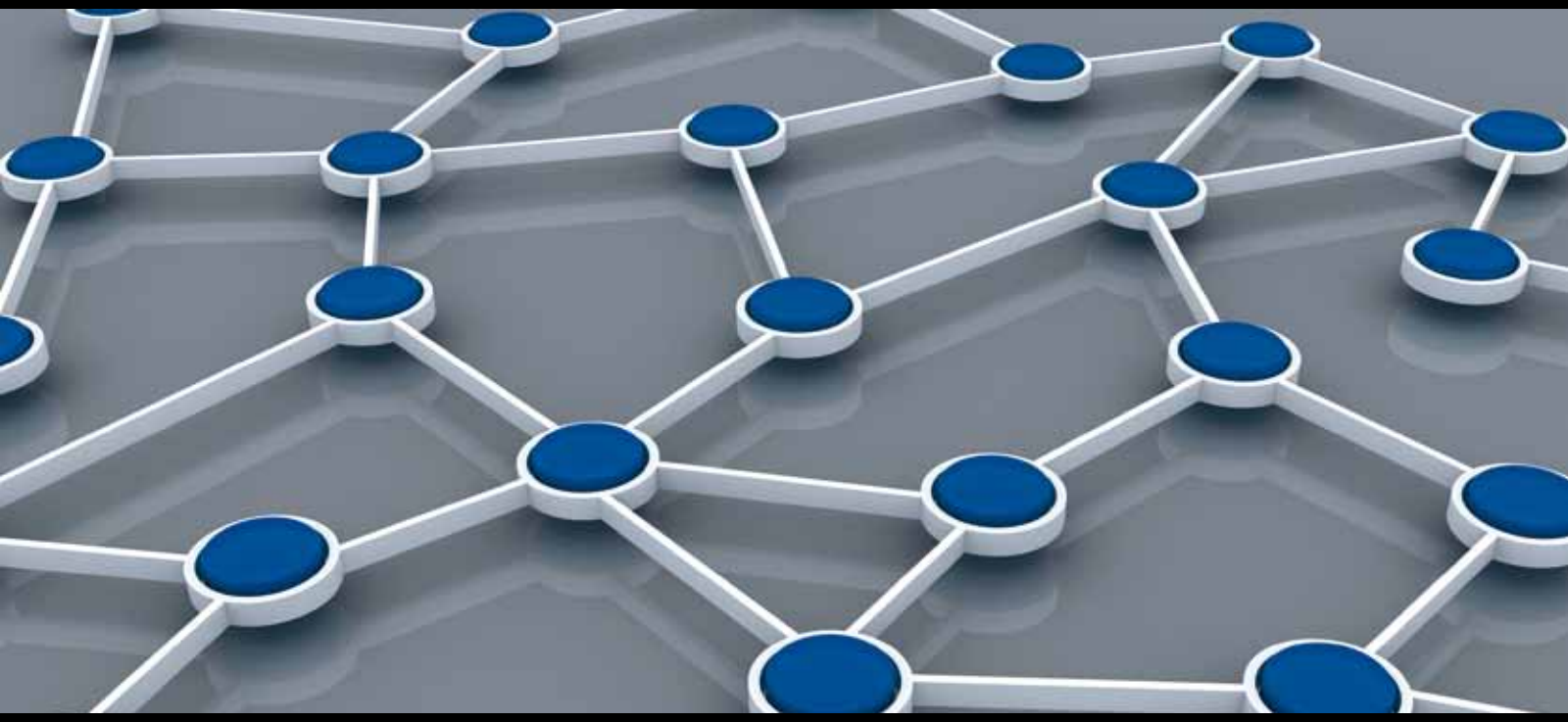
International Journal of Distributed Sensor Networks

IP-enabled Wireless Sensor Network

Guest Editors: Deyun Gao, Chuan Heng Foh, Oliver W. W. Yang, Xusheng Sun, and Chin-Feng Lai



IP-EnabledWireless Sensor Network

International Journal of Distributed Sensor Networks

IP-EnabledWireless Sensor Network

Guest Editors: Deyun Gao, Chuan Heng Foh,
Oliver W. W. Yang, Xusheng Sun, and Chin-Feng Lai

Editorial Board

Contents

IP-enabled Wireless Sensor Network, Deyun Gao, Chuan Heng Foh, Oliver W. W. Yang, Xusheng Sun, and Chin-Feng Lai
Volume 2012, Article ID 851426, 2 pages

DACH: An Efficient and Reliable Way to Integrate WSN with IPv6, Dequan Yang and Qiao Guo
Volume 2012, Article ID 714786, 12 pages

An Effective Find and Replicate Strategy for Data Communication in Intermittently Connected Wireless Ad Hoc and Sensor Networks, C. Poongodi and A. M. Natarajan
Volume 2012, Article ID 452686, 11 pages

A Group Key Distribution Scheme for Wireless Sensor Networks in the Internet of Things Scenario, Hong Yu, Jingsha He, Ting Zhang, and Peng Xiao
Volume 2012, Article ID 813594, 12 pages

Improving Driving Safety Based on Safe Distance Design in Vehicular Sensor Networks, Chen Chen, Lei Liu, Xiaobo Du, Qingqi Pei, and Xiangmo Zhao
Volume 2012, Article ID 469067, 13 pages

Interconnection between IP Networks and Wireless Sensor Networks, Brandon Keith Maharrey, Alvin S. Lim, and Song Gao
Volume 2012, Article ID 567687, 15 pages

Broadcasting with Least Redundancy in Wireless Sensor Networks, Ruiqin Zhao, Xiaohong Shen, Zhe Jiang, and Haiyan Wang
Volume 2012, Article ID 957606, 11 pages

Remote Industrial Sensor Network Monitoring Using M2M Based Ethical Sniffers, Syed Muhammad Asad Zaidi, Jieun Jung, Minsoo Kang, Byunghun Song, and Ki-Hyung Kim
Volume 2012, Article ID 286424, 9 pages

A Routing Scheme for IPv6-Based All-IP Wireless Sensor Networks, Wang Xiaonan and Zhong Shan
Volume 2012, Article ID 750681, 9 pages

Security and Vulnerability of SCADA Systems over IP-Based Wireless Sensor Networks, HyungJun Kim
Volume 2012, Article ID 268478, 10 pages

Wireless Sensor Networks for Vital Signs Monitoring: Application in a Nursing Home, Yuan-Jen Chang, Chin-Hsing Chen, Li-Feng Lin, Ruo-Ping Han, Wen-Tzeng Huang, and Guey-Chuen Lee
Volume 2012, Article ID 685107, 12 pages

Environment-Aware CMT for Efficient Video Delivery in Wireless Multimedia Sensor Networks, Yuanlong Cao, Changqiao Xu, Jianfeng Guan, Fei Song, and Hongke Zhang
Volume 2012, Article ID 381726, 12 pages

Holes Detection in Anisotropic Sensornets: Topological Methods, Wei Wei, Xiao-Lin Yang, Pei-Yi Shen, and Bin Zhou
Volume 2012, Article ID 135054, 9 pages

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 851426, 2 pages
doi:10.1155/2012/851426

Editorial
IP-Enabled Wireless Sensor Network

Deyun Gao,[1] Chuan Heng Foh,[2] Oliver W. W. Yang,[3] Xusheng Sun,[4] and Chin-Feng Lai[5]

[1] *School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2] *School of Computer Engineering, Nanyang Technological University, Singapore 639798*
[3] *School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, ON, Canada K1N 6N5*
[4] *School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*
[5] *Institute of Computer Science and Information Engineering, National Ilan University, Ilan 26041, Taiwan*

Correspondence should be addressed to Deyun Gao, gdeyun@gmail.com

Received 17 December 2012; Accepted 17 December 2012

In wireless sensor networks (WSNs), the use of the IP technology has always been considered inadequate due to its high overhead in processing and memory usage. However, the use of IP technology in WSNs can provide basis for transparent communication among sensor nodes and seamless interconnection between WSNs and the Internet-based infrastructure network, thus eliminating the need for translation gateways or similar devices that are used in current WSNs for this purpose. Besides, many devices in a WSN make network autoconfiguration and statelessness highly desirable for the network operator. For this, IPv6 has ready solutions. Moreover, the large number of devices poses the need for a large address space, which is well met by IPv6. Application of IP technology to IEEE 802.15.4 will be a very important research direction. To accelerate the adoption of these new emerging techniques, a number of important issues must be addressed such as the routing and security of IP-enabled wireless sensor network and applications. In this special issue, we have invited a few papers that address such issues.

