Wireless Communications and Mobile Computing

Real-Time Communication in Wireless Sensor Networks

Lead Guest Editor: Ki-Il Kim
Guest Editors: Babar Shah, Jeongcheol Lee, Giovanni Pau, and Javier Prieto

Real-Time Communication in Wireless Sensor Networks

Real-Time Communication in Wireless Sensor Networks

Lead Guest Editor: Ki-Il Kim
Guest Editors: Babar Shah, Jeongcheol Lee, Giovanni Pau, and Javier Prieto

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9612631, 2 pages
https://doi.org/10.1155/2018/9612631

Editorial

Real-Time Communication in Wireless Sensor Networks

Jeongcheol Lee,[1] Babar Shah ,[2] Giovanni Pau ,[3] Javier Prieto ,[4] and Ki-Il Kim [5]

[1]*Department of Scientific Platform Development, Computational Science & Engineering Center, Korea Institute of Science and Technology Information (KISTI), Daejeon, Republic of Korea*
[2]*College of Technological Innovation, Zayed University, Abu Dhabi, UAE*
[3]*Kore University of Enna, Enna, Italy*
[4]*BISITE Research Group, University of Salamanca, Salamanca, Spain*
[5]*Department of Computer Science and Engineering, Chungnam National University, Daejeon, Republic of Korea*

Correspondence should be addressed to Ki-Il Kim; kikim@cnu.ac.kr

Received 9 January 2018; Accepted 11 January 2018; Published 8 February 2018

Through wireless sensor networks (WSN), we can acquire the various interesting event information around sensor nodes through multihop communications. In WSN, there are two types of applications, that is, event or query based. Commonly, in these application, the value on each sensor node is very sensitive to delay or latency. So, it is strongly required to deliver data to sink node within the deadline since data received after the deadline is not acceptable at all in WSN. The good example of application demanding real-time communication in WSN includes tracking of moving object and intrusion detection.

However, compared to typical networks, it is very difficult to achieve real-time communication in WSN. Severe constraints such as limited computing power and narrow bandwidth are not suitable to provide real-time communication accordingly. So, a number of important issues and research challenges have to be addressed to provide real-time communication in WSN. Based on this demand, this special issue is planned to contribute to advances in real-time communications in WSN.

While considering our objective, editors believe that this special issue provides collection of articles on networking technique in real-time communications. We have selected 4 valuable papers by evaluating several aspects such as relevance to special issue and novelty of solution. The topic of these papers is roughly categorized into the following areas: multichannel transmission, MAC protocol, testbed for routing protocol, and comprehensive survey for real-time in WSN.

In the paper entitled "Priority-Based Dynamic Multichannel Transmission Scheme for Industrial Wireless Networks," Y. Igarashi et al. proposed priority-based dynamic multichannel transmission scheme for industrial wireless sensor networks (IWSN) where applications are required to provide precise measurement functions as feedback for controlling devices. In order to guarantee latency for unpredictable on-demand communications, a root node controls the transmission timing of high-priority packets, while other nodes autonomously decide what channel to use and when to transmit packets to a neighbor. In the proposed scheme, packet priority is determined in accordance with application requirements. The proposed scheme operates over a MAC layer and does not rely on any specific MAC protocol. Since there are some standard protocols for real-time communications in IWSN such as ISA100.11a and wirelessHART, the authors discuss compatibility with ISA100.11a in this paper.

Another paper is related to MAC protocol. However, instead of general MAC protocol, T. Kim et al. proposed an efficient MAC protocol for radio frequency (RF) energy harvesting in WSN, called REACH. Unlike conventional RF energy harvesting methods, an Energy Transmitter (ET) in the proposed scheme can actively send RF energy signals without Request-for-Energy (RFE) messages. An ET determines the active energy signal transmission according to the consequence of the passive energy harvesting procedures. The other feature of the proposed scheme is that an ET participates contention-based channel access procedure. The simulation results reveal that the proposed protocol can increase the energy harvesting rate and the lifetime of WSN by the help of the active energy signal transmission method.

In addition to approaches in physical and data link, W. Jiang et al. addressed the testbed for multipath routing protocol to manage deployment cost. To achieve this goal, a reconfigurable testbed, which supports dynamic protocol switching by creating a novel architecture and experiments with several different protocols, is presented. Based on architecture to separate control and data plane in this testbed, independent routing configuration and data transmission are assumed. Moreover, a programmable flow table provides the testbed with the ability to switch protocols dynamically. The experiments demonstrate that the testbed can reduce the complexity of network management and updating by separating the control and data planes. A user can manage the network easily by simply configuring different behaviors, without redeploying the whole network. Therefore, this reconfiguration function can support WSN virtualization when the scale is growing up and can be used in a multitenant architecture for cloud service.

Another paper is a comprehensive survey paper for literature reviews in real-time communications in WSN. Even though few survey literatures were published, they are out-of-date and fail to provide overview of diverse protocols. To address this problem, this survey paper is likely to collect the research work since 2010 and include the potential real-time application and applicable platform for real-time communication in WSN. In addition, unlike previous survey papers, B.-S. Kim et al. included firm real-time model with hard and soft real-time model in WSN. Since several research works for (m, k)-firm model in WSN have been proposed recently, it is recommended to analyze and compare them accordingly. Related to rule for categorization, research area on layered architecture has been chosen. In each category, recent research work is described and explained. Finally, the open issues and further research challenges are discussed.

Acknowledgments

Finally, we appreciate all authors, reviewers, and editorial members for their invaluable contribution. Without their hard work and dedication, it would not have been possible to select these highly qualified papers within the given time limits of this special issue.

Jeongcheol Lee
Babar Shah
Giovanni Pau
Javier Prieto
Ki-Il Kim

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 1594270, 10 pages
https://doi.org/10.1155/2017/1594270

Research Article

A Dynamically Reconfigurable Wireless Sensor Network Testbed for Multiple Routing Protocols

Wenxian Jiang,[1] Chenzhe Gu,[1] and Jingjing Wu[2,3]

[1] *College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China*
[2] *College of Mathematics and Computer Science, Quanzhou Normal University, Quanzhou 362000, China*
[3] *Fujian Provincial Key Laboratory of Data Intensive Computing, Quanzhou 362000, China*

Correspondence should be addressed to Wenxian Jiang; jwx@hqu.edu.cn

Received 20 May 2017; Accepted 23 August 2017; Published 8 November 2017

Academic Editor: Javier Prieto

Because wireless sensor networks (WSNs) are complex and difficult to deploy and manage, appropriate structures are required to make these networks more flexible. In this paper, a reconfigurable testbed is presented, which supports dynamic protocol switching by creating a novel architecture and experiments with several different protocols. The separation of the control and data planes in this testbed means that routing configuration and data transmission are independent. A programmable flow table provides the testbed with the ability to switch protocols dynamically. We experiment on various aspects of the testbed to analyze its functionality and performance. The results demonstrate that sensors in the testbed are easy to manage and can support multiple protocols. We then raise some important issues that should be investigated in future work concerning the testbed.

1. Introduction

The self-organization and multihop features of wireless sensor networks (WSNs) allow nodes to communicate with each other quickly. Nodes in WSNs, which are different from traditional network nodes, often carry out information collection, signal processing, and wireless communication simultaneously [1]. Meanwhile, new network structures and algorithms have been constructed and tested on simulators. Simulation allows WSNs research to be carried out quickly and easily but may provide inaccurate results because of the lack of a real environment. This has motivated the design and implementation of a number of WSN testbeds by researchers across the globe. However, most testbeds concentrate on specific applications or routing protocols, resulting in a lack of abstraction and reconfiguration capabilities, which makes policy changing and network management rigid.

Network reconfiguration allows an operator to change functions and update policies at any time. These dynamic characteristics make a network flexible and scalable. Meanwhile, owing to the resource limitations of WSNs, it is important to create an adaptive architecture to reduce the complexity of deployment and execution. In general, there are three forms of reconfiguration for WSNs: software, data routing, and hardware [2]. Hardware reconfiguration provides flexibility on a sensor itself, such as activity or energy consumption. In terms of functionality, software and protocol reconfiguration can support various scenarios at runtime. For most networks, reconfiguration can be implemented by dynamic reprogramming [3, 4] or the loading of different components [5].

In this study, a dynamically reconfigurable WSN testbed is designed and it is easy to manage and reconfigure with different routing protocols. Rather than simply implementing a testbed, the target focus is on the architecture of software and protocol reconfiguration. The testbed touches data and routing control traffic (including forwarding policies and protocol reconfiguration). In the design and experiment, the separation of the data and control planes can help to reduce the complexity of the platform and allow the support of various routing protocols. Meanwhile, a programmable *flow table* of the data plane makes the implementation of policy changes fast and clear. Our work presents a concrete form of the method and concept for establishing the testbed and demonstrates its flexibility and scalability by several protocols in the experiment.

The remainder of this paper is organized as follows. In Section 2, we describe the motivation for this work. An architecture overview of the testbed is presented on the framework and operating system in Section 3. The method of routing reconfiguration and data processing is described in Section 4. Section 5 describes the general method for implementing a protocol for the testbed. Section 6 presents some experiments based on different protocols, in order to evaluate the testbed. Conclusions and directions for future work are presented in Section 7.

2. Motivation

WSN testbeds usually consist of a set of sensor nodes, a backchannel for testbed management connectivity, a database for data logging, and a management or debugging software application deployed on a server. Previous research has focused on the scale, speed, mobility, network architectures, and so on [6]. For example, [7] presented one of the first fully developed WSN testbeds and provided an open source tool to users for creating their own experiments. Reference [8] can support different network architectures and topologies by reprogramming or suspending some motes. By deploying two different radios on one mote, [9] implemented a heterogeneous and complex testbed to support various speed rates in the network. For high performance hardware and the mass deployment of nodes, researchers have utilized different testbeds to complete various tasks, such as system design, protocol testing, and performance evaluation. Table 1 provides a lateral comparison of the main characteristics of some important testbeds.

According to Table 1, WSN testbeds have been designed to achieve various functions, scales, and communication layers, but most of these do not have the ability to support multiple and dynamic protocol tests. Although there are some testbeds that can support different network architectures, the routing layer still needs to be reconfigured or even redeployed for every experiment. It is worth mentioning that IOT-Lab provides three layers of API: drivers, operating systems, and communication libraries, allowing users to define their own network policies. However, regarding the following demands, there are significant disparities and challenges present in existing testbeds.

Network Programmability. The support of multiple applications or protocols: this is achieved by programmable hardware or new design structures between the data and control planes [4, 13]. However, existing WSN testbeds always concentrate on a specific application or scenario, which leads to incompatible policy changes or function that are too simple.

Packet Processing. Data traffic and control traffic should be separated by different modules. In some situations, the data process consists of nodes performing flow-based packet forwarding, and the network contains one or more controllers (or sinks) managing all other nodes. The network will send unknown packets or commands to the controller to avoid excess processing logic, which reduces the complexity of network.

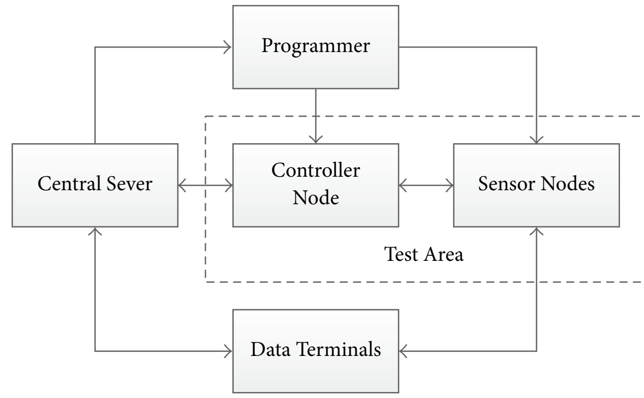

Figure 1: Testbed framework.

Multiple Applications. A centralized network control provides an application-independent method of management, so that ordinary nodes do not need to care about specific processes. In traditional network devices, the function of a hardware component or software module is designed for known protocols or application scenarios. However, applications are transparent to a node, which means that multiple applications or protocols could be mixed without mutual interference.

Another challenge is that embedded operating systems for WSNs do not support dynamic reconfiguration effectively. Although there are some operating systems that can load applications dynamically, such as Contiki [14], these always lack protocol or algorithm reconfiguration. The task scheduling for WSNs is usually linear and single threaded, which does not distinguish traffic with different demands. This high degree of coupling makes it hard to change policies in real time [15].

The challenges described above are inherent to WSNs and deeply rooted in the architecture. A reconfigurable testbed should support runtime reconfigurations to the application and data routing. Although existing architectures support some kinds of well-designed applications and algorithms, they still lack good abstraction and make networks rigid and inelastic.

3. Architecture and Features of the Testbed

This section provides an overview of the implementation of the proposed testbed on real hardware. Various components of the testbed are described, which include the control workstation, the OpenFlow controller, the node programmer, the sensor nodes, and a group of computers that will collect the run data by wired connections. A schematic diagram of the testbed is presented in Figure 1.

3.1. Testbed Framework and Hardware. The network controller shown in Figure 1 is different from the sink in the original WSNs. It is part of the test area that manages the

Table 1: A comparison of WSN testbeds.

Testbed	Nodes and scales	Primary features	Multiprotocol support
MoteLab [7]	190 TmoteSky motes	Aiming at programming environment, communication protocols, system design, and applications	Single protocol
SignetLab [8]	48 EyesIFXv2 motes	Software to control motes and dynamically creating architectures	Single protocol with variable network architectures
Sensei-UU [10]	TelosB motes and mobile robots	Smart phone support and repeatable experiments are at different locations	Single protocol
NetEye [11]	130 TelosB motes	Web-interface to create and schedule experiments	Single protocol
EasiTest [9]	EZ271 and EZ521 motes	Heterogeneous multiradios to support various speed rates	Single protocol
IOT-Lab [12]	Over 2700 motes and mobile robots	Heterogeneous platform, mobility support, and different APIs	Single protocol with programmable routing layer

whole network, not only the communication interface. In this design, every sensor node can be configured as a sink, which allows the network to adapt to some routing protocols with one or more data centers, such as the directed diffusion (DD) protocol [16] and the low energy adaptive clustering hierarchy (LEACH) protocol [17]. One problem with the testbed concerns the data collection of sensor nodes. Most WSN testbeds transmit the testbed metadata (the testbed configuration commands, internal data of nodes, statistical information, etc.) by the wireless channel, which takes significant channel resources and memory space away from nodes and results in a higher latency or loss rate when a test is run. Therefore, we set a group of data terminals to collect sensor data instantaneously by a UART (universal asynchronous receiver/transmitter) or USB (universal serial bus) port. For a sensor node, UART/USB transmission requires fewer computing resources than radio frequency (RF).

On the hardware side, the testbed consists of a CC2530 and CC2531 system on chip (SoC), developed by Texas Instruments. CC2530 is based on an enhanced 8051 core using the standard 8051 instruction set, which can operate at 32 KHz, 16 MHz, or 32 MHz to support different power modes (fully functional or ultralow-power modes). It has an 8 KB static random access memory (SRAM) and a programmable Flash (32/64/128/256 KB) to support an embedded operating system. CC2530 supports a variety of serial interfaces, including the serial peripheral interface (SPI) and UART (CC2531 supports USB control) and features an IEEE 802.15.4/ZigBee RF transceiver in the 2.4 GHz band. In the context of our testbed, the controller is built on CC2531 and the sensor nodes are built on CC2530 or CC2531. The nodes of both types can all be powered by USB interface. In addition, for the convenience of the testbed deployment, in this work CC2530 will be equipped with three AA batteries when required.

3.2. Testbed Operating System. All of the nodes run TinyOS [18], a specific embedded operating system for WSNs. TinyOS

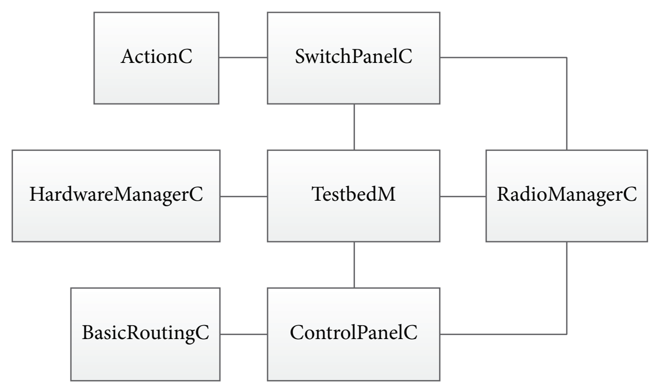

Figure 2: Operating system component relations graph for the testbed.

is a component-based operating system, which provides a fast method of updating the system and reducing the coupling between modules. In this work, there are several components to separate the control and data planes and accomplish different functions such as data transmission, routing configuration, and user application. By overwriting the hardware interface, TinyOS is transplant to CC2530. Figure 2 presents a component graph of the system. The system utilizes two important components, “SwitchPanelC” and “ControlPanelC,” to implement different network policies. The component “SwitchPanelC” provides a set of operations, which are stored in the component “ActionC,” to modify the flow table. The component “ControlPanelC” is a part of the control plane of the network (the system structure of the controller is similar to that of the sensor node, but its functionality is more complicated), which handles and stores information used in the network control. The component “BasicRoutingC” is maintained by “ControlPanelC,” and most routing policies are stored in this component. The component “RadioManagerC” provides a common interface

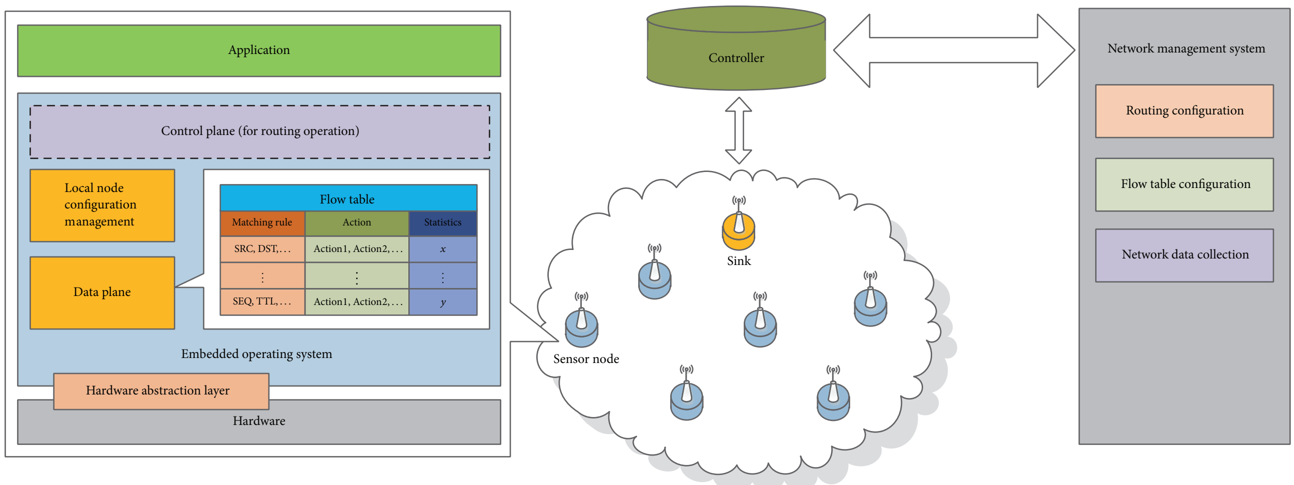

Figure 3: Platform framework based on OpenFlow.

for accessing the data transmission and reception directly. With this independent access control, routing information can be separated from general data, which makes dynamic policy changing possible.

4. Reconfiguration Kernel

On the software and protocol levels, software-defined networking (SDN) is developed to enhance the programmability and flexibility of wired networks [19]. As a realization of SDN, the starting point of OpenFlow is to allow researchers to develop and verify new network architectures and protocols conveniently on campus networks. OpenFlow provides a simple implementation for different applications, protocols, and network management systems by separating the control and data planes. Meanwhile, OpenFlow allows administrators to use the standardized interface to manage the network and change policies. For WSNs, SDN provides a method of reconfiguring the network dynamically, without reprogramming or redeploying [20].

In traditional networks, dynamic protocol reconfiguration is implemented by software and flexible protocol stacks. The processing of many network parameters is required, which conflicts to the demands of WSNs. Recent work has demonstrated the potential of SDN in sensor networks and has discussed basic considerations regarding implementation, including architecture and deployment [21, 22].

In comparison with general network platforms based on OpenFlow, a platform in WSNs exhibits some different characteristics, such as the networking mode, topology, transmission rate, and fault tolerance [23]. Therefore, a node in WSNs must be able to support several network layer functions. Therefore, the concept of flows is incorporated into WSN to simplify the architecture of the testbed. This should be constructed from several parts, as illustrated in Figure 3. In this paper, the control and data planes are separated, not only for unknown packets and the flow table configuration, but also for routing policy management. This separation allows the extraction of routing information from data transmission and the reduction of the coupling of the two planes.

4.1. Basic Routing Control Plane. For most routing protocols, some defined routing policies require the cooperation of different nodes or control by a specific node (i.e., a sink or cluster head). However, this could result in high-coupling relations between routing information exchange and data forwarding. To separate routing operations and data processing, the control plane manages routing operations to configure the current network state, such as MAC learning, clustering, and signal-aware topology discovering. These operations are independent of forwarding policies in the data plane, even though they are always different for each node. In particular, the control plane marked by the dashed line in Figure 3 separates routing messages from data flows, which means that the control plane can handle not only unknown packets, but also routing policies.

4.2. Programmable Flow Table. In the data plane, all packets are handled as flows. The forwarding policy of flows is controlled by a *flow table*, which is defined by an OpenFlow controller. Each item or flow entry in the table contains three basic fields: a set of rules related to the flow, one or more actions that can be executed when a flow is matched to an entry, and some statistical information regarding the entry. The content of an entry depends on the running protocol in the network and can be updated to adopt changes to the protocol if necessary. In addition, because of the special character of WSNs, data generated by a node itself must also be reasonably processed, even it is a routing node. Therefore, nodes and flow tables in a WSN must have the ability to handle locally generated data. This can be solved by adding an additional flow entry to the flow table.

4.3. Actions and Miss Rule. Each flow entry has an *action field*, which contains one or more operations that could be

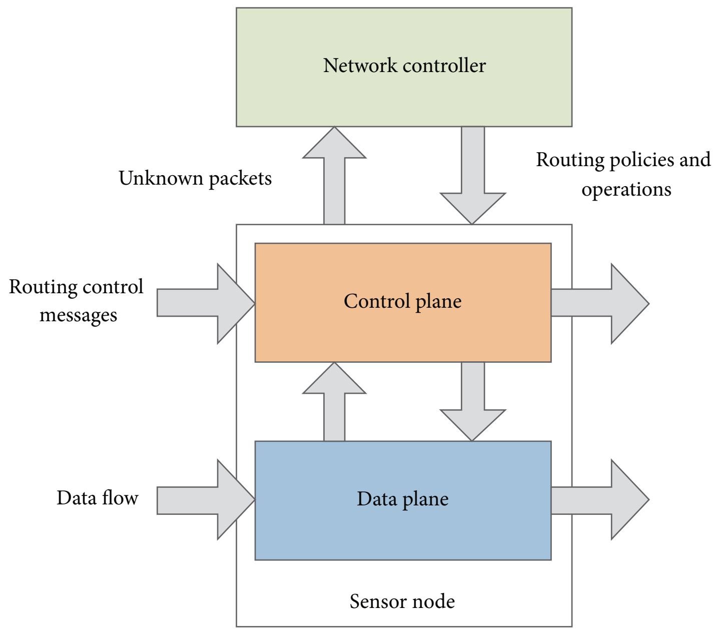

FIGURE 4: Relation between the control and data planes.

applied to the packet belonging to the flow. We argue that there are two kinds of action: *required actions* and *optional actions*. The flow tables must implement required actions, such as dropping, broadcasting, forwarding, and uploading the packet to the local network stack. Optional actions can enhance the functionality of the table, including modifying specific packet fields, sending requests to the controller, or performing other optimizing operations. The optional actions could be different for one node than for others for a specific network role or requirement (i.e., data collection or routing).

In traditional OpenFlow, if a packet cannot be matched to any flow entry, then it will be dropped by a *miss rule* or uploaded to the controller. In comparison with the flow entry in OpenFlow, the miss rule in our work contains one or more operations regarding coping with packets: such as dropping it, modifying (including creating) the flow entry, or uploading it to the controller. Like other flow entries, a miss rule can be updated independently for different protocol policies.

5. Protocol Implementation with Separation of Control and Data Planes

Figure 4 illustrates the relation between the control and data planes in one sensor node. As previously mentioned, a protocol will be configured in these two planes, so the design can be classified into two parts (the routing operation and flow table parts). In order to switch protocols dynamically, the flow table and basic routing control must be configured independently. Unlike traditional wireless network protocols, some protocols in WSNs have to accomplish other tasks before starting the routing algorithm, such as clustering, time synchronization, and energy-awareness. Therefore, we can specify a routing operation sequence in the control plane to implement these required tasks. These operations can be assembled together randomly (unless the control plane does not need any operations, such as flooding) to support different routing protocols. Table 2 presents some examples of different operation sequences.

TABLE 2: Routing operation sequences.

Protocol	Routing operation sequence
Flooding	No routing operation
LEACH	(1) Time synchronization (2) Cluster head election (3) Cluster head broadcasting
Directed diffusion	Sink information broadcasting

5.1. Data Flows in the Network. Figure 5 illustrates the manner in which a packet is processed by the network. For the received packet, the treatment process can be considered as a function of action, which uses the packet itself and the matched flow entry as arguments. Thus, the following process formula can be applied when a packet is to be forwarded or received by the function:

$$\mathrm{Act}_X\left(\mathrm{Pkt}_m, \mathrm{FE}_m\right) \longrightarrow \mathrm{Node}_Y, \tag{1}$$

where the packet, the matched flow entry, and actions are represented by Pkt_m, FE_n, and the function Act_X. The result generated by Act_X will be attached to the next hop or local

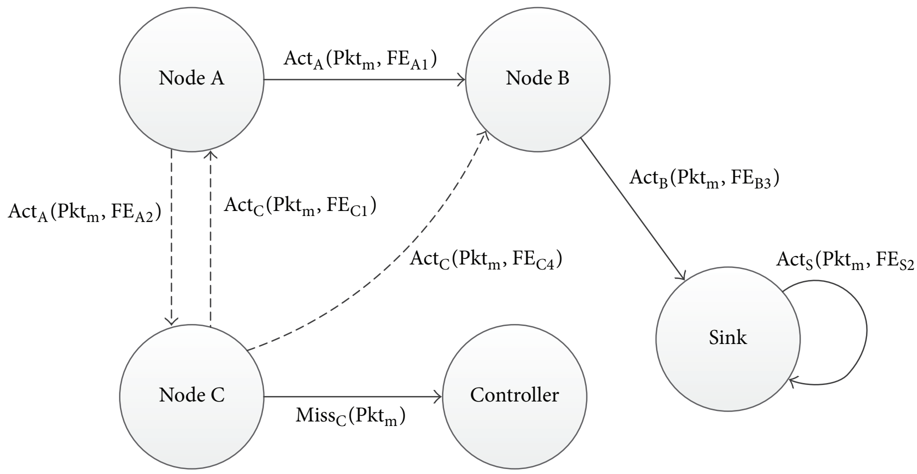

Figure 5: Topology of the network and flow table processing a packet. Broken lines present optional methods of processing the packet.

node. Therefore, the final result can be obtained by a recursive application of formula (1):

$$\text{result} = \mathrm{Act}_{X+1}\left(\mathrm{Act}_X\left(\mathrm{Pkt}_m, \mathrm{FE}_n\right) \longrightarrow \mathrm{Node}_{X+1}, \mathrm{FE}_{(X+1)_n}\right) \longrightarrow \mathrm{Node}_Y. \tag{2}$$

Furthermore, a packet that does not belong to any flow entry will be processed by a miss rule as follows:

$$\mathrm{Miss}_X\left(\mathrm{Pkt}_m\right) \longrightarrow \mathrm{Node}_{\mathrm{Controller}}, \tag{3}$$

where $\mathrm{Node}_{\mathrm{Controller}}$ can also be the local node itself if the miss rule involves dropping the packet or other operations.

