_{level} of radio transmitter accordingly. The relationship between required transmitter P_{level} and $\text{RSSI}_{\text{loss}}$ is formulated using least square approximation [12] as follows:

Table 1: Estimated parameters.

N(A, B, C)	46, 30, 24
n_d(A, B, C)	41, 25, 19
n_c(A, B, C)	41, 22, 17
Threshold power level (A, B, C)	43.24, 31.77, 22.21
Nodes above threshold $\text{RSSI}_{\text{loss}}$(A, B, C)	23, 11, 8
Nodes below threshold $\text{RSSI}_{\text{loss}}$ (A, B, C)	18, 11, 9
PRR (A, B, C)	(80–98), (70–96), (63–97)%
Threshold $\text{RSSI}_{\text{loss}}$ (A, B, C)	3.78, −0.61, −5.17 dBm

$$P_{\text{level}} = \left[\frac{(\text{RSSI}_{\text{loss}} + 40)}{12}\right]^{2.91}. \qquad (2)$$

Based on (1) and (2), we obtain appropriate P_{level} to compensate $\text{RSSI}_{\text{loss}}$ due to temperature variation. To compensate path loss due to distance between each sensor node in WSN, free-space model helps to estimate actual required transmitter power. After the addition of $\text{RSSI}_{\text{loss}}$ due to temperature variation in (3), we estimate actual required transmitter power between each sensor node. For free-space path loss model, we need the number of nodes in a network (N), the distance between each node (d), (E_b/N_o) depending upon (SNR), spectral efficiency (η), frequency (f), and Receiver Noise Figure (RNF):

$$P_t\,[\text{dBm}] = \left[\eta * \left(\frac{E_b}{N_0}\right) * mkTB * \left(\frac{4\pi d}{\lambda}\right)^2 + \text{RNF}\right] + \text{RSSI}_{\text{loss}}. \qquad (3)$$

Parameters for propose scheme are (1) threshold $\text{RSSI}_{\text{loss}}$ for each region, (2) desired nodes in each region $n_d(t) = n_c(t) - 5$, and (3) transmission power level P_{level} for each region (Table 1).

Threshold $\text{RSSI}_{\text{loss}}$ is the minimum value required to maintain link reliability. A reference node broadcasts beacon message periodically to nodes and waits for ACKs. If ACKs are received from nodes, then $\text{RSSI}_{\text{loss}}$ is estimated for logical division of network, number of nodes with high $\text{RSSI}_{\text{loss}}$ considered in region A, medium $\text{RSSI}_{\text{loss}}$ considered in region B, and low $\text{RSSI}_{\text{loss}}$ in region C. If $\text{RSSI}_{\text{loss}} \geq \text{RSSI}_{\text{loss}}$ threshold and $n_c(t) \geq n_d(t)$, then threshold transmitter P_{level} is assigned if for similar case $n_c(t) < n_d(t)$ then similar transmitter P_{level} is assigned and if $\text{RSSI}_{\text{loss}} < \text{RSSI}_{\text{loss}}$ threshold then by default keep the same transmitter P_{level} as shown in Algorithm 1.

Figure 2 shows complete flow chart for reference node. A node senses temperature by using locally installed sensor and checks if a temperature change is detected. If there is any temperature change, compensation process is executed