The paper titled as "*A routing scheme for IPv6-based all-IP wireless sensor networks*" addresses the routing for IPv6 WSNs. The paper creates the IPv6 address structure and the IPv6 address configuration algorithm for all-IP wireless sensor networks. Based on the IPv6 address structure, the paper proposes the routing algorithm in the link layer for all-IP wireless sensor networks. The paper titled as "*Broadcasting with least redundancy in wireless sensor networks*" presents the study on the broadcast algorithm with least redundancy for WSNs, which optimizes broadcasting by reducing redundant rebroadcasts and balancing energy consumption among all nodes.

The paper titled as "*Interconnection between IP networks and wireless sensor networks*" proposes a middleware layer that provides transparent communication between IP-based hosts and sensor nodes via gateway nodes, which are employed as access points for the purpose of interoperable information retrieval from WSNs and seamless interconnection between IP networks and wireless sensor networks. The paper titled as "*DACH: an efficient and reliable way to integrate WSN with IPv6*" presents an adaptive clustering hierarychy algorithm. The communications in cluster are based on IEEE 802.15.4, and one cluster connects with other clusters or with base stations through IPv6 technique.

The paper titled as "*Security and vulnerability of SCADA systems over IP-based wireless sensor networks*" identifies and removes various vulnerabilities and threats to supervisory control and data acquisition systems over IP-based wireless sensor networks, especially for 6LoWPAN applications. The paper titled as "*A group key distribution scheme for wireless sensor networks in the Internet of Things scenario*" presents a group key distribution scheme for WSNs in the Internet of Things scenario, which organizes sensor nodes into groups in a hierarchical structure. The paper titled as "*Holes detection in anisotropic sensornets: topological methods*" gives out a method to detect the holes in the topological architecture of sensornets only by connectivity information, which can properly detect holes on the topological surfaces and connect them into meaningful boundary cycles. The paper titled as "*An effective find and replicate strategy for data communication in intermittently connected wireless ad hoc and sensor networks*" presents a routing protocol to reduce the number

of replications in turn, decrease the delay, and increase the throughput.

The remaining several papers address some issues for many applications of wireless sensor network. The paper titled as "*Environment-aware CMT for efficient video delivery in wireless multimedia sensor networks*" presents a transport protocol for multimedia traffic delivery over WSN. The paper titled as "*Remote industrial sensor network monitoring using M2M-based ethical sniffers*" considers some security issues in industrial sensor network. The papers titled as "*Improve driving safety based on safe distance design in vehicular sensor networks*" and "*Wireless sensor network for vital signs monitoring: application in a nursing home*" address some issues of WSNs applied to intelligent transportation system and monitoring in a nursing home respectively.

Deyun Gao
Chuan Heng Foh
Oliver W. W. Yang
Xusheng Sun
Chin-Feng Lai

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 714786, 12 pages
doi:10.1155/2012/714786

Research Article

DACH: An Efficient and Reliable Way to Integrate WSN with IPv6

Dequan Yang and Qiao Guo

School of Automation, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Dequan Yang, yangdequanbit@gmail.com

Received 27 June 2012; Revised 25 September 2012; Accepted 31 October 2012

Academic Editor: Chuan Foh

IPv6 has many advantages such as the massive amount of addresses, high security, and high robustness, which are beneficial for wireless sensor networks (WSNs). However, it is almost impossible to use IPv6 directly in WSN due to its huge energy consumption. This paper proposes a double adaptively clustering hierarchy (DACH) algorithm which enables using IPv6 in WSN in an efficient and reliable way. Firstly, we present a clustering method to adaptively divide the whole sensor network into clusters according to its energy consumption in the last round. Then we propose an adaptive cluster head selection algorithm which employs a strategy to choose the most suitable cluster heads; meantime, this selection algorithm is integrated into DACH. Finally, the complete framework is built between headers and their slave nodes based on IEEE 802.15.4, and IPv6 is used to connect the headers and the base stations. Experimental and simulation results demonstrate that the DACH algorithm has lower time and energy consumption. Moreover, it is more reliable and applicable than many other IP-based WSN algorithms.

1. Introduction

One of the most important techniques of this decade is wireless sensor networks (WSNs). In the last twenty years, interpersonal communication has become very popular with the booming internet technology. Similarly, with the development of WSNs [1], the same phenomenon will occur, and people will benefit a lot from this new information exchange technology. When WSN is as widely used as the internet, people can turn on their air conditioners at home when they are still on their way; the information of snow depth of every valley of Alps can be measured and collected by sensors and sent to people for making decisions about holiday skiing; any equipment of a city can send an alarm to the fire station automatically when the temperature is beyond the normal range, and so forth.

Without access to the internet, WSN is just a usual local network with its limited power. However, when IPv6 joins, WSN becomes magic and powerful, for IPv6 has a lot of advantages, such as massive addresses, high security, and good QoS service [2]. Since TCP/IP is limited with factors like too much energy cost and low battery frequent data transmission at the sensor nodes, IPv6-based WSN is more favorite for the researchers. However, for WSN, header overhead problem in IPv6 is more serious than that in IPv4. Usually, the monitoring signal, control signal, and measured data of a sensor is no more than 10 bytes [3]. If IPv6 is introduced directly, the header overhead will consume more than 80% of the total energy and drastically shorten the lifetime of WSN.

As is shown in Figure 1, the header length of IPv6 is nearly one-third of the upper bound of the total frame length under IEEE 802.15.4, which takes up too much resource and energy consumption. To solve this problem, many researchers are devoted to finding efficacious ways to integrate WSN with IPv6.

To make WSN access to the internet, [4] proposes an all-IP scheme in which all the senor nodes take IPv6 as the internet protocol. In this way, WSN can be integrated into the internet seamlessly, and WSN becomes part of the internet. However, the all-IP scheme often leads to a large energy consumption, and when the energy cost of all-IP WSN is too high to be tolerated, the whole network will be down.

In [5], a high energy efficiency strategy is employed to make WSN access to the internet. Although the lifetime of the nodes in [5] is ideal for applications in reality, the strategy is based on IPv4. Therefore, this strategy cannot be used in IPv6-based sensor networks.

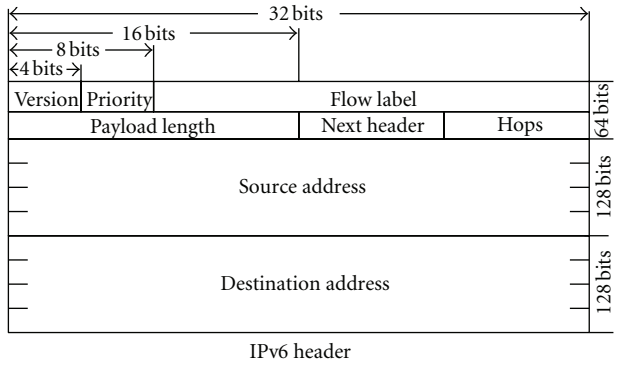


Figure 1: Structure of IPv6 header.

In [6, 7], the scholars try to build a data-centric IP-based WSN routing protocol, in which the algorithms have acceptable time and computational overhead. But these methods are not well adapted to the address-centric internet.

In [8], a mixed scheme is proposed, in which it combines agent-based IP WSN and TCP/IP-WSN to make WSN access to the internet. Theoretically, this method is perfect, but no details of this algorithm is given, and plus, no experiment results and even simulations are provided to support this theory.