5.2. Procedure for Processing Packets. Forwarding policies are implemented as a set of flow entries in the flow table. Flow entries decide how packets will be treated, as well as other operations regarding the node itself. Flow entries have to be configured before a protocol is deployed. Rules in a flow entry are some conditions that data flow fields have to match [24]. Each field is given by two thresholds (the top and bottom thresholds) to maintain the matching strategy. The flow is considered as a match for a specific flow entry if the value of every field in the flow is located between these two thresholds. Table 3 presents some examples of different operation sequences. We extract the flow mask of every packet and use a resulting code to judge the match results of actions by Algorithm 1.

When the network is running, these operations will be executed at the appropriate time (i.e., the clustering operations should be executed before each running period). To avoid a situation where a flow is matched to several entries, we argue that every entry has a priority in the table. In addition, a field called the *flow mask* could decide which field should be added to the flow, thus reducing the length of a flow entry and rendering the checking of every field unnecessary. Moreover, protocols can be distinguished by their masks. When more than one protocol is run in the network, different flows could have the same mask, because

Table 3: Example of rules and actions of flow table.

Rule	Actions	Stats
$x \leq$ SEQ $\leq$ MAX	(1) Update the flow table	s_1
DST = Local Address	(2) Receive	
DST $\neq$ Local Address	(1) Update the packet field (2) Forward	s_2
$\vdots$	$\vdots$	$\vdots$
TTL = 0	(1) Drop	s_n

of the requirements of protocols. Therefore, there will be another field to separate from these if necessary. The function GetFlowMask is used to achieve the masking of a flow and the function GetMatchField to locate the first match field. Each effective part of the mask represents a specific match field that will be checked by the function CheckMatchField. If all match fields conform to a flow entry, then the corresponding actions will be executed, and other entries will not be checked anymore. A miss rule will be applied when no entry matches the current flow.

6. Experiment and Evaluation

The principal feasibility of the testbed is evaluated on several features, such as the transmission efficiency, flow table processing capability, and energy consumption. By filling flow tables with different parameters, different routing protocols are tested to evaluate the effectiveness and correctness of the testbed.

6.1. Experimental Setup. The application scenario in our experiment consists of a group of sensor nodes and computers. Figure 6 illustrates a small section of the testbed that contains 16 sensors in a lab. In this scenario, sensor data and system information are collected by the terminal computer connected to a backbone Ethernet, which makes it convenient for the testbed to upload data and receive commands. A

```
Input: Payload; FlowTable[]; TableLength mask=GetFlowMask(Payload);
offset=GetMatchField(Payload);
for each i TableLength do for each bit in the mask do
        if bit = 1 then
            state=CheckMatchField(offset, FlowTable[i]); if state is true then
                update offset to next match field; end
            else state=false;
                break; end
        end end
    if state is true then
        ExecuteAction(Payload, FlowTable[i]);
        break; end
end
if no action executed then
    ExecuteMiss(Payload); end
```

Algorithm 1: Packets process in switch plane.

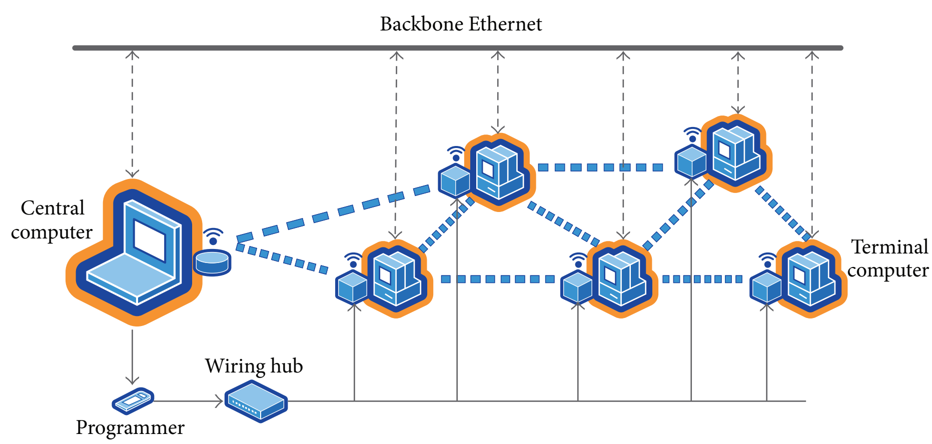

Figure 6: Testbed scenario and setup.

wiring hub is attached to the controller to program sensor nodes as a whole. When the testbed is loaded, all nodes can be directly taken over by terminal computers.

6.2. Statistics of the Flow Table. In order to evaluate the validity in a common scenario, different protocols are transferred and applied to this platform and two indices are defined: the network transmission efficiency (NTE) and the average matching number (AMN). When a protocol is running, it can generate some redundant packets, such as beacons; thus NTE describes the proportion of the real data that is actually required by the network, reflecting the efficiency of the communication (including the two planes). AMN describes the average retrieval times of packets being processed by the flow table, which can be used to evaluate the processing speed and capability of a protocol. Figure 7 presents the results of our experiment for different protocols (flooding, LEACH, and directed diffusion). The expected manifestation of every protocol is given, and LEACH indicates the best performance (a high NTE and low AMN value). This can be reduced to 61.90% when the network recreates the route for each communication, although DD with the fixed path achieves the highest NTE value (92.03%). This test proves that the testbed based on OpenFlow can support one or more protocols, and a horizontal comparison is made.

6.3. Energy Consumption. The energy consumption of a node, or even the whole network, can be effected by the protocol, topology, physical environment, and so on. From Figure 8, the influence of each protocol and the flow table content on the network energy consumption are presented. In the context of the same topology and communication frequency, because of the low-power consumption of WSNs, trends in energy among these protocols are implicit. However, the average energy consumption obviously varies. Meanwhile, because the configuration of a flow table determines how a flow is processed, some energy-intensive actions (such as forwarding or uploading a packet to the controller) can determine the energy consumption, which means there is a novel method for reducing the energy consumption by controlling and configuring the actions of a flow table.

6.4. Throughput of Control and Data Planes. In general, a control plane separates the network management flow from

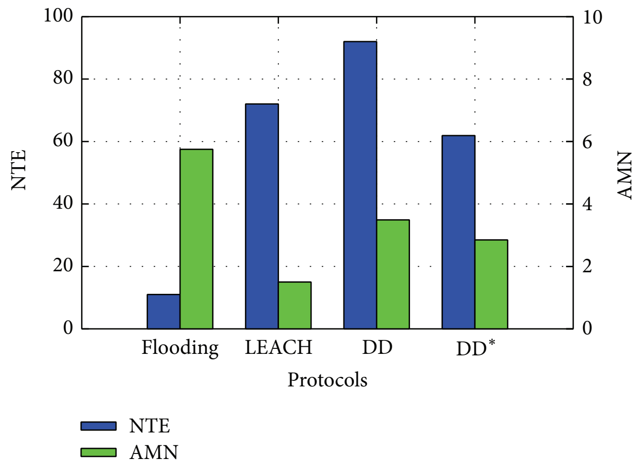

Figure 7: NTE and AMN of protocols. We suppose that the topology of DD is fixed, and once all transmission paths are created they will not be modified further. On the contrary, the topology of DD* is dynamic, so it will recreate some paths for reliable communication.

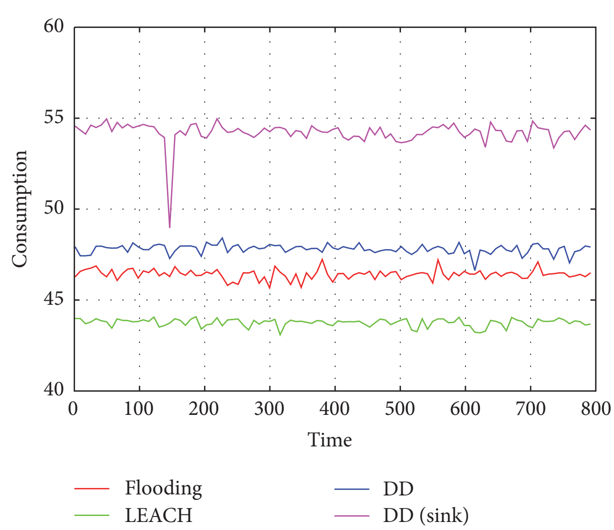

Figure 8: Energy consumption with different protocols.

the data transmission. Furthermore, in this testbed it can cope with routing operations such as the flow table configuration, clustering, and route requests. To illustrate the effect of the separation of the control and data planes in this testbed, the throughput of two planes with different protocols is evaluated. Figure 9 presents the statistics for each protocol in 500 s. Under the circumstance that the same data generation rate is maintained, the flooding protocol achieved a higher throughput than the other protocols on the data plane. This is because excessive redundant packets are present during the transmission. For the DD and DD* protocols with routing maintenance, the number of redundant packets is much lower than for flooding. On the control plane, the LEACH protocol achieves the highest throughput for the clustering operation but expends vast amounts of network resources. In fact, layered routing protocols (i.e., LEACH) consume more

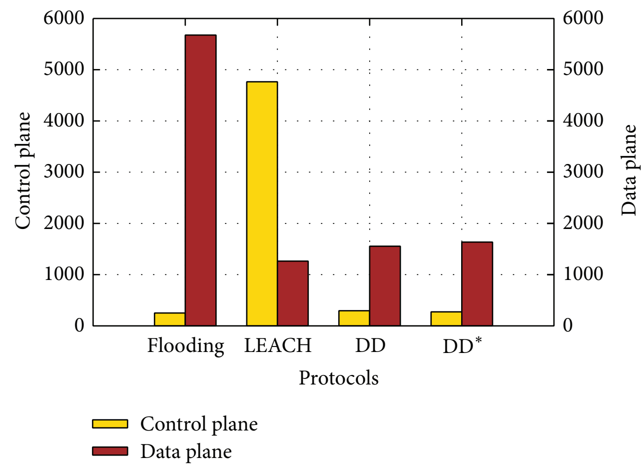

Figure 9: Throughput of the two planes with different protocols.

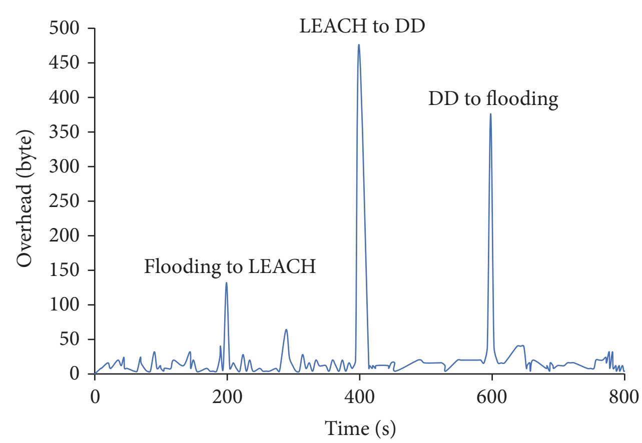

Figure 10: Network traffic during protocol switching.

controlling flow on the transmission than planar routing protocols. In this experiment, the testbed clearly separated the two planes and reflected the respective characteristics of different protocols with OpenFlow.

6.5. Dynamic Reconfiguration with Different Protocols. Although most nodes in the network share common structures and functions, they can exhibit extremely different transmission policies when given different configurations in the flow table. All data packets sent (or received) by a node should be processed by the flow table, and this can affect the performance of the transmission instantaneously. A protocol switch can be completed by updating one or more flow tables with new contents. The overhead of deploying a protocol to one flow table is presented in Table 4 (the extra network traffic is not contained in the table). In Figure 10, the network traffic is tested when the running protocol is switched. The network load was recorded by the testbed during protocol configuration. The switch to the DD protocol consumes noticeable network traffic to update the flow tables. In addition, as the network grows in size, an inappropriate

Table 4: Net overhead of filling the flow table with a specific protocol.

Protocol	Overhead (byte)
Flooding	56
LEACH	23
DD (sensor node)	89
DD (sink)	31

protocol selection could lead to a network load increase. Therefore, the testbed based on OpenFlow can not only switch routing protocols dynamically without redeploying the whole network, but also choose an optimal protocol to reduce the network load.

7. Conclusions and Future Work

In this paper, a dynamically reconfigurable WSN testbed is established. To implement a software-based reconfiguration method in real time, a novel architecture using OpenFlow constructs the data plane in a protocol-independent manner. The experiments demonstrated that the testbed can reduce the complexity of network management and updating by separating the control and data planes. A user can manage the network easily by simply configuring different behaviors, without redeploying the whole network. The dynamic switching between protocols was implemented by a programmable flow table, which makes the network reconstruction faster and more convenient. Furthermore, it is obvious that large scale WSN will be applied to various scenarios which have different routing demands. So this ability of reconfiguration can support WSN virtualization when the scale is growing up and can be used in a multitenant architecture for cloud service.

However, there still remain issues that should be investigated. In OpenFlow, a controller should support extra functions for the network to process unknown flows or update flow tables. However, in some networks with fast topology changing, a controller cannot handle all messages instantly and exactly. Sometimes this could result in performance degradation, because of the uncertainty in deployment and network topology. Furthermore, it is difficult to deploy all protocols by one universal flow table configuration method. For example, the networking processes for a layered protocol and planar protocol differ significantly, which could result in a significant overhead for dynamic protocol switching.

There can be three sections to resolve these problems in future work. Firstly, a controller group can be applied to the network rather than only one controller, which can offload specified traffic or protocol to different controllers and enhance the functionality of control plane to optimize routing operations. Secondly, common features of transmission policies among different protocols will be extracted by analysis and experiment, so that flow table (including flow actions) can be simplified for packet process. Finally, there are some search algorithms and they will decrease the cost of flow table matching.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Plan Key Project of the Fujian Province of China under Grant no. 2014H0030, the Science and Technology Plan Key Project of Quanzhou City, Fujian Province of China, under Grant no. 2014Z102, the Natural Science Foundation of Fujian Province of China (no. 2017J01776), the Special Scientific Research Project of Provincial Colleges and Universities of Fujian Province of China under Grant no. JK2015037, and the Young Dr. Project of Quanzhou Normal University under Grant no. 2015QBKJ02.

References

[1] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.

[2] H. Grichi, O. Mosbahi, M. Khalgui, and Z. Li, "RWiN: New methodology for the development of reconfigurable WSN," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 109–125, 2017.

[3] P. Latha and M. A. Bhagyaveni, "Reconfigurable FPGA based architecture for surveillance systems in WSN," in *Proceedings of the 2010 International Conference on Wireless Communication and Sensor Computing, ICWCSC 2010*, January 2010.

[4] P. Bosshart, G. Gibb, H.-S. Kim et al., "Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN," in *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, ACM SIGCOMM 2013*, pp. 99–110, August 2013.

[5] S. Bagchi, "Nano-kernel: A Dynamically Reconfigurable Kernel for WSN," in *Proceedings of the 1st International ICST Conference on Mobile Wireless Middleware, Operating Systems and Applications*, Innsbruck, Austria, Feburary 2008.

[6] L. P. Steyn and G. P. Hancke, "A survey of wireless sensor network testbeds," in *Proceedings of the IEEE Africon'11*, zmb, September 2011.

[7] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: a wireless sensor network testbed," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 483–488, April 2005.

[8] R. Crepaldi, S. Friso, A. Harris et al., "The design, deployment, and analysis of signetlab: A sensor network testbed and interactive management tool," in *Proceedings of the 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, TridentCom 2007*, May 2007.

[9] Z. Zhao, G.-H. Yang, Q. Liu, V. O. K. Li, and L. Cui, "EasiTest: a multi-radio testbed for heterogeneous wireless sensor networks," in *Proceedings of the IET International Conference on Wireless Sensor Network (IET-WSN '10)*, pp. 104–108, November 2010.

[10] O. Rensfelt, F. Hermans, P. Gunningberg, L.-Å. Larzon, and E. Björnemo, "Repeatable experiments with mobile nodes in a relocatable WSN testbed," *The Computer Journal*, vol. 54, no. 12, pp. 1973–1986, 2011.

[11] X. Ju, H. Zhang, and D. Sakamuri, "NetEye: a user-centered wireless sensor network testbed for high-fidelity, robust experimentation," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1213–1229, 2012.

[12] C. Adjih, E. Baccelli, E. Fleury et al., "FIT IoT-LAB: A large scale open experimental IoT testbed," in *Proceedings of the 2nd IEEE World Forum on Internet of Things, WF-IoT 2015*, pp. 459–464, ita, December 2015.

[13] L. De Carli, Y. Pan, A. Kumar, C. Estan, and K. Sankaralingam, "PLUG," in *Proceedings of the the ACM SIGCOMM 2009 conference*, p. 207, Barcelona, Spain, August 2009.

[14] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki—a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the 29th IEEE Annual International Conference on Local Computer Networks (LCN '04)*, pp. 455–462, November 2004.

[15] M. R. Senouci, A. Mellouk, L. Oukhellou, and A. Aissani, "WSNs deployment framework based on the theory of belief functions," *Computer Networks*, vol. 88, pp. 12–26, 2015.

[16] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

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

[18] Tinyos home page, 2013, http://www.tinyos.net/.

[19] O. N. Foundation, *Software-Defined Networking: The New Norm for Networks*, White Paper, 2012.

[20] T. Luo, H.-P. Tan, and T. Q. S. Quek, "Sensor openflow: enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.

[21] A. Mahmud, R. Rahmani, and T. Kanter, "Deployment of Flow-Sensors in Internet of Things' Virtualization via OpenFlow," in *Proceedings of the 2012 3rd FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing, MUSIC 2012*, pp. 195–200, June 2012.

[22] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and OpenFlow: from concept to implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.

[23] J. Horneber and A. Hergenroder, "A survey on testbeds and experimentation environments for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1820–1838, 2014.

[24] A. De Gante, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in *Proceedings of the 27th Biennial Symposium on Communications (QBSC '14)*, pp. 71–75, Kingston, Canada, June 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 1864847, 14 pages
https://doi.org/10.1155/2017/1864847

Review Article

A Survey on Real-Time Communications in Wireless Sensor Networks

Beom-Su Kim,[1] HoSung Park,[1] Kyong Hoon Kim,[1] Daniel Godfrey,[2] and Ki-Il Kim[2]

[1]*Department of Informatics, Gyeongsang National University, Jinju, Republic of Korea*
[2]*Department of Computer Science and Engineering, Chungnam National University, Daejeon, Republic of Korea*

Correspondence should be addressed to Ki-Il Kim; kikim@cnu.ac.kr

Received 3 June 2017; Revised 15 September 2017; Accepted 3 October 2017; Published 31 October 2017

Academic Editor: Paul Honeine

Generally, various traffic requirements in wireless sensor network are mostly dependent on specific application types, that is, event-driven, continuous, and query-driven types. In these applications, real-time delivery is one of the important research challenges. However, due to harsh networking environment around a node, many researchers usually take different approach from conventional networks. In order to discuss and analyze the advantage or disadvantage of these approaches, some comprehensive survey literatures were published; however they are either out of date or compiled for communication protocols on single layer. Based on this deficiency, in this paper, we present the up-to-date research approaches and discuss the important features related to real-time communications in wireless sensor networks. As for grouping, we categorize the approaches into hard, soft, and firm real-time model. Furthermore, in all these categories, research has been focused on MAC and scheduling and routing according to research area or objective in second level. Finally, the article also suggests potential directions for future research in the field.

1. Introduction

As the wireless sensor network (WSN) becomes pervasive, lots of recent research works tend to focus on application specific properties. Among them, real-time communication remains one of research challenges depending on application types such as event-driven, continuous, and query-driven types. In these applications, data packets beyond deadline are regarded as affecting the system performance and quality. However, real-time communication is challenging problem in wireless networks which are subject to fading, interference, being unreliable, and rapid varying channel quality [1]. Particularly, if real-time and non-real-time applications coexist, deadline of the real-time traffic cannot be met often due to shared wireless medium with non-real-time traffic as the amount of traffic increases. In addition to property of wireless link, severe constraints on node in WSN make it hard to support real-time communications. To deal with impact of traffic over wireless link, usually higher priority is given to real-time traffic than non-real-time one to minimize the contention on shared medium [2] or contention-free scheme is usually employed in WSN [3]. Moreover, practical real-time communication issues in WSN are well presented and introduced in [4]. In addition, serious unreliability problem of the contention-based medium access control (MAC) protocol and its default parameter values was mentioned in [5]. The authors addressed the impact of power management system and low density for reliability with extensive analysis based on both simulation and experiments.

Based on the above motivation, most of research takes different approaches to focus on application specific property. Moreover, some literatures and one special issue were published and organized, recently. However, they are either out of date or mainly focus on communication protocols in single layer. Thus, they are not sufficient to provide insight into real-time communications in WSN and provide recent research trends. For example, Alanazi and Elleithy [6] presented state-of-the-art research work based on real-time QoS routing protocols for wireless multimedia sensor networks. Moreover, three comprehensive survey literatures for real-time routing [7–9] were published to provide the comparison in the aspects of energy efficiency and reliability. Teng and Kim [10] addressed the specific requirement of real-time MAC protocol in WSN by surveying recent wireless sensor real-time MAC protocols. However, mentioned research work focuses on network or MAC protocol, respectively, and does not provide any classification principle. Another survey paper authored by Li et al. [11] addresses the real-time communication protocols and data processing scheme in cross-layer designs. But, this paper was published in 2007 so it is too out of date. In addition, special issue titled "New Challenges of Real-Time Wireless Sensor Networks: Theory and Applications" [12] was recently organized in 2016 but survey literature for real-time communications in WSN was not included. Moreover, both hard and soft real-time models were explained and analyzed in their work.

According to above motivation, in this paper, we present the up-to-date research approaches and discuss strength as well as weakness of them by collecting research literatures since 2010 with the exception of a few fundamental papers. We categorize the existing scheme according to general classifications such as hard, soft, and firm real-time model. Particularly, unlike the previous survey which does not include firm approach, we collect research for the (m, k)-firm stream model in WSN and analyze their applicability to WSN. Prior to description of the details of each scheme, we introduce the example of real-time application and available platform for real-time communications in WSN. Finally, we conclude this article with a discussion of potential open issue.

2. Real-Time Applications and Platform

In this section, we present the representative applications which require real-time delivery. In addition, several sensor node platforms to support real-time communications are briefly described.

2.1. Real-Time Applications. We choose four examples for potential and practical real-time applications. In this section, we describe how real-time communication in each application is provided.

2.1.1. Health Monitoring. One of the emerging and promising applications to demand real-time communications in WSN is health monitoring system which consists of monitoring and alarming system for patient health. First, a patient's monitoring system embedded with a set of medical sensors and wireless communication module was proposed by Al-Aubidy et al. [13]. In this system, the patient health status is reported to medical center and checked by the doctor who is in charge of sending medical advice. The experimental results show the effectiveness of the implemented prototype in the aspects of accuracy, intelligence for making decision, and reliability. In addition, Li-Wan et al. [14] proposed a new type of wireless network monitoring systems to collect patients' physiological indicators by adapting the multichannel high-frequency wireless data transmission. The last example is about a portable real-time wireless health monitoring system which was proposed by Choudhary et al. [15]. This system consists of ZigBee wireless standard and demonstration of pulse oximetry data monitoring on the patients.

2.1.2. Target Tracking. VigilNet [16] was proposed to track, detect, and classify targets in a timely as well as energy efficient manner by introducing both a deadline partition method and theoretical derivations to guarantee each sub-deadline. Since end-to-end deadline is affected by many system parameters, system-wide global optimal solution was proposed in this work. In the proposed scheme, end-to-end deadline is divided into multiple subdeadlines. To meet real-time requirements, activation, sentry detection, wake-up, aggregation, communications, and base processing delay and their tradeoff were analyzed and brought insights. For example, in the case of slow moving target tracking, the deadline is guaranteed by considering several factors such as a higher node density, increased wake-up delay, and fast detection algorithm. The VigilNet was implemented and tested for various cases to prove suitability for real-time communications.

2.1.3. Environmental Tracking. Pozzebon et al. [17] proposed the new architecture of a heterogeneous WSN to monitor coastal sand dunes where three different typologies of integrated sensors were employed. The proposed architecture consists of Sand Level Sensor Network, environmental monitoring node, and Gateway. The ZigBee radio module transmits the collected data while anemometric station is in charge of data processing. In addition, a Gateway node provides external connection with GSM connection. Tse and Xiao [18] proposed WSN system which is able to sense multiple environmental factors and aggregate collected data in real-time. The data transmission is accomplished by the Wi-Fi module using the UDP protocol. Moreover, environmental monitoring system for air quality was presented in two papers [19, 20], respectively.

2.1.4. Control System. Georgoulas and Blow [21] proposed In-Motes EYE application based on In-Motes platform to obtain acceleration variations in an environments for automobile. Four different categories of In-Motes agent are the actor of application under layered architecture. Communications are performed by federation communication scheme. Based on this In-Motes platform, the application is to allow a user to monitor the acceleration pattern of a moving car by injecting a new job agent to the vehicles sensor and checking whether a car breaches the critical parameter of the application in test scenarios. By this experiment, In-Motes applications show possibilities to monitor real-time operation.

2.2. Platform for Real-Time Communications

2.2.1. FireFly. FireFly [22] is one of the well-known WSN platforms for monitoring, surveillance, and voice communication with battery-operated node through multihop mesh communications. Particularly, each node operates over

IEEE 802.15.4 protocols over Nano-Resource Kernel real-time operating system. Global time synchronization works in energy efficient way by maximizing both common sleep time and throughput within bound end-to-end delay. In the aspects of real-time, each sensor node transmits and receives data packet within predetermined time on dedicated time slots. Thus, FireFly can be used for real-time applications such as delay sensitive voice communication through Real-Time Link (RT-Link) protocol, one of the TDMA MAC protocols, running over a network of FireFly nodes. Moreover, a new extended platform of FireFly Mosaic [23] was proposed to run on vision-enabled sensor networks with application to monitor people's daily activities at home. In this application, frequent particular activities were observed by multiple overlapping cameras to extract wanted information.

2.2.2. PAVENET OS. PAVENET OS [24] is a compact hard real-time operating system for WSN. To be optimized for real-time and best-effort tasks, preemptive multithreading and cooperative multithreading are employed. Both higher compactness and lower overheads than typical TinyOS are supported by hybrid multithreading. To realize the hard real-time feature, PAVENET OS is designed in accordance with a thread model and enabling preemption. PAVENET OS provides a wireless communication stack for hiding the exclusive controls to users. The protocol stack realizes modularity at each communication layers so the user can develop various communication protocols according to application demands easily. PAVENET OS employs a buffer management mechanism called *pbuf* to exchange data among layers. The experimental results show that PAVENET OS achieves 0.01% jitter while performing wireless communication tasks with low overhead in the aspects of size of RAM and ROM and minimum task switching time.