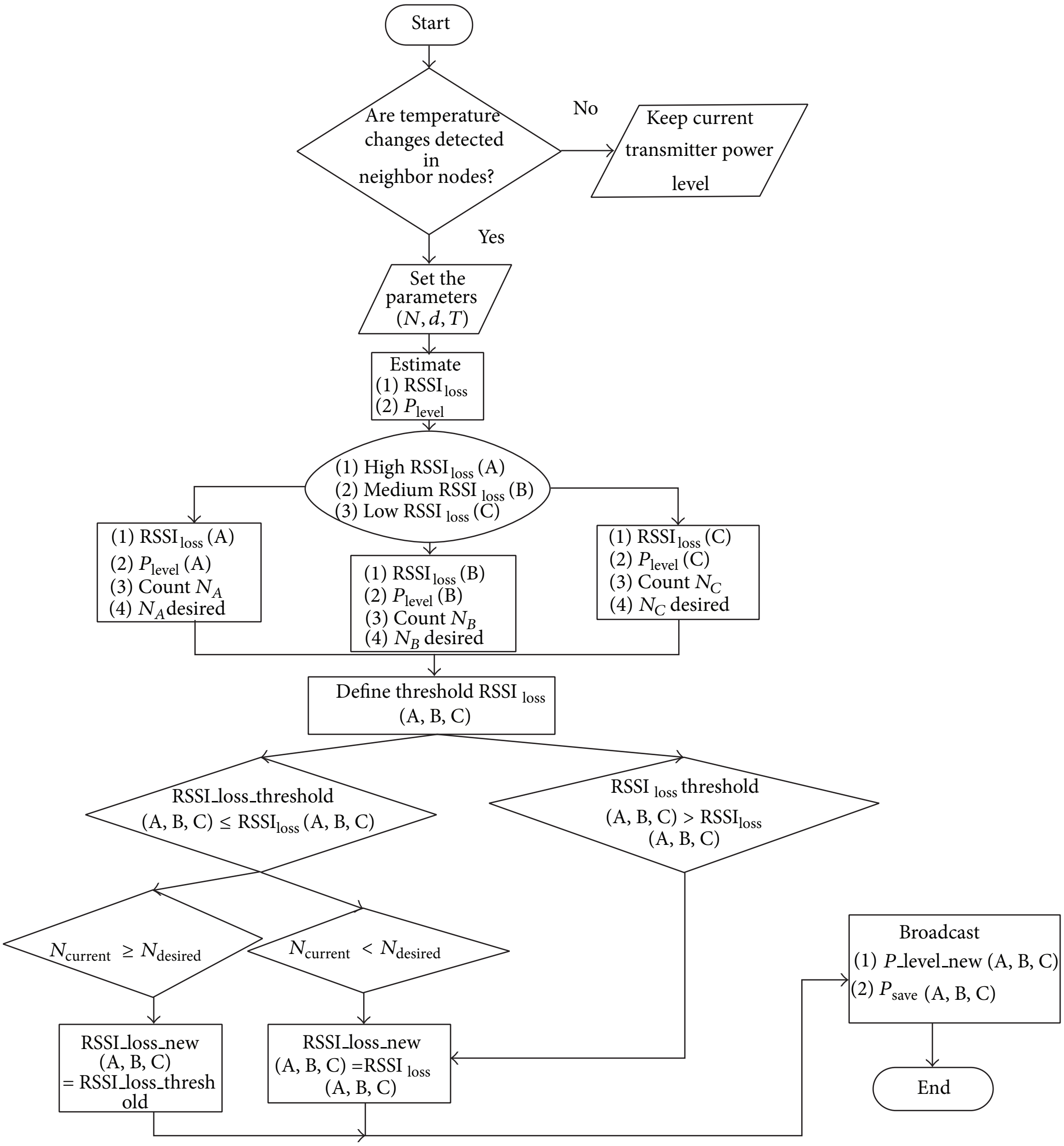

Start
Are temperature changes detected in neighbor nodes?
No
Keep current transmitter power level
Yes
Set the parameters (N, d, T)
Estimate
(1) $\text{RSSI}_{\text{loss}}$
(2) P_{level}
(1) High $\text{RSSI}_{\text{loss}}$ (A)
(2) Medium $\text{RSSI}_{\text{loss}}$ (B)
(3) Low $\text{RSSI}_{\text{loss}}$ (C)
(1) $\text{RSSI}_{\text{loss}}$ (A)
(2) P_{level} (A)
(3) Count N_A
(4) N_A desired
(1) $\text{RSSI}_{\text{loss}}$ (B)
(2) P_{level} (B)
(3) Count N_B
(4) N_B desired
(1) $\text{RSSI}_{\text{loss}}$ (C)
(2) P_{level} (C)
(3) Count N_C
(4) N_C desired
Define threshold RSSI $_{\text{loss}}$ (A, B, C)
RSSI_loss_threshold (A, B, C) ≤ $\text{RSSI}_{\text{loss}}$ (A, B, C)
RSSI $_{\text{loss}}$ threshold (A, B, C) > $\text{RSSI}_{\text{loss}}$ (A, B, C)
$N_{\text{current}} \geq N_{\text{desired}}$
$N_{\text{current}} < N_{\text{desired}}$
RSSI_loss_new (A, B, C) = RSSI_loss_thresh old
RSSI_loss_new (A, B, C) =RSSI $_{\text{loss}}$ (A, B, C)
Broadcast
(1) P_level_new (A, B, C)
(2) P_{save} (A, B, C)
End

Figure 2: Flow chart of reference node.

on the basis of (1) and (2). Nodes send an ACK message including temperature change information with a newly calculated P_{level}. Applying this temperature-aware compensation scheme, we can reduce overhead caused by conventional scheme in changing temperature environments.

4. Mathematical Representation of the Proposed Scheme

Let suppose that we have 100 nodes in a network that are randomly deployed represented as (N_i). Nodes are placed at different locations in a square area of $100*100$ m, and distance (d_i) between them is from 1 to 100 m. For the given environment, temperature (T_i) can have values in the range $-10^{\circ}\text{C} \leq T_i \leq 53^{\circ}\text{C}$ for all $i \epsilon N$.

$\text{RSSI}_{\text{loss}}$ due to the temperature variation can be formulated using the relation between $\text{RSSI}_{\text{loss}}$ and the temperature experimented in [12]. Equation for the $\text{RSSI}_{\text{loss}}$ for the temperature variation is as follows:

$$\text{RSSI}_{\text{loss}}(i)\,[\text{dBm}] = 0.1996 * \left(T_i\left[{}^{\circ}\text{C}\right] - 25\left[{}^{\circ}\text{C}\right]\right). \quad (4)$$

The relation between P_{level} and $\text{RSSI}_{\text{loss}}$ is formulated by using a least square approximation [12]:

$$P_{\text{level}}(i) = \left[\frac{\text{RSSI}_{\text{loss}}(i) + 40}{12}\right]^{2.91}. \quad (5)$$

```
$r \leftarrow$ Number of rounds
$N \leftarrow$ Number of nodes in Network
$d \leftarrow$ Distance between each node and reference node
$T \leftarrow$ Temperature for each node
$RSSI_{loss} \leftarrow$ Transmission power loss for each node
$P_{level} \leftarrow$ Power level for each node
$P_t \leftarrow$ Transmitter power for each node
Region $A \leftarrow HighRSSI_{loss}$
Region $B \leftarrow MediumRSSI_{loss}$
Region $C \leftarrow LowRSSI_{loss}$
$n_c(t) \leftarrow$ Current number of nodes
$n_d(t) \leftarrow$ Desired number of nodes
if $RSSI_{loss}(A,B,C) \geq RSSI_{loss}(Threshold)$ then
    if $n_c(t)(A,B,C) \geq n_d(t)(A,B,C)$ then
        $RSSI_{loss}(new)(A,B,C) = RSSI_{loss}(Threshold)$
    else
        $RSSI_{loss}(new)(A,B,C) = RSSI_{loss}(A,B,C)$
    end if
end if
if $RSSI_{loss}(A,B,C) < RSSI_{loss}(Threshold)$ then
    $RSSI_{loss}(new)(A,B,C) = RSSI_{loss}(A,B,C)$
end if
$P_{Save}(A,B,C) = P_{level} - P_{level}(new)(A,B,C)$
```

Algorithm 1: EAST algorithm.