In [9], the stack is modified to integrate IPv6 into sensor network. The simulation verifies its good performance. However, the plan is too complicated and difficult to deploy in reality.

Although there are a lot of studies on the solution for the application of compressed IPv6 to WSN, compressed IPv6 is also a high energy consumption protocol, which needs the support of inner-cluster communications. Hence, an algorithm based on inner-cluster connection for high energy efficiency WSN is needed to save more energy for the heads to support IPv6 communications.

In this paper, a double adaptive clustering hierarchy (DACH) algorithm is proposed to introduce IPv6 into WSN to make sensor networks able to access to the internet. Firstly, a clustering algorithm is designed to divide WSNs into several clusters adaptively according to the number of hops and energy consumption. Secondly, a cluster head selection algorithm is constructed to select clusters adaptively, which can further reduce the energy consumption. Finally, communications in cluster are based on IEEE 802.15.4, and one cluster connects with other clusters or with base stations through IPv6. The paper contributes the following:

(1) a clustering algorithm which makes clustering adaptively and saves energy cost;

(2) an adaptive cluster head selection algorithm which enables the network to choose headers with minimum energy consumption;

(3) a reliable IPv6-based WSN framework which makes WSN access to the internet in a reliable way with high energy efficiency;

(4) simulations and experiments are present to demonstrate the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 reviews the problems of WSN clustering. In Section 3, the adaptive clustering algorithm for dividing WSN into groups is presented. In Section 4, a weighing mechanism is designed, which takes the parameters such as remaining capacity, number of hops, and cluster head history into consideration, to build a cluster head selection algorithm. The low energy consumption access to internet of WSN is achieved by a switching strategy between IEEE 802.15.4 and IPv6 in Section 5. In Section 6, simulations and experiments are presented and analyzed. And Section 7 is the summary of this paper.

2. Background

2.1. Why Clustering? Since the survival time of nodes in WSN is limited by the remaining capacity of their batteries, one of the most significant challenges in WSN design is how to save energy as much as possible [10].

Clustering is an important topology control method in WSN. It can significantly reduce average energy consumption of nodes and improve network throughput. In clustering routing protocols, sensors networks are usually divided into clusters. A cluster is a node set in which the nodes are connected together in a certain way. Each cluster contains a cluster head and a number of cluster members. Cluster heads of lower-level networks are the cluster members of higher-level networks. The highest level of cluster head works for communicating with base station [11]. The topology of cluster-based WSN is shown in Figure 2.

In clustering algorithm, the whole sensor network is divided into a number of subsets, which are connected with each other by the heads. In each cluster, the head is selected by a certain algorithm which can manage or control all the members of the cluster. In addition, the head should be responsible for inner-cluster information collection, data fusion, and intercluster data forwarding [12].

The advantages of clustering routing are as follows.

(1) Most of the time, cluster members could turn off their communication models because the cluster

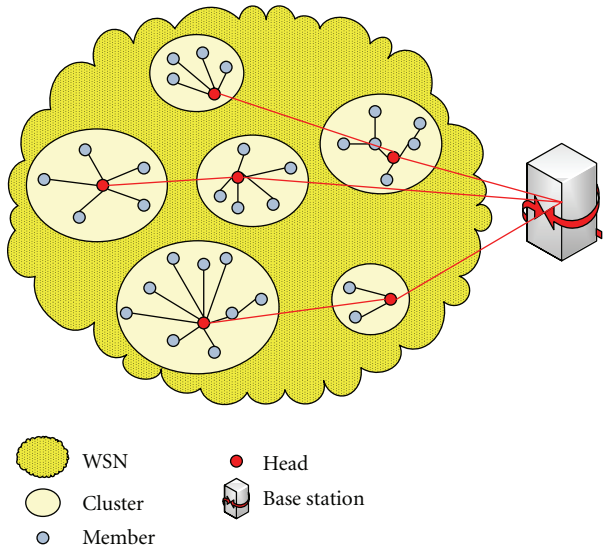


Figure 2: Topology of the cluster-based WSN.

heads constitute a higher hierarchy connectivity network and take the work of long-distance routing forwarding.

(2) Data collected from members will be forwarded after it is fused. This mechanism reduces communication energy cost because the data traffic is reduced.

(3) Without the need of maintaining complex routing information, member nodes have simpler function. This characteristic will further reduce the energy consumption of communication.

(4) Clustering routing has good scalability, and it is suitable for large scale networks and distributed algorithm [13].

(5) It is easy to overcome problems caused by mobile sensor nodes [14].

Because clustering strategy has the above benefits, it attracts extensive attention from researchers and a lot of clustering algorithms are proposed in the previous literatures. The most widely used clustering methods are LEACH [15] and its variants.

2.2. LEACH and Its Variants. Low Energy Adaptive Clustering Hierarchy (LEACH) is probably the most famous clustering protocol for WSN. The basic idea of LEACH is to randomly select the cluster head nodes in a circulating way and make the energy load of the entire network evenly distributed to each sensor node to achieve the purpose of reducing network energy consumption and enhancing the overall survival time of the network. Simulation result reveals LEACH can lead an additional 15% lifetime [16].

There are two procedures in LEACH clustering. In the first procedure, each node generates a random number $\sigma \in (0, 1)$, and if σ is smaller than the threshold $T(n)$, the node will be selected as the cluster head. The threshold can be calculated as follows:

$$T(n) = \begin{cases} \dfrac{p}{1 - p[r \bmod (1/p)]}, & n \in G \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where p is the expected proportion of cluster heads in all the nodes, r is the number of current round, G is the set of nonhead nodes in the former $1/p$ round, and mod is the Modulo operator. The selected head broadcasts an Advertisement Message (ADV) using Carrier Sense Multiple Access (CSMA) MAC protocol to inform that it is a cluster head. Cluster members will decide which cluster it should belong to according to its received signal strength (RSS). Then the member nodes will send Joint-REQ messages to the cluster head. Members will connect with their head using Time Division Multiple Access (TDMA) technique. The final step of the first procedure is to send a timing schedule from heads to their slave nodes.

The second procedure of LEACH is called stable procedure. In this stage, cluster members will send their perceived data to the head node within their shared time slot. In other time, member nodes will switch to sleep state for energy saving. Once the head received all the data it will make data fusion immediately to strengthen the public signal and weaken the noise. Then the head will send fused data to sink node within single hop. After the stable stage continues a period of time, LEACH will repeat the clustering stage.

Although LEACH has higher energy efficiency than many other protocols, it still has shortcomings which limits its performance. First of all, because it assumes that each node can connect with sink node directly, LEACH is not perfectly suitable for large scale networks as other clustering algorithms. Secondly, LEACH is not suitable for sensor networks with unevenly distributed energy. Thirdly, it does not define the distribution of clusters heads in the whole network; therefore, some members may not be able to find any head in their neighborhood [17]. Moreover, the single hop communication between heads and sink node may lead to a large amount of energy consumption [18].

To overcome the drawbacks above, researchers proposed many variants of LEACH, and we call them LEACH protocol family. The most important members of LEACH family include LEACH-C [19], TEEN [20], SEP [21], mobile LEACH [22], and LEACH-EEC [23]. The performance of these improved versions of original LEACH is shown in Table 1.

As shown in Table 1, no variant of LEACH can completely solve all the problems above. However, these problems are very important for WSN routing. Therefore, routing strategy of wireless sensor networks is still an open issue especially on IPv6-based routing in WSN.

Actually, LEACH protocol is quite an effective ideology in initialing clustering. Its main problem is in the later iterative clustering loops. Besides, it is not able to take full advantage of the previous clustering information. In other words, the stochastic strategy makes it difficult to make a comprehensive decision in a cluster head selection. Moreover, selecting heads before clustering may lead to an unreasonable distribution

Table 1: Performance of LEACH protocol family.