2.3. A-LNT. A-LNT [25] is a lightweight low-speed and low-power WSN platform for voice communications. In order to meet requirement of voice communications and sensing data transmission, clock synchronous MAC protocol and data noninterference mechanisms are employed under star network topology. Moreover, A-LNT supports three types of voice communications in most conditions in order to reduce wireless transmission pressure. The audio channel capacity and delay are enough to support emergency voice communication, audio/sound sensor network, and health monitoring system.

2.3.1. A-Stack. A-Stack [26] is real-time protocol stack for time-synchronized, multichannel, and slotted communication in multihop wireless networks to meet latency requirement for industry automation and structural health monitoring. To bound delay within deadline, there are several factors such as MAC, routing, and clock synchronization protocols to be considered. A-Stack operates with multichannel TDMA, global time synchronization, and source routing. To be more detailed, A-Stack runs on the FreeRTOS real-time kernel; however, it can easily be transferred to any preemptive multitasking OS. Through experimental results, it is proven that A-Stack is suitable for low latency and high reliability real-time WSN applications and protocols.

3. Real-Time Communications in WSN

Despite its importance and necessity, real-time communications in WSN have faced many research challenges and misunderstanding. For this issue, Oliver and Fohler [4] have analyzed the suitability and applicability of the real-time communications in WSN with impacts of the number of assumptions and different evaluation criteria to infer a number of basic considerations. Even though it is difficult to provide real-time communications in WSN, a lot of researches have been conducted. Therefore, it is essential to identify the existing schemes with proper good insight into real-time communications in WSN.

To achieve above goal, real-time communications in WSN are classified into the categories hard, firm, and soft real-time without loss of generality like conversational classification. Hard real-time communications indicate missing deadline affects the operation on the system by causing failure of the whole system. So, it takes the worst case times to bound end-to-end delay within the deadline. Soft real-time communications attempt to reduce deadline miss so probabilistic guarantee can meet requirement where some misses are tolerable. Firm real-time communications have similar features from both hard and soft real-time model in that it allows for infrequently missed deadlines while the system can survive task failures so long as they are adequately spaced.

In this paper, we categorize the existing schemes for real-time communications into the above three classes. Particularly, as for firm real-time model, we explain well-known (m, k)-firm model which guarantees requirement that at least m out of any k consecutive messages from the stream must meet their deadlines, to ensure adequate quality of service. Figure 1 illustrates the classification of the existing real-time approaches. As for the second level classification, we focus on the research objective in homogenous way. As a result, we classify the existing schemes in two parts. One is MAC and scheduling and the other is routing.

3.1. Hard Real-Time Communications in WSN. When it comes to taking into account various constraints in WSN, hard real-time communications are very difficult to implement. Particularly, since multihop communications cannot bound deterministic end-to-end delay, star topology in single hop is preferred in most research. Based on this analysis, there are a few hard real-time schemes which are categorized into MAC and scheduling in TDMA as well as routing protocol. Moreover, hard real-time communications in WSN are usually employed in very critical applications such as automobile or industrial application. In this model, as for platform through integrated project, REWIN [28], real-time guarantees in wireless sensor networks, was launched to offer hard real-time guarantees to individual real-time flows over multihop WSN of arbitrary node deployments and arbitrary traffic pattern. These newly studied methods were proven to

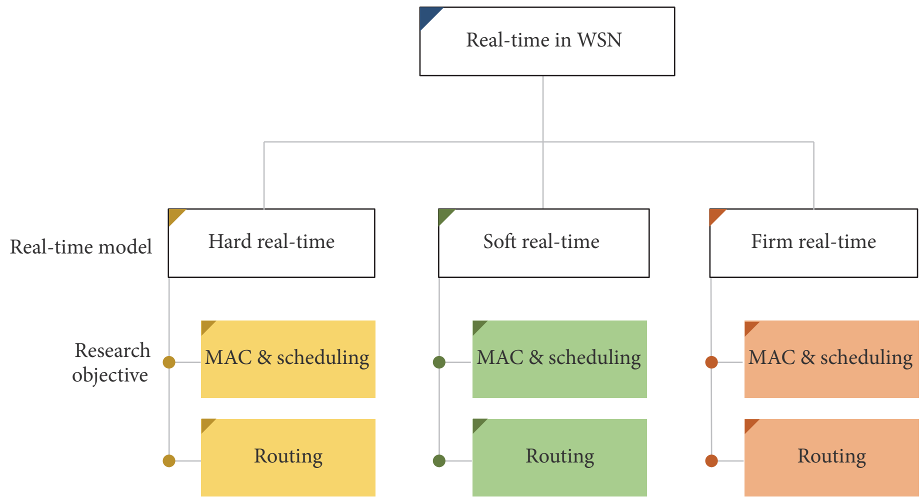

Figure 1: Classification.

guarantee a small delay for disseminating the occurrence of critical events.

3.1.1. MAC and Scheduling. In this section, we briefly describe MAC and scheduling algorithm for hard real-time model.

First, Caccamo and Zhang [29] proposed Implicit Earliest Deadline First (I-EDF) based on famous Earliest Deadline First scheduling algorithm. In this scheme, each node is grouped in a form of hexagonal cells and supports intracellular communication and intercellular communication. I-EFP prevents collision through time-based scheduling based on multiple frequencies in intercellular communication. This time-based scheduling together with multiple frequencies guarantees the collision-free nature of I-EDF. Another scheduling algorithm, Source Aware Scheduling (SAS-TDMA) [30], was proposed to reduce overhead through information under cross-layered architecture. It includes the priority queuing model at the node based on average waiting time. N queue saturation point depends on the sum of the loads of the classes of priority up to N. In addition, Enhanced Distributed Channel Access (EDCA) method is employed at the MAC layer. As for hard real-time protocol for WSN, a multipath routing protocol discovers disjoint paths where a source node selects its route dynamically and checks the quality of the alternative routes with delay metric.

Beside scheduling algorithm, Watteyne et al. [31] proposed real-time MAC protocol under random liner networks. The proposed Dual-MAC regulates medium access according to node's position. An unprotected mode prone to collisions and a slower protected mode free for collision were considered to guarantee worst case times. This property was validated by a formal model. Moreover, Kieckhafer [32] studied Wireless Architecture for Hard Real-Time Embedded Networks (WAHREN) under hard real-time deadline for national project, Pierre Auger Cosmic Ray. At the MAC layer, hybrid TDMA/CSMA window is employed where all infrastructure nodes transmit only within their preassigned TDMA slots and noninfrastructure nodes transmit only within the CSMA to avoid interference between them. Another applicable real-time communication to WSN was proposed by Aísa and Villarroel [33]. Even though Wireless Chain networK Protocol (WICKPro) was proposed for wireless mesh networks with chain topology, its token passing approach and time-token protocol as well as cyclic executive can be implemented in WSN. Moreover, EchoRing is proposed by Dombrowski and Gross [34] to address communication at very short latencies together with high reliabilities for wireless industrial network. It introduces cooperative communication and improved fault tolerance functionality in decentralized way. The measured latency is maintained below 10 ms. At last, unlike single hop TDMA, Ergen and Varaiya proposed multihop TDMA scheme with access point which performs scheduling with gathered topology information. PEDAMACS [35] employs a polynomial-time scheduling algorithm which guarantees a delay proportional to the number of nodes instead of optimization problem as known NP-complete one.

In addition to MAC layer approach, Cherian and Nair [36] presented the priority queuing model at the node based on average waiting time. N queue saturation point depends on the sum of the loads of the classes of priority up to N. In addition, Enhanced Distributed Channel Access (EDCA) method is employed at the MAC layer. As for hard real-time protocol for WSN, a multipath routing protocol discovers disjoint paths where a source node selects its route dynamically and checks the quality of the alternative routes with delay metric. Moreover, as an extension of IEEE 802.15.4, a new mode for Deterministic and Synchronous Multichannel Extension (DSME) and Low Latency Deterministic Network (LLDN) in IEEE 802.15.4e [37] can be regarded as acceptable solution for real-time communications in WSN. The former supports deterministic delay and high reliability to time-varying traffic and operating conditions while LLDN defines a fine granular deterministic TDMA access over star topology. Based on this feature, they are supposed to

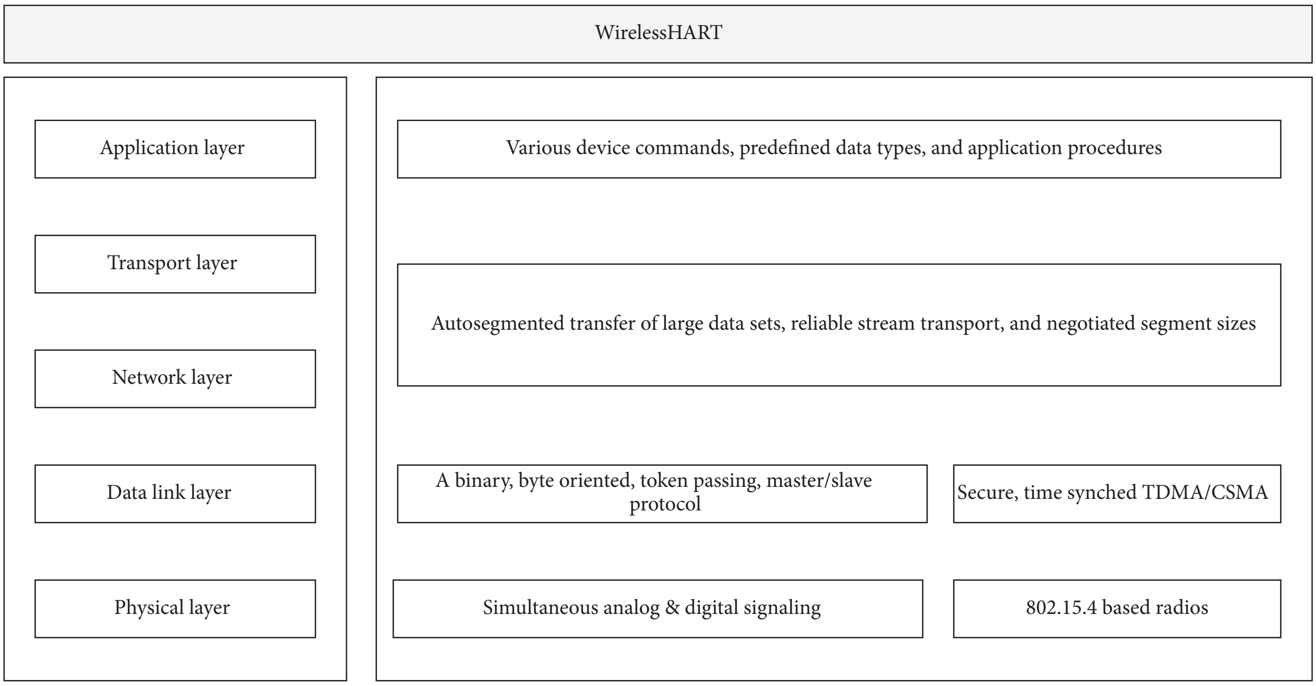

Figure 2: Layering of WirelessHART.

be implemented for healthcare application. In addition to communication protocol, few platforms for real-time communications were proposed. Related to DSME mode, a new channel access scheme and beacon scheduling schemes were proposed by Sahoo et al. [38] to reduce the network discovery time and energy consumption. A new dynamic guaranteed slot allocation algorithm leads to reducing retransmission delay significantly. As for LLDN, Ouanteur et al. [39] presented a three-dimensional Markov chain model to analyze the performance of LDDN in industrial environments. The analysis mode targets estimating reliability, energy consumption, throughput, delay, and jitter. Moreover, they conduct a comparative study between the IEEE 802.15.4e LLDN and the IEEE 802.15.4 slotted carrier sense multiple access with collision avoidance (CSMA/CA) to prove deterministic behavior of the LLDN.

WirelessHART [40], extended from Highway Addressable Remote Transducer Protocol (HART) framework, has a centralized network management architecture, multichannel TDMA transmission, redundant routes, and avoidance of spatial reuse of channels for enhanced reliability and real-time performance. The important device, network manager, is responsible for controlling of scheduling and configuring the routing in the network. By the help of network manager, hard real-time communication is achieved by receiving data from each of the WirelessHART nodes in the network. In addition, network manager runs source and graph routing to meet real-time constraints. TDMA with 10 ms time lost is used at data link layer. Figure 2 illustrates the layering of WirelessHART. In addition to above layers, physical layer defines radio characteristics such as signaling method, signaling strength, and device sensitivity. Moreover, application layer defines various device commands, data types, and response.

In conformity with these rules, there are a few research works over WirelessHART platform. Nobre et al. [41] presented literature review of routing and scheduling for WirelessHART. In addition, some open issues in WirelessHART routing and scheduling algorithms were discussed. Moreover, Lu et al. [42] presented real-time Wireless Sensor-Actuator Networks (WSAN) for industrial control systems through WirelessHART. For real-time service, experimental WSAN testbed and scheduling algorithms were implemented and reviewed. Finally, some issues such as rate control for wireless control systems were presented.

3.1.2. Routing. There is one protocol to address hard real-time communications in WSN since most of schemes assume one-hop communications. Ergen and Varaiya [43] proposed new real-time routing protocol to consider network lifetime. At first, a scheme to maximize the minimum lifetime of each node is presented without considering delay as programming problem. The second approach is to incorporate delay guarantee into energy efficient routing by constraining the length of the routing paths.

3.2. Soft Real-Time Communications in WSN. Followed by few hard real-time protocols in WSN, routing and MAC protocol for soft real-time communication are described in this section. In this model, three platforms have been presented. RAP [44] architecture attracts researcher's interest. Since RAP provides the query/event service API, hence it is used for registering the query for specific sensing event. The query stores timing constraints, transmission period, and

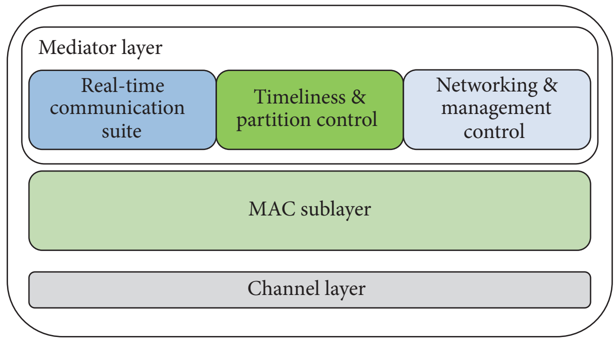

Figure 3: Architecture of Wi-STARK.

destination location information. Hence, when a registered event occurs, the query result is automatically transmitted to the destination. RAP uses velocity monotonic scheduling to match the end-to-end deadline of query result packets. It uses the requested deadline and the distance to the destination to obtain the requested velocity, assigns a high priority to the packet with the higher requested velocity, and ensures that transmission is done within the packet's deadline. Another framework to consider both real-time communications and energy efficiency is Real-Time Power-Aware (RTPAW) framework [45]. Key feature of RTPAW includes aggregation layer between MAC and routing layer under cluster architecture. Aggregation layer is responsible for creating and maintaining cluster to balance energy consumption by control of active and sleep period. The other architecture called Wi-STARK [46] was proposed to consider both fault tolerance and real-time communication at the same time. It has compliance with standard IEEE 802.15.4 and provides service interface which can be used in building control application. Guarantee of timeliness as well as resilient communication services is achieved in one-hop communication domain by real-time communication suite which consists of message request dispatcher and protocol bundle. To be detailed, Figure 3 shows architecture of Wi-STARK with three different layers. There are three major components such as real-time communication suite, timeliness and partition control (TPC), and networking and management control. Particularly, networking and management control in mediator layer is responsible for integrating all functionalities as well as providing management services.

3.2.1. MAC and Scheduling. Several soft real-time MAC protocols were proposed. Matischek et al. proposed Real-Time Hybrid MAC (RTH-MAC) protocol [47] to combine both TDMA and FDMA to offer soft real-time communications. To meet real-time requirement, centralized approach that eliminates collisions, minimizes interferences, and ensures a small bounded end-to-end delay was taken. Moreover, RTH-MAC employed an acknowledgment mechanism and duty cycling ratio for reliability and adaptability. As another type of combined MAC, Abdeli et al. [48] presented MAC protocol which provides network traffic prioritization in order to guarantee worst case message delays for a set of high-prioritized nodes automotive applications. The proposed, Soft Real-Time Shared Time slot (SRTST), uses a shared time slot method by combining TDMA and CSMA/CA mechanisms in a special two-step way.

In addition, GinLITE [49] was developed as one of the components of GINSENG system to offer time-critical and reliable data delivery by utilizing a purely static topology with precomputed and static TDMA schedule. Basically, GinLITE is a mesh under TDMA MAC protocol which operates through static topology/schedule information. A new MAC protocol with black-burst (BB) mechanism was proposed to provide real-time access in [50]. But, in order to decrease energy loss and latency caused by long length, a binary coding scheme is applied to coding-black-burst-based protocol. Zhang et al. [27] presented real-time MAC protocol to meet high throughput, low latency, and energy consumption by accurate time synchronization. To achieve this goal, hybrid approach to combine TDMA with novel time synchronization approach and Frequency Hopping Spread Spectrum (FHSS) was presented to include antijamming and collision prevention. Particularly, for real-time communication, command and data packets are delivered in a bucket brigade-like manner for bandwidth utilization. Figure 4 shows the example of tree topology to communicate with sink node. Each sensor node is grouped by cluster so it transmits sensed data to cluster header. As shown in Figure 4, each sensor node communicates with parent node during phase 1 while communication with child node is performed during phase 2. Moreover, Shukeri et al. [51] studied cluster architecture for adaptive TDMA scheduling in WSN. Through the adaptive scheme to type of flows, channel is dynamically allocated to achieve better utilization by minimizing the number of unused channels.

In addition to new MAC protocol development, Ali et al. [52] presented experimental results for prototype of TSMAC which is implementation of a multihop mesh topology real-time IEEE 802.15.4-based MAC protocol for Contiki OS. To implement TSMAC, radio duty cycle was modified to realize the slotted transmission mode. In the testbed, the network operates in Beaconless mode and all nodes are powered by USB hub. Throughout the experimental results, TSMAC shows higher throughput and lower collision and jitter than existing CSMA.

Furthermore, Mouradian et al. presented MAC and routing protocol under cross-layered architecture. The proposed

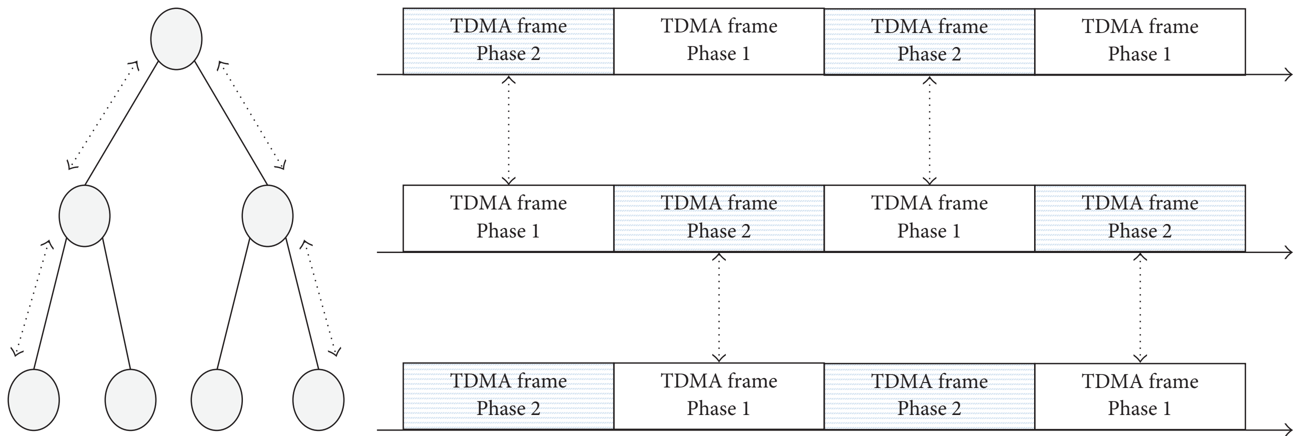

FIGURE 4: Staggered scheme based on tree topology in [27].

Real-Time X-layer Protocol (RTXP) [53] aims at guaranteeing an end-to-end requirement while keeping energy consumption. It relies on a hop-count-based Virtual Coordinate System (VCS) which classifies nodes having the same hop-count from the sink, allows forwarder selection, and gives to the nodes a unique identifier in a 2-hop neighborhood allowing deterministic medium access. Another scheme to feature cross-layer forwarding and medium access control was presented in Service-Differentiated Real-time Communication Scheme (SDRCS) [54]. It integrates real-time routing and prioritized MAC by performing packet speed estimation and admission control in a distributed way under dynamic network topologies. To be detailed, local prioritized packet forwarding is performed to maximize packet speed.

3.2.2. Routing. First of all, various types of soft real-time routing protocol have been proposed. Among them, SPEED [55] has good reputation and diverse variations for real-time communication. SPEED is a real-time communication protocol, which supports feedback control and nondeterministic geography forwarding so as to guarantee the desired delivery speed. In the stateless nondeterministic geographic forwarding (SNGF) module used by SPEED, the node with the highest relay speed is being selected as the next hop. In order to calculate the relay speed in SNGF module, neighbor node distance and location information are collected by exchanging beacon. Moreover, in order to determine transfer delay between nodes, data packet is sent to the neighbor node. After the ACK message is received, it is possible to acquire delay estimation through the round trip time. SNGF offers not only Back-pressure Rerouting (BR) but also Neighborhood Feedback Loop (NFL) models which provides traffic and congestion control in order to guarantee desired delivery speed. NFL uses neighbor nodes miss rate information to decide if a node cannot maintain more than a single hop relay speed. When NFL has been activated, instantly the average transfer delay value of each node within the congestion area is added to back-pressure beacon and transferred into the upstream node. After receiving this beacon packet, the upstream node controls the sending rate by stopping to send the packets into the area of congestion.

In parallel with SPEED, MMSPEED [56] creates a virtual multiple speed layer and performs virtual isolation to classify the incoming packets into the appropriate priority queues according to the required delivery velocity to prevent packets from being delayed by slow packets in the queue. MMSPEED groups single or multiple forwarding path of the packet depending on the required reliability level of the packet. In other words, packet layering is performed according to the reliability of a packet. The lower reliability packet is transmitted to a single path while the high reliability packet is transmitted to a destination via multiple paths. Moreover, energy efficient SPEED [57] was proposed. Like the existing SPEED routing protocol, it transmits routing information and its own residual energy information to neighbor nodes through beacon exchange. In the SPEED routing protocol, it is highly likely that the node with the highest relay speed among the other neighbor nodes will be selected as the next hop. But, in the proposed routing protocol, the weight function of the neighboring nodes is calculated based on the information exchanged with the beacon exchanging method and the neighboring node with the highest value in weight function is the one that is selected as the next hop. Furthermore, Zhou et al. proposed POWER-SPEED [58] which does not use a control packet but uses the upstream hop-by-hop delay in the data packet at each relaying node to the destination to perform future hop-by-hop delay estimation. In addition, to support energy efficient routing in POWER-SPEED, the minimum value of total energy consumption required for transmission is calculated, and then the transmitter power level required for the relay to the neighboring node is adaptively set according to this value. Furthermore, remaining energy was concerned in [59] by introducing weight which is calculated by adding delay with remaining energy of neighbor node. Furthermore, Aissani et al. [60] presented EA-SPEED to drop the delayed packets in early time and extend the stateless nondeterministic geographic forwarding (SNGF) of the SPEED protocol. In this extension, next hop was decided while considering speed and residual energy instead of random selection in original SPEED. Another energy aware real-time protocol, EARQ [61], was proposed to support real-time, reliable delivery of a packet and energy awareness in wireless industrial sensor networks. To do this, each node exchanges beacon messages with its neighbor nodes and records the energy cost, time delay, and reliability needed to reach the sink node into its routing table. Since the path with less energy cost is more likely to be selected than another path, if the reliability of the selected next hop is lower than the required reliability, a new next hop is selected and then the redundant packet is transmitted so as to ensure reliable packet delivery.

In the aspects of fault tolerance, another extension of SPEED, FT-SPPED [62], focused on fault tolerant property. FT-SPEED solves the void problem which existed in the previous SPEED protocol by adding Void Announce Scheme (VAS) and a Void Bypass Scheme (VPS). In addition to fault tolerance and energy efficiency in SPEED, IMMSpeed [63] transmits the other copy of the remaining packet to the remaining nodes if the required number of neighbors is insufficient. IMMSpeed does not select the neighboring node with the fastest forwarding speed as the next hop; rather it selects the node with the highest energy among the neighboring nodes that can keep the real-time deadline as the next hop. Although the energy is the same then the neighboring node with the fast forwarding speed is selected as the next hop. In addition, alternative path selection algorithm based on Neural Network [64] was additionally applied to SPEED for the case of path failure and sleep node. In this work, Neural Network is applied to evaluate QoS parameter and get the optimized path. Some of these addressed protocols were compared and analyzed in [65] through proposed energy model to verify the acceptable performance.

Besides SPEED and its variants, some soft real-time routing protocols have been proposed. First, Enhanced Real-Time with Load Distribution (ERTLD) [66] selects optimal forwarding node based on Received Signal Strength Indicator (RSSI), remaining battery level of sensor nodes, and packet delay. For this purpose, the ERTLD uses the corona mechanism, which computes the corona level of the mobile node according to the distance from the mobile sink, and each mobile node sets one-hop neighbors with a corona level smaller than or equal to itself as forwarding candidate nodes. Each mobile node computes the optimal forwarding (OF) value using three parameters among the selected forwarding candidate nodes with mentioned procedures. These parameters are packet rate, RSSI as link quality, and remaining power. Therefore, the neighbor node with the highest OF value is the one that ends up being selected as the next hop. Another protocol called Potential-based Real-Time Routing (PRTR) [67] divides the packet into real-time and non-real-time packets to prevent the shortest path from being congested by nondelay sensitive packets in WSN. The real-time packet selects the shortest path and the non-real-time packet selects another routing path; thereby these approaches reduce the congestion of the shortest path and reduce the transmission delay of the real-time packets. For this operation, PRTR uses the flag field of each packet header

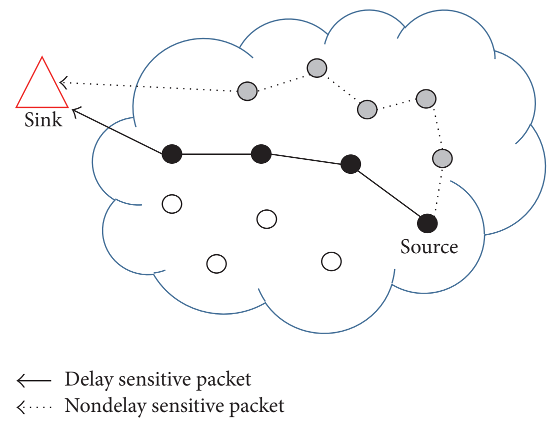

Figure 5: Data delivery in PRTR.

to distinguish whether the packet is a delay sensitive packet or a nondelay one and allows the delay sensitive packet to be transmitted ahead of other packets using the priority queue. Figure 5 shows the example of data forwarding in PRTR. As illustrated in Figure 5, nondelay sensitive packet is forwarded along the different path to prevent congestion. Each packet is identified by flag field in the packet header. Quang and Kim [68] proposed combining a Two-Hop Velocity based Routing (THVR) algorithm and a gradient-based network to reduce deadline miss ratio (DMR) and improve energy efficiency in industrial wireless sensor networks. Moreover, Mahapatra et al. [69] proposed an energy aware dual-path routing scheme considering packet delivery deadlines, efficient energy consumption, and reliability in WSN. Each node periodically exchanges HELLO_PKT as beacon message and calculates the location, remaining energy, and estimated time delays of neighboring nodes. After that, each node calculates an urgency factor based on the remaining distance and slack time information until the packet arrives at the destination. Moreover, urgent packets based on the urgency factor are transmitted to the boundary of the transmission range. At last, Rachamalla and Kancharla [70] proposed Energy Efficient Adaptive Routing Protocol (EE-ARP) by combining adaptive transmission power algorithm with any geographic routing to improve energy efficiency.