Maximum, minimum, and average value of $\text{RSSI}_{\text{loss}}$ for all nodes in network can be formulated as follows:

$$\begin{aligned} \text{RSSI}_{\text{loss}}(\min) &= \min\left(\text{RSSI}_{\text{loss}}(i)\right), \\ \text{RSSI}_{\text{loss}}(\max) &= \max\left(\text{RSSI}_{\text{loss}}(i)\right), \\ \text{RSSI}_{\text{loss}}(\text{avg}) &= \frac{\left(\min\left(\text{RSSI}_{\text{loss}}(i)\right) + \max\left(\text{RSSI}_{\text{loss}}(i)\right)\right)}{2}. \end{aligned} \tag{6}$$

After finding maximum and minimum values of $\text{RSSI}_{\text{loss}}$, we will define upper and lower limits of $\text{RSSI}_{\text{loss}}$ to divide the network into three regions. A counter is also initialized at zero to count the number of nodes in each region. Then we define upper and lower bounds and check condition; nodes that follow this condition are considered to be in region A for all $i \in N$:

$$\begin{aligned} \text{RSSI}_{\text{loss}}(\text{A max}) &= \max\left(\text{RSSI}_{\text{loss}}(i)\right), \\ \text{RSSI}_{\text{loss}}(\text{A min}) &= \text{RSSI}_{\text{loss}}(\text{avg}) + 2 \end{aligned} \tag{7}$$

count = 0;

count_{A} = count + 1

Given that for all $i \in N$;

$\text{RSSI}_{\text{loss}}(i) \leq \text{RSSI}_{\text{loss}}(\text{A max})$ and $\text{RSSI}_{\text{loss}}(i) > \text{RSSI}_{\text{loss}}(\text{A min})$.

Similarly, we define upper and lower limits for regions B and C and also check nodes that follow given conditions are said to be in region, B and C, respectively:

$$\begin{aligned} \text{RSSI}_{\text{loss}}(\text{B max}) &= \text{RSSI}_{\text{loss}}(\text{avg}) + 2, \\ \text{RSSI}_{\text{loss}}(\text{B min}) &= \text{RSSI}_{\text{loss}}(\text{avg}) - 2. \end{aligned} \tag{8}$$

count = 0;

count_{B} = count + 1.

Given that for all $i \in N$,

$\text{RSSI}_{\text{loss}}(i) \leq \text{RSSI}_{\text{loss}}(\text{B max})$ and $\text{RSSI}_{\text{loss}}(i) > \text{RSSI}_{\text{loss}}(\text{B min})$

$$\begin{aligned} \text{RSSI}_{\text{loss}}(\text{C min}) &= \min\left(\text{RSSI}_{\text{loss}}(i)\right) \\ \text{RSSI}_{\text{loss}}(\text{C max}) &= \text{RSSI}_{\text{loss}}(\text{avg}) - 2 \end{aligned} \tag{9}$$

count = 0;

count_{C} = count + 1.

Given that forall $i \in N$,

$\text{RSSI}_{\text{loss}}(i) \leq \text{RSSI}_{\text{loss}}(\text{C max})$ and $\text{RSSI}_{\text{loss}}(i) \geq \text{RSSI}_{\text{loss}}(\text{C min})$.

To apply our proposed scheme EAST, we need to define threshold on $\text{RSSI}_{\text{loss}}$ for each region for energy-efficient communication between sensor nodes. Threshold on $\text{RSSI}_{\text{loss}}$ for each region depends upon $\text{RSSI}_{\text{loss}}$ of all nodes in a particular region and the number of nodes in that region. Threshold on $\mathrm{RSSI}_{\mathrm{loss}}$ for each region is defined as follows:

$$\mathrm{RSSI}_{\mathrm{loss}}(\mathrm{Threshold}_{\mathrm{A}}) = \sum_{i=1}^{\mathrm{count}_{\mathrm{A}}} \frac{(\mathrm{RSSI}_{\mathrm{loss}}(i))}{\mathrm{count}_{\mathrm{A}}},$$
$$\mathrm{RSSI}_{\mathrm{loss}}(\mathrm{Threshold}_{\mathrm{B}}) = \sum_{i=1}^{\mathrm{count}_{\mathrm{B}}} \frac{(\mathrm{RSSI}_{\mathrm{loss}}(i))}{\mathrm{count}_{\mathrm{B}}}, \quad (10)$$
$$\mathrm{RSSI}_{\mathrm{loss}}(\mathrm{Threshold}_{\mathrm{C}}) = \sum_{i=1}^{\mathrm{count}_{\mathrm{C}}} \frac{(\mathrm{RSSI}_{\mathrm{loss}}(i))}{\mathrm{count}_{\mathrm{C}}}.$$

PRR is also an important metric to measure link reliability. Here, $\mathrm{count}_{\mathrm{A}}$ is $n_d(t)$, and $\mathrm{count}_{\overline{\mathrm{A}}}$ is number of nodes not present in the region due to mobility, and $(\mathrm{count}_{\mathrm{A}} - \mathrm{count}_{\overline{\mathrm{A}}})$ is $n_c(t)$. It is defined as the number of nodes present in a region at a particular time $n_c(t)$ to the number of desired nodes $n_d(t)$ in a region. Similarly we can define PRR for regions B and C. PRR for all three regions is defined as given below:

$$\mathrm{PRR}_{\mathrm{A}} = \frac{\mathrm{count}_{\mathrm{A}} - \mathrm{count}_{\overline{\mathrm{A}}}}{\mathrm{count}_{\mathrm{A}}},$$
$$\mathrm{PRR}_{\mathrm{B}} = \frac{\mathrm{count}_{\mathrm{B}} - \mathrm{count}_{\overline{\mathrm{B}}}}{\mathrm{count}_{\mathrm{B}}}, \quad (11)$$
$$\mathrm{PRR}_{\mathrm{C}} = \frac{\mathrm{count}_{\mathrm{C}} - \mathrm{count}_{\overline{\mathrm{C}}}}{\mathrm{count}_{\mathrm{C}}}.$$