Protocols	Features			
	Large scale	Energy uneven	Low cost	Robustness
LEACH-C	Yes	No	Yes	No
TEEN	No	Yes	Yes	No
SEP	Yes	Yes	No	Yes
Mobile LEACH	No	Yes	No	Yes
LEACH-EEC	Yes	No	Yes	No

of cluster heads, thus increases the energy consumption. Therefore, a novel WSN routing method should be proposed, which firstly makes clustering and then selects heads in the clusters. In addition, the clustering and head selection should take many indicators into account such as node power, number of hops, RSS, head selection history, and sum of nodes.

3. Adaptive Clustering

3.1. Clustering Initialization. As the analysis in the former section, LEACH is an excellent clustering initialization method. Therefore, we use LEACH in the initial round of clustering. Thence different from later rounds of clustering, in the first round, we make head selection before clustering.

The cluster head is selected after the indicator σ is generated randomly. In LEACH, each slave node connects with its head in a single hop so it is enough to consider the received signal strength. Obviously, not in all cases communications in inner clusters are single hop connections. According to the research in [24], when the distance between members and their head is no longer than two hops, the cluster has the minimal energy consumption. Therefore, we modified the clustering mechanism in LEACH to constitute clusters. The number of hops is taken into account when a slave node makes decision about which cluster head it should follow. When the broadcasted ADV messages by heads are received, members will check the RSS and the number of hops.

There are three possibilities for ADV messages: (1) all single hop packets, (2) all multihops packets, and (3) mixed packets.

In case (1), member nodes choose their head according to the RSS, which is similar to LEACH.

In case (2), member nodes select their heads which could minimize the overall number of hops of the network as below:

$$\frac{\sum_{i=1}^{N}\{\min[h(i,1),h(i,2),\ldots,h(i,M)]\}}{N+M}, \tag{2}$$

where N is the number of nonhead nodes in the whole sensor network, M is the number of cluster heads, and $h(i, j)$ is the number of hops from slave node i to cluster head j.

In case (3), we introduce a lower bound of RSS to determine whether to use a single hop or multihops in communications. In most cases, the energy consumption is lower when the number of hops is less. However, when the received signal is too weak to be processed, it may lead nodes re-send the message. Moreover, long-distance communication itself is a kind of high energy cost connection. When the RSS of a member node from a head is lower than the empirical threshold, it will be ignored. After the RSS is filtered, if only multihops ADV remains, the problem will be transformed into case (2) and the slave nodes will make their decision according to function (2). When there are still both single hop ADV and multihops ADV, member nodes will choose single hop to connect to the head as in the LEACH protocol. The schematic of clustering initialization is shown in Figure 3.

3.2. Adaptive Iterative Clustering. In the later rounds, different clustering strategy should be used; otherwise it is still a variant of LEACH. However, clustering information of the initial round should be used for the second round clustering. The number of clusters is a crucial factor which largely affects the energy efficiency of WSN. Unfortunately, there is no universal ideal proportion of heads in WSN clustering. In different situations, the same heads percentage may have a different performance, which can be influenced by application, data type, terrain, physical layer protocols, and so forth. Therefore, empirical percentage of cluster heads is not feasible, and to adjust the number of clusters adaptively according to the last round of clustering seems a reasonable idea.

First of all, when there are too many multihop connections and the RSS of most single hop connections is low in some clusters, it means the distance between members and the head is too long. In other words, the number of clusters is lower than that it should be in the normal state. In this case, the overlarge clusters should be split into two or more subclusters.

To evaluate whether a cluster is overlarge, the overall hops of the clusters, the average RSS, the energy cost in one round, and the amount of data transmission should be taken into account. When we assume the criteria is F, it can be calculated as follow:

$$F = \sum_{i=1}^{n+m} t(k)\frac{\sum_{i=1}^{n}\mathrm{RSS}(i)}{\sum_{j=1}^{m}h(j)+\sum_{j=1}^{n+m}\Delta E(j)}, \tag{3}$$

where n is the number of single hop slave nodes, m is the number of multihop slave nodes, RSS(i) is the signal strength received by head from ith single hop node, $h(j)$ is the number of hops between head and jth multihops node, $t(k)$ is the times of sending data of kth node, and $\Delta E(j)$ is the difference of the remaining capacity of jth node between the beginning and ending of the round.

Obviously, smaller value of F reveals a larger scale of the clusters. A threshold θ could be used to evaluate whether the scale of cluster is reasonable or not. When F is lower than θ, the cluster should be split. θ is an empirical parameter which is mainly depended on the energy consumption of inner-cluster communication. It is usually between $1.5\sum \mathrm{RSS}(i)$

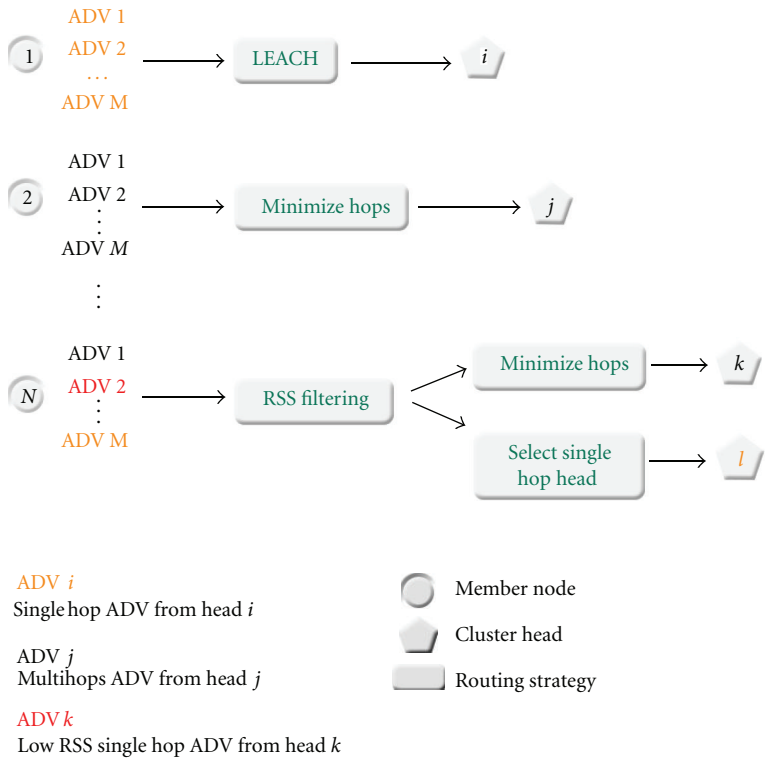


Figure 3: Schematic of clustering initialization.

and $3.5 \sum \mathrm{RSS}(i)$. The number of split clusters could be determined by the following formula:

$$C_n = \left\lfloor \frac{\theta}{F} \right\rfloor, \tag{4}$$

where C_n is the number of clusters after the split. The cluster head selection in the split clusters is similar to that in other clusters which will be presented in the next section.

Secondly, too many clusters in the WSN will also reduce the energy efficiency of the nodes, because the nodes will be selected as clusters more frequently, when the number of clusters is larger. Thence the small clusters should be merged when the proportion of heads is too high.

Different from the *split* situation, the average number of nodes in the clusters of the entire network and the frequency of the nodes selected as heads in a certain cluster should be taken into consideration. When we assume the merge criterion of the ith cluster is $F'(i)$, it can be computed as below:

$$F'(i) = \frac{N \sum_{k=1}^{n} \tau(k)}{(N+M) \sum_{k=1}^{n} t(k)}, \tag{5}$$

where n is the number of nodes in cluster $i$and $\tau(k)$ is the times node k has been selected as cluster head. N, M, and $t(k)$ have the same definition with the former functions. Similar to the split situation, a lower bound θ' could be used to determine whether to merge the cluster with its neighbor cluster or not. When $F' < \theta'$, the cluster will be considered to be too small.

When a cluster head finds that the F' is lower than θ' it will ask the adjacent heads to check whether another too-small cluster exists in its neighborhood. If there is another too-small cluster in the bordering area, the two clusters will merge and construct a new cluster. Otherwise, the too-small cluster will merge with a normal one and then be split into two clusters. The procedure is shown in Figure 4.