3.3. Firm Real-Time Communications in WSN. The research for (m, k)-firm communications in WSN was conducted in two directions. One is to extend the existing scheme to accommodate (m, k)-firm requirement streams and the other is to develop new communication protocols to meet (m, k)-firm requirement. Even though two types of mechanisms have different objectives, most approaches have in common the fact that they make use of Distance Based Priority (DBP) value to differentiate priority or choose the next hop. DBP value is used to indicate the current stream status.

In this model, Lee et al. [71] presented new architecture to solve scalability problem by new architecture for (m, k)-firm streams. In the new integrated architecture, flow aggregation scheme derived from compositional hierarchical model and velocity based protocol were proposed to solve

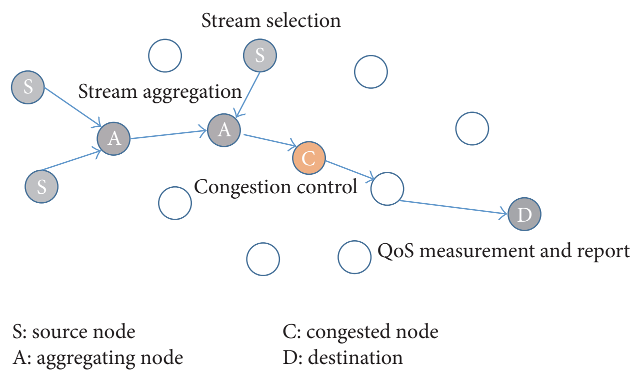

Figure 6: Example of operation.

the scalability problem. In addition, hybrid MAC protocol and congestion control scheme were proposed to meet (m, k)-firm constraints in efficient way. Figure 6 shows the example of operation in the proposed scheme. All source nodes first select the adequate streams. If this stream is delivered to the same node, streams are aggregated into one stream. If the congestion happens at node j, a Level_Adjustment message is back to j to reduce sending traffic rate. The sink or destination node measures the QoS level periodically and sends reply to each source node. This reply triggers stream selection procedure if the condition is met. As an sequential approach, optimization solution for (m, k)-firm stream was not presented yet.

3.3.1. MAC and Scheduling. There are several approaches to develop new MAC and scheduling mechanism to meet (m, k)-firm streams. As for scheduling, Kim [72] proposed a new scheduling algorithm for (m, k)-firm stream. The priority of packet was dynamically assigned by DBP value, slack time, distance to sink node, and link stability. Particularly, network congestion around the sink node is crucially concerned by adjusting weighting factor. Moreover, Zhao and Xiong [73] proposed a channel-aware scheduling algorithm through packet partitioning and real-time delivery. The former combines static assignment and dynamic adjustment to minimize the number of packets over bad channel state while the latter is accomplished by automatic repeat request. Moreover, Semprebom et al. [74] presented dynamic Guaranteed Time Slot (GTS) scheduling approach based on the (m, k)-firm task model to prevent starvation problem as well as ensure delivery of at least m messages in each window. Due to inherent limitations in processing power and energy consumption, a fixed priority scheduling algorithm rather than dynamic one was proposed. At last, even though Matusovsky [75] proposed a recovery from data losses to support real-time application in wireless networks, this proposed scheme can be easily applicable in WSN. In this work, a recovery was accomplished by retransmission of failed packets under a point to multipoint network with fixed number of nodes. The proposed algorithm was derived from Reinforcement Learning theory.

Moreover, a new MAC protocol to enhance IEEE 802.15.4/ZigBee was proposed by Semprebom et al. [76]. The proposed scheme aimed at improving CSMA/CA algorithm by assigning decentralized priorities based on the (m, k)-firm task model. So, occurrences of dynamics failures can be reduced by assigning the highest priority level to node which is close to a dynamic failure condition. By the help of the highest priority, success probability increases in the next transmission.

3.3.2. Routing. An approach to extend SPEED described in previous section was presented in [77]. In the extended (m, k)-firm SPEED, the next hops selection depended on velocity of link and current DBP value. If DBP value is greater than 0, probabilistic selection is performed among the possible links having higher speed than threshold in order to distribute the load. Otherwise, a suitable link is selected according to DBP value. The next extension is made on Plum-Slowly Fetch-Quickly (PSFQ) to support the (m, k)-firm stream efficiently in [78]. To meet (m, k)-firm requirement, a segment is divided into m subsegments. If the current DBP value carried on the packet is negative, more strict requirement is temporarily made to ensure the packet delivery as well as control the retransmission. Unlike the addressed approach which is based on real-time scheme, extension of general ad hoc routing was presented by Tekaya et al. [79]. The objective of this extension is to introduce scheduling policy to increase admission rate of real-time traffic over AODV protocol.

Beside extension of the existing protocol, Jiang et al. [80] proposed Link Quality Estimation based Routing (LQER) protocol to monitor environmental monitoring in wetlands through (m, k)-firm link quality estimating. To achieve energy efficiency, LQER employed minimum hop field based routing protocol by limiting the number of participating nodes in the transmission of packets. Even though LQER introduced (m, k)-firm model, its applicability was limited to use of (m, k)-firm as metric of the link. Thus, a new approach to use (m, k)-firm as application requirement emerged. Moreover, Kim and Sung [81] proposed a new geographical routing protocol to meet (m, k)-firm requirements. The new protocol took delay, distance, and remaining slack time into account for priority-based scheduling and a geographic forwarding scheme. For the latter, new measurement for link quality and adaptive next hop selection algorithm were presented.

In addition to previous communication protocols, the following papers focused on recovery scheme for (m, k)-firm stream. Li and Kim [82] proposed a new fault recovery mechanism by employing a local status indicator (LSI) to adjust transmission capability. LSI is used to make the intermediate nodes that were bared aware of their local transmission conditions. By the help of LSI and streams DBP, three different major sources for packet loss and long delay are identified clearly. They are congestion, link failure, and void in the networks, respectively. Depending on one of them, different recovery scheme is adaptively applied. This LSI based routing protocol for (m, k)-firm stream was improved by the same authors in [83]. The extension is for energy efficiency so next hop is selected by node with maximum remaining energy. Furthermore, Nam [84] proposed a load balancing routing protocol (LBRP) to meet

(m, k)-firm constraints while taking into account energy efficiency and extended network lifetime. In this mechanism, each node maintains two different groups for forwarding. The next hop is chosen from each group sequentially to prevent energy consumption by multiple forwarding. The candidate set for next hop consists of nodes having higher speed than threshold.

Unlike previous general layered architecture, cross-layered approach was presented by Kim and Sung [85]. In this approach, new scheme was developed in each layer with (m, k)-firm requirement passed from the application. Based on this information, adjustment of transmission range in physical layer was developed. Also, prioritization of packet in MAC layer and multiple paths establishment in network layer are completed to meet (m, k)-firm constraints dynamically. Moreover, one framework that integrates each scheme shows the low failure probability for real-time requirements on (m, k)-firm stream by the help of the proposed scheme. However, since there are researches for communication protocol, Kim and Sung [86] presented traffic model and new routing protocol together. Particularly, application and clustering scheme were firstly mentioned and addressed in this work. According to DBP value, duplicated data packets were assumed to be transmitted to recover negative DBP status and link stability was presented in a form of (m, k). High performance was measured and obtained by these two schemes. Despite the above approaches, there is applicability problem to introduce (m, k)-firm in WSN. To address this problem, Azim et al. [87] presented multicriteria system for forwarding for (m, k)-firm stream since current existing protocols applied these parameters sequentially without any prioritization. This implied that there are many perspective parameters to be considered in forwarding scheme. Optimization is achieved by two approaches, fuzzy interference system and analytical hierarchical process in conjunction with the gray relational analysis. These two protocols took delivery ratio, energy, speed, and (m, k)-firm stream requirement as well as current stream status to select the next hop. As final approach, a new clustering scheme for (m, k)-firm stream was proposed by Kim [88]. For the clustering scheme, header and members are chosen by (m, k)-firm requirement or deadline.

4. Open Issue

4.1. Multicast and Broadcast. As we explained in the previous section, most of real-time communication is performed in unicast communication. However, there are increasing demands for either group communications or data delivery to whole nodes. For example, a sink node delivers new mission to whole nodes or inform some nodes for object information to be tracked. Thus, real-time multicast and broadcast are critical and essential research challenge. However, fewer researches for real-time multicast and broadcast have been conducted than unicast. To be detailed, since multicast and broadcast are largely dependent on delivery tree, tree initialization and maintenance procedure should take deadline into account.

4.2. Energy Efficiency. Due to node operation with battery, energy efficiency issue is always given to higher priority than other features in WSN. Particularly, most schemes for real-time communications are likely to choose a path with least cost repeatedly. In this case, a node's battery along the path will be quickly drained so it becomes unavailable at early time. Consequently, failure on node results in short network lifetime. So, real-time communications protocol should be designed in energy efficient way. Duty cycle in MAC layer and utilizing multiple paths are good approach to achieve energy efficiency. In addition, energy aware scheduling for TDMA and QoS routing needs to be explored as further study.

4.3. Simulation Model. Performance of most research for real-time communication is accomplished by simulation. Currently, various simulation platforms including TOSSIM, OMNeT++, and NS-2/3 are general frameworks to conduct simulation for WSN. Particularly, Lalomia et al. [89] proposed hybrid simulation model with augmented version of TOSSIM by merging actual and virtual nodes seamlessly as well as interacting with each other. To ensure soft real-time in WSN, simulation timing is constrained to handle simultaneous events by scheduler. In addition, Rousselot et al. [90] presented OMNeT++ simulation models based on the IEEE 802.15.4 with four evaluation models to validate the timeliness. However, a validated simulation model for real-time communications for WSN is not released yet. So, adding on module for real-time communications should be implemented and integrated with current simulator.

4.4. Network Architecture. Most of protocols for real-time communications for WSN are designed by assuming flat network architecture. On the other hand, several clustering schemes for real-time communications have been proposed in mobile ad hoc networks. Clustering scheme results in low energy consumption and routing overhead. Also, since clustering can solve scalability problem, real-time communication for large scale WSN should take clustering into account. In addition, end-to-end delay is closely related to number of hops so reduced hops contribute to meeting deadline requirement. Consequently, details to create and maintain clustering as well as path selection are worth being studied. In this aspect, cross-layered architecture and approach are another promising research area to improve the performance. Related to this issue, it is worth mentioning that superframe duration allocation schemes for cluster-heads lead to improvement throughput for cluster-tree [91]. Through the proposed allocation scheme, network congestion around the PAN coordinator, high message communication delays, and a high number of discarded messages due to buffer overflows are significantly reduced. Thus, it is very suitable for wide-scale networks with energy efficiency QoS.

4.5. Programming Models and Tools. Even though many programming models and tools were introduced in [92] for WSN, there are no programming model and tool for real-time communications yet. Moreover, since they are related to operating system and debugging tools together, programming tool-chain needs to be explored. Moreover, programming tools should consider the specific application for real-time requirements; appropriate Application Programming Interface (API) to manage component needs to be developed. Moreover, Integrated Development Environment (IDE) is demanded to allow user to develop real-time application easily.

4.6. Applicability to Ad Hoc Networks. Since WSN is based on ad hoc network technology, most of the presented schemes work in ad hoc networks without significant changes even though the opposite is not feasible. However, some schemes which are designed to be specific to WSN need to be modified or extended to be applied in ad hoc networks. Thus, protocols assuming high density of nodes and data aggregation in WSN need to be modified to reduce the interference. For example, in case of original SPEED [55] protocol, each node keeps a neighbor table with nodes in transmission range for reliability. However, the limited number of nodes around boundary of transmission range is enough for table in case of ad hoc networks where wider range and bandwidth than WSN are given. On the other hand, if such limited resource constraints on sensor node get loosen, current protocols can support real-time communications easily in ad hoc networks.

4.7. Deployment and Applications. Even though FireFly was reported to be deployed in coal mine for people tracking with voicemail communication, there is no scheme overwhelming others. Also, there is only one application, voice communications, which makes use of real-time communications in WSN. This indicates that more optimization and customization for the protocol and framework are demanded for killer application. Related to this issue, since real operation is affected by limitation on wireless link and energy constraints, integration with wireless cellular networks or LAN needs to be studied. For example, Al-Rousan and Kullab [93] presented two-tiered architecture for real-time communications in WSN. In their approach, WLAN serves as a backbone to an adaptively clustered Low Energy Adaptive Clustering Hierarchy- (LEACH-) based wireless sensor network. Through this architecture, reliable data delivery with reduced delay bounds and lower energy consumption is observed in WSN.

5. Conclusion

In this paper, we reviewed recent literatures for real-time communications in WSN. Even though it is not easy to provide real-time communication in WSN when it comes to take harsh environments into account, the demands for real-time delivery are more increasing. In order to meet this demand, various approaches based on hard and soft real-time model were taken. In addition, we explained existing research work for (m, k)-firm model in WSN. Each protocol was briefly introduced and explained. Finally, further research challenges and issues were presented to give guideline for research trend.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant no. NRF-2015R1D1A3A01019680).

References

[1] I.-H. Hou and P. R. Kumar, "Real-time communication over unreliable wireless links: A theory and its applications," *IEEE Wireless Communications Magazine*, vol. 19, no. 1, pp. 48–59, 2012.

[2] Z. Shen, P. Xu, and X. Xu, "A feedback-based timeout packets dropping strategy in real-time wireless sensor networks," *Lecture Notes in Electrical Engineering*, vol. 127, no. 4, pp. 207–212, 2012.

[3] C. Busch, M. Magdon-Ismail, F. Sivrikaya, and B. Yener, "Contention-free MAC protocols for asynchronous wireless sensor networks," *Distributed Computing*, vol. 21, no. 1, pp. 23–42, 2008.

[4] R. Oliver and G. Fohler, "Timeliness in Wireless Sensor Networks: Common Misconceptions," in *Proceedings of International Workshop on Real-Time Networks*, July 2010.

[5] G. Anastasi, M. Conti, and M. Di Francesco, "A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 52–65, 2011.

[6] A. Alanazi and K. Elleithy, "Real-time QoS routing protocols in wireless multimedia sensor networks: study and analysis," *Sensors*, vol. 15, no. 9, pp. 22209–22233, 2015.

[7] P. Chennakesavula, J. Ebenezer, and S. Satya Murty, "Real-time Routing Protocols for Wireless Sensor Networks: A Survey," in *Proceedings of International Workshop on Computer Networks Communications*, pp. 141–158, October 2012.

[8] S. Rachamalla and A. Kancharla, "Survey of Real-time Routing Protocols for Wireless Sensor Networks," *International Journal of Computer Science & Engineering Survey*, vol. 4, no. 3, pp. 35–44, 2013.

[9] A. Zhan, T. Xu, G. Chen, B. Ye, and S. Lu, "A Survey on Real-time Routing Protocols for Wireless Sensor Networks," in *In Proceedings of China Wireless Sensor Network Conference*, 2008.

[10] Z. Teng and K. Kim, "A Survey on Real-Time MAC Protocols in Wireless Sensor Networks," *Communications and Network*, vol. 02, no. 02, pp. 104–112, 2010.

[11] Y. Li, C. Chen, Y. Song, and Z. Wang, "Real-time QoS Support in Wireless Sensor Networks: A Survey," in *Proceedings of International Conference on Field buses Networks in Industrial Embedded Systems*, 2007.

[12] M. Collotta, D. G. Costa, F. Falcone, and X. Kong, "New challenges of real-time wireless sensor networks: Theory and applications," *International Journal of Distributed Sensor Networks*, vol. 12, no. 9, 2016.

[13] K. M. Al-Aubidy, A. M. Derbas, and A. W. Al-Mutairi, "Real-time patient health monitoring and alarming using wireless-sensor-network," in *Proceedings of the 13th International Multi-Conference on Systems, Signals and Devices, SSD 2016*, pp. 416–423, deu, March 2016.

[14] C. Li-Wan, C. Qiang, and L. Hong-Bin, "Wireless sensor network system for the real-time health monitoring," *Lecture Notes in Electrical Engineering*, vol. 97, no. 1, pp. 9–14, 2011.

[15] D. Choudhary, R. Kumar, and N. Gupta, "Real-time health monitoring system on wireless sensor network," *International Journal of Advance Innovations, Thoughts & Ideas*, vol. 1, no. 5, 2015.

[16] T. He, P. Vicaire, T. Yan et al., "Achieving real-time target tracking usingwireless sensor networks," in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '06)*, pp. 37–48, IEEE, April 2006.

[17] A. Pozzebon, C. Bove, I. Cappelli, F. Alquini, D. Bertoni, and G. Sarti, "Heterogeneous Wireless Sensor Network for Real Time Remote Monitoring of Sand Dynamics on Coastal Dunes," in *Proceedings of the World Multidisciplinary Earth Sciences Symposium, WMESS 2016*, cze, September 2016.

[18] R. T. Tse and Y. Xiao, "A portable Wireless Sensor Network system for real-time environmental monitoring," in *Proceedings of the 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2016*, prt, June 2016.

[19] Y. Cheng, X. Li, Z. Li et al., "AirCloud: a cloud-based air-quality monitoring system for everyone," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys '14)*, pp. 251–265, Memphis, TN, USA, November 2014.

[20] D. M. Holstius, A. Pillarisetti, K. R. Smith, and E. Seto, "Field calibrations of a low-cost aerosol sensor at a regulatory monitoring site in California," *Atmospheric Measurement Techniques*, vol. 7, no. 4, pp. 1121–1131, 2014.

[21] D. Georgoulas and K. Blow, "In-Motes EYE: A Real Time Application for Automobiles in Wireless Sensor Networks," *Journal of Wireless Sensor Network*, vol. 03, no. 05, pp. 158–166, 2011.

[22] A. Rowe, R. Mangharam, and R. Rajkumar, "FireFly: a time synchronized real-time sensor networking platform," in *Wireless Ad Hoc Networking: Personal-Area, Local-Area, and the Sensory-Area Networks*, CRC Press Book, Chapter, 2006.

[23] A. Rowe, D. Goel, and R. Rajkumar, "FireFly Mosaic: A vision-enabled wireless sensor networking system," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium, RTSS 2007*, pp. 459–468, usa, December 2007.

[24] S. Saruwatari, M. Suzuki, and H. Morikawa, "PAVENET OS: a compact hard real-time operating system for precise sampling in wireless sensor networks," *SICE Journal of Control, Measurement, and System Integration*, vol. 5, no. 1, pp. 24–33, 2012.

[25] Y. Fu, Q. Guo, and C. Chen, "A-LNT: a wireless sensor network platform for low-power real-time voice communications," *Journal of Electrical and Computer Engineering*, vol. 2014, Article ID 394376, 19 pages, 2014.

[26] E. I. Cosar, A. Mahmood, and M. Björkbom, "A-stack: A real-time protocol stack for IEEE 802.15.4 radios," in *Proceedings of the 36th Annual IEEE Conference on Local Computer Networks, LCN 2011*, pp. 1020–1023, deu, October 2011.

[27] J. Zhang, J. Wu, Z. Han, L. Liu, K. Tian, and J. Dong, "Low power, accurate time synchronization mac protocol for real-time wireless data acquisition," *IEEE Transactions on Nuclear Science*, vol. 60, no. 5, pp. 3683–3688, 2013.

[28] REWIN Project. http://www.cister.isep.ipp.pt/projects/rewin/.

[29] C. Caccamo and L. Y. Zhang, "The capacity of implicit EDF in wireless sensor networks," in *Proceedings of the 15th Euromicro Conference on Real-Time Systems, ECRTS 2003*, pp. 267–275, prt, July 2003.

[30] W. Shen, T. Zhang, M. Gidlund, and F. Dobslaw, "SAS-TDMA: a source aware scheduling algorithm for real-time communication in industrial wireless sensor networks," *Wireless Networks*, vol. 19, no. 6, pp. 1155–1170, 2013.

[31] T. Watteyne, I. Augé-Blum, and S. Ubéda, "Dual-mode real-time MAC protocol for wireless sensor networks: A validation/simulation approach," in *Proceedings of the 1st International Conference on Integrated Internet Ad hoc and Sensor Networks*, fra, May 2006.

[32] R. M. Kieckhafer, "Hard real-time wireless communication in the northern Pierre Auger observatory," in *Proceedings of the 17th IEEE-NPSS Real Time Conference, RT10*, prt, May 2010.

[33] J. Aísa and J. L. Villarroel, "WICKPro: a hard real-time protocol for wireless mesh networks with chain topologies," in *Proceedings of the European Wireless Conference (EW '10)*, pp. 163–170, April 2010.

[34] C. Dombrowski and J. Gross, "EchoRing: A low-latency, reliable token-passing MAC protocol for wireless industrial networks," in *Proceedings of the 21st European Wireless Conference*, May 2015.

[35] S. C. Ergen and P. Varaiya, "PEDAMACS: power efficient and delay aware medium access protocol for sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 7, pp. 920–930, 2006.

[36] M. Cherian and T. Nair, "Priority based bandwidth allocation in wireless sensor networks," *International Journal of Computer Networks & Communications*, vol. 6, no. 6, pp. 119–128, 2014.

[37] D. De Guglielmo, S. Brienza, and G. Anastasi, "IEEE 802.15.4e: A survey," *Computer Communications*, vol. 88, pp. 1–24, 2016.

[38] P. K. Sahoo, S. R. Pattanaik, and S.-L. Wu, "A novel IEEE 802.15.4e DSME MAC for wireless sensor networks," *Sensors*, vol. 17, no. 1, article no. 168, 2017.

[39] C. Ouanteur, D. Aïssani, L. Bouallouche-Medjkoune, M. Yazid, and H. Castel-Taleb, "Modeling and performance evaluation of the IEEE 802.15.4e LLDN mechanism designed for industrial applications in WSNs," *Wireless Networks*, vol. 23, no. 5, pp. 1343–1358, 2017.

[40] IEC, IEC 62591: Industrial Communication Networks - Wireless Communication Network and Communications Profiles - WirelessHART; 2010.

[41] M. Nobre, I. Silva, and L. A. Guedes, "Routing and scheduling algorithms for wirelesshart networks: A survey," *Sensors*, vol. 15, no. 5, pp. 9703–9740, 2015.

[42] C. Lu, A. Saifullah, B. Li et al., "Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1013–1024, 2016.

[43] S. C. Ergen and P. Varaiya, "Energy efficient routing with delay guarantee for sensor networks," *Wireless Networks*, vol. 13, no. 5, pp. 679–690, 2007.

[44] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: a real-time communication architecture for large-scale wireless sensor networks," in *Proceedings of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '02)*, pp. 55–66, September 2002.

[45] E. Toscano, O. Mirabella, and L. Bello, "An Energy-efficient Real-Time Communication Framework for Wireless Sensor

Networks," in *Proceedings of International Workshop on Real-Time Networks*, 2007.

[46] J. L. Souza and J. Rufino, "The Wi-STARK architecture for resilient real-time wireless communications," *ACM SIGBED Review*, vol. 11, no. 4, pp. 61–66, 2015.

[47] R. Matischek, T. Herndl, C. Grimm, and J. Haase, "Real-time wireless communication in automotive applications," in *Proceedings of the 14th Design, Automation and Test in Europe Conference and Exhibition, DATE 2011*, pp. 1036–1041, fra, March 2011.

[48] D. Abdeli, S. Zelit, and S. Moussaoui, "RTH-MAC: A real time hybrid MAC protocol for WSN," in *Proceedings of the 11th International Symposium on Programming and Systems, ISPS 2013*, pp. 153–162, April 2013.

[49] J. Brown and U. Roedig, "GinLITE - A MAC Protocol for Real-Time Sensor Networks," in *Proceedings of IEEE European Workshop on Wireless Sensor Networks*, December 2012.

[50] F. Yu, L. Wang, D. Gao, Y. Wang, and X. Zhang, "Real-time MAC protocol based on coding-black-burst in wireless sensor networks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E97A, no. 11, pp. 2279–2282, 2014.

[51] N. M. Shukeri, M. A. Rahim, and T. Wan, "Empirical testing of prototype real-time multi-hop MAC for Wireless Sensor Networks," in *Proceedings of the 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 10–15, Penang, Malaysia, November 2016.

[52] G. Ali, K. H. Kim, and K.-I. Kim, "Adaptive TDMA scheduling for real-time flows in cluster-basedwireless sensor networks," *Computer Science and Information Systems*, vol. 13, no. 2, pp. 475–492, 2016.

[53] A. Mouradian, I. Augé-Blum, and F. Valois, "RTXP: A localized real-time MAC-routing protocol for wireless sensor networks," *Computer Networks*, vol. 67, pp. 43–59, 2014.

[54] Y. Xue, B. Ramamurthy, and M. C. Vuran, "SDRCS: a service-differentiated real-time communication scheme for event sensing in wireless sensor networks," *Computer Networks*, vol. 55, no. 15, pp. 3287–3302, 2011.

[55] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: a stateless protocol for real-time communication in sensor networks," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems*, pp. 46–55, Providence, RI, USA, May 2003.

[56] 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.

[57] M. S. Kordafshari, A. Pourkabirian, K. Faez, and A. M. Rahimabadi, "Energy-efficient speed routing protocol for wireless sensor networks," in *Proceedings of the 5th Advanced International Conference on Telecommunications, AICT 2009*, pp. 267–271, ita, May 2009.

[58] Y. Zhou, E. C.-H. Ngai, M. R. Lyu, and J. Liu, "POWER-SPEED: A power-controlled real-time data transport protocol for wireless sensor-actuator networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC 2007*, pp. 3739–3743, chn, March 2007.

[59] I. Memon, N. Memon, and F. Noureen, "Modified SPEED protocol for wireless sensor networks," *QUAID-E-AWAM University Research Journal of Engineering, Science & Technology*, vol. 13, no. 2, pp. 29–33, 2014.

[60] M. Aissani, S. Bouznad, A. Fareb, and M. A. Laidoui, "EA-SPEED: Energy-aware real-time routing protocol for wireless sensor networks," *International Journal of Information and Communication Technology*, vol. 5, no. 1, pp. 22–44, 2013.

[61] J. Heo, J. Hong, and Y. Cho, "EARQ: energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 3–11, 2009.

[62] L. Zhao, B. Kan, Y. Xu, and X. Li, "FT-SPEED: A fault-tolerant, real-time routing protocol for wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2007*, pp. 2531–2534, chn, September 2007.

[63] S. Saqaeeyan and M. Roshanzadeh, "Improved Multi-Path and Multi-Speed Routing Protocol in Wireless Sensor Networks," *International Journal of Computer Network and Information Security*, vol. 4, no. 2, pp. 8–14, 2012.

[64] M. Kaur and A. Sharma, "Routing in WSN network using neural Network (NN) and SPEED protocol," in *Proceedings of the 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 161–167, Greater Noida, India, December 2016.