Here $\mathrm{PRR}_{\mathrm{A}}$, $\mathrm{PRR}_{\mathrm{B}}$, and $\mathrm{PRR}_{\mathrm{C}}$ are packet reception ratios for regions A, B, and C respectively. $\mathrm{RSSI}_{\mathrm{loss}}$ for each region on basis of the propose scheme for given conditions like threshold $\mathrm{RSSI}_{\mathrm{loss}}$, and $n_c(t)$ is formulated as follows:

$$\mathrm{RSSI}_{\mathrm{loss}}(\widetilde{\mathrm{A}}, \widetilde{\mathrm{B}}, \widetilde{\mathrm{C}})(i) = \mathrm{RSSI}_{\mathrm{loss}}(\mathrm{Threshold\ A, B, C}). \quad (12)$$

Given that for all $i \in N$,

$\mathrm{RSSI}_{\mathrm{loss}}(\mathrm{Threshold\ A, B, C}) \leq \mathrm{RSSI}_{\mathrm{loss}}(\mathrm{A, B, C})(i)$ and $n_c(t)(\mathrm{A, B, C}) \geq n_d(t)(\mathrm{A, B, C})$

$$\mathrm{RSSI}_{\mathrm{loss}}(\widetilde{\mathrm{A}}, \widetilde{\mathrm{B}}, \widetilde{\mathrm{C}})(i) = \mathrm{RSSI}_{\mathrm{loss}}(\mathrm{A, B, C})(i). \quad (13)$$

Given that for all $i \in N$,

$\mathrm{RSSI}_{\mathrm{loss}}(\mathrm{Threshold\ A, B, C}) \leq \mathrm{RSSI}_{\mathrm{loss}}(\mathrm{A, B, C})(i)$ and $n_c(t)(\mathrm{A, B, C}) \leq n_d(t)(\mathrm{A, B, C})$ or $\mathrm{RSSI}_{\mathrm{loss}}$ $(\mathrm{Threshold\ A, B, C}) > \mathrm{RSSI}_{\mathrm{loss}}(\mathrm{A, B, C})(i)$.

Estimation of P_{level} for new $\mathrm{RSSI}_{\mathrm{loss}}$ is formulated as forall $i \in N$:

$$P_{\mathrm{level}}(\widetilde{\mathrm{A}}, \widetilde{\mathrm{B}}, \widetilde{\mathrm{C}})(i) = \left[\frac{(\mathrm{RSSI}_{\mathrm{loss}}(\widetilde{\mathrm{A}}, \widetilde{\mathrm{B}}, \widetilde{\mathrm{C}})(i) + 40)}{12}\right]^{2.91}. \quad (14)$$

P_{save} is defined as the difference between P_{level}s assigned before and after applying the proposed scheme:

$$P_{\mathrm{save}}(\mathrm{A, B, C}) = \sum_{i=1}^{N} (P_{\mathrm{level}}(\mathrm{A, B, C})(i)) - \sum_{i=1}^{N} (P_{\mathrm{level}}(\widetilde{\mathrm{A}}, \widetilde{\mathrm{B}}, \widetilde{\mathrm{C}})(i)). \quad (15)$$

Network lifetime can be enhanced by maximizing P_{save}. The aim of the proposed scheme is to save maximum power with link reliability. Objective function formulation for P_{save} is defined forall $i \in N$:

$$\mathrm{Maximize} \sum_{i=1}^{N} (P_{\mathrm{save}}(i)). \quad (16)$$

Constraints to save maximum power are given below for all $i \in N$:

$$\sum_{i=1}^{N} \mathrm{RSSI}_{\mathrm{loss}}(\mathrm{A,B,C})(i) \geq \mathrm{RSSI}_{\mathrm{loss}}(\mathrm{Threshold\ A, B, C}), \quad (17)$$

$$\sum_{i=1}^{N} n_c(t)(\mathrm{A, B, C})(i) \geq \sum_{i=1}^{N} n_d(t)(\mathrm{A, B, C})(i), \quad (18)$$

$$\sum_{i=1}^{N} \mathrm{count}_{\mathrm{AT}}(\mathrm{A, B, C})(i) \geq \sum_{i=1}^{N} \mathrm{count}_{\mathrm{BT}}(\mathrm{A, B, C})(i). \quad (19)$$

Here, $\mathrm{count}_{\mathrm{AT}}$ and $\mathrm{count}_{\mathrm{BT}}$ are the number of nodes above and below the threshold in each region, respectively.

5. Results and Discussions

In this section, we describe simulation results of the proposed technique for energy-efficient transmission in WSNs. Simulation parameters are as follows: rounds 1200, temperature −10–53°C, distance (1–100) m, nodes 100, regions A, B, and C, η 0.0029, SNR 0.20 dB, bandwidth 83.5 MHz, frequency 2.45 GHz, RNF 5 dB, T 300 k, and E_b/N_0 8.3 dB. In Figure 3, we have shown values of meteorological temperature for one round that each sensor node has sensed. Let us suppose that we have 100 nodes in $100*100\ \mathrm{m}^2$ region and temperature can have values in range (−10–53)°C [13] for given meteorological condition of Pakistan. Reference node is placed at the edge of this region.