There is a time when the size of each cluster is appreciated but some nodes are in the wrong clusters. In this case, we need to adjust the affiliation of some nodes rather than the number of clusters. To evaluate the unsuited affiliation, the number of hops and RSS are helpful. In a multihop cluster, if the number of hops between a member and the head is significantly larger than the other slave nodes, it is better if the slave node belongs to another cluster. The significant low RSS in single hop clusters has the similar meanings. The following inequalities can be used as the criteria:

$$\begin{aligned} h(i) &> \frac{n \cdot \max_{1 \le j \le n}[h(j)] + \sum_{j=1}^{n} h(j)}{2n}, \\ \mathrm{RSS}(i) &< \frac{n \cdot \min_{1 \le j \le n}[\mathrm{RSS}(j)] + \sum_{j=1}^{n} \mathrm{RSS}(j)}{2n}, \end{aligned} \tag{6}$$

where $h(j)$ is the number of hops between the head and member j in the cluster. Other symbols have the same definitions with the former formulas. With the above inequalities, the nodes can make judgment whether its cluster affiliation is suitable or not. If the above inequalities are met, the nodes should switch to another cluster. These nodes will send a multicast request to the heads of the bordering clusters.

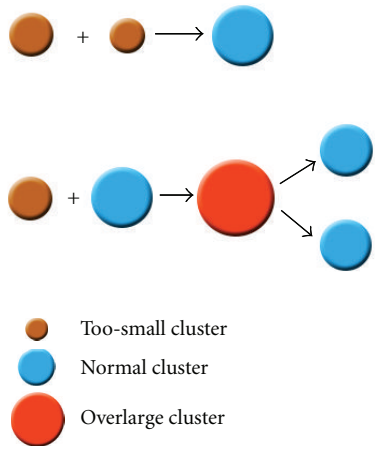


Figure 4: Merging the too-small clusters.

Then they will make a decision following the above principle according to the ACK of the adjacent heads:

$$C_{\text{NEW}} = \min_{1 \le i \le n} [\text{RSS}(i) \cdot h(i) \cdot n]. \quad (7)$$

Function (7) reveals that the nodes select the nearer and smaller cluster as its new work group in the neighborhood.

3.3. Enable Scalability. Scalability is a key aspect of WSN applications. In a WSN, when the proportion of the power-off nodes is higher than an upper bound, we need to add new nodes to the network. Sometimes we want to deploy more nodes to increase the coverage of the sensor network or to bring new functions. And sometimes we need to fuse two or multiple sensor networks into one network. All of the above cases need a scalable clustering algorithm.

To enable the WSN to be scalable, we need to provide the initial clustering strategies in the adaptive clustering algorithm, because the new added nodes and the old nodes are in different rounds.

Usually, there are two kinds of new added sensor nodes. Because the issues which should be addressed are different, new sensors scattered among the old nodes should be treated differently with the added sensors which are deployed spate with old nodes in a new area.

The two kinds of new added sensor nodes are as shown in Figure 5.

An important issue is how to detect the distribution of the new nodes. It is clear that when the new nodes are deployed manually, their distribution is known and could be used directly. However, in many situations, sensor nodes are sowed by airplanes or robots. In these cases, some strategies should be used to make the system detect the distribution of the new nodes automatically.

On this problem, head history is helpful because it not only reflects the history of how the nodes are selected as cluster heads but also reflects whether it is a new joined member or not. When a new joined node knows its neighbors' situation, it can detect where it is deployed.

According to the received head history, a new node may be classified into one of the following three cases.

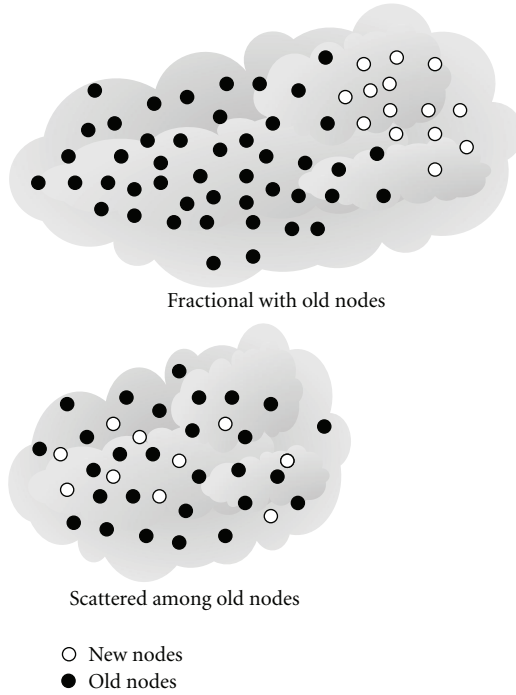


Figure 5: Two kinds of new added nodes distribution.

Case 1. All or most neighboring nodes are new joined members.

Case 2. All or most neighboring nodes are old nodes.

Case 3. In its neighboring area, new joined members and old nodes are counterparts.

The three cases above correspond to the three distribution modes of the new joined sensor nodes. Especially, Case 3 is the distribution of the boundary node in fractional distributed new nodes, and the process of distribution detection is shown in Figure 6.

The new added nodes distributed fractionally with old nodes can be considered as a new sensor network. Therefore, the strategy of its initial clustering is similar to the workflow proposed in Section 3.1. LEACH protocol is called to make the initial random head selection and sensor converge into clusters according to RSS and the number of hops.

After the clustering initialization, there is an important step named time synchronization. Time synchronization will be achieved in the second round of the new nodes by the CTSS [25] mechanism. Then the new added sensor nodes (subnetwork) can work together with the old nodes at the same rounds. During the third clustering round, the new added sensors can cluster together with the old nodes using the strategy for splitting, merging, or adjustment.

A different method is applied in the case when the new sensors are scattered among the old ones. The new nodes have to work together with the old nodes from the outset, and the initial clustering procedure of these nodes is as follows.

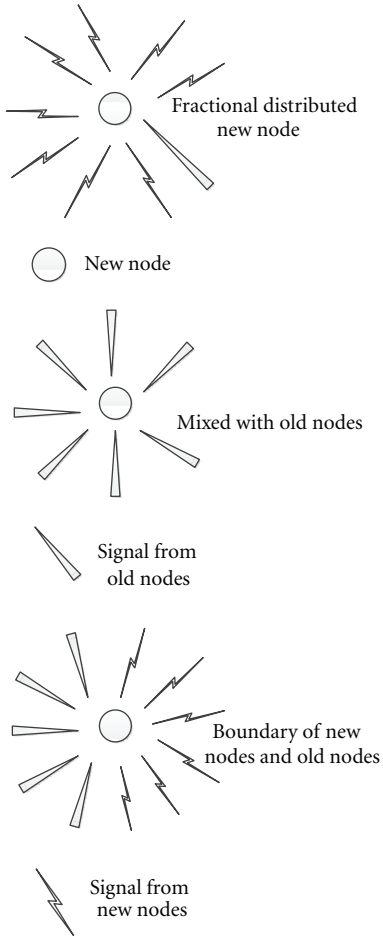


Figure 6: Distribution detection of the new nodes.

(1) The new node broadcasts an ADV to all the heads of a WSN.

(2) The ADV will be discarded when forwarded more than three hops.

(3) Each head that received the ADV will send an ACK to the new node.

(4) The new node will select a cluster according to the RSS or the number of hops of the ACK.

Following the steps above, the new nodes will find suitable clusters as their work groups and achieve Plug and Play. In the next round, the new nodes will be regarded as old nodes, so the clustering strategy proposed in former subsection can be deployed seamlessly. In this way, the WSN can achieve scalable clustering. No matter what the distribution of the new added nodes is, the new nodes can work and cluster together with the old sensors.

Hitherto, an adaptive clustering algorithm for WSN is proposed, which is shown in Figure 7.