[65] H. Thakar, S. Vhatkar, and M. Atique, "Comparative study of speed protocols in wireless sensor network," *International Journal of Computer Applications*, vol. 120, no. 16, pp. 8–13, 2015.

[66] A. Ali Ahmed, "An enhanced real-time routing protocol with load distribution for mobile wireless sensor networks," *Computer Networks*, vol. 57, no. 6, pp. 1459–1473, 2013.

[67] Y. Xu, F. Ren, T. He, C. Lin, C. Chen, and S. K. Das, "Real-time routing in wireless sensor Networks: A potential field Approach," *ACM Transactions on Sensor Networks*, vol. 9, no. 3, Article ID 2480738, 2013.

[68] P. T. A. Quang and D.-S. Kim, "Enhancing real-time delivery of gradient routing for industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 61–68, 2012.

[69] A. Mahapatra, K. Anand, and D. P. Agrawal, "QoS and energy aware routing for real-time traffic in wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 437–445, 2006.

[70] S. Rachamalla and A. S. Kancherla, "A two-hop based adaptive routing protocol for real-time wireless sensor networks," *SpringerPlus*, vol. 5, no. 1, article no. 1110, 2016.

[71] C. Lee, B. Shah, and K.-I. Kim, "An architecture for (m, k)-firm real-time streams in wireless sensor networks," *Wireless Networks*, vol. 22, no. 1, pp. 69–81, 2016.

[72] K. Kim, "A Novel Scheduling for (m, k)-firm Streams in Wireless Sensor Networks," in *Proceedings of International Conference on Networked Computing and Advanced Information Management*, September 2010.

[73] C. Zhao and H. Xiong, "A channel-aware scheduling scheme for (m,k)-firm streams in wireless multimedia sensor networks," *IEICE Transactions on Communications*, vol. E95-B, no. 10, pp. 3312–3315, 2012.

[74] T. Semprebom, C. Montez, and F. Vasques, "(m,k)-firm pattern spinning to improve the GTS allocation of periodic messages in IEEE 802.15.4 networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, article no. 222, 2013.

[75] Y. Matusovsky, *Reliability-Focused Scheduling with (m, k)-firm Deadlines over Wireless Channels - A Reinforcement-Learning Approach*, University of Canterbury, 2016.

[76] T. Semprebom, C. Montez, R. Moraes, F. Vasques, and R. Custódio, "Distributed DBP: A (m,k)-firm based distributed approach for QoS provision in IEEE 802.15.4 networks," in *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, ETFA 2009*, esp, September 2009.

[77] K.-I. Kim, "A revised SPEED protocol for (m, k)-firm streams in wireless sensor networks," in *Proceedings of the International Conference on ICT Convergence, ICTC 2011*, pp. 267-268, kor, September 2011.

[78] K. Kim, "Evaluating an (m, k)-firm deadline real-time stream based on a reliable transport protocol in wireless sensor networks," *Journal of Information and Communication Convergence Engineering*, vol. 10, no. 2, pp. 129–134, 2012.

[79] M. Tekaya, N. Tabbane, and S. Tabbane, "Enhance (m,k)-firm Constraint on the Real Time Streams Applied to AODV Protocol," in *Proceedings of International Conference on Advanced Computing and Communications*, October 2013.

[80] P. Jiang, Q. Huang, J. Wang, x. Dai, and R. Lin, "Research on Wireless Sensor Networks Routing Protocol for Wetland Water Environment Monitoring," in *Proceedings of the First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, pp. 251–254, Beijing, China.

[81] K.-I. Kim and T.-E. Sung, "Network layer approaches for (m, k)-Firm stream in wireless sensor networks," *IEICE Transactions on Communications*, vol. E93-B, no. 11, pp. 3165–3168, 2010.

[82] B. Li and K.-I. Kim, "An (m, k)-firm real-time aware fault-tolerant mechanism in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 905740, 12 pages, 2012.

[83] B. Li and K.-I. Kim, "A real-time routing protocol for (m,k)-firm streams in wireless sensor networks," in *Proceedings of the IEEE 8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing: Sensing the Future, ISSNIP 2013*, pp. 129–134, April 2013.

[84] J. Nam, "Load balancing routing protocol for considering energy efficiency in wireless sensor network," *Advanced Science and Technology Letters*, vol. 44, pp. 28–31, 2013.

[85] K.-I. Kim and T. E. Sung, "Cross-layered approach for (m, k)-firm stream in wireless sensor networks," *Wireless Personal Communications*, vol. 68, no. 4, pp. 1883–1902, 2013.

[86] K.-I. Kim and T.-E. Sung, "Modeling and routing scheme for (m, k)-firm streams in wireless multimedia sensor networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 3, pp. 475–483, 2015.

[87] M. A. Azim, B.-S. Kim, B. Shah, and K.-I. Kim, "Real-time routing protocols for (m,k)-firm streams based on multi-criteria in wireless sensor networks," *Wireless Networks*, vol. 23, no. 4, pp. 1233–1248, 2017.

[88] K. Kim, "Clustering Scheme for (m,k)-Firm Streams in Wireless Sensor Networks," *Journal of Information and Communication Convergence Engineering*, vol. 14, no. 2, pp. 84–88, 2016.

[89] A. Lalomia, G. Lo Re, and M. Ortolani, "A hybrid framework for soft real-time WSN simulation," in *Proceedings of the 13th IEEE/ACM Symposium on Distributed Simulation and Real-Time Applications, DS-RT 2009*, pp. 201–207, October 2009.

[90] J. Rousselot, J.-D. Decotignie, M. Aoun, P. Van Der Stok, R. S. Oliver, and G. Fohler, "Accurate timeliness simulations for real-timewireless sensor networks," in *Proceedings of the UKSim 3rd European Modelling Symposium on Computer Modelling and Simulation, EMS 2009*, pp. 476–481, November 2009.

[91] E. Leão, C. Montez, R. Moraes, P. Portugal, and F. Vasques, "Superframe duration allocation schemes to improve the throughput of cluster-tree wireless sensor networks," *Sensors*, vol. 17, no. 2, article no. 249, 2017.

[92] L. Mottola and G. P. Picco, "Programming wireless sensor networks: fundamental concepts and state of the art," *ACM Computing Surveys*, vol. 43, no. 3, article 19, 2011.

[93] M. AL-Rousan and D. Kullab, "Real-Time Communications for Wireless Sensor Networks: A Two-Tiered Architecture," *International Journal of Distributed Sensor Networks*, vol. 5, no. 6, pp. 806–823, 2009.

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 9124858, 14 pages
https://doi.org/10.1155/2017/9124858

Research Article

Priority-Based Dynamic Multichannel Transmission Scheme for Industrial Wireless Networks

Yuichi Igarashi,[1] Yoshiki Matsuura,[1] Minoru Koizumi,[1] and Naoki Wakamiya[2]

[1] *Hitachi, Ltd., Research & Development Group, Center for Technology Innovation, Tokyo, Japan*
[2] *Osaka University, Osaka, Japan*

Correspondence should be addressed to Yuichi Igarashi; yuichi.igarashi.hb@hitachi.com

Received 31 March 2017; Accepted 15 August 2017; Published 27 September 2017

Academic Editor: Giovanni Pau

Industrial wireless sensor network (IWSN) applications are required to provide precise measurement functions as feedback for controlling devices. Current industrial wireless communication protocols, such as ISA100.11a and wirelessHART, have difficulty, however, in guaranteeing latency for unpredictable on-demand communications. In this paper, a priority-based dynamic multichannel transmission scheme is proposed for IWSNs. In the proposed scheme, a root node controls the transmission timing of high-priority packets, while other nodes autonomously decide what channel to use and when to transmit packets to a neighbor. Simulation results show that real time control is possible where a response delay from transmission of a request to reception of a reply at a root node is within 1,140 ms at per-link communication success probability with a retry of higher than 93%.

1. Introduction

Industrial wireless sensor networks (IWSNs) have been emerging as a new means of communication for social infrastructure applications like Advanced Metering Infrastructure (AMI), Distribution Automation, Optimized Factory, Predictive Maintenance, Building Automation, and so on. These applications basically gather information from remote devices, that is, sensors, check device status or circumstance, and control devices, that is, actuators, based on the gathered information. For remote monitoring devices, the main purpose is periodic collection of device status or sensor data. At the same time, industrial applications also require on-demand communication for data collection and operation of devices by a remote control server, within specific end-to-end deadlines. For instance, an AMI system requires a deadline of 20–60 seconds when a remote control server requests on-demand meter reading. Although such request/reply type of communication is unpredictable, IWSNs must guarantee a maximum communication delay for both periodic and unpredictable packets.

Wireless networks using legacy Media Access Control (MAC) protocols based on Carrier Sense Multiple Access/ Collision Avoidance (CSMA/CA), such as WiFi or Zigbee [1], sometimes drop packets because of interference from other wireless networks. To increase the reliability of wireless networks for industrial applications such as remote monitoring devices, several protocols, such as Wireless HART [2], ISA100.11a standard [3], and IEEE802.15.4e [4], have been developed and standardized.

These standards use Time Division Multiple Access (TDMA) and multiple channel hopping technologies for their MAC protocols. Both WirelessHART and ISA100.11a for industrial applications support a centralized network management architecture in which all nodes in an IWSN are time-synchronized and assigned communication timings for periodic data gathering by a central network manager. It decreases both interference within the IWSN and interference from other wireless networks using the same channels, thus increasing the rate of successful communication.

However, high success rate of packet transmission of periodic communication alone is not sufficient for industrial applications. Assume that an on-demand downward packet from a root node to a sensor and upward packets from sensors to the root node are generated on the same path in a multihop TDMA-based wireless network. They compete for a channel and an opportunity of packet transmission and thus some packets will have to wait until the next transmission opportunity. This may cause quite large accumulated end-to-end delay, making it difficult to guarantee a maximum delay to applications. To mitigate delay, one of solutions is to assign transmission timings to all nodes for unpredictable communication. However, because of unpredictability, such assignment is very likely to be redundant and as a result precious network bandwidth will be considerably wasted.

To guarantee a deadline for such unpredictable packets, we propose a priority-based dynamic multichannel transmission scheme. Our scheme prioritizes packets in accordance with application requirements. Packet transmission is scheduled in a slotted manner, but detailed slot allocation as in usual TDMA-based protocols is not performed. More specifically, a root node only determines both when it transmits on-demand request packets for remote control and when nodes transmit data packets for periodic monitoring. On the other hand, packet forwarding is not scheduled at all. In a slot, which we call SlotFrame, priority CSMA/CA-like packet transmission is performed at each node. Our scheme operates over a MAC layer and does not rely on any specific MAC protocol. We discuss compatibility with ISA100.11a in Section 6.2.

We consider one type of packets for periodic communication and three types of packets for unpredictable communication. We first define three priorities of packets depending on their type and then assign one dedicated channel to each priority. The highest priority is given to unpredictable and on-demand packets that a root node sends to a node for control or information retrieval, that is, downward packets. Upward reply packets are also given the highest priority, because request/response communication for device control requires a very short end-to-end delay. The second priority is given to periodic packets used for regular data collection. Network control packets are then set to the lowest priority. In our proposal, a root node can thus transmit the highest-priority packets at any time but still control the transmission timing through centralized administrative control, as in ISA100.11a. More specifically, a node replies to a request packet at the time specified by the root node.

On the other hand, each node decides the time of forwarding a packet by autonomous decentralized radio channel control. A node scans three communication channels in descending order of priority and dynamically decides which channel to use. For example, when a node having a reply packet finds that a request packet is to be sent by a neighbor, it defers transmission of the packet for a certain duration of time to avoid collision among downward and upward packets over the high-priority channel. If there is no transmission of high-priority packets in the vicinity, a node can transmit a periodic data packet using another channel for the middle priority. Only when there is no packet transmission on either high or middle priority channel, a node can transmit a network control packet. This mechanism ensures that the highest-priority packets are preferentially transferred without unexpected delay.

The contribution of this paper is proposing a priority-based multichannel transmission scheme that determines when and what packets should be transmitted on which channel. Through simulation, we validate our proposal for two industrial applications: AMI and industrial process monitoring and control. We also theoretically estimate the lower bounds of available bandwidth for middle- and low-priority packet transmission. Moreover, TDMA-based protocols like ISA100.11a typically have a scheduler for allocating network resources such as time slots to all nodes. This scheduling process is often complicated and the scheduler has to deliver the information to all nodes whenever a new node joins the network or a network topology changes. In contrast, since our scheme only determines when to generate and transmit a packet at a root node for remote control and at nodes for periodic monitoring, it does not need to adjust a schedule as far as the maximum number of hops and the number of nodes do not change. We discuss this advantage in more detail in Section 6.3.

Table 1: Application and system requirements.

	920-MHz band	2.4-GHz band
Applications	(i) AMI (ii) Distribution automation	(i) Optimized factory (ii) Predictive maintenance
Application processing	(i) Remote monitoring (ii) Remote operation	(i) Remote monitoring (ii) Remote operation
Communication type	(i) Publish/subscribe (ii) Request/response	(i) Publish/subscribe (ii) Request/response
Remote monitoring cycle	30 min (periodic data)	1–5 min (periodic data)
Maximum delay for remote operation	20–60 sec (unpredictable data)	5–20 sec (unpredictable data)
Number of nodes in wireless network	1–500 nodes	1–500 nodes
Packet length	500–600 B	90 B
Communication speed	50–100 kbps	250 kbps
Protocols	(i) IEEE802.15.4g/e (ii) 6tisch (iii) 6LoWPAN	(i) ISA100.11a (ii) WirelessHART

The rest of the paper is structured as follows. We first describe requirements and challenges in Section 2, and Section 3 presents an overview of related work. In Section 4, we propose the priority-based transmission scheme. Then, we evaluate the communication delay for highest-priority packets and available bandwidth for other packets in Section 5. In Section 6, we discuss compatibility with ISA100.11a and overhead incurred in implementing our proposal before discussing our conclusions and future work in Section 7.

2. Requirements and Challenges

2.1. System Requirements from Industrial Applications. Table 1 lists a summary of industrial applications and system requirements for IWSN [5–9]. AMI, mentioned in Section 1, has been intensively deployed in Japan since the Great East Japan

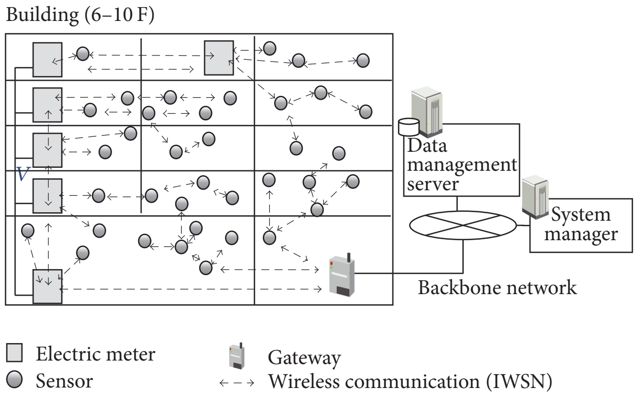

Figure 1: Target system configuration.

Earthquake in 2011. Industrial manufacturing companies have recently faced a need to increase productivity and optimize factories. To achieve this, IWSN systems require a variety of data and means of field data collection, like sensor data from factories and buildings, smart meters, and so on. To collect such large data, standard Internet of things (IoT) wireless technologies have provided large, dense wireless networks that contain several hundred nodes and form at most 8–16 multihop networks. However, collection of periodic data with high success in communication is necessary [7, 8].

Moreover, unpredictable and on-demand communication occurs in an IWSN. For example, eco-friendly systems such as distribution automation and demand response systems have been deployed in order to use energy efficiently. These systems request communication of contingency packets from a remote server to an end device in an unpredictable way. Industrial applications such as industrial process control [5] also require controlling devices remotely with a maximum delay of several seconds. Figure 1 shows an example of a target system configuration for the application of monitoring and controlling a building.

2.2. Challenge. The packet error rate (PER) is normally one of the most important parameters for evaluating reliability of a wireless network. In addition, guaranteed deadlines should be considered for IWSNs, because industrial applications require real time processing. Several wireless protocols have been already standardized and developed for industrial applications, including WirelessHART and ISA100.11a. To decrease the PER for collecting periodic data from sensors in dense and lossy wireless networks, these standards use TDMA-based MAC protocols. Such protocols overcome the problem of packet collisions in the network. In fact, the data collection ratio for WirelessHART reaches more than 99%, because SmartMesh WirelessHART devices basically perform retransmission twice at most [10]. This performance seems sufficiently high for remote monitoring purposes.

On the other hand, when an unpredictable on-demand packet has to be sent to or from a node, the packet either consumes assigned bandwidth or waits several seconds until the next assigned bandwidth becomes available. This can cause random latency, which further depends on TDMA scheduling, retransmission timing, and wireless radio conditions. Most WSNs do not support real time communication [11] and it is difficult for even TDMA-based MAC protocols such as IEEE802.15.4e to support real time communication for large scale networks like AMI [12].

As described below in Section 3, when we use normal ISA100.11a for both remote monitoring and remote control of devices, it does not guarantee the latency of on-demand and multihop communication at any time, although it can transfer a higher-priority packet by applying the priority CSMA/CA scheme among single-hop neighbors. If such on-demand communication can be expected, then a system manager with an optimized scheduler may enable ISA100.11a to allocate communication timing for all nodes in order to transfer a higher-priority packet within a certain delay and with due consideration to maintain a high data collection ratio. Whenever the network topology is changed, because of the instability of the radio environment, however, the schedule must be updated, so this is an unrealistic solution.

As another solution to the problem, we could use multiple ISA100.11a network functions on different network interfaces, that is, one ISA100.11a function for remote monitoring and another for remote control, for example. In this case, each function would concentrate on scheduling transmission of a single application packet with a high end-to-end path success probability. Unfortunately, this approach faces the same problem that a system manager must deliver an optimized schedule to all nodes whenever the network topology is changed. Moreover, each node would have to control multiple ISA100.11a network functions precisely, but the standard does not describe how an application can manage multiple ISA100.11a networks.

Therefore, our challenges in this paper are to mitigate unexpected latency for unpredictable and high-priority IWSN communications and to show how to meet system requirements for high end-to-end success probability of periodic communication.

3. Related Work

In wireless sensor networks, MAC is a key technology that determines channel access delay and utilization. MAC protocols are roughly classified into three types: contention-based, contention-free, and hybrid.

First, contention-based schemes (using CSMA/CA) such as IEEE802.15.4 determine transmission timing by checking existence of carrier signals, that is, carrier sense. When a network is large or dense, the PER is normally high and as such CSMA/CA-based MAC protocols cannot guarantee latency [13–15].

Second, contention-free MAC protocols using TDMA implement scheduled communication with a centralized coordinator, such as a network manager. In TDMA-based MAC protocols, a node transmits and forwards a packet to a neighbor according to an allocated time slot schedule. When packet transmission fails, a node should wait until the next assigned time slot to resend the packet. Therefore, the end-to-end delay depends on the whole schedule and its cycle

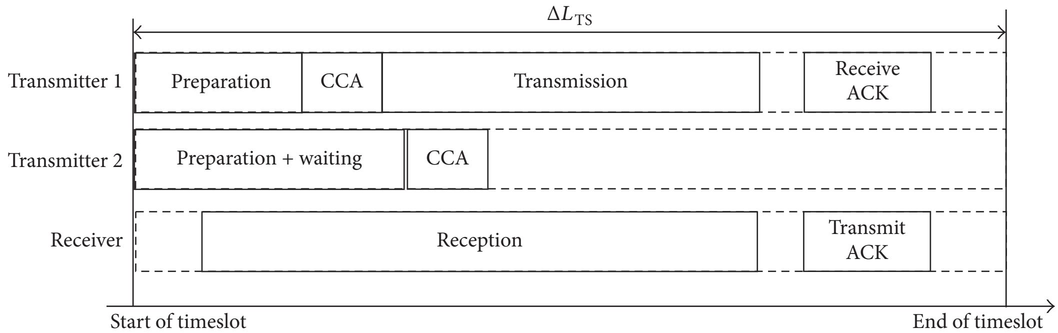

Figure 2: A shared time slot using the CSMA/CA technology of ISA100.11a.

length called superframe. To reduce latency in industrial networks, Saputra and Shin proposed a scheduling scheme for ISA100.11a superframes [16]. This scheme specifies how to build a superframe to guarantee the delay for periodic upward packets from sensors to a root node and how to check the schedulability of a superframe.

Finally, IWSNs often adopt hybrid schemes [2, 3, 17, 18]. While the hybrid standard schemes such as ISA100.11a and WirelessHART use a TDMA-based MAC protocol, they also provide periodic data communication at low PER. At the same time, these schemes adopt a CSMA/CA-based MAC protocol for unpredictable transmission requirements, such as network control packets, alert information, on-demand requests, and retransmission of data packets.

During a CSMA/CA period in a hybrid scheme, ISA100.11a nodes can use a priority CSMA/CA scheme, as shown in Figure 2. Waiting time proceeding to transmission of a high-priority packet is shorter than that of a low-priority packet as shown in Figure 2, where transmitter 1 has a high-priority packet and transmitter 2 has a low-priority one. Because of difference in waiting time, transmitter 2 can detect transmission of a high-priority packet during its CCA (Clear Channel Assessment) and stop the transmission attempt. This scheme enables priority control within single-hop communication and decreases the probability of collision among transmission of packets of different priority [19]. We also use this priority CSMA/CA-like scheme in our approach.

4. Priority-Based Transmission Scheme with Dynamic Channel Shift

In this section, we provide assumptions and terminologies at first. Then, we present an outline of our priority-based transmission scheme with dynamic channel shift. We also give detailed algorithms for priority-based channel selection and the transmission and reception mechanisms.

4.1. Assumptions

Industrial Applications Features. As noted above, our target applications are AMI, Distribution Automation, Optimized Factory, and so on. These typical industrial applications normally collect field data and store them at a remote server like cloud. Industrial applications often involve real time communication to react or respond to user queries within a predetermined deadline in order to perform timely control and avoid failures. Since most of existing WSNs cannot satisfy the requirement, we propose a scheme to provide real time communication over a large scale IWSN.

MAC Protocols. Recently, many new industrial wireless systems have been deployed. Most of them use TDMA-based protocols such as ISA100.11a, wirelessHART, and IEEE802.15.4e/g based protocols in order to avoid interference among internal nodes and keep communication success probability high. Those protocols provide multihop and time-synchronized networks that consist of a central manager and other nodes that are synchronized with the central manager. We assume that our targeted network is also multihop and time-synchronized, but our proposal operates over a MAC layer to decide what channel to use and when to transmit packets and does not rely on any specific MAC protocols.

Network Topology and Its Condition. Similarly to other TDMA-based WSN protocols, we also assume a tree topology whose root is a central manager. Our proposal does not specify any routing protocols as far as a stable tree-based routing topology is established and maintained for a large scale WSN. In simulation experiments, we consider a network of 500 nodes with 8 hops for 920 MHz and 16 hops for 2.4 GHz at maximum.

Priority Level of Packets. In an IWSN, multiple applications would simultaneously operate such as periodic data gathering and remote control. In addition, networking functions such as routing and time synchronization are also running. Among them, remote control is the most crucial and must be given the highest priority to guarantee real time communication. Furthermore, its responses from nodes to a root node should have the higher priority than those packets belonging to periodic data gathering. Although frequent loss of control packets affects stability and reliability of a WSN, a best-effort service is enough. We evaluate the lower bound of available bandwidth for lower priority packets in Section 6. Details of prioritization will be given in the next subsection.

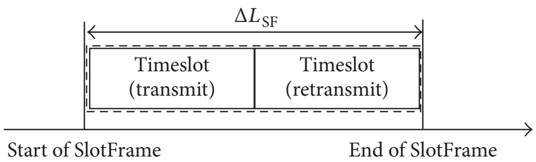

Figure 3: A composition of a SlotFrame.

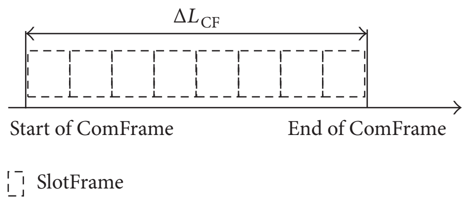

Figure 4: A composition of a ComFrame.

Multiple Communication Channels. TDMA-based MAC protocols for IWSNs have channel hopping functions to enable coexistence of multiple networks in the same area and dynamic bandwidth allocation. In this paper, we assume that three channels are available to use.

4.2. Terminologies. We define our terminologies as follows.

4.2.1. Frame Composition. First, we define three types of frames over a MAC protocol.

The first type is the *SlotFrame*, which consists of two timeslots as shown in Figure 3. Both ISA100.11a and IEEE802.15.4e technologies divide time into timeslots of configurable length, with typical durations ranging from 10 to 14 ms. These technologies do not, however, support MAC layer retransmission within a timeslot. The SlotFrame enables transmission of a packet within the 1st timeslot and retransmission within the 2nd timeslot. ΔL_{SF} denotes the length of a SlotFrame, for example, 20 to 28 ms.

The second type is the *ComFrame*, which consists of SlotFrames, as shown in Figure 4. The number of SlotFrames in a ComFrame is calculated as the maximum hop count in a multihop wireless network plus 3. In ISA100.11a, a root node knows the whole network topology. The number "3" is a key number that was chosen to avoid hidden terminal problems on a multihop route, as described in detail later. In this paper, we assume that the number of SlotFrames in a ComFrame is 11 (8 hops + 3). In addition, through centralized administrative control, a root node assigns a ComFrame to a node when it joins the network. The assignment does not change even if the network topology changes.

The final frame type is the *AppFrame*, which consists of ComFrames on multiple channels, as shown in Figure 5. The number of ComFrames in an AppFrame is a system parameter that depends on the application requirements. ΔL_{AF} denotes the length of a AppFrame. For example, if an application remotely operates all devices in 30 min and collects data from all devices in 30 min, then $\Delta L_{AF} = 30$ min. In Figure 5, the network manager divides the AppFrame to two blocks. In this example, block 1 is used for controlling all nodes ($L1$ and $L2$ packets), collecting data from all nodes ($L3$ packets), and transmitting network control packets ($L4$ packets). Other blocks, such as block 2 in Figure 5, are used for bidirectional communication ($L1$ and $L2$ packets) needed for repeat attempts at remote control or data collection from devices, and for transmitting network control packets ($L4$ packets). A system manager determines the number of blocks in an AppFrame.

4.2.2. Priority Level and Communication Channels. As mentioned above, we define 4 priority levels. The highest level ($L1$) is for downward packets from a root node to a sensor node (end device) that an application controls. The second ($L2$) is for upward packets in response to $L1$ packets. The third ($L3$) is for periodically collected data transferred from an end device to a root node, for example, a network health report or sensing data. The lowest priority level ($L4$) is for network control packets, for example, routing packets, time synchronization packets, or beacon packets. In our proposed scheme, $L1$ and $L2$ packets are transferred over communication channel 1 (Ch1), $L3$ packets are transferred over channel 2 (Ch2), and $L4$ packets are transferred over channel 3 (Ch3). Here, Ch1 and Ch2 are contention-free channels like TDMA-based communication, whereas Ch3 is a contention-based channel like CSMA/CA-based communication.

4.3. Outline. We give an outline of how our scheme simultaneously fulfills several requirements of industrial wireless communications: a guaranteed deadline for on-demand communication, data collection at low PER, and communication of network control packets among neighbors.

In our scenario, there are three kinds of packets. The first kind is unpredictable packets for on-demand control. The second is periodic packets generated by sensors for periodic data collection. The third is network control packets that build multihop routes from sensors to a root node and exchange time information for synchronization among nodes.