Different values of temperature for each sensor node based on meteorological condition help to estimate $\mathrm{RSSI}_{\mathrm{loss}}$(dBm). Figure 4 shows $\mathrm{RSSI}_{\mathrm{loss}}$(dBm) due to temperature variation in any environment using the relationship between $\mathrm{RSSI}_{\mathrm{loss}}$(dBm) and temperature (°C) given by Bannister et al. High $\mathrm{RSSI}_{\mathrm{loss}}$(dBm) means that the sensor node is placed in a region where temperature is high so the link does not have good quality. For temperature (−10–53)°C $\mathrm{RSSI}_{\mathrm{loss}}$(dBm) has value in range between (−6 dBm) and (5 dBm).

From Figure 4, it is also clear that link quality and $\mathrm{RSSI}_{\mathrm{loss}}$ have inverse relation, when temperature is high

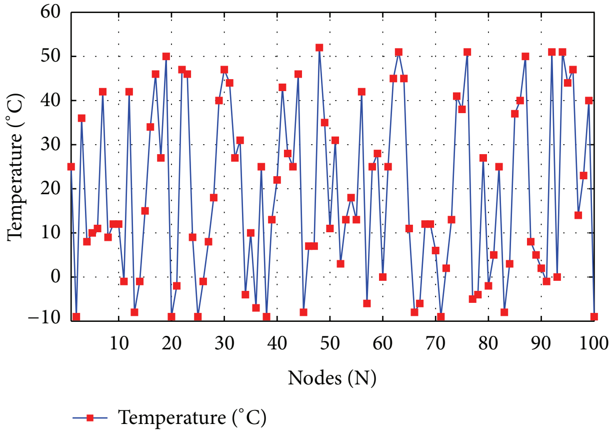


Figure 3: Temperature sensed at each sensor node.

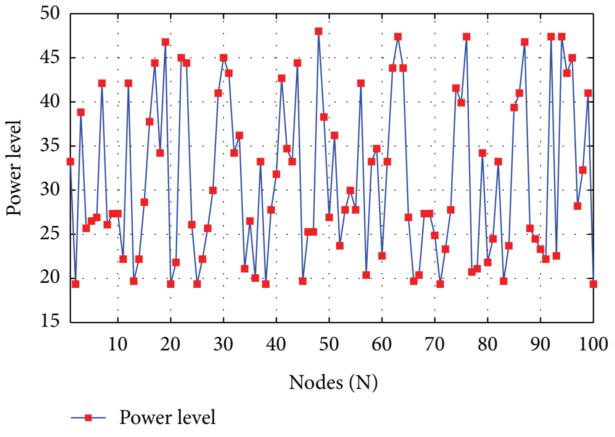


Figure 5: Required power level.

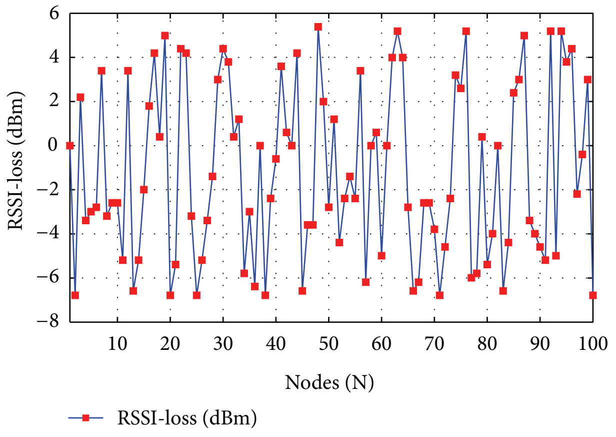


Figure 4: Estimated transmission power loss.

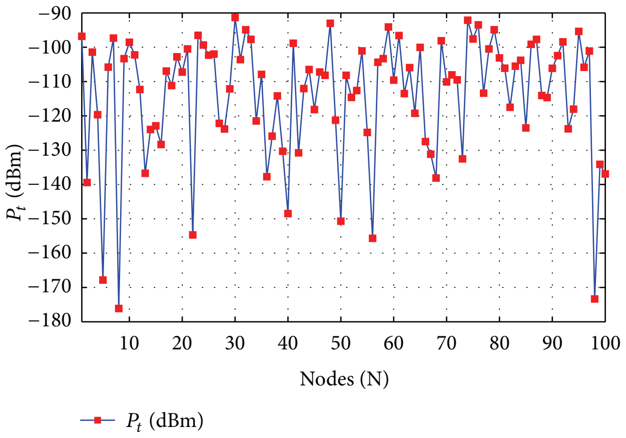


Figure 6: Transmitter power.

$\text{RSSI}_{\text{loss}}$ has high value which means low quality link and vice versa. After estimating $\text{RSSI}_{\text{loss}}$ for each node in WSN, we compute corresponding transmitter P_{level} to compensate $\text{RSSI}_{\text{loss}}$. Figure 5 shows the range of P_{level} on y-axis for given $\text{RSSI}_{\text{loss}}$, that is, between 20 and 47, and also variation of required P_{level} for sensor node with changing temperature, that is, at low temperature required P_{level} is low and for high temperature required P_{level} is high.

Estimated $\text{RSSI}_{\text{loss}}$ for each sensor node on the basis of given temperature helps to estimate corresponding P_{level}. That power level only helps to compensate $\text{RSSI}_{\text{loss}}$ due to temperature variations. To compensate path loss due to distance between each sensor node in WSNs, free-space model helps to estimate actual required transmitter power. After the addition of required P_{level} due to temperature variation and distance, we estimate actual required P_t between each sensor node. Figure 6 shows required P_t including both $\text{RSSI}_{\text{loss}}$ due to temperature variation and free-space path loss for different nodes. We clearly see from figure that P_t lies between −175 and 90, dBm and most of the time it is above −120 dBm.

