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

Copyright © 2013 Md. M. Islam and E.-N. Huh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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

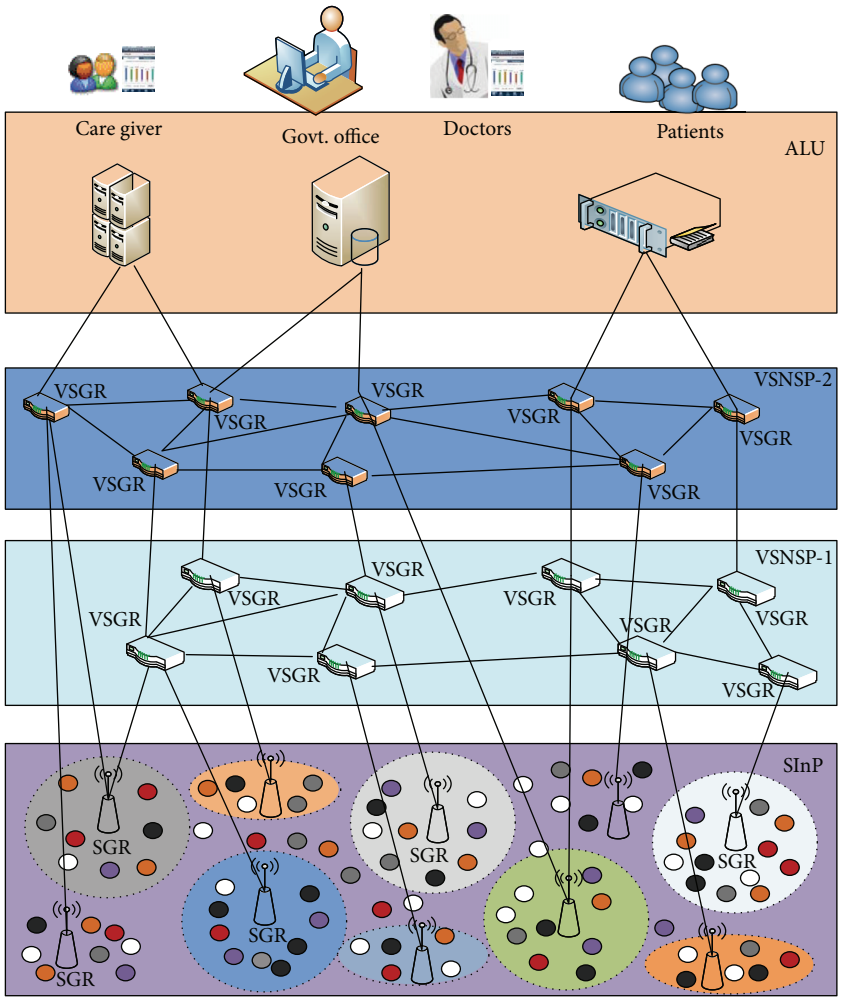


Figure 1: Business model for VSN.

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.

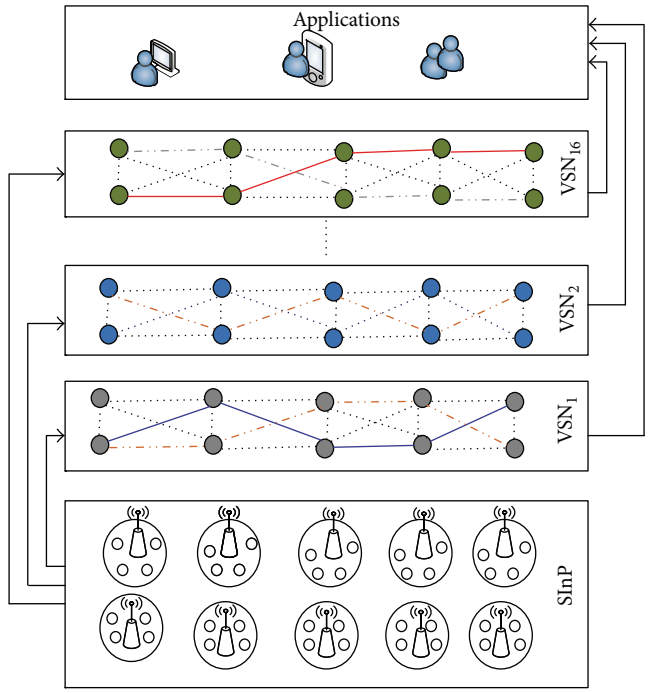


Figure 2: VSN architecture.