This algorithm can adjust the scale of clusters and the members in the clusters adaptively according to the energy consumption in the former round. Therefore, it can achieve high energy efficiency. Because there is no missing condition in this algorithm; it is reliable and all functions are bounded.

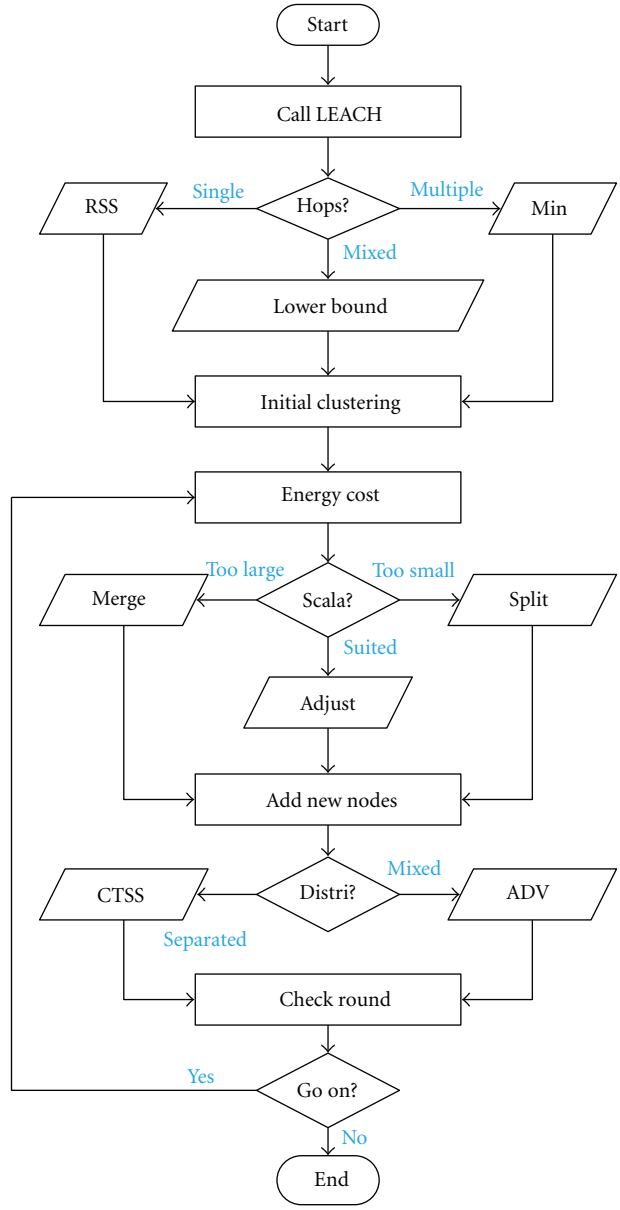


Figure 7: Flowchart of adaptive clustering algorithm.

Moreover, a comprehensive strategy is introduced in the algorithm to ensure its scalability.

4. Adaptive Head Selection

In traditional LEACH based routing protocols, head selection is the first step of clustering. This kind of strategy is a semirandom selection method and requires that the selected heads broadcast ADV to the whole network. The semirandom mechanism makes it hard to get an optimization solution. The broadcast leads to an additional energy consumption. To solve the problems mentioned above, we design an adaptive head selection algorithm which can be used to select heads based on the principle of a minimum energy consumption after the clusters is built.

4.1. Head Selection in Unbalanced Stage. Unbalanced stage is the phase of the first few rounds when not every node in the cluster has been a head. At this stage, average hops and the RSS between a node and each other cluster member are unknown. So the remaining capacity of nodes and the head selection history is the most important reference for head selection.

All the nodes except the current head will be the new head candidate. They send their information to the current head. Assuming the weight of node i for head competition is $\delta(i)$, in the simplest case, we select the head according to the following formula:

$$\delta(i) = \frac{E_r(i)}{t_h(i) \cdot (r - r_i')}, \tag{8}$$

where $E_r(i)$ is the remaining capacity of node i, $t_h(i)$ is the times node i has been selected as a head in history, r is the current round of the cluster, and r_i' is the most recent round in which node i is a head.

The known RSS and number of hops can be regarded as the sample of overall RSS and number of hops. Image the situation that there is a lot of power in the battery of a node that has not been selected as a head for a long time, intuitively it is the ideal choice for head selection. However, this kind of node may have a long distance from the other members in the cluster. The only reason they have high remaining power is that they perceived and transported little data. Using these nodes as heads will not only enhance energy cost of the whole cluster but also make them die sooner for power exhaustion. Therefore the threshold of maximum number of hops and minimum RSS should be introduced to avoid the wrong selection of the heads. The thresholds are defined as follows:

$$\begin{aligned} \mathrm{RSS}(t) &= \frac{\sum_{i=1}^{N} \mathrm{RSS}(i)}{2n}, \\ h(t) &= \frac{2\sum_{i=1}^{N'} h(i)}{n}, \end{aligned} \tag{9}$$

where N is the number of RSS recorded by node i and N' is the sum of the number of hops recorded by node i. When the thresholds of RSS and number of hops are taken into account, the weight $\delta(i)$ of node i for the head selection in a cluster can be defined as

$$\begin{aligned} &0, \quad \mathrm{RSS}(i) < \mathrm{RSS}(t) \cup h(i) > h(t), \\ &\delta(i) = \frac{E_r(i)}{t_h(i) \cdot (r - r_i')}, \quad \text{otherwise.} \end{aligned} \tag{10}$$

In this way, the current head could select its successor through their weights. According to Bayesian Theory, the average duration of an unbalanced stage T_U is

$$T_U = \frac{m!}{m^k}. \tag{11}$$

The performance of impaction in unbalanced stage is analyzed latter.

4.2. Head Selection in Balanced Stage. When the clustering iteration runs many rounds, every node will have the experience of being a cluster head. When the head selection executes to this phase, each node in a cluster has the information of RSS and the number of hops of all the other cluster members. As we know, RSS and the number of hops have a major impact on energy consumption, so they must be considered when we make the head selection.

The average received signal strength of node j from other cluster members $R_a(j)$ can be calculated as

$$R_a(j) = \frac{\sum_{i=1}^{n} \sum_{k=1}^{T_i} \mathrm{RSS}(i,k)}{\prod_{i=1}^{n} T_i}, \tag{12}$$

where T_i is the times of single hop transmissions between node j and i, $\mathrm{RSS}(i,k)$ is the received signal strength of node j from node i's kth transmission, and n is the number of members in the cluster.

Similar to function (12), the average number of hops between node j and all the other nodes $h_a(j)$ can be calculated as

$$h_a(j) = \frac{\sum_{i=1}^{n} \sum_{k=1}^{T_i'} h(i,k)}{\prod_{i=1}^{n} T_i'}. \tag{13}$$

In (13), T_i' is the number of multihop transmissions between node j and i. The definitions of other symbols are the same as (12).

In addition, the proportion $P_M(j)$ of single hop transmissions of node j can be computed as

$$P_M(j) = \frac{\sum_{i=1}^{n} T_i}{\sum_{i=1}^{n} T_i + \sum_{i=1}^{n} T_i'}. \tag{14}$$

Greater number of hops means a higher energy consumption, and a higher RSS means a higher energy efficiency [26]. Therefore, formula (8) can be modified when R_a, h_a, and P_M are taken into account. The new head selection weight of node j is

$$\begin{aligned} &0, \quad \mathrm{RSS}(j) < \mathrm{RSS}(t) \cup h(j) > h(t), \\ &\delta(j) = \frac{P_M(j) \cdot R_a(j) \cdot E_r(j)}{h_a(j) \cdot t_h(j) \cdot \left(r - r_j'\right)}, \quad \text{otherwise.} \end{aligned} \tag{15}$$

Before the head selection weight calculation, nodes with the maximal h_a and minimal R_a will be excluded from the candidate list. Then the current head will select heads according to the weights of the members. Node with the highest weight will be selected as the head of the cluster in the next round.

4.3. Deal with New Members. The former discussion solved the problems of head selection for the old nodes in a cluster. Clusters often have new members because of the scalability of the system and the clustering strategies for *merging* and *adjustment*. There is no information available, such as average RSS, average number of hops, and percentage of single hop transmissions, for head competition. However,

sometimes new members are ideal candidates for the cluster heads. Therefore, methods should be proposed to deal with the new members of clusters.

When new members are the minority of a cluster, it is easy to be treated. Because only a few new nodes exist in the cluster, we keep them out of the head competition until next round unless they satisfy a strong constraint as below:

$$C_{\text{NEW}} = \max_{1 \le i \le n} [C_i], \tag{16}$$

where C_{NEW} is the remaining capacity of the new joined node and C_i is the remaining capacity of node i in the cluster. If the power of a new sensor is not the highest of the cluster, it will be excluded from the candidate list in the first round. And the unbalanced strategy will be used for head selection from the second round until each node has the experience of working as a head.

When two or more clusters merge into one cluster, the selection history of each cluster is useful for the head competition of the new merged cluster. And a two-stage head selection mechanism is used for this case. The heads in the last round before merging will select their successors, respectively, and make data exchange to compare the remaining capacity of the candidate successors. Finally the node which has the most power will be selected as the head of the new cluster.

The simplest situation is when the new nodes are deployed fractionally with the old nodes. In this case, head selection is similar to other part of the network using unbalanced strategy. After the fusion of the new jointed part, the problem is converted to the same problem described in the former paragraph.

Hitherto, the adaptive head selection algorithm is presented completely with no missing condition. It will run together with the adaptive clustering algorithm in a cross-iterative way which is called double adaptive clustering hierarchy (DACH) in this paper. The initial round of head selection is based on adaptive clustering. When the entire network is divided into clusters, the head selection algorithm will operate in each cluster. After the data transmission of the first round, the second round clustering algorithm will function based on the head selection of the first round. The iteration will be executed. The detailed workflow of the adaptive head selection is shown in Pseudocode 1.

The inner-cluster communication is achieved by DACH algorithm based on IEEE 802.15.4. Clusters will play a key role in intercluster communication based on IPv6 to make the WSN access to the internet easily and keep energy consumption in an acceptable scale at the same time.

```
Input: cluster c = {n1, n2, ..., nn} of a sensor network
Output: cluster head h of c
1  begin
2    initial head selection using LEACH
3    clustering initialization
4    for (i = 0, i <= n − 1, i + +)
5      send clustering information to current cluster head.
6      if ni is a new cluster member of minority situation
7        make head selection according to formula (16)
8      else if ni is new cluster member of merge case
9        goto step 2
10     else
11       unbalance? selection by formula (10): continue
12     send the full information to current head
13   head selection according to received information
14   check the timestamp
15     if (timestamp <= Δt)
16       data transmission
17     else
18       clustering using adaptive clustering algorithm
19   check whether the ending of the task
20     if (not)
21       goto step 2
22     else
23       break the algorithm
24 end
```

Pseudocode 1: Adaptive head section algorithm for WSN.

5. Intercluster Communication Based on IPv6

Traditionally, WSN accesses to the internet through gateway or directly access to the internet with full IP. Access to the internet through gateway is a low energy and low computational cost solution. However, with this method, only one access point exists in the sensor network. Once the gateway fault happens, the connection between the internet and WSN will be cut off and the WSN becomes blind. Moreover, single access point method tends to cause data congestion. On the other hand, full IP sensor networks will lead to huge energy consumption and greatly shorten the lifetime of WSN. To solve the problems above, we make the cluster heads communicate with the internet based on IPv6 and use IEEE 802.15.4-based inner-cluster connection.

Because the protocol header of IPv6 is overlong for the MAC protocol of WSN, the protocol stake optimization is needed for implementing IPv6 in sensor nodes.

The head structure of IPv6 is shown in Figure 1. Therefore, the compression is focused on address. IPv6 address is composed of address prefix and subnet address. The address prefix is the same when get IPv6 address using the stateless automatic address configuration. Moreover, stateless automatic address configuration achieved according to the data link address of nodes. Therefore, the header compression algorithm proposed by Junwei et al. [27] is a wonderful method. The procedures to implement IPv6 in cluster heads are as follows:

(1) header compression using Xu's algorithm;

(2) IPv6 package fragmentation;

(3) communication with sink nodes, gateways, or other access points through IPv6;

(4) IPv6 package fragments reorganization;

(5) decompression of the received data.

Following the above steps, the cluster heads can achieve IPv6-based communication with the internet through Xu's

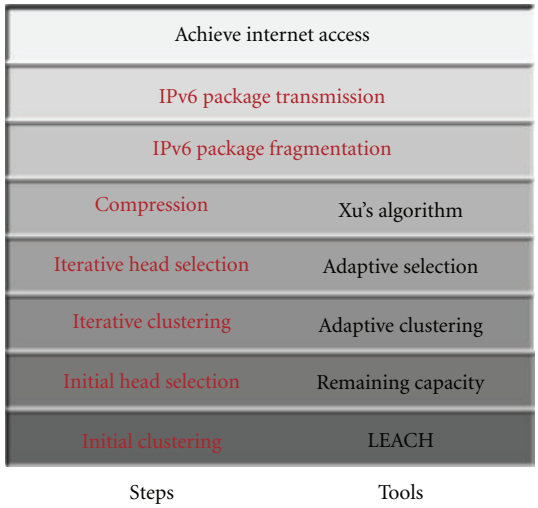


Figure 8: Framework of the whole system.

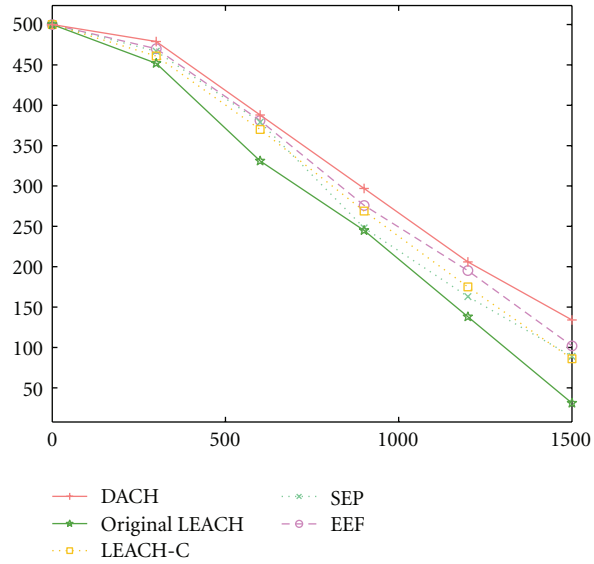


Figure 9: Energy efficiency of different algorithm.

fragmentation algorithm. The complete framework of communication between WSN and IPV6 presented in this paper is shown in Figure 8.

As is shown in Figure 8, a novel WSN routing algorithm is proposed in this paper. The structure of sensor networks is cluster based. IEEE 802.15.4 protocol is used in inner-cluster communication to save energy and IPv6 protocol is used in intercluster communication to make the sensor network access to the internet. Comprehensive factors, such as remaining capacity, RSS, number of hops, and head history, are considered when making the clustering decision and head selection to minimize energy consumption and prolong the network's lifetime.

6. Simulation, Experiment, and Analysis

We make simulations and experiments to evaluate the performance of the novel algorithm. The simulation and experimental results were compared with other algorithms to get an objective and comprehensive conclusion.

6.1. Simulations for Energy Consumption. Lifetime of nodes is an important indicator of energy consumption. We make simulation using Matlab and compare the results of 1500 rounds with several classic WSN clustering algorithms such as original LEACH, LEACH-C, SEP, and EEF [28]. The simulation results are shown in Figure 9.

As is shown in Figure 9, DACH has the longest lifetime in the comparison group. Its energy efficiency is significantly higher than the original LEACH protocol. Moreover, after 1500 rounds of clustering, the number of survived nodes using DACH is 25 percent more than the second best algorithm.

Some WSN routing algorithms are suffering a large scale sensitive problem, which means the energy consumption of a round will increase exponentially with the increase of node number. This problem will seriously affect the performance

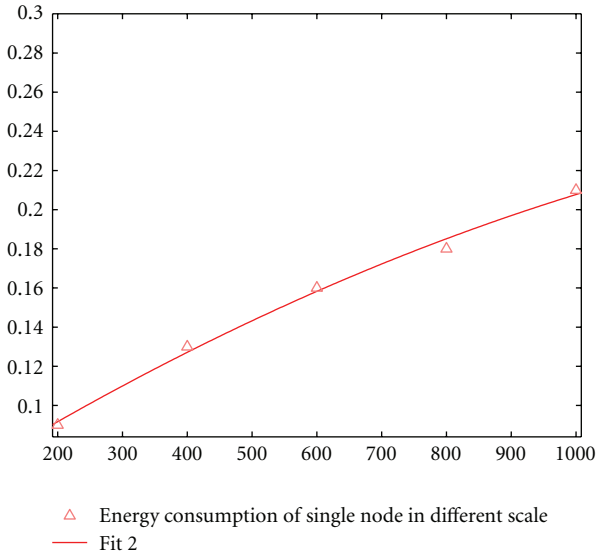


Figure 10: Energy consumption of single node using DACH in different sizes of WSN.

of clustering algorithm and significantly shorten the lifetime of sensor networks.

We simulate the case of energy consumption of a node sending 10 packages when the scale of the whole network is 200 nodes, 400 nodes, 600 nodes, 800 nodes, and 1000 nodes, respectively. The average energy consumption of a single node in WSN with a different scale is shown in Figure 10.

As is shown in Figure 10, the average node energy consumption grows slowly with the increase of the network scale. It means nodes using DACH algorithm do not face a large scale sensitive problem. The algorithm is reliable with different size.

Another important issue that should be discussed is the performance impaction of an unbalanced stage which

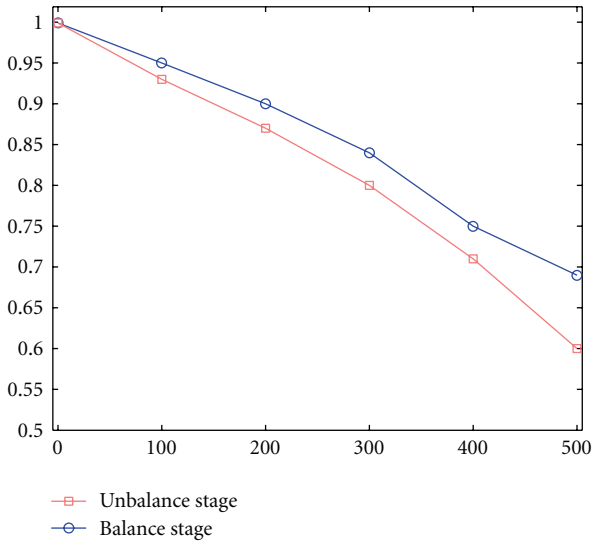


Figure 11: Impaction of unbalanced and balanced stages.

Figure 12: Sensor node used in the experiment.

is analyzed in Section 3.3. To evaluate its impaction, we simulated the remaining battery capacity in balanced and unbalanced stages after 100 rounds, 200 rounds, 300 rounds, 400 rounds, and 500 rounds, respectively. The simulation results are shown in Figure 11.

Figure 11 reveals that the energy consumption of the unbalanced stage is higher than that of the balanced stage. After 500 rounds, the nodes in the unbalanced stage have a capacity nearly 10% less than nodes in the balanced stage. An alternative strategy is making the n nodes of the cluster be heads ordinarily in the first n round.

6.2. Experimental Results and Analysis. We use 50 sensor nodes to form a real-world system using IRIS. Each of these node uses two 1.5 V, 1500 mA AA batteries. The picture of sensor nodes is shown in Figure 12.

We make the sensor network connect with the internet in different ways including full IP connection, static heads connection, LEACH-based connection, and DACH-based

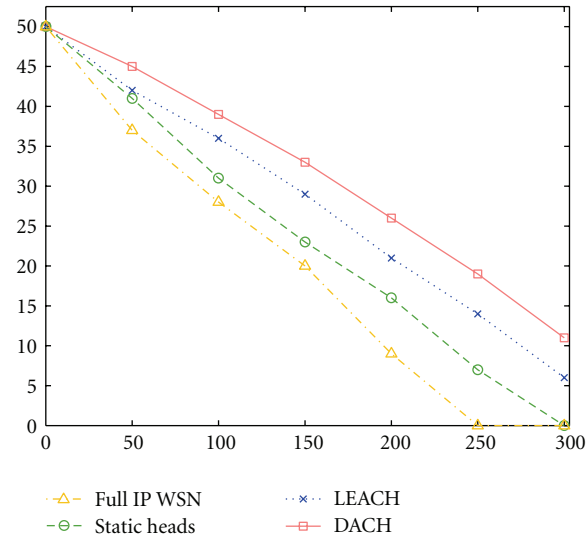


Figure 13: Lifetime of different algorithms.

connection. Each algorithm was tested by 300 rounds. Each round contains 10 measurements. Therefore, each node sends 3000 packages. The experimental results are shown in Figure 13.

Figure 13 reveals that DACH has the lowest energy consumption. WSN based on DACH which contains 50 sensor nodes has more than 100 rounds longer lifetime than the full IP-based routing.

7. Conclusion

IPv6 may bring a lot of benefits to sensor networks because of its high security, high robustness, mass amount of address, and so forth. However, the high energy cost of IPv6-based communication makes it difficult to be used in WSN. This paper proposed a double adaptive clustering hierarchy algorithm. After LEACH is called for clustering initialization, DACH will make clustering and head selection adaptively according to RSS, number of hops, remaining capacity, and head selection history. The comprehensive consideration which takes all energy cost factors into account leads to high energy efficiency and makes the energy consumption distribution even more. Therefore, the nodes and the whole sensor networks based on DACH can achieve access to the internet through IPv6 with a significantly longer lifetime.

The mobility of this algorithm is not verified. In addition, package loss rate should be further reduced. This will be undertaken as a future work on this topic.

Acknowledgment

This research was supported by CNGI.

References

[1] M. B. Krishna and M. N. Doha, "Self-organized energy conscious clustering protocol for wireless sensor networks," in

Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT '12), pp. 521–526, March 2012.

[2] L. D. Paulson, "Will wireless be IPv6's killer app?" *Computer*, vol. 34, no. 1, pp. 28–29, 2001.

[3] E. Lee, S. Park, F. Yu, and S. H. Kim, "Data gathering mechanism with local sink in geographic routing for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1433–1441, 2010.

[4] W. Xiaonan and G. Demin, "An IPv6 address configuration scheme for All-IP wireless sensor networks," *Ad-Hoc and Sensor Wireless Networks*, vol. 12, no. 3-4, pp. 209–227, 2011.

[5] A. Dunkels, J. Alonso, and T. Voigt, "Making TCP/IP viable for wireless sensor networks," in *Proceedings of the 1st European Workshop on Wireless Sensor Networks*, pp. 1–9, November 2003.

[6] P. A. C. S. Neves, A. Esteves, R. Cunha, and J. J. P. C. Rodrigues, "User-centric data gathering multi-channel system for IPv6-enabled wireless sensor networks," *International Journal of Sensor Networks*, vol. 9, no. 1, pp. 13–23, 2011.

[7] W. Xiaonan and Q. Huayan, "An IPv6 address configuration scheme for wireless sensor networks," *Computer Standards and Interfaces*, vol. 34, no. 3, pp. 334–341, 2012.

[8] Y. A. Liu, L. Ye, Q. M. Shao, and B. H. Tang, "Integrating wireless sensor networks with the TCP/IP networks," *Journal of Beijing