In Figure 7, we have shown P_{level} using classical approach for three regions and in Figure 8 P_{level} for the proposed technique, EAST. We can clearly see the difference between P_{level} assigned. To show P_{level} for each region, we take the difference between the assigned P_{level}s using EAST and classical technique, as can be seen in Figures 9, 10, and 11. As we know that in classical approach, there is no concept of subregions, for the sake of comparison with the proposed technique, EAST, we have shown P_{level} for different regions using classical approach.

After estimating $\text{RSSI}_{\text{loss}}$ for nodes of each region, we have estimated required P_{level} for nodes of each region that we clearly see in Figure 7; in region A, P_{level} lies 40–45, for region B 30–35 and for region C 20–25. It means that for region A required P_{level} is higher than both other regions that also shows that for that region temperature and $\text{RSSI}_{\text{loss}}$ are large. For regions B required P_{level} is between both region A and C, and for C region required P_{level} is less than both other two regions. We have earlier seen in Figure 7 that P_{level} for each region is assigned using classical approach. After applying the proposed technique, we see what P_{level} required for each region. We can clearly see difference from between P_{level} as shown in Figure 8 that the required P_{level} decreases for each region and for region A it decrease at its maximum. Figures 9, 10, and 11, respectively, show required P_{save} for regions A, B,

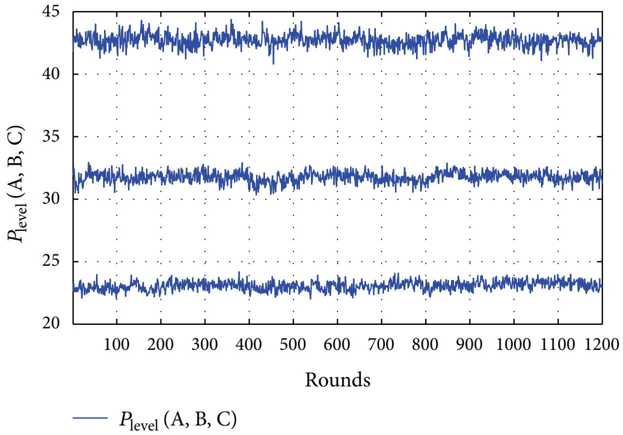


Figure 7: Power level using classical approach for regions A, B, and C.

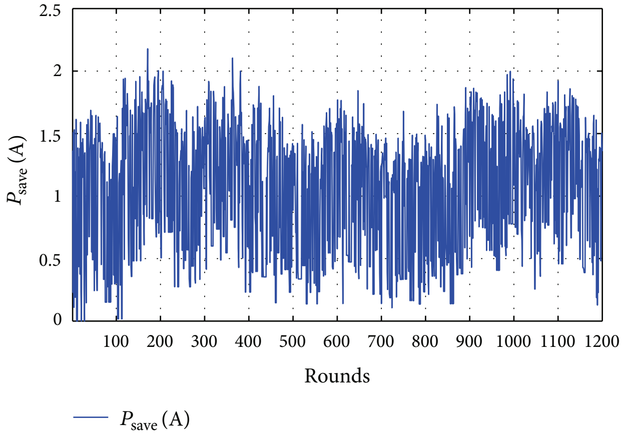


Figure 9: Difference of Power level save between classical technique and EAST for region A.

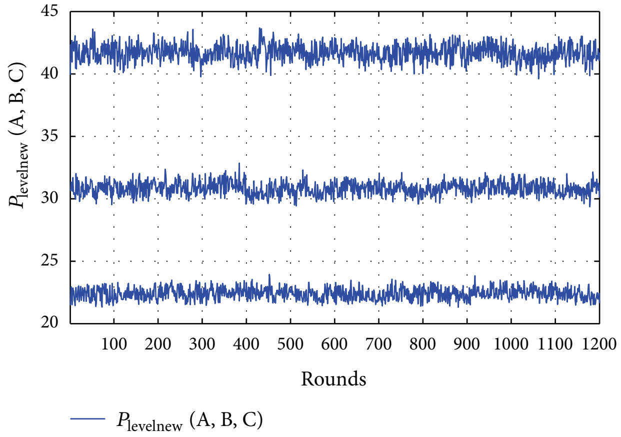


Figure 8: Power level using EAST for regions A, B, and C.

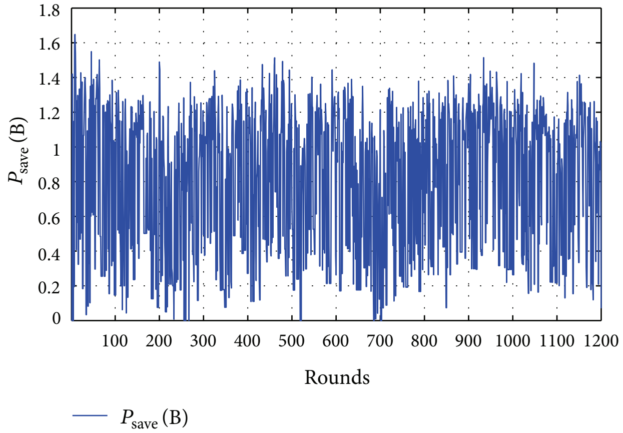


Figure 10: Difference of power level save between classical technique and EAST for region B.

and C after implanting the proposed technique. P_{save} values reach 2.3 for region A, 1.7 for B and 1.5 for C.

Figure 12 describes the effect of reference node mobility on P_{save} for region A. Reference node moves around boundaries of square region, and nodes in a region are considered to be static. When a reference node is at center location (50,50) of network, maximum nodes around reference node have large $\text{RSSI}_{\text{loss}}$ than threshold so we need to reduce P_{level} to meet threshold P_{level} requirements that cause maximum P_{save}. We can clearly see maximum P_{save} 12 dBm to 20 dBm for center location. When a reference node moves from center to one of the corner (0,0) of square region P_{save} remains constant approximately around 1 dB, because the nodes near reference node having same $\text{RSSI}_{\text{loss}}$ bear constant temperature and they need approximately the same P_{level} near threshold. P_{save} for reference node movement from (0,0) to (0,100) fluctuates between −5 dBm and 6 dBm, and at two moments we observe maximum P_{save} because a number of nodes near reference node have to increase their P_{level} to meet threshold's minimum.