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

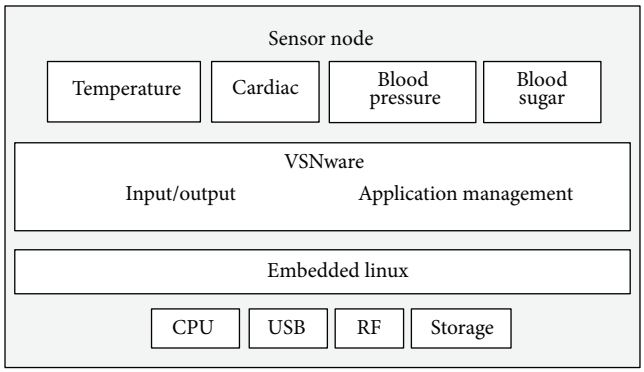


Figure 3: Architecture of sensor node.

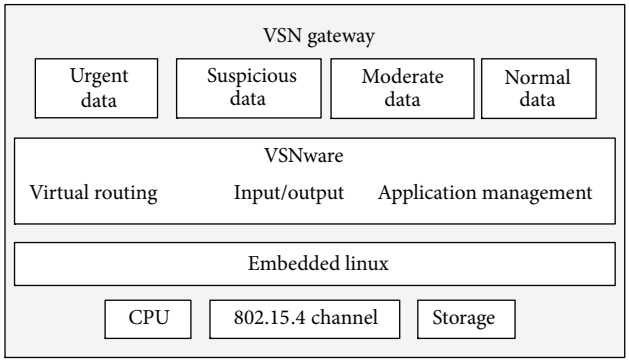


Figure 4: Architecture of SGR.

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.

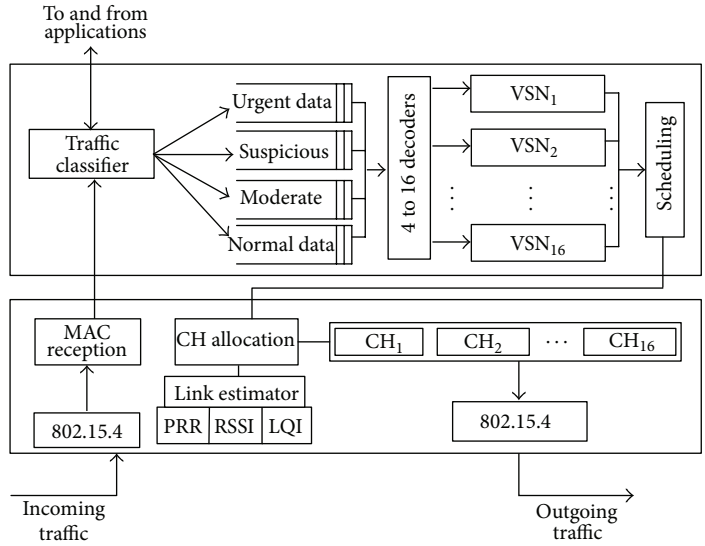


Figure 5: Protocol architecture of SGR.