We first rank packets according to industrial application requirements. To provide a guaranteed deadline, we define an on-demand downward packet from a root node to a sensor to have the highest priority ($L1$) and an on-demand upward reply packet from a sensor to a root node to have the second-highest priority ($L2$). The third priority ($L3$) is for periodic data collection packets from any sensor, and the lowest priority ($L4$) is for network control packets.

In addition, our priority-based dynamic multichannel transmission scheme uses three communication channels. The $L1$ and $L2$ packets between a root node and sensor nodes share a communication channel (Ch1). The periodic $L3$ packets use another communication channel (Ch2) for a certain period of time, and the $L4$ packets use a third channel (Ch3). In other words, a root node sends an on-demand request packet with priority $L1$ while waiting to receive a reply packet ($L2$) for a previous request packet. During the same period, a sensor node sends a periodic data collection packet ($L3$) to a root node on Ch2. The timing for a sensor node to transfer such a periodic packet to a root node is decided by the

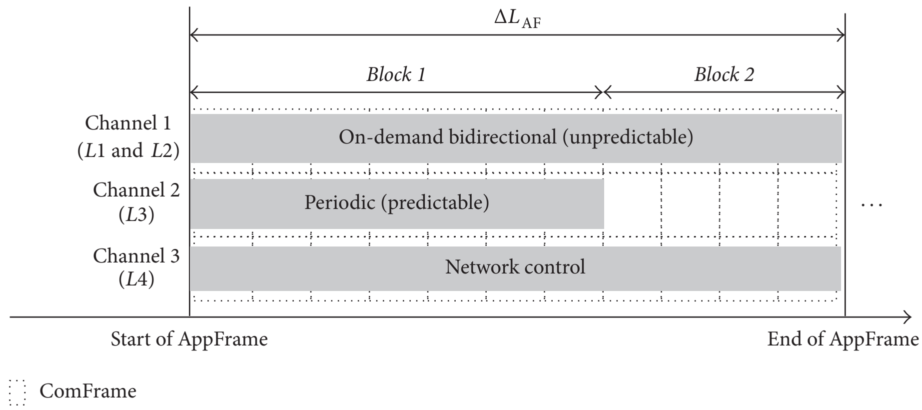

Figure 5: A composition of a AppFrame.

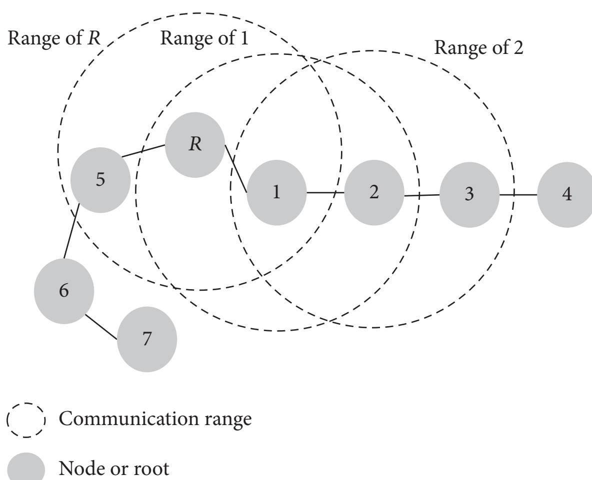

Figure 6: A simple example of network topology.

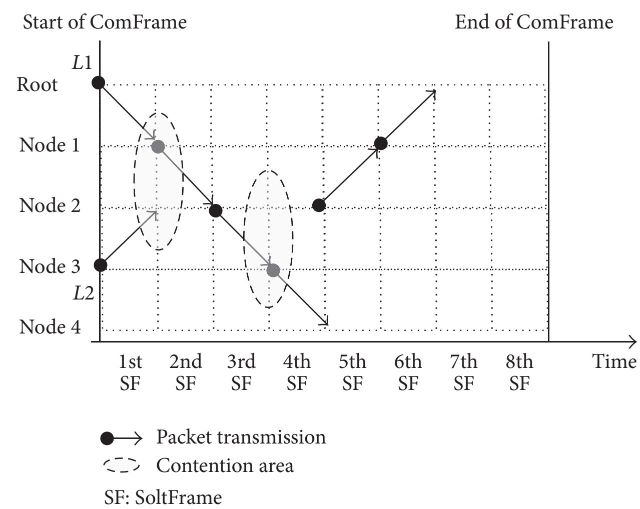

Figure 7: An example of packet flow over highest priority channel (Ch1).

root node when the sensor node joins the network. Network control packets ($L4$) should be transferred only when no neighbors have to transfer higher-priority packets. In each SF, nodes scan channels in order of priority. When a node detects any packet is being transmitted in a higher-priority channel, it stays at the channel and receives the packet. Otherwise, it moves to a lower priority channel and checks the existence of packets. We describe details later in this section.

4.4. Example of Priority-Based Dynamic Multichannel Transmission Mechanism. We next provide an example of how to ensure preferential communication of a downward packet from a root node to an end device ($L1$) and how to avoid contention between a downward packet and an upward packet ($L2$) in response to a previous downward packet. As noted above in Sections 4.1 and 4.3, the order of the priority is predetermined and all nodes share the information. Figure 6 shows a simple example of a network topology. The network consists of 8 nodes and has a maximum of 4 hops. Figure 7 shows an example of packet flow, in which the root generates an $L1$ packet and node 3 generates an $L2$ packet. At the 2nd SlotFrame in the ComFrame, node 2 cancels forwarding of the $L2$ packet to node 1. Then, node 2 waits two SlotFrames to avoid collisions due to the hidden terminal problem. While node 2 waits, the $L1$ packet is delivered to node 4 without any delays. Then, the $L2$ packet is eventually transferred to the root node at the 6th SlotFrame. This crossed-transfer mechanism guarantees a maximum latency for transmission of the highest-priority information.

In addition, our scheme uses a dynamic channel shift mechanism to communicate information about other priority levels, as shown in Figure 8. As noted above in Section 4.1, our proposal uses three communication channels and nodes share the number of channels and the order of scanning channels as well as packet priority. All nodes choose Ch1 ($L1$ or $L2$) at first. Then, if they do not detect any packets over Ch1 during ΔL_{wait}, they move to Ch2 and scan the channel again. For example, in Figure 6, nodes 6 and 7 shift from Ch1 ($L1$ or $L2$) to Ch2 ($L3$), and node 4 does not detect any packets over Ch2 and so shifts to Ch3 ($L4$), while the other nodes stay at Ch1. For this example, Table 2 summarizes the

Table 2: An example of SlotFrame usage in a ComFrame.

	1st SF	2nd SF	3rd SF	4th SF	5th SF	6th SF	7th SF
Root	**Ch1 (L1) R → 1**	Ch3	Ch2 (*L*3) 5 → *R*	Ch3	Ch3	Ch1 (*L*2) 1 → *R*	Ch3
Node 1	Ch1 (*L*1) *R* → 1	Ch1 (*L*1) 1 → 2	Ch3	Ch3	Ch1 (*L*2) 2 → 1	Ch1 (*L*2) 1 → *R*	Ch3
Node 2	Ch1 (*L*2) 3 → 2	Ch1 (*L*1) 1 → 2	Ch1 (*L*1) 2 → 3	Ch3	Ch1 (*L*2) 2 → 1	Ch3	Ch3
Node 3	**Ch1 (L2) 3 → 2**	Ch3	Ch1 (*L*1) 2 → 3	Ch1 (*L*1) 3 → 4	Ch3	Ch3	Ch3
Node 4	Ch1	Ch3	Ch3	Ch1 (*L*1) 3 → 4	Ch3	Ch3	Ch3
Node 5	Ch1	Ch2 (*L*3) 6 → 5	Ch2 (*L*3) 5 → *R*	Ch3	Ch3	Ch3	Ch3
Node 6	Ch2 (*L*3) 7 → 6	Ch2 (*L*3) 6 → 5	Ch3	Ch3	Ch3	Ch3	Ch3
Node 7	**Ch2 (L3) 7 → 6**	Ch3	Ch3	Ch3	Ch3	Ch3	Ch3

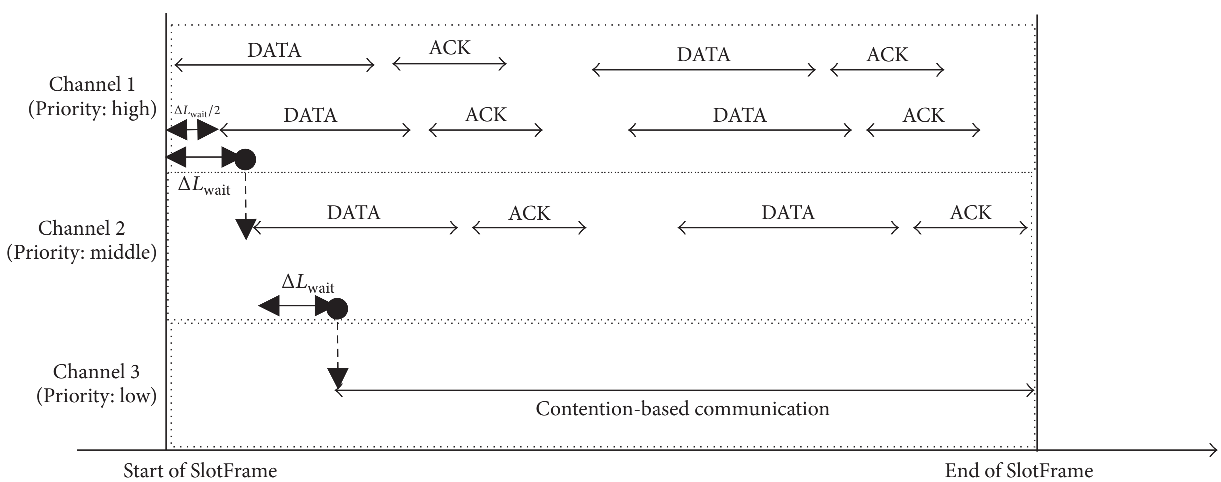

Figure 8: Outline of dynamic channel shift.

channel usage for all nodes and SlotFrames. Most TDMA-based protocols predetermine such a complete and detailed schedule as Table 2 and send it to all nodes to follow the same schedule. On the other hand, emission of a request packet from a root node to node 4 and that of a data packet from node 7 to a root node are predetermined, but other detailed slot and channel usages are autonomously and dynamically decided by our dynamic channel shift mechanism. Since control packets belonging to *L*4 use remainder of network resources, we evaluate the available bandwidth for *L*4 in Section 5.

4.5. Detailed Mechanism

4.5.1. Transmission Policy. Every node transmits a packet according to its SlotFrame usage policy and ComFrame usage policy.

4.5.2. SlotFrame Usage Policy. All nodes select a communication channel for each SlotFrame by a dynamic multichannel transmission mechanism. Then, nodes transmit *L*1, *L*2, and *L*3 packets over Ch1 or Ch2 as shown in Figure 8. They can also retransmit a packet once per SlotFrame according to the retransmission policy below. Nodes transmit *L*4 packets by CSMA/CA over Ch3.

4.5.3. ComFrame Usage Policy. A root node uses the 1st or 2nd SlotFrame to transmit an *L*1 packet to a node (final destination) in a ComFrame. It first checks the hop count to the final destination in the current network topology and the hop count of an *L*1 packet in a previous ComFrame. When the hop count of the previous *L*1 packet is even, the root node uses the 2nd SlotFrame to avoid packet collisions due to hidden terminal problems on a multihop route. A node transmits an *L*2 packet at the 1st SlotFrame when it received

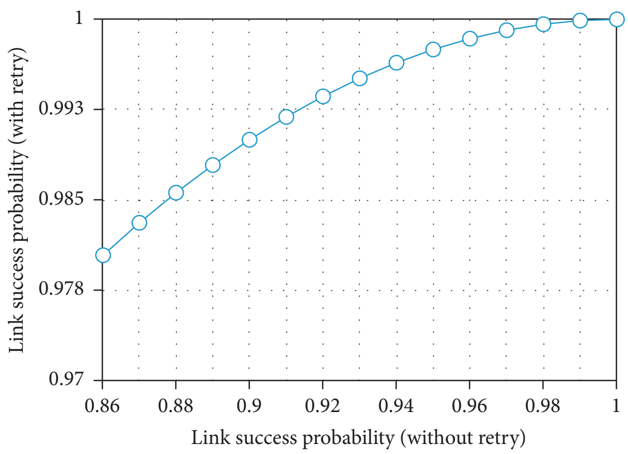

Figure 9: Communication success probability comparison in a SlotFrame between one hop communication with retry and without retry.

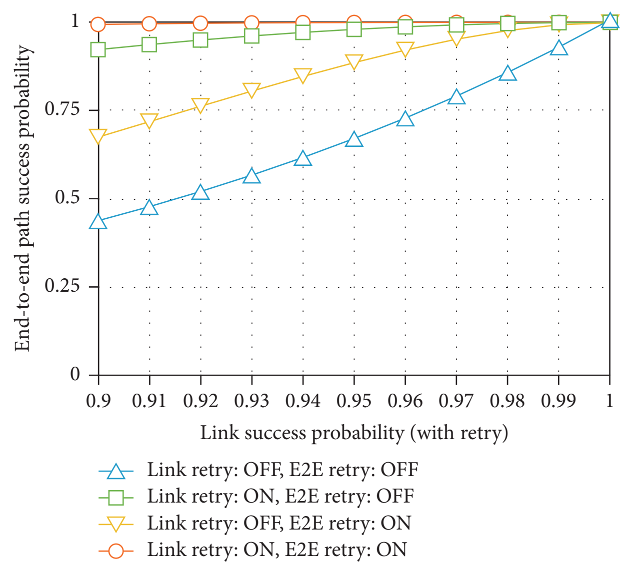

Figure 10: End-to-end path success probability.

an $L1$ packet in a previous *ComFrame*. Thus, an $L1$ packet and an $L2$ packet are transmitted in the same ComFrame. Regarding $L3$ packets, the root node notifies a node of a ComFrame to use for $L3$ packet transmission when a node joins the network. Each node transmits an $L3$ packet at the 1st SlotFrame of its own ComFrame.

Since the length of ComFrame is large enough for a packet sent by a node at any hop distance to reach a root node, ComFrame assignment can be maintained and fixed as far as the maximum hop count does not increase.

4.5.4. Retransmission Policy. The length of a time slot in a TDMA scheme like ISA100.11a is just long enough for a MAC frame of maximum size and its acknowledgement (ACK). Normally, TDMA schemes do not permit any retries in a time slot. For lossy networks, however, link quality (i.e., the communication success ratio) is significantly improved by permitting a node to send a retry packet, as shown in Figure 9. In our proposal, transmission of a retry packet is permitted for every one hop communication of $L1$, $L2$, and $L3$ packets.

Figure 10 shows a comparison of successful path transmission probabilities among the following four retransmission policies: the first policy does not support retry for either link communication or end-to-end communication; the second one supports link retry but not end-to-end retry; the third one supports end-to-end retry but not link retry; and the fourth policy supports both link and end-to-end retry. The figure shows that both retry types are effective even if the retry is only attempted once.

4.5.5. Packet Forwarding Policy. As described above for the SlotFrame usage policy, our proposed scheme does not allocate the intermediate SlotFrames of all ComFrames. For example, only the 3 bold SlotFrames are allocated in advance in Table 2. Every node basically seeks to forward a packet received at a previous SlotFrame with a dynamic channel shift for the transmitting process rule when it does not detect any higher-priority packets.

4.5.6. Reception Process with Dynamic Channel Shift. A root node transmits an $L1$ packet to a node at the 1st or 2nd SlotFrame in each ComFrame. As shown in Figure 11, a nonroot node first checks Ch1 over a period of ΔL_{wait}. If it detects an $L1$ packet, it maintains the channel to receive the packet. It determines the priority level of a packet by detecting the timing. If the timing is $0 \le \Delta L_{\text{wait}}/2$, the packet is treated as $L1$; otherwise, it is treated as $L2$. After searching Ch1, the node checks Ch2 over another period of ΔL_{wait}. If it detects a packet, it maintains the channel to receive the packet as $L3$. Otherwise, it chooses Ch3 as the communication channel at the SlotFrame.

4.5.7. Transmission Process with Dynamic Channel Shift. For transmission all nodes must check for packet existence over channels in order of priority until reaching the usage channel, as in the reception process with dynamic channel shift. The transmission process works as follows by packet priority level:

$L1$: A root node (network manager) knows the current network topology and the hop count of a node that is the destination of a previous $L1$ packet. If the hop count is even, the root node cancels transmission of an $L1$ packet at the 1st SlotFrame and reserves the 2nd SlotFrame in order to avoid the hidden terminal problem on the path.

$L2$: A node that transmits an $L2$ packet to a root node checks for packet existence over Ch1 for a period of $\Delta L_{\text{wait}}/2$. If it does not detect any packets, it transmits an $L2$ packet over Ch1.

$L3$: A node checks for packet existence over all channels in order of priority until Ch2, as in the reception process with dynamic channel shift. If the node detects no higher-priority packets, it transmits an $L3$ packet over Ch2. Otherwise, it cancels transmission of the $L3$ packet at the current SlotFrame and reserves

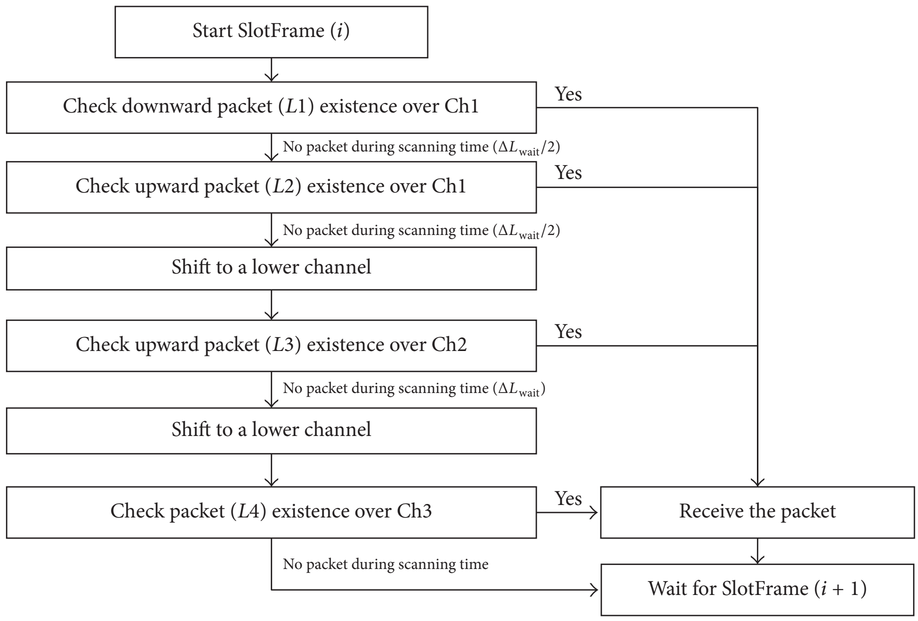

Figure 11: Outline of reception process with channel shift.

the next SlotFrame when the number of remaining SlotFrames in the ComFrame is greater than the hop counts.

*L*4: A node checks for packet existence over all channels in order of priority until Ch3, as in the reception process with dynamic channel shift. If the node detects no higher-priority packets, it transmits an *L*4 packet over Ch3 by CSMA/CA.

4.5.8. Forwarding Process with Dynamic Channel Shift. All nodes must check for packet existence, as in the transmission process with dynamic channel shift. The forwarding process works as follows by packet priority level:

*L*1: A node forwards a packet received at a previous SlotFrame to the next-hop node.

*L*2: A node checks for packet existence over Ch1 for a period of $\Delta L_{\text{wait}}/2$. If the node does not detect any packets, it forwards an *L*2 packet over Ch1. Otherwise, it cancels forwarding of the *L*2 packet at the current SlotFrame and reserves the next 3 SlotFrames.

*L*3: A node follows the *L*3 behavior in the transmission process with dynamic channel shift.

*L*4: A node follows the *L*4 behavior in the transmission process with dynamic channel shift.

5. Simulation Evaluation

5.1. Simulation Settings. To evaluate the performance impact of our priority-based dynamic multichannel transmission scheme, we performed a set of simulations with 501 nodes placed statically and randomly in a square field. A root node was placed at the lower left-hand corner of the field, and a routing protocol for low-power and lossy networks (LLNs) [20] was applied to create routes from all nodes to the root node with a shortest-path metric. Figure 13 shows an example of the resulting network topology. This served to simulate our target system, like that shown in Figure 1. The network topology was fixed during a simulation, and a total of 10 network topologies were tested to take into consideration localization of sensors. We also assume that packet loss among neighbors is caused by several factors such as propagation models, signal processing technology, transmitting power, antenna characteristics, and reception sensitivity, except signal interference from other nodes because of TDMA-like transmission. We then determined the link PER at random as shown in Table 3. Although the link PER dynamically changes in reality, we assume it is stable and constant in this paper. Evaluation under dynamic environment is left as future work.

The network is subject to three traffic packets: request/response type traffic from a root node as unpredictable packets, sensor-to-root traffic as periodic packets, and network control traffic that exchanges information among neighbors. In our proposal, transmission of *L*1 and *L*2 packets for remote control and response is scheduled by a root node to guarantee real time communication. On the other hand, *L*3 packets for periodic data gathering are emitted at predetermined intervals and *L*4 control packets are generated irregularly. Therefore, the worst case scenario is that all of those packets are generated in a certain short period.

In this paper, we evaluate the worst case performance. More specifically, we define an AppFrame accommodating three traffic classes as shown Figure 12. An AppFrame consists of three blocks. The first block is used for a root node to send requests (*L*1) to all 500 nodes for remote control. Responses (*L*2) from nodes are also accommodated in the

Table 3: Simulation conditions.

Item	Notation	Value (920 MHz)	Value (2.4 GHz)
Number of nodes	N_{node}	500 nodes	500 nodes
$L2$ packet length	L_{data}	500 B	127 B
$L2$ ACK length	L_{ack}	100 B	40 B
Communication speed	v	100 kbps	250 kbps
Data collection cycle	Δ_{AF}	30 min	5 sec
Max hop count	$H_{\max}$	8 hops	16 hops
Average hop count	H_{ave}	30 hops	8 hops
Link PER with retry	Per	0–10%	0–9%
Times lot length	ΔL_{TS}	100 ms	10 ms
Wait time for channel shift	ΔL_{wait}	5 ms	1 ms

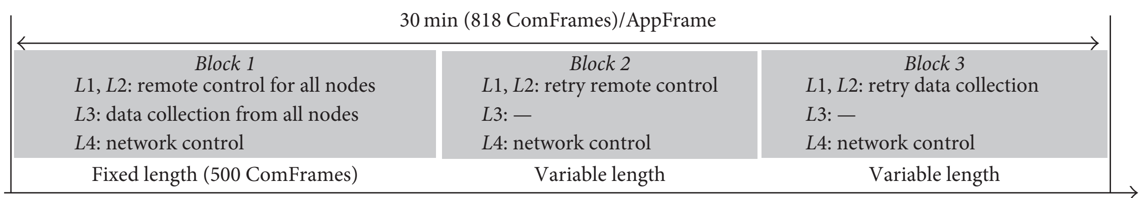

Figure 12: AppFrame composition for simulations.

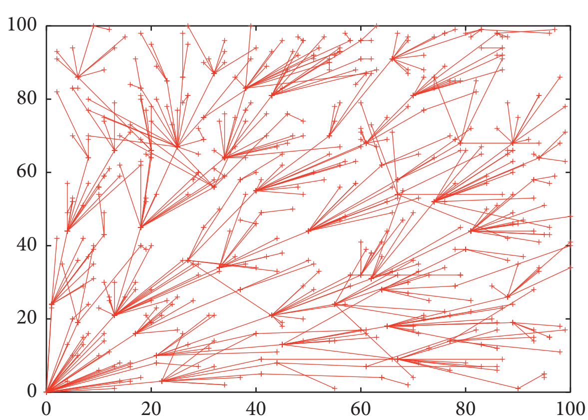

Figure 13: Simulated network topology ($N_{\text{node}} = 500$).

same block. Block 1 is also used for periodic packets ($L3$) and control packets ($L4$). The length of Block 1 is the same as the number of nodes in ComFrames. The second block is used for retransmission of requests to those nodes from which a root node does not receive any response in Block 1. On the contrary, polling-based retransmission of periodic packets is deferred to Block 3, because periodic data gathering is more delay tolerant than remote control. A root node sends a request to resend a data packet to each node from which it fails in receiving a report in Block 1. At this time, requests and responses are given priorities $L1$ and $L2$, respectively. Control packets ($L4$) are irregularly generated in both Block 2 and 3 as in Block 1. Table 3 summarizes the details of the other parameter settings.

5.2. Simulation Results

5.2.1. End-to-End Delay of High-Priority Packets (L1 and L2). All packets are transmitted by ComFrame. At a time when an application queues an $L1$ packet but ComFrame n is already in process, the $L1$ packet should stay in the queue until the head of the next ComFrame $n+1$. The request is transmitted to the destination node at ComFrame $n+1$, and the root node receives the reply packet from the destination node at ComFrame $n+2$. Therefore, the following defines the range for the end-to-end delay time:

$$2 \times \Delta L_{\text{CF}} \leq \text{Delay}_{\text{E2E}} < 3 \times \Delta L_{\text{CF}}. \tag{1}$$

In the 920-MHz simulation case, the end-to-end delay was less than 6.6 sec (=3 × 11 SlotFrames × 200 ms), while in the 2.4-GHz case, it was 1,140 ms (=3 × 19 SlotFrames × 20 ms). Our proposal guarantees the deadline for remote operation, and these simulation results meet our target requirements, as listed in Table 1.

Figure 14 shows end-to-end delay comparison. We conducted field experiments to obtain delay samples of WirelessHART. In the experiments, a root node of WirelessHART received 90.2% packets (5,481 packets (received)/6,088 packets (total)) from nodes and the delay considerably fluctuates. The average delay was 1.309 sec. The theoretical maximum delay of our proposal in the similar condition is 1.14 sec and smaller than the average delay of WirelessHART. To derive this, we substitute the average link success probability of 95% in the experiment to (1).

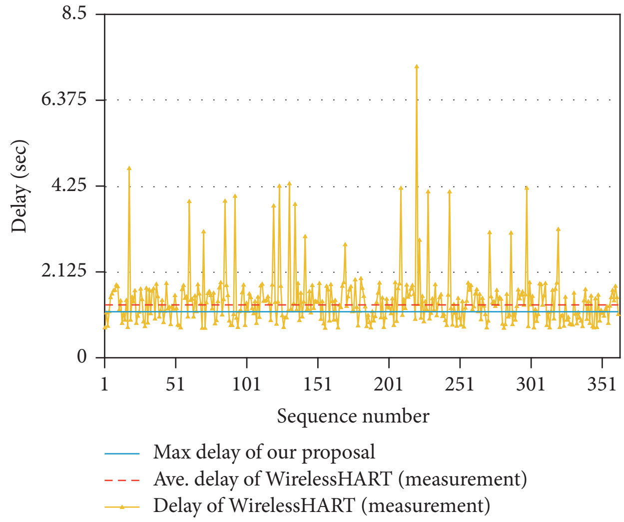

FIGURE 14: End-to-end delay comparison.