Movement of reference node from (0, 100) to (100, 100) causes P_{save} between −4 dBm and 12 dBm, and only one time peak P_{save}. Similarly, when a reference node moves from (100, 100) to (100, 0), P_{save} remains in limits between −4 dBm and 7 dBm, and only one time maximum P_{save}. From this figure it is also clear that for region A reference node location at center gives maximum P_{save} that enhances network lifetime. We can also see that variation of P_{save} depends upon the distance of nodes from reference node, as if nodes have shorter $\text{RSSI}_{\text{loss}}$ than threshold, then we have to increase P_{level} that enhances P_{save} and vice versa. It is also clear from result that peak maximum and minimum P_{save} comes at a same time.

Similarly we can see P_{save} for similar pattern of reference node mobility considering regions B and C. For region B in Figure 13 when reference node at center location (50, 50) P_{save} remains between 14 dBm and 20 dBm, from center to (0, 0) P_{save} remains between 0 and 1dBm. When a reference node moves from (0, 0) to one of the corner of square regions (0, 100) P_{save} fluctuates between 0 and 4 dBm. Reference node's movement from (0, 100) to (100, 100) causes P_{save} to change from 1 dBm to 5 dBm and from (100, 100) to (100, 0) increases P_{save} change from 4 dBm to 5 dBm.

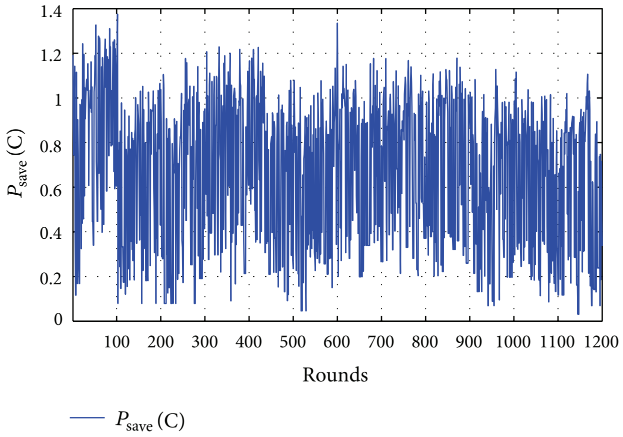


FIGURE 11: Difference of power level save between classical technique and EAST for region C.

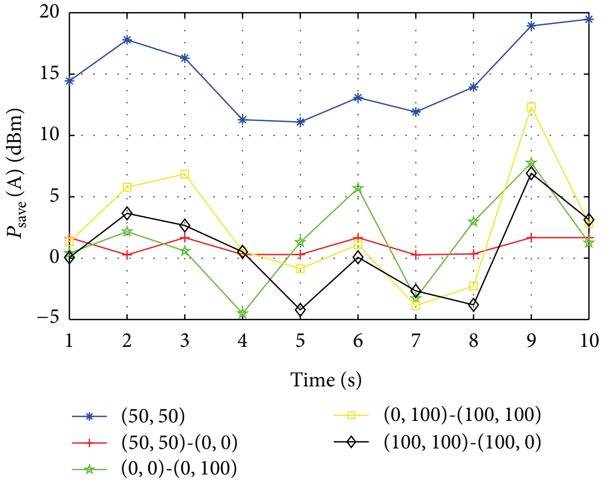


FIGURE 12: Transmitter power save in region A for different reference node locations.

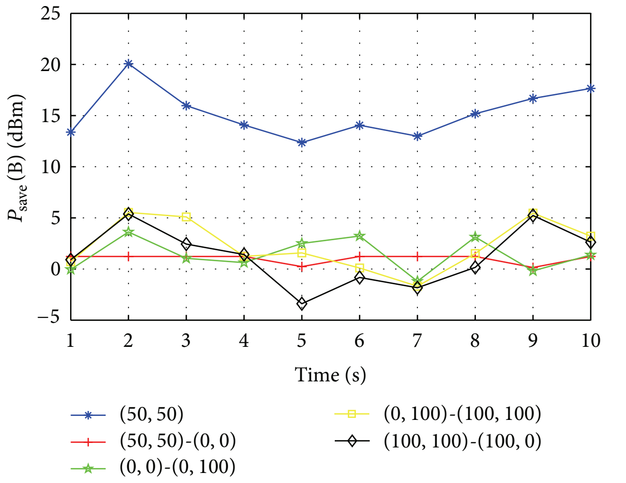


FIGURE 13: Transmitter power save in region B for different reference node locations.

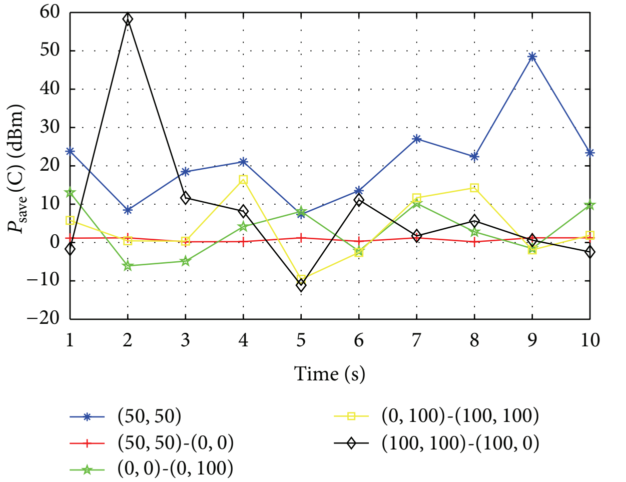


FIGURE 14: Transmitter power save in region C for different reference node locations.