(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:

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

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:

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

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:

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

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:

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

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:

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

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

$$p(m,n) = 1 - \left[\prod_{i=1}^{16} \left[1 - p_i(k_i)\right]\right]\left[\prod_{j=1}^{m} \left[1 - p_j\left(k_j\right)\right]\right],$$

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

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

$$p(m) = 1 - \left[\prod_{j=1}^{m} \left[1 - p_j\left(k_j\right)\right]\right],$$

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

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

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

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

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

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, $p_{s\text{Path}}^{s\text{VSN}}$, is

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

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

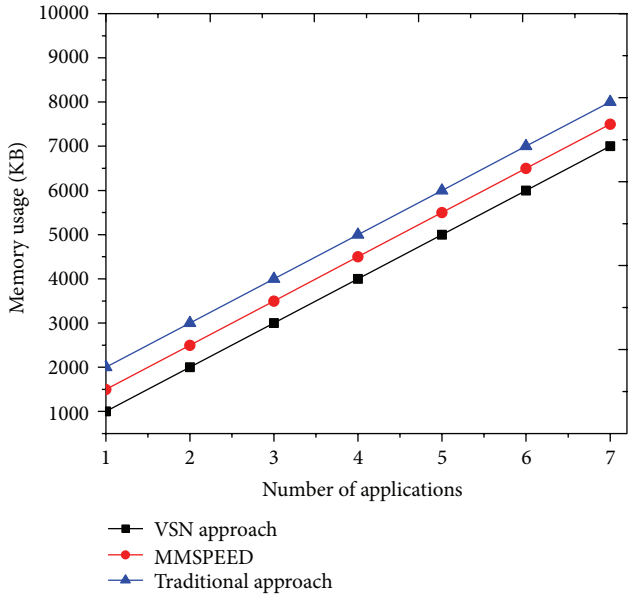


Figure 6: Comparative memory usage in VSN approach.

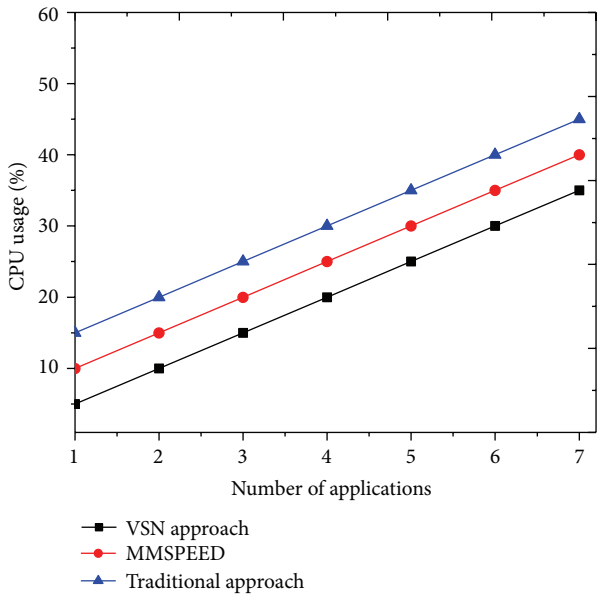


Figure 8: Comparative CPU usage in VSN approach.

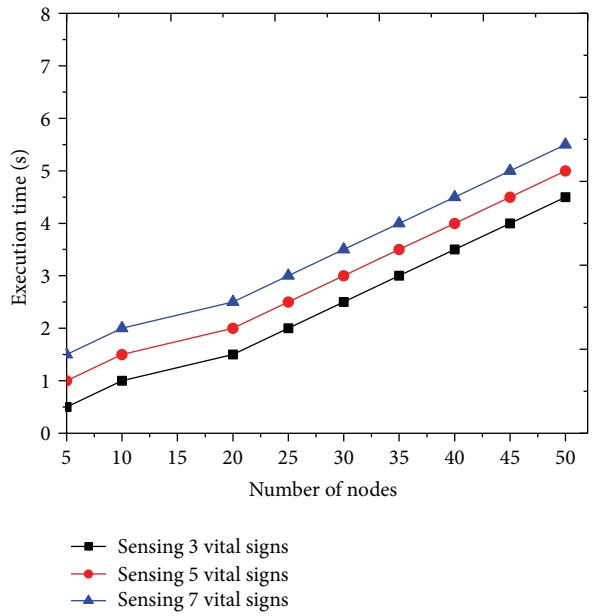


Figure 7: Execution time versus number of nodes.

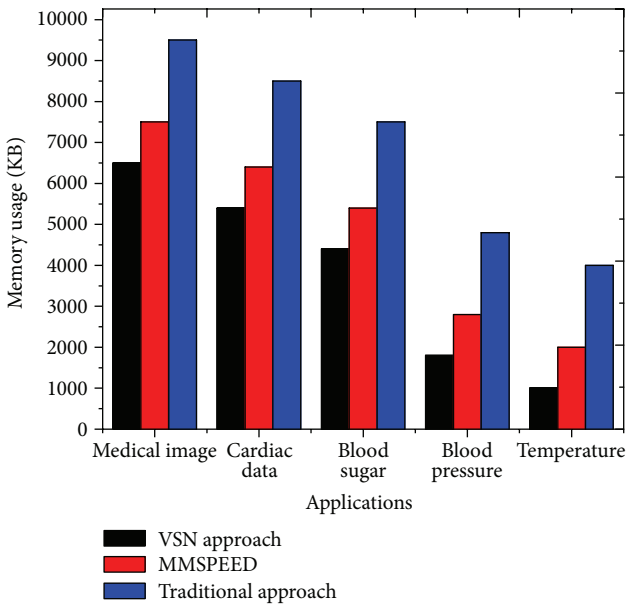


Figure 9: Comparative memory usage by applications.

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.

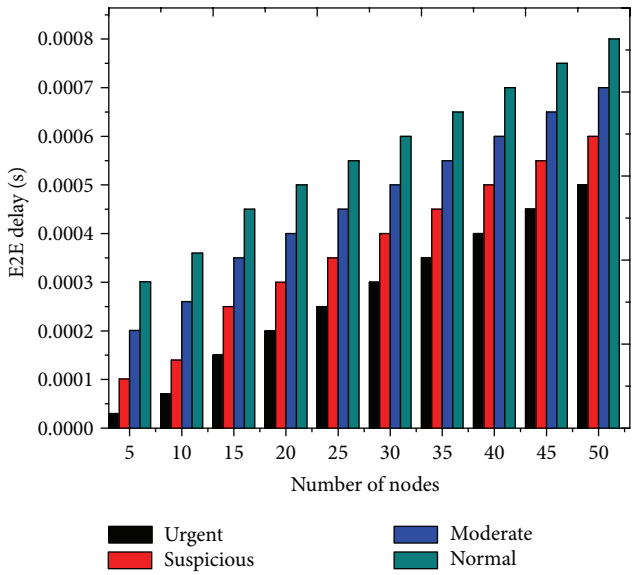


Figure 10: End-to-end delay versus number of nodes.

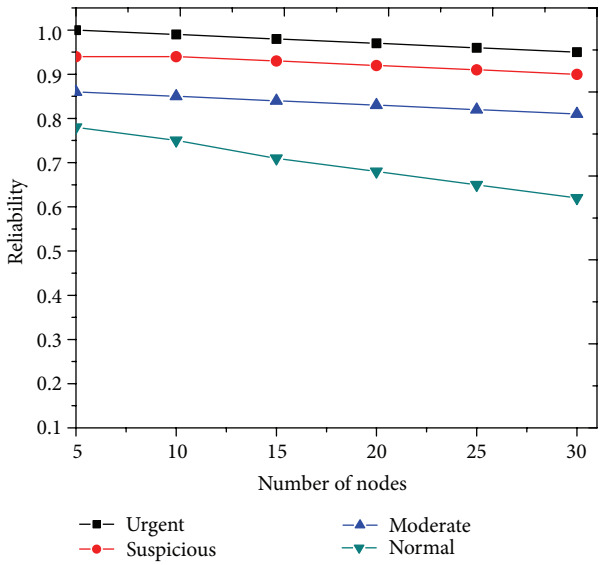


Figure 12: Reliability versus number of nodes.

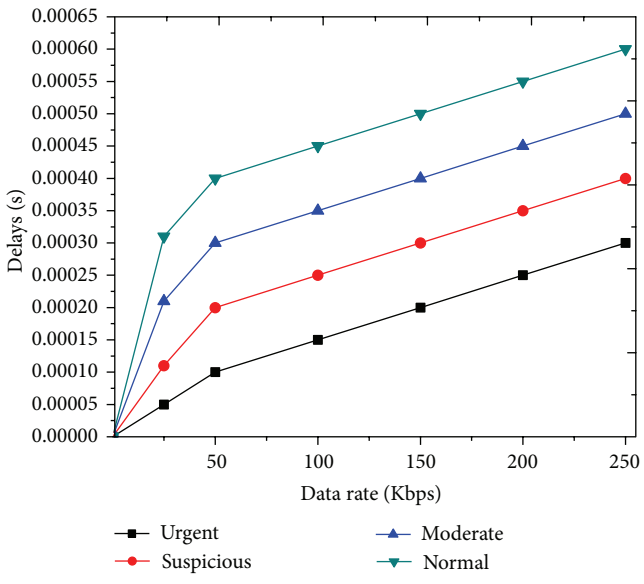


Figure 11: Delay versus data rate.

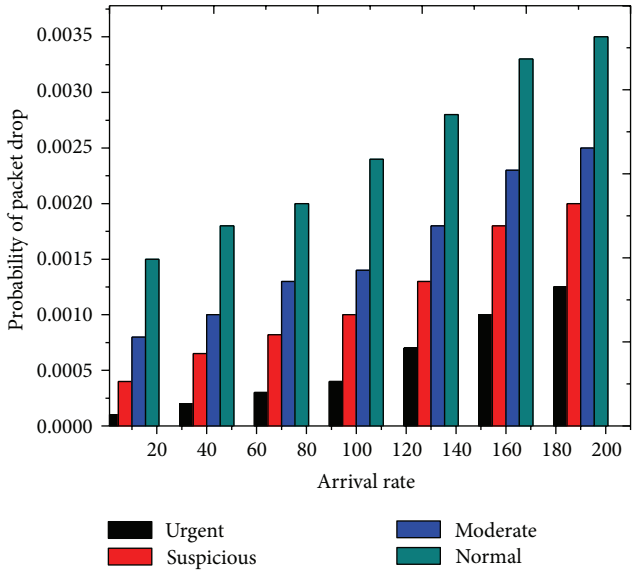


Figure 13: Packet dropping probability versus arrival rate.

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.

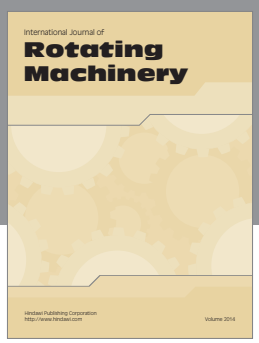
[19] N. M. Mosharaf, K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proceedings of the IEEE 28th Conference on Computer Communications (INFOCOM '09)*, pp. 783–791, April 2009.

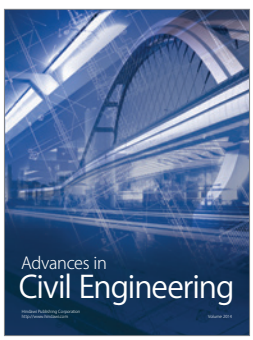
[20] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, 2012.

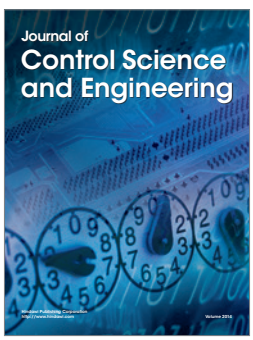
[21] A. H. Shuaib and A. H. Aghvami, "A routing scheme for the IEEE-802.15.4-enabled wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5135–5151, 2009.

[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
Submit your manuscripts at
http://www.hindawi.com