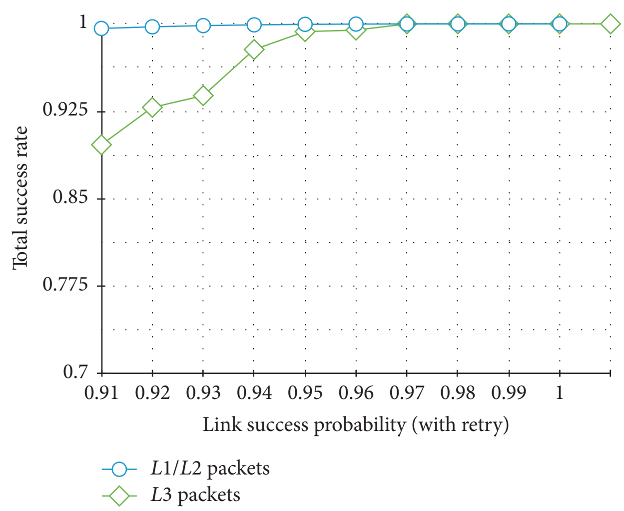

FIGURE 15: Total number of nodes from which the root node successfully received periodic data in an AppFrame.

5.2.2. Success Rate of High-Priority Packets. Figure 15 shows the simulation results for the success rate of high-priority packets. The root node received the highest-priority packets ($L1/L2$) from all nodes when the link success probability (with retry in a SlotFrame) was greater than 93%. It also received all health report data ($L3$) from almost all nodes when the link success probability (with retry) was greater than 96%. The important points here are that our scheme guarantees the maximum delay for getting information from a node and provides a high success rate for getting packets of different priority at the same time.

5.2.3. Available Bandwidth for L4 Packets. Figure 16 shows the simulation results for the ratio of the available $L4$ channel usage time in which a node can totally transmit $L4$ packets in an AppFrame to the length of the AppFrame. According to the figure, the sum of the $L4$ packet bandwidth from Block 1

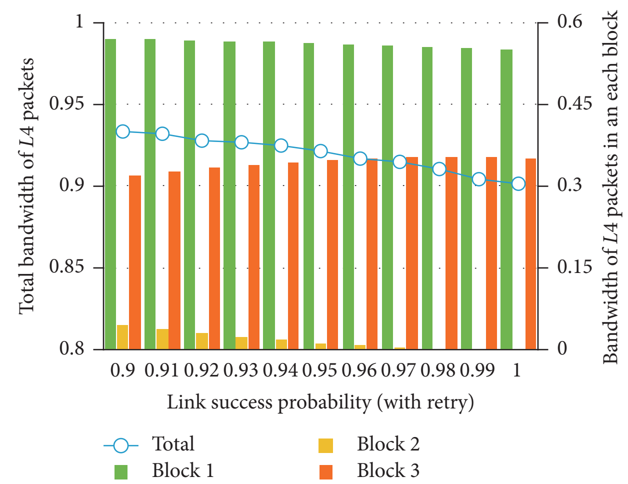

FIGURE 16: Total available bandwidth for $L4$ packets and breakdown by block.

to Block 3 was almost 90%. Although up to three instances of higher-priority traffic were generated in Block 1, the impact of the traffic was very limited. Overall, our scheme provides sufficient bandwidth, because WirelessHART requires about 30% of the bandwidth for network control packets. In addition, when the link PER becomes high, more $L1$, $L2$, and $L3$ packets might drop on a multihop route. This means that the total usage for $L1$, $L2$, and $L3$ packets would drop, and the total available bandwidth for $L4$ packets would increase. When the network is not stable, more network control packets should be generated in order to repair routes than when the network is stable. Therefore, this approach is appropriate for an LLN.

6. Discussion

6.1. Lower Bound of Available Bandwidth for L3 and L4 Packets. As we show above, almost all $L3$ packets can be delivered when the link success probability (with retry) is greater than 96%. To achieve that level, in Block 1, $L1$ or $L2$ packets and $L3$ packets are not often generated on the same path. ISA100.11a can allocate all time slots for all nodes in order to reduce the traffic pattern. Our proposal, however, does not schedule time slots. Instead, the root node notifies each node of a sequence number in the ComFrame at which it can transmit $L3$ packets. The sequence number does not depend on either the traffic pattern or network topology but may purely be in order of nodes joining the network. Therefore, we evaluated the worst case scenario in which all $L1$, $L2$, and $L3$ packets are transmitted on the same path or nearby paths at the same time. In this case, the end-to-end path success probability for $L3$ packets in Block 1 is 0% because all nodes on the path are in use. Consequently, the number of data collection packets from sensors depends on the length of Block 3 (ΔL_{Block3}).

The length of Block 3 is calculated by

$$\Delta L_{\text{Block3}} = \Delta L_{\text{AF}} - \left\{\Delta L_{\text{Block1}} + \Delta L_{\text{Block2}}\right\}, \qquad (2)$$

Table 4: Communication patterns for $L1$ and $L2$ packets with E2E retry.

Number	Appearance pattern				Required # of ComFrames
	$L1$ (1st)	$L2$ (1st)	$L1$ (retry)	$L2$ (retry)	
(1)	Pass	Pass	—	—	1.0
(2)	Pass	Fail	Pass	Pass	2.0
(3)	Pass	Fail	Pass	Fail	2.0
(4)	Pass	Fail	Fail	—	1.5
(5)	Fail	—	Pass	Pass	1.5
(6)	Fail	—	Pass	Fail	1.5
(7)	Fail	—	Fail	—	1.0

Table 5: Expected number of ComFrames for $L1/L2$ packets with an E2E retry.

Index i	1	2	3	4	5	6	7
E2E success prob.	1.0	0.99	0.98	0.97	0.96	0.95	0.94
E_i	1.0	1.014	1.029	1.043	1.057	1.070	1.083

where ΔL_{Block1} is $N_{\text{node}} \times (H_{\max}+3)$, and $\Delta L_{\text{Block1}} + \Delta L_{\text{Block2}}$ is the number of devices successfully controlled with link retry and end-to-end path retry. Therefore, the length of Block 2 (ΔL_{Block2}) is calculated by

$$\Delta L_{\text{Block2}} = E_i \times N_{\text{node}} - \Delta L_{\text{Block1}}, \tag{3}$$

where E_i is the expected number of ComFrames per node for $L1/L2$ packets with both link retry and end-to-end retry. Table 4 summarizes the communication patterns for $L1/L2$ packets with end-to-end retry. The expected number of ComFrames for $L1/L2$ packets with end-to-end retry is calculated as shown in Table 5. Finally, the number of successfully received $L3$ packets (N_{L3}) is

$$N_{L3} = \Delta L_{\text{Block3}} \times \frac{1}{E_i}, \tag{4}$$

where $1/E_i$ is the success probability of round-trip end-to-end communication with end-to-end retry. In the worst case, $L3$ packets can be collected at a rate of 45.52–63.63% ($0.9 \le 1/E_i \le 1$).

On the other hand, $L4$ packet traffic used almost 90% of the bandwidth in our simulation scenario. In the same worst case, we assume that the communication pattern in Block 1 is similar to those in Blocks 2 and 3. Consequently, the lower bound on the available bandwidth for $L3$ and $L4$ packets (P_{L4}) is 36.36%, as calculated by

$$P_{L4} = \frac{\{(H_{\max}+3) - (H_{\text{ave}}+3)\}}{\Delta_{L_{\text{CF}}}} = \frac{H_{\max} - H_{\text{ave}}}{\Delta_{L_{\text{CF}}}}. \tag{5}$$

6.2. Compatibility with ISA100.11a Standard. Figure 17 shows how to adapt our proposal to the ISA100.11a standard protocol. Basically, our proposal is a technology between the network and data link layers, so that it does not directly affect processing in those layers. We do have to specify the operation mode and adjust some parameters of the data link layer to compose our own frames. We use priority CAMSA/CA, select slow-hopping mode as the channel hopping pattern, and bundle time slots defined by ISA100.11a to compose SlotFrame, ComFrame, and AppFrame logically. Our scheme decides what channel (Ch1, Ch2, or Ch3) each node should use at each SlotFrame. The important point is that selecting a channel from among these three in our scheme is equivalent to deciding the operation mode in the data link layer: transmitting a packet, receiving a packet, forwarding a packet, or waiting to forward a packet. If we implement our scheme over one ISA100.11a data link function over one physical interface, the bandwidth for $L3$ and $L4$ packets will decrease. For example, in Table 2, hidden terminal problems could result. An $L4$ packet from the root node to node 5 at the 2nd SlotFrame would collide with an $L3$ packet from node 6 to node 5. Also, an $L1$ packet from node 2 to node 3 at the 3rd SlotFrame would collide with an $L4$ packet from node 4. To avoid these collisions, we can define a longer length of SlotFrame so as not to overlap the times at which all levels of packets are transmitted. Or, more specifically, our proposal implements three ISA100.11a data link functions over one physical interface in order to guarantee the maximum delay for $L1$ and $L2$ packets and keep the bandwidth for $L3$ and $L4$ packets high.

6.3. Strengths and Weaknesses of Our Proposal. Our proposal is very lightweight and much simpler than usual TDMA-based protocols like ISA100.11a. They typically have a scheduler for allocating time slots to meet application requirements. The network manager has to determine and deliver the schedule to all nodes whenever a new node joins the network or the network topology changes. It consumes considerable bandwidth and causes extra delay especially in a lossy and unstable network. Our proposal defines a ComFrame whose length is fixed during network operation. The length depends on maximum multihop count that is one of predetermined system parameters. In a ComFrame, there are at most two high primal packets (an $L1$ packet and an $L2$ packet). Moreover, the 1st SF of a ComFrame is assigned to a node to transmit an $L3$ packet. Under these setting, all nodes autonomously determine when and what packets should be transmitted on which channels to avid packet collisions. Then, from scheduling point of view, our proposal is simple and does not need to reschedule and redeliver a

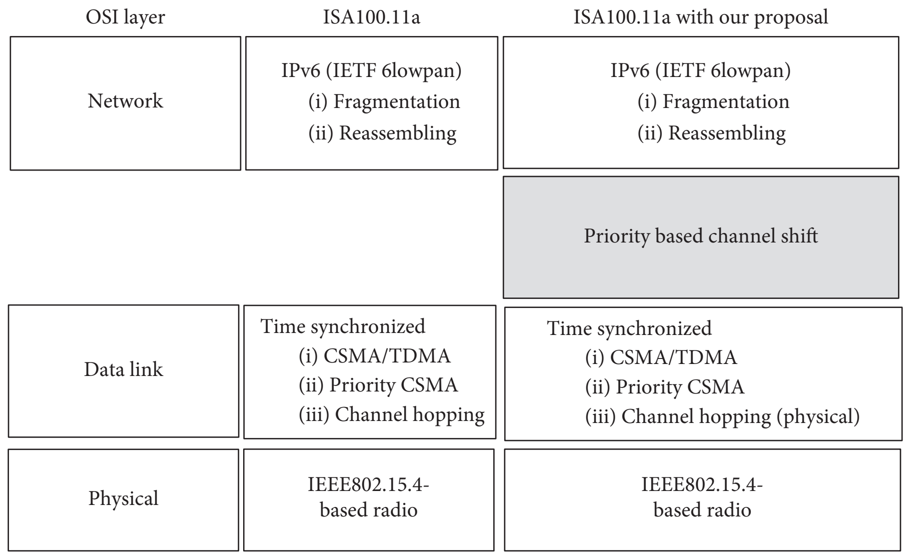

Figure 17: Comparison of functions between ISA100.11a and our proposal.

schedule even if a network topology changes. Above features are strong points of our protocols.

On the other hand, as described above in Section 6.2, our proposal requires more hardware resources than a normal ISA100.11a, when our scheme operates over ISA100.11a. Our proposal needs at least three channels to avoid collisions among different priority packets. Then a node should have three physical interfaces each of which runs full functions of ISA100.11a or have virtual communication interfaces that operate independently over a physical interface to meet our requirements. In either case, hardware cost for a node becomes more expensive than a normal of ISA100.11a. It may hinder deployment of our proposal, but IWSN should be designed to meet real time requirement to guarantee interaction within a predetermined deadline.

7. Conclusions and Future Work

This paper introduced a priority-based dynamic multichannel transmission scheme for IWSNs. Our algorithm enables transmission of packets of different priority level in the same period without collisions. The highest-priority packets for remote control can be delivered within a guaranteed deadline through a hybrid control scheme that combines centralized control by a root node and autonomous decentralized radio channel shift by nonroot nodes. At the same time, lower priority packets belonging to periodic data gathering and control can receive the satisfactory quality of service, where the collection ratio of periodic data packets is higher than 45% and the lower bound of bandwidth available to control packets is larger than 36% at the worst case scenario.

In this paper, we do not address dynamic adaptation of our scheme to handle dynamic or unexpected changes in application and system requirements. For example, composition of AppFrame must be predetermined at the deployment phase under assumptions on system configurations, but it should be dynamically regulated to fit to actual traffic demand. In a case of an unstable network, control packets would be transmitted more frequently. Therefore, we need to organize an AppFrame to spare more bandwidth for $L4$ packets. We plan to tackle these issues as future work.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Z. Alliance, “Zigbee,” 2017, http://www.zigbee.org.

[2] FieldComm Group, “Hart communication,” 2015, https://fieldcommgroup.org.

[3] ISA, “Isa100, wireless systems for automation,” 2017, http://isa100wci.org.

[4] Institute of Electrical and Electronics Engineers (IEEE), “IEEE Standard for Local and metropolitan area networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer IEEE Std,” 802. 15.4e-2012, 2012.

[5] G. Kalani, *Industrial Process Control: Advances And Applications*, Gulf Professional Publishing, Houston, Tex, USA, 2002.

[6] D. J. Olsen, N. Matson, M. D. Sohn et al., “Grid integration of aggregated demand response,” Part 1: Load Availability Profiles and Constraints for the Western Interconnection LBNL-6417E, 2013, Lawrence Berkeley National Laboratory, http://escholarship.org/uc/item/6ps4r3xp.

[7] TEPCO, “Basic concept for smart meter specification based on rfc,” 2012.

[8] D. F. Ramírez and S. Céspedes, “Routing in neighborhood area networks: a survey in the context of ami communications,” *Journal of Network and Computer Applications*, vol. 55, pp. 68–80, 2015.

[9] DLMS, “Dlms user association: Cosem identification system and interface classes,” 2010.

[10] Lineartechnology, "Smartmesh wirelesshart application notes," 2016, http://cds.linear.com/docs/en/application-note/SmartMesh_WirelessHART_Application_Notes.pdf.

[11] Y. Li, C. S. Chen, Y. Song, and Z. Wang, "Real-time qos support in wireless sensor networks: a survey," *IFAC Proceedings Volumes*, vol. 40, no. 22, pp. 373–380, 2007.

[12] H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, "Worst-case bound analysis for the time-critical MAC behaviors of IEEE 802.15.4e," in *Proceedings of the 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pp. 1–9, Trondheim, Norway, May 2017.

[13] D. Vassis and G. Kormentzas, "Throughput analysis for IEEE 802.11 ad hoc networks under the hidden terminal problem," in *Proceedings of the 2006 3rd IEEE Consumer Communications and Networking Conference, CCNC 2006*, pp. 1273–1276, Las Vegas, Nev, USA, January 2006.

[14] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[15] B. Shrestha, E. Hossain, and K. W. Choi, "Distributed and centralized hybrid CSMA/CA-TDMA schemes for single-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 4050–4065, 2014.

[16] O. D. Saputra and S. Y. Shin, "Real-time based Superframe for ISA100.11a in Wireless Industrial Network," *Journal of Communication and Computer*, vol. 12, no. 1, pp. 28–32, 2015.

[17] I. Rhee, A. Warrier, and M. Aia, "Z-MAC: a hybrid MAC for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511–524, 2008.

[18] R. Costa, P. Portugal, F. Vasques, and R. Moraes, "A TDMA-based mechanism for real-time communication in IEEE 802.11e networks," in *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*, Bilbao, Spain, September 2010.

[19] N. Q. Dinh and D.-S. Kim, "Performance evaluation of priority CSMA-CA mechanism on ISA100.11a wireless network," *Computer Standards and Interfaces*, vol. 34, no. 1, pp. 117–123, 2012.

[20] J. Yi, T. Clausen, and Y. Igarashi, "Evaluation of routing protocol for low power and Lossy Networks: LOADng and RPL," in *Proceedings of the 2013 IEEE Conference on Wireless Sensor, ICWISE 2013*, pp. 19–24, Kuching, Malaysia, December 2013, http://www.thomasclausen.net/wp-content/uploads/2015/12/2013-ICWiSE-Evaluation-of-Routing-Protocol-for-Low-Power-and-Lossy-Networks-LOADng-and-RPL.pdf.

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 6438726, 8 pages
https://doi.org/10.1155/2017/6438726

Research Article

REACH: An Efficient MAC Protocol for RF Energy Harvesting in Wireless Sensor Network

Teasung Kim, Joohan Park, Jeehyeong Kim, Jaewon Noh, and Sunghyun Cho

Department of Computer Science and Engineering, Hanyang University, Ansan, Republic of Korea

Correspondence should be addressed to Sunghyun Cho; chopro@hanyang.ac.kr

Received 9 June 2017; Accepted 3 August 2017; Published 24 September 2017

Academic Editor: Jeongcheol Lee

This paper proposes a MAC protocol for Radio Frequency (RF) energy harvesting in Wireless Sensor Networks (WSN). In the conventional RF energy harvesting methods, an Energy Transmitter (ET) operates in a passive manner. An ET transmits RF energy signals only when a sensor with depleted energy sends a Request-for-Energy (RFE) message. Unlike the conventional methods, an ET in the proposed scheme can actively send RF energy signals without RFE messages. An ET determines the active energy signal transmission according to the consequence of the passive energy harvesting procedures. To transmit RF energy signals without request from sensors, the ET participates in a contention-based channel access procedure. Once the ET successfully acquires the channel, it sends RF energy signals on the acquired channel during Short Charging Time (SCT). The proposed scheme determines the length of SCT to minimize the interruption of data communication. We compare the performance of the proposed protocol with RF-MAC protocol by simulation. The simulation results show that the proposed protocol can increase the energy harvesting rate by 150% with 8% loss of network throughput compared to RF-MAC. In addition, the proposed protocol can increase the lifetime of WSN because of the active energy signal transmission method.

1. Introduction

A Wireless Sensor Network (WSN) is a motive power for implementing Internet of Things (IoT) technologies and is used in many systems [1]. As a representative example, a system has been developed that periodically monitors and manages information about a target environment (e.g., temperature, humidity, and illumination) around the sensor [2–4]. There is a critical problem that the lifetime of the WSN is limited because of limited sensor batteries [5]. Also, it cannot be assumed that all sensors are easily physically accessible. As a result, researches have been conducted to increase the energy efficiency of sensor components in order to increase the limited lifetime of the WSN [6, 7]. Previous works showed that the power consumption of sensors can be reduced. However, they do not consider charging the battery, so the battery will be discharged inevitably. A sensor should be able to charge itself using a specific energy source. Radio Frequency (RF) energy harvesting has been proposed [8, 9] as a new energy source of sensors. It supplies stable power to peripheral sensors through Energy Transmitters (ETs) without being affected by the physical environment [10]. With these advantages, a lot of related works have been proposed [11] for determining the routing path [12], data aggregation method [13], improving the energy conversion efficiency [14], and duty-cycle control method [15]. But still, there has not been enough research on the new MAC protocol considering RF energy harvesting. Current researches of the MAC protocol are only related to energy harvesting time. In [16], a method is proposed to adjust the charging time of the sensor. The charging time is changed adaptively according to the data traffic pattern. In [17], a method is proposed to enable for a long time charging of sensors actively involved in data communication. The charging time is determined by an Important Index (IDX) of the sensor that requests energy. Both protocols use methods to allocate the time and channel for charging or requesting energy. They inevitably delay data communication. It is difficult to guarantee real-time communication in WSN when using the existing MAC protocol. To overcome this problem, we propose a method called RF Energy Autocharging and Harvesting (REACH). When there are no packets to be transmitted in the channel, an idle time continues. In REACH, ETs automatically transmit energy to charge the sensors during the idle time. This allows sensors to maintain long data communication times. The contributions of this paper are as follows:

(i) We propose REACH to charge automatically when idle time continues on the channel.

(ii) An improved MAC protocol considering RF energy harvesting is proposed to prevent real-time communications from being disrupted by charging.

(iii) We design REACH that shows 150% performance improvement in harvested energy with backward compatibility.

The rest of this paper is organized as follows. Section 2 introduces the existing MAC protocol considering RF energy harvesting. We explain the proposed REACH algorithm in Section 3. The simulation environment is described in Section 4. The results of performance analysis are presented in Section 5. Finally, conclusion is presented in Section 6.

2. Related Work

There are two representative MAC protocols considering RF energy harvesting: RF-AASP (RF-based energy harvesting technique and the Adaptive, Active Sleeping Period) and RF-MAC (Radio Frequency-Medium Access Control). The RF-AASP determines charging time depending on data traffic and RF-MAC determines it depending on how much the sensor has participated in data communication. However, they still have a problem that charging delays data communication. We describe the RF-AASP and RF-MAC in this section.

2.1. RF-AASP. In [15], an algorithm is proposed that adaptively changes the sleeping period of a sensor. The period changes depending on the traffic pattern and residual energy of sensors. A sensor with low energy checks traffic pattern and satisfaction of Quality of Service (QoS). Based on the result, the sensor adjusts variables like BO (Beacon Order) and SO (Superframe Order). The two adjusted variables determine sleeping period by the equation $t_{\text{sleep}} = 2^{\text{BO}} - 2^{\text{SO}}$. The sensor performs energy harvesting during the sleeping period to charge energy. When the traffic load is large, the RF-AASP may not guarantee a sufficient sleeping period for charging. In other words, the sensor may fail to charge sufficient energy and continues to request energy.

2.2. RF-MAC. In [16], an algorithm is proposed with a new procedure of energy harvesting. Energy harvesting occurs with the Request-for-Energy (RFE) packet. A sensor with low energy broadcasts the RFE packet. Peripheral ETs respond with Cleared-for-Energy (CFE) packets. When the sensor broadcasts the ACK, the ETs emit energy. In this case, the charging time of the sensor depends on the value of the Important Index (IDX). The IDX indicates how much the sensor has participated in data communication in the channel. This algorithm guarantees continuous data communication when there is no energy request. However, when the energy request occurs frequently, starvation can occur. A sensor to send data falls into starvation because of the higher priority of energy request.

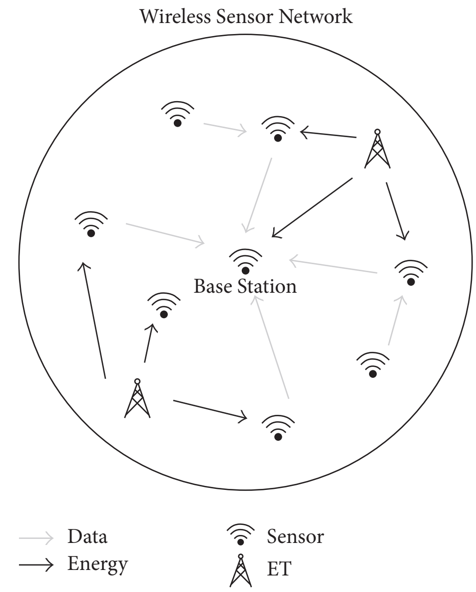

Figure 1: Target WSN.

3. The Proposed REACH Protocol

3.1. Target System. Notations and descriptions used in this paper are shown in the "Notations and Descriptions" section. Figure 1 shows a target network. A WSN consists of a large number of sensors and ETs. ETs are hardware that transmit energy to sensors. Sensors are the subjects of data communication and contend with each other through the CSMA/CA method. All the data from the sensors are transmitted to the Base Station (BS). ETs and sensors share the same frequency band and they have omnidirectional antennas. Therefore, energy and data signals cannot be transmitted at the same time. An energy request has a higher priority than data communication to prevent sensor outage and increase stability of the WSN.

3.2. REACH Algorithm. REACH consists of three steps. Algorithm 1 describes a pseudocode of REACH. First, a sensor gives $\text{CW}_{\text{random}}$ and $\text{SCT}_{\text{origin}}$ (Short Charging Time) to ETs using the existing energy harvesting process. ETs set parameters required for the REACH process based on the received values. The ETs determine the backoff to participate in channel contention. REACH process has a lower priority than data communication because ETs have longer backoff. Finally, the ETs determine the next action depending on whether they acquired the channel. They transmit RF energy signals to nearby sensors for $\text{SCT}_{\text{origin}}$ when they acquire the channel. They adjust the parameters after transmission so that REACH can be performed while reducing the interruption to data communication. Figure 2 shows a flow diagram of REACH. Dashed-line boxes represent the newly proposed process in REACH. Solid-line boxes represent the existing RF-MAC process.

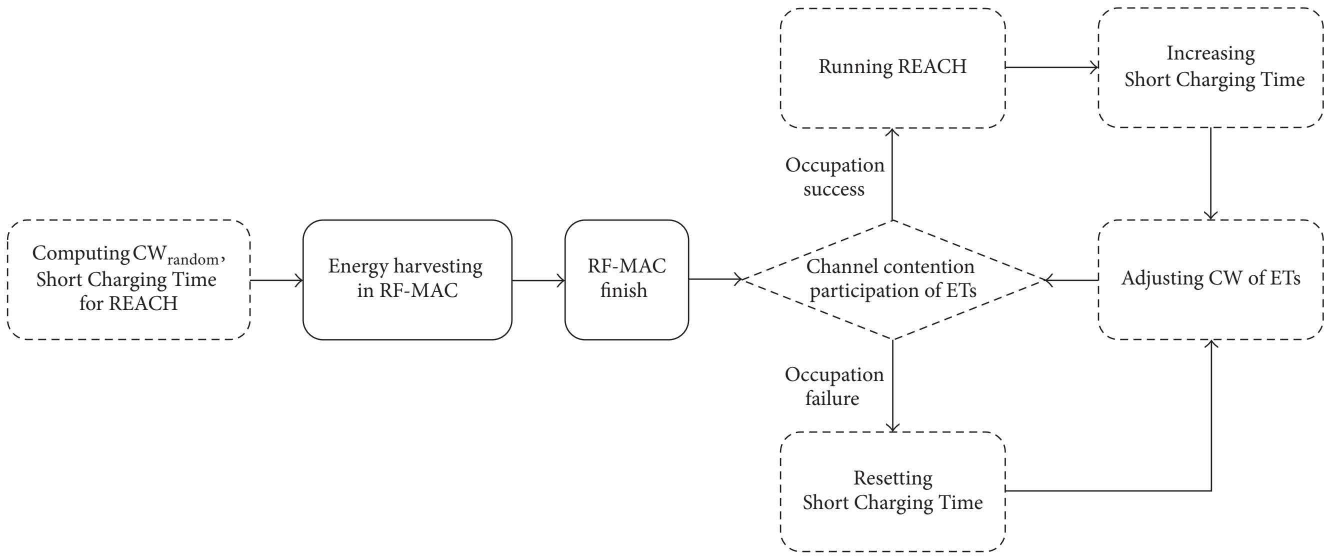

Figure 2: Flow diagram.

```
Step 1
    Compute CW_random, SCT_origin
    Transmit values to ETs
    Save CW_random, SCT_origin with SCT_default, SCT_unit
    While:
Step 2
        Determine backoff
Step 3
        Participate in channel contention
        If success to channel occupation then:
            Auto Charging as SCT_origin
            SCT_origin ⟵ SCT_origin + SCT_unit
            CW_random ⟵ CW_random + 1
        Else:
            SCT_origin ⟵ SCT_default
            CW_random ⟵ CW_random − 1
        End If
    End While
```

Algorithm 1: REACH.