This figure also indicates that P_{save} for region B is maximum when a reference node is at center location. For reference node mobility from center to (0, 0), P_{save} remains constant due to constant $\text{RSSI}_{\text{loss}}$ near reference node region. For other reference node movements, P_{save} remains approximately constant due to less variations in $\text{RSSI}_{\text{loss}}$. Compared to region A where P_{save} goes to peak maximum and minimum values in region B, P_{save} remains on average approximately constant and less variation occurs; fact is that nodes in region B have approximately the same $\text{RSSI}_{\text{loss}}$ near threshold.

P_{save} for reference node mobility in region C around square is shown in Figure 14. When a reference node is at center (50, 50), P_{save} fluctuates between 8 dBm and 50 dBm. From center to edge (0, 0), reference node mobility causes P_{save} around 0 dBm. When a reference node moves from a corner of square (0, 0) to corner (0, 100), P_{save}−5 dBm–12 dBm. Similarly from (0, 100) to (100, 100), P_{save} remains between −10 dBm and 18 dBm. Finally when the reference node is location changes from (100, 100) to (100, 0) and P_{save} goes to maximum value 60 dBm that shows that nodes near reference node have larger $\text{RSSI}_{\text{loss}}$ than threshold $\text{RSSI}_{\text{loss}}$ at that moment. Figures 12, 13, and 14 also elaborate that on average P_{save} is maximum for reference node location at center. Compared to region B, in this region peak maximum and minimum P_{save} exists the reason is that nodes in this region have larger $\text{RSSI}_{\text{loss}}$ than threshold at that moment.

6. Conclusion and Future Work

In this paper, we presented a new proposed technique, EAST. It shows that temperature is one of the most important factors impacting link quality. Relationship between $\text{RSSI}_{\text{loss}}$ and temperature has been analyzed for our transmission power control scheme. The proposed scheme uses open-loop control to compensate for changes of link quality according to temperature variation. By combining both open-loop temperature-aware compensation and close-loop feedback control, we can significantly reduce overhead of transmission power control in WSN; we further extended our scheme by dividing network into three regions on basis of threshold $\mathrm{RSSI}_{\mathrm{loss}}$ and assigned P_{level} to each node in three regions on the basis of current number of nodes and the desired number of nodes, which helps to adapt P_t according to link quality variation and increase network lifetime. We have also evaluated the performance of the proposed scheme for reference node mobility around square region that shows P_{save} up to 60 dBm. But in case of static reference node, P_{save} goes maximum to 2 dBm.

In future, firstly, we are interested to work on Internet Protocol (IP) based solutions in WSNs [14]. Secondly, as sensors are usually deployed in potentially adverse environments [15], we will address the security challenges using the intrusion detection systems because they provide a necessary layer for the protection.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[2] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, article 16, 2010.

[3] K. Lin, M. Chen, S. Zeadally, and J. J. Rodrigues, "Balancing energy consumption with mobile agents in wireless sensor networks," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 446–456, 2012.

[4] K. Lin, J. J. Rodrigues, H. Ge, N. Xiong, and X. Liang, "Energy efficiency qos assurance routing in wireless multimedia sensor networks," *IEEE Systems Journal*, vol. 5, no. 4, pp. 495–505, 2011.

[5] M. Kubisch, H. Karl, A. Wolisz, L. Zhong, and J. Rabaey, "Distributed algorithms for transmission power control in wireless sensor networks," in *Wireless Communications and Networking, (WCNC '03)*, vol. 1, pp. 558–563.

[6] J. Jeong, D. Culler, and J. Oh, "Empirical analysis of transmission power control algorithms for wireless sensor networks," in *Fourth International Conference on Networked Sensing Systems (INSS '07)*, pp. 27–34, IEEE, 2007.

[7] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "ATPC: adaptive transmission power control for wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 223–236, November 2006.

[8] M. Meghji and D. Habibi, "Transmission power control in multihop wireless sensor networks," in *Proceedings of the 3rd International Conference on Ubiquitous and Future Networks (ICUFN '11)*, pp. 25–30, June 2011.

[9] F. Lavratti, A. R. Pinto, D. Prestes, L. Bolzani, F. Vargas, and C. Montez, "A transmission power self-optimization technique for wireless sensor networks," in *Proceedings of the 11th Latin-American Test Workshop (LATW '10)*, March 2010.

[10] V. G. Douros and G. C. Polyzos, "Review of some fundamental approaches for power control in wireless networks," *Computer Communications*, vol. 34, no. 13, pp. 1580–1592, 2011.

[11] X. Cui, X. Zhang, and Y. Shang, "Energy-saving strategies of wireless sensor networks," in *Proceedings of the IEEE International Symposium on Microwave, Antenna, Propagation, and EMC Technologies for Wireless Communications (MAPE '07)*, pp. 178–181, August 2007.

[12] K. Bannister, G. Giorgetti, and S. Gupta, "Wireless sensor networking for hot applications: effects of temperature on signal strength, data collection and localization," in *Proceedings of the 5th Workshop on Embedded Networked Sensors (HotEmNets '08)*, Citeseer, 2008.

[13] S. Cheema, G. Rasul, and D. Kazmi, "Evaluation of projected minimum temperatures for northern pakistan," *Pakistan Journal of Meteorology*. In press.

[14] J. J. P. C. Rodrigues and P. A. C. S. Neves, "A survey on ip-based wireless sensor networks solutions," *Communication Systems*, vol. 23, no. 8, pp. 963–981, 2010.

[15] G. Han, J. Jiang, W. Shen, L. Shu, and J. J. P. C. Rodrigues, "Idsep: a novel intrusion detection scheme based on energy prediction in cluster-based wireless sensor networks," *IET Information Security*. In press.