3.2.1. Step 1: Parameter Setting. This step uses a process in the existing charging when a sensor transmits optimization values to nearby ETs. The sensor calculates additional values to be used in the REACH process. Additional calculated values are CW_{random} and SCT_{origin}. These values are included in the ACK packet and transmitted to nearby ETs. CW_{random} is the number of contention windows used when ETs determine backoff to participate in channel contention. After the existing energy harvesting process, ETs must have the same backoff period as each other to set the same start timing of autocharging. CW_{random} cannot be the same without communication between ETs. For this reason, the sensor in this process determines the random value and transmits it to ETs. CW_{random} is randomly determined in the range of CW ($[CW_{min}, CW_{max}]$). CW_{min} and CW_{max} are the minimum and maximum number of contention windows, respectively. SCT_{origin} means the time that ETs automatically transmit energy to nearby sensors if ETs acquire the channel. Autocharging can interrupt data communications because of longer charging time if SCT_{origin} is too large. For this reason, SCT_{origin} should be determined to be of a reasonable length considering current channel conditions. SCT_{origin} expression is as follows:

$$SCT_{origin} = \frac{\sum t_{IDLE}}{\sum N_{data} + 1}. \quad (1)$$

($\sum N_{data} + 1$) represents the maximum number of times the channel was idle for a unit time. SCT_{origin} is calculated to the average of the idle times that existed between data communications. The initial value of SCT_{origin} becomes small since it is divided by the maximum number. SCT_{origin} does not significantly interrupt the data communication time. The sensor passes the random values to nearby ETs after the calculations are complete. The ETs store the values. Then, $SCT_{default}$ and SCT_{unit} are calculated and stored by the ETs. $SCT_{default}$ is the same as the initial SCT_{origin}. $SCT_{default}$ is used to initialize SCT_{origin} again when the channel cannot be acquired. SCT_{unit} is set to SCT_{origin}'s largest decimal unit. SCT_{unit} is used to increase SCT_{origin} after acquiring the channel. Once the transmission and storage process is complete, the ETs proceed with the existing RF-MAC charging process. The ETs proceed to the next step of the REACH process without deleting the received values after the charging process is finished.

3.2.2. Step 2: Backoff Decision. The ETs compute backoff to participate in channel contention after the charging process is finished. ETs must have the same backoff period to match the start timing of autocharging. This is because all ETs must transmit energy for charging at the same time to maximize the constructive interference. Besides, there is a disadvantage that the autocharging time is lengthened when ETs do not match the timing of transmitting energy. The following expression is proposed for synchronizing backoff of all ETs:

$$\text{Backoff} = \text{DIFS}_{\text{AC}} + \text{CW}_{\text{random}} \times \text{Slot}_{\text{AC}}. \quad (2)$$

In order to not have a serious impact on network throughput, the REACH process should have a lower priority than data communications. DIFS_{AC} and Slot_{AC} of backoff should be larger than the value of data communications. We set the two values to $\text{DIFS}_{\text{AC}} = \text{DIFS}_{\text{data}} + \text{DIFS}_{\text{energy}}$ and $\text{Slot}_{\text{AC}} = \text{Slot}_{\text{data}} + \text{Slot}_{\text{energy}}$. $\text{DIFS}_{\text{data}}$ and $\text{DIFS}_{\text{energy}}$ denote DIFS of data communications and energy requests, respectively. $\text{Slot}_{\text{data}}$ and $\text{Slot}_{\text{energy}}$ denote slot time of data communications and energy requests, respectively. All ETs have the same value of DIFS_{AC} and Slot_{AC} because these values are defined by the protocol. $\text{CW}_{\text{random}}$ is broadcasted by the sensor that issued the RFE packet as mentioned in Section 3.2.1, so all ETs have the same $\text{CW}_{\text{random}}$ value. As a result, all ETs have the same backoff period. ETs participate in channel contention like other sensors after backoff is calculated. $\text{SCT}_{\text{origin}}$ and $\text{CW}_{\text{random}}$ are then adjusted according to results of channel contention.

3.2.3. Step 3: Channel Contention. There are two cases of channel contention. The first case is a situation in which the channel is acquired by ETs. Autocharging is executed in this case. The other case is a situation in which the channel is not acquired by ETs. The stored values are reset and ETs rejoin the channel contention in this case.

If there is no data to send in the channel, ETs acquire the channel. The ETs will start autocharging for $\text{SCT}_{\text{origin}}$ after waiting for backoff. ETs transmit energy at the frequencies used in the most recent existing charging process. Therefore, it can skip some processes like RFE-CFE exchange and frequency optimization. $\text{SCT}_{\text{origin}}$ is increased by SCT_{unit} which was calculated beforehand after the autocharge is finished. ETs increase $\text{CW}_{\text{random}}$ to make backoff for autocharging longer than data communication. ET's success in channel acquirement means that there was idle time for ETs to acquire the channel. It can be inferred that there is no data waiting for transmission on the current channel. There is also a high possibility that the channel will be idle in the future. Therefore, ETs increase $\text{SCT}_{\text{origin}}$ to improve channel efficiency. $\text{SCT}_{\text{origin}}$ becomes larger if the ETs frequently acquire the channel. If a particular sensor wants to send data while autocharging is executing, there will be a long delay. It is necessary to prevent the ET from frequently acquiring the channel. Therefore, ETs increase the number of contention windows to wait for a longer backoff time than before. $\text{CW}_{\text{random}}$ is increased to not exceed CW_{max}. In summary, ETs increase the charging time to improve the efficiency when autocharging is executed. ETs also increase backoff at the same time to prevent the frequent autocharging occurrence.

If the channel is not acquired by ETs, the ETs adjust the values. $\text{SCT}_{\text{origin}}$ is initialized to $\text{SCT}_{\text{default}}$ which was saved. $\text{CW}_{\text{random}}$ is decreased. ETs again calculate backoff to try to acquire the channel. Failure to acquire the channel means that there is data to send in the channel and the REACH process is not available. In addition, it can be inferred that there is a high probability that there will be data to be transmitted. It is needed as a way to minimize the effect on data communication even if the channel is acquired by ETs. ETs initialize the charging time to the smallest value to solve this problem. However, if the backoff has increased previously, it will take a long time to wait for a short period of autocharging. The efficiency is very low. $\text{CW}_{\text{random}}$ is decreased to shorten the backoff of ETs. $\text{CW}_{\text{random}}$ is decreased to not less than CW_{max}. In summary, ETs decrease the charging time to minimize effect on data communication. ETs also decrease the backoff to increase the efficiency of autocharging.

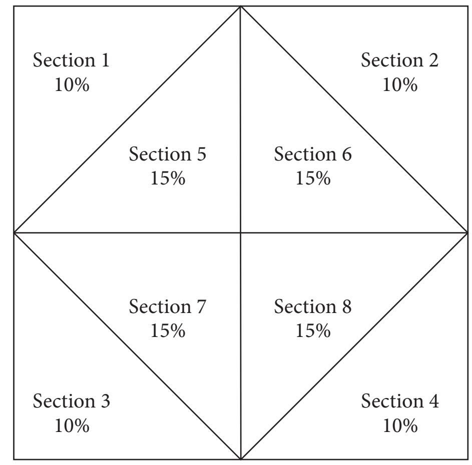

Figure 3: Sectioning of plane and distribution of nodes.

4. Simulation Environment

4.1. Node Setting. First, all nodes are randomly distributed evenly by dividing a $50 \times 50\,\text{m}^2$ grid plane into 8 zones. The reason for dividing the plane into 8 zones is to solve a problem of node gathering on one place in the plane. Figure 3 shows the specific zone on the plane and percentages of the number of nodes. A node is selected as a BS based on the location information of all sensors. The BS role will be taken by a node closest to the geographic center $(25, 25)$. Data generated by sensor is transmitted to the BS node through the forwarding path. All sensors relay the data immediately without compressing or gathering the data. In this simulation, we assume that frame synchronization on all nodes is exactly the same.

Figure 4 shows how a sensor determines a sensor to forward data. Sensor A searches for sensors presented within 10 m from itself. Sensor A finds sensor B which is closest to the BS among the sensors existing within 10 m. And sensor A forwards the data to sensor B. The sensor selected as BS receives data but does not transmit data. We assume that there is one channel in the plane. Table 1 lists system parameters used by the simulator.

4.2. Numerical Modeling

4.2.1. Energy Consumption Model. We refer to sensor motes called Mica2 to set the energy consumption model of sensor

Table 1: System parameters.

Parameter	Symbol	Value
Slot time for energy (s)	$\text{Slot}_{\text{energy}}$	10μ
Slot time for data (s)	$\text{Slot}_{\text{data}}$	20μ
Minimum contention window	CW_{min}	32
Maximum contention window	CW_{max}	1024
SIFS for energy (s)	$\text{SIFS}_{\text{energy}}$	5μ
SIFS for data (s)	$\text{SIFS}_{\text{data}}$	10μ
DIFS for energy (s)	$\text{DIFS}_{\text{energy}}$	25μ
DIFS for data (s)	$\text{DIFS}_{\text{data}}$	50μ
Minimum voltage (v)	V_{min}	1.8
Maximum voltage (v)	V_{max}	3.0
Threshold for harvesting (v)	$V_{\text{threshold}}$	2.3
Channel speed (kbps)		250

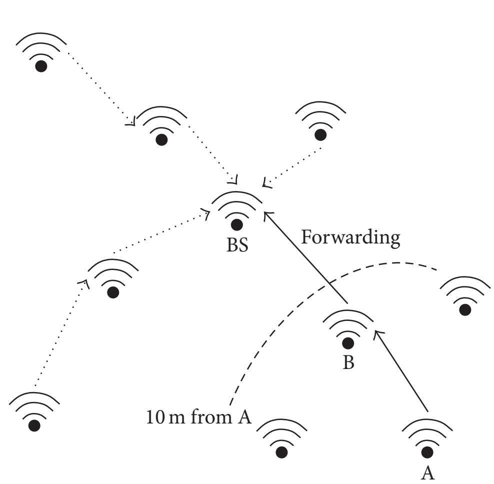

Figure 4: The forwarding node setup process.

used in simulation [18]. Table 2 shows currents at specific state of sensor. An I notation is the amount of consumed current in each state. In sleep mode, currents of 15 μA flow through a circuit. 8 mA currents flow in the circuit while a sensor listens to a channel. 25 mA currents flow while a sensor sends data and 8 mA currents flow while a sensor receives data. E_{consume} is the summation of the amount of consumed energy in each state (R represents the resistance of the circuit):

$$E_{\text{consume}} = E_{\text{sleep}} + E_{\text{tx}} + E_{\text{rx}} + E_{\text{listen}} = R \times \left\{\left(I_{\text{sleep}} \times t_{\text{sleep}}\right) + \left(I_{\text{tx}} \times t_{\text{tx}}\right) + \left(I_{\text{rx}} \times t_{\text{rx}}\right) + \left(I_{\text{listen}} \times t_{\text{listen}}\right)\right\}. \quad (3)$$

4.2.2. Energy Harvesting Model. ETs transmit energy to nearby sensors at a power of 3 W. The propagation loss of energy sent from ETs is calculated by the Friis transmission equation [19]. Friis transmission equation shows a relationship between transmitting power (TP) and receiving power (RP). Suppose a transmitting ET is i and a receiving sensor is j. The Friis transmission equation between i and j is as follows:

Table 2: Currents at each state.

State	Symbol	Value (A)
Sleep	I_{sleep}	15μ
Waiting	I_{waiting}	8m
Transmit packet	I_{tx}	25m
Receive packet	I_{rx}	8m
Listening	I_{listen}	8m

$$\text{RP}_j = \text{TP}_i G_i G_j \left(\frac{\lambda}{4\pi d_{(i,j)}}\right)^2. \quad (4)$$

Since we assume the omnidirectional antenna, the two antenna gain values are always 1. A sensor can receive energy from multiple ETs located at various distances. In the RF-MAC charging process, ETs are separated into two groups according to the phase of wave. As a result, only constructive interference is caused by the energy waves at the sensor. An actual amount of harvested energy should add RP_j of the signal from each ET during a charging time t:

$$E_{\text{harvest}} = \sum_{i \in A_j} \int_0^t \text{TP}_i \left(\frac{\lambda}{4\pi d_{(i,j)}}\right)^2. \quad (5)$$

5. Performance Analysis

In this section, we compare the performances between RF-MAC and REACH. We built the simulation environment mentioned in Section 4 by using Java Language. Each experiment was performed 20 times and average values of the results were compared. In experiments of changing the number of sensors, the number of ETs is fixed at 100. Likewise, in experiments of changing the number of ETs, the number of sensors is fixed at 250. It is assumed that all data has a size of 50 bytes.

5.1. The Number of RFE Packet Generations. We compared the number of RFE generations by the number of sensors. Figure 5 shows the average number of RFE generations according to the number of sensors. Residual energy of the sensor drops below a threshold of energy harvesting; RFE generation is inevitable. When a RFE packet is generated, energy harvesting is performed. In this process, energy harvesting delays data communications because of higher priority. In the graph, REACH has substantially fewer RFE generations than the existing RF-MAC. Due to autocharging, the lifetime of sensors is increased and RFE generation is delayed. The number of RFE generations is reduced within the same time and delaying data communications will be reduced. However, when the number of sensors is large, the number of RFE generations becomes similar. This is because there are a lot of data communications in the channel. Autocharging cannot occur frequently because of lower priority than data communications. The amount of

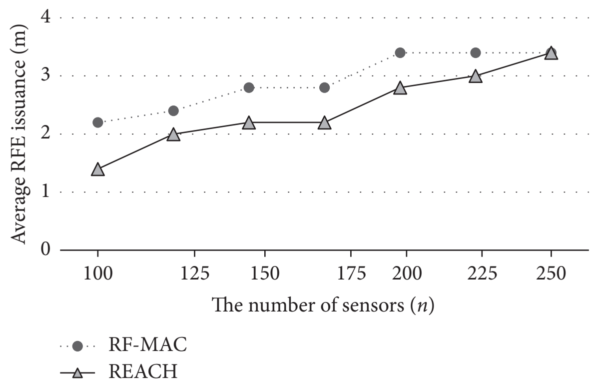

Figure 5: The number of generations of RFE packet.

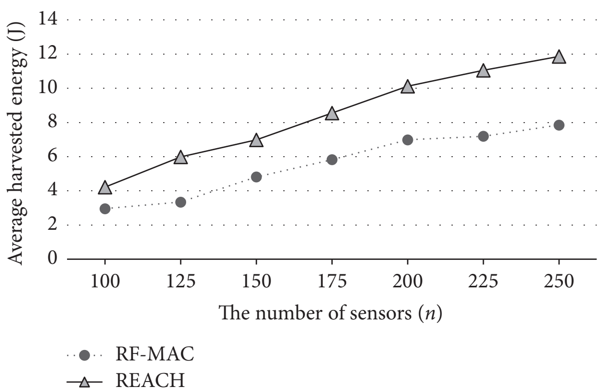

Figure 6: The amount of harvested energy by the number of sensors.

autocharging is reduced, and RFE is generated to charge sensors. As a result, the average number of RFE generations was reduced from 18% up to 36%.

5.2. The Amount of Harvested Energy. We compared the amount of harvested energy by the number of sensors in WSN. The harvested energy is the summation of the amount of RF energy received. Figure 6 shows the amount of harvested energy by the number of sensors in RF-MAC and REACH. Basically, both algorithms show that the amount of harvested energy increases as the number of sensors increases. As the number of sensors increases, sensors are located more closely on the experimental plane. In other words, there are many sensors around one ET. When the ET emits energy, a large number of sensors can be charged. REACH showed an overall 150% increase in harvested energy compared with the existing RF-MAC. Even though the number of RFE generations is reduced as shown in Section 5.1, the amount of harvested energy is rather higher.

We compared the amount of harvested energy by the number of ETs in WSN. Figure 7 shows the amount of harvested energy by the number of ETs in RF-MAC and REACH. As the number of ETs increases, both algorithms show increases in the amount of harvested energy. If the number of ETs increases, the probability that a sensor and an ET are close together also increases. Therefore, the amount of

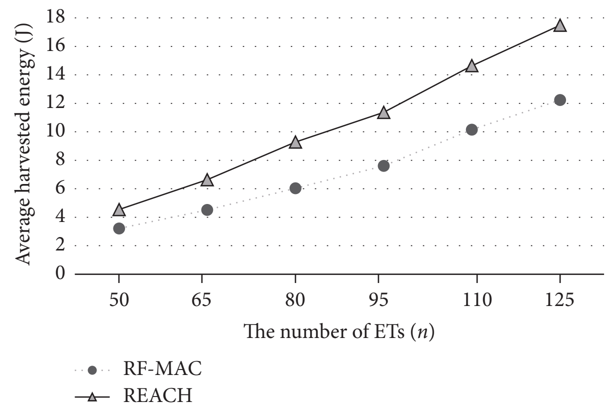

Figure 7: The amount of harvested energy by the number of ETs.

harvested energy is increased as the number of ETs increases. The difference between RF-MAC and REACH is not large when the number of ETs is small. This is due to the fact that a small number of ETs cannot charge a large amount of energy during autocharging. The efficiency of autocharging increases and the difference between REACH and RF-MAC increases when the number of ETs is large. Results show that REACH harvests 140% more energy than RF-MAC.

5.3. The Network Throughput. We compared the network throughput according to the number of sensors constituting WSN. Figure 8 shows the comparison of network throughput. As shown in Section 5.2, network throughput also tends to increase as the number of sensors increases. When the number of sensors is increased, sensors are located more closely. In the forwarding path setup process, there are many sensors within 10 m of one sensor. A straight line connecting the sensor and BS is the most optimal path for forwarding data from one sensor to the BS. When the number of sensors is great, the probability that a sensor closer to the optimal path is determined as the next hop sensor increases. An efficient forwarding path is set, and network throughput is increased. Regardless of the number of sensors, REACH shows a throughput loss of about 8%. This can be inferred by the inevitable impact of autocharging being set to minimize the impact of data communication.

We compared the network throughput by the number of ETs. Figure 9 shows the simulation results. Both algorithms tend to show a constant throughput regardless of the number of ETs. They emit energy simultaneously to increase the efficiency of charging. This behavior prevents the number of ETs from affecting network throughput. It can be inferred that as the number of ETs increases, the charging efficiency increases and no additional network throughput loss occurs. As with the previous simulation, the network throughput of REACH is lower by 10% than that of RF-MAC.

5.4. The Average Residual Energy of Sensors. We compared the average residual energy of sensors at every unit time. In this simulation, the unit time is 5 seconds. Figure 10 shows the average of residual energy of sensors every 5 seconds. The point where the residual energy in the graph soars

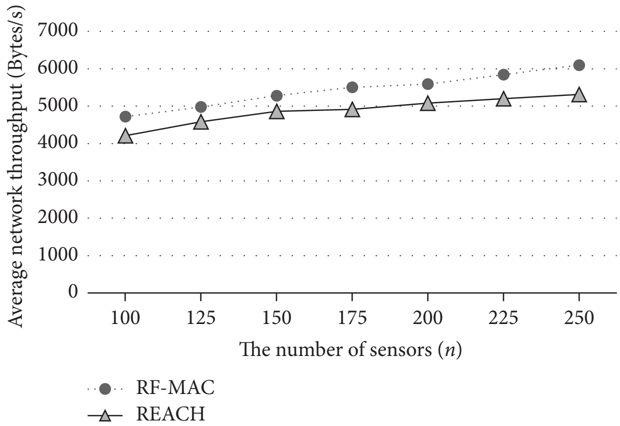

Figure 8: Network throughput by the number of sensors.

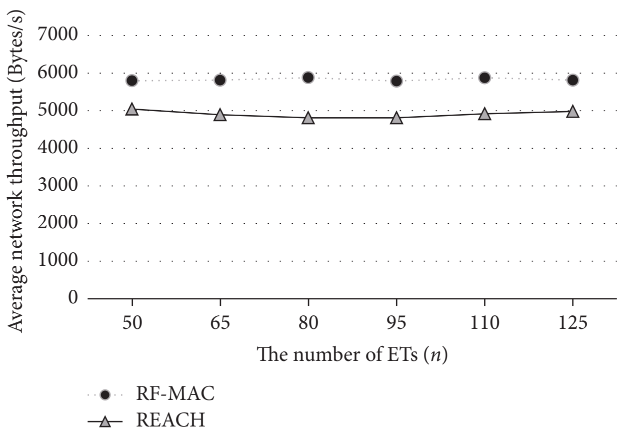

Figure 9: Network throughput by the number of ETs.

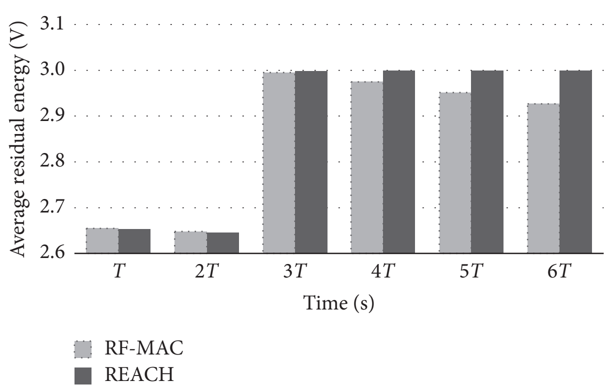

Figure 10: Average residual energy.

greatly is after the energy harvesting in the existing RF-MAC. Before energy harvesting, there are no significant differences between RF-MAC and REACH. There is a difference after the energy harvesting. The average of residual energy in the existing RF-MAC is reduced. However, the average of residual energy in REACH is still maintained as high energy because of autocharging. Autocharging affects data communication, but the overall lifetime increases. If there are many data communications, the average of residual energy becomes the same. Until then, REACH has maintained a high energy. As a result, REACH differs by 0.8% from the existing RF-MAC in the average of residual energy.

6. Conclusion

We have proposed a protocol that uses idle time of the channel to autocharge batteries of sensors. If the sensors do not have data to send, ETs emit energy. The sensors automatically charge by using the energy. Sensors can be held for longer periods of time without artificial charging requests when data communication is active. We compared the performance between the existing RF-MAC and REACH. As a result, REACH showed an increase in energy charging of 150% and decrease in network throughput of 8%. Also, we can see that the residual energy of the sensors is more than 0.8% at every point in time. Using REACH may result in a small loss of network throughput but it can increase the lifetime of the WSN in a stable manner. The proposed REACH still has some problems. Some network throughput is lost for autocharging. When there are a lot of data communications, eventually, the RF-MAC charging procedure is processed. The future work is the process of improving REACH that can be executed even in a situation when there are a lot of data communications.

Notations and Descriptions

CW_{random}: The number of contention windows of ET
SCT_{origin}: Current Short Charging Time
$SCT_{default}$: Default value of SCT_{origin}
SCT_{unit}: Incremental unit of SCT_{origin}
$\sum t_{IDLE}$: Total amount of idle time in unit time
$\sum N_{data}$: The number of data communication occurrences
$DIFS_{AC}$: The DIFS value for autocharging
$Slot_{AC}$: The slot time for autocharging
$E_{consume}$: The amount of consumed energy by sensor
$E_{harvest}$: The amount of harvested energy by sensor
E_{state}: The amount of consumed energy by sensor at each state
TP_i: Transmission power of object i
RP_i: Received power of object i
G_i: Antenna gain of object i
$d_{(i,j)}$: Distance between i and j
A_j: Set of ETs that received RFE sent from sensor j.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. NRF-2015R1D1A1A01059473) and was supported by the research fund of Hanyang University (HY-2015-N).

References

[1] Y. Zhan, L. Liu, L. Wang, and Y. Shen, "Wireless sensor networks for the internet of things," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[2] T. Torfs, T. Sterken, S. Brebels et al., "Low power wireless sensor network for building monitoring," *IEEE Sensors Journal*, vol. 13, no. 3, pp. 909–915, 2013.

[3] V. Jelicic, M. Magno, D. Brunelli, G. Paci, and L. Benini, "Context-adaptive multimodal wireless sensor network for energy-efficient gas monitoring," *IEEE Sensors Journal*, vol. 13, no. 1, pp. 328–338, 2013.

[4] N. Sakthipriya, "An effective method for crop monitoring using wireless sensor network," *Middle-East Journal of Scientific Research*, vol. 20, no. 9, pp. 1127–1132, 2014.

[5] L. Rosyidi and R. F. Sari, "Energy harvesting aware protocol for 802.11-based Internet of Things network," in *Proceedings of the 2016 IEEE Region 10 Conference, TENCON 2016*, pp. 1325–1328, November 2016.

[6] S. Guo, L. He, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2787–2802, 2014.

[7] D. Zhang, G. Li, K. Zheng, X. Ming, and Z.-H. Pan, "An energy-balanced routing method based on forward-aware factor for wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 766–773, 2014.

[8] T.-Q. Wu and H.-C. Yang, "On the performance of overlaid wireless sensor transmission with rf energy harvesting," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 8, pp. 1693–1705, 2015.

[9] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless networks with RF energy harvesting: a contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 757–789, 2015.

[10] D. Mishra, S. De, S. Jana, S. Basagni, K. Chowdhury, and W. Heinzelman, "Smart RF energy harvesting communications: Challenges and opportunities," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 70–78, 2015.

[11] X. Lu, P. Wang, D. Niyato, and Z. Han, "Resource allocation in wireless networks with RF energy harvesting and transfer," *IEEE Network*, vol. 29, no. 6, pp. 68–75, 2015.

[12] A. A. Nasir, X. Zhou, S. Durrani, and R. A. Kennedy, "Relaying protocols for wireless energy harvesting and information processing," *IEEE Transactions on Wireless Communications*, vol. 12, no. 7, pp. 3622–3636, 2013.

[13] S. Jeong, H. Kim, D. K. Noh, and I. Yoon, "Energy-aware data aggregation scheme for energy-harvesting wireless sensor networks," in *Proceedings of the 1st IEEE International Conference on Computer Communication and the Internet, ICCCI 2016*, pp. 140–143, October 2016.

[14] M. M. Ababneh, S. Perez, and S. Thomas, "Optimized power management circuit for RF energy harvesting system," in *Proceedings of the Wireless and Microwave Technology Conference (WAMICON)*, pp. 1–4, April 2017.

[15] T. D. Nguyen, J. Y. Khan, and D. T. Ngo, "Energy harvested roadside IEEE 802.15.4 wireless sensor networks for IoT applications," *Ad Hoc Networks*, vol. 56, pp. 109–121, 2017.

[16] T. D. Nguyen, J. Y. Khan, and D. T. Ngo, "An adaptive MAC protocol for RF energy harvesting wireless sensor networks," in *Proceedings of the 59th IEEE Global Communications Conference, GLOBECOM 2016*, pp. 1–6, December 2016.

[17] M. Y. Naderi, P. Nintanavongsa, and K. R. Chowdhury, "RF-MAC: A medium access control protocol for re-chargeable sensor networks powered by wireless energy harvesting," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 3926–3937, 2014.

[18] Z. Dian, M. Zhong, L. Gang et al., "An empirical study of radio signal strength in sensor networks using mica2 nodes," *Journal of Shenzhen University Science and Engineering*, vol. 1, article 009, 2014.

[19] P. Nintanavongsa, "A survey on RF energy harvesting: circuits and protocols," in *Proceedings of the Eco-Energy and Materials Science and Engineering, EMSES 2014*, vol. 56, pp. 414–422, December 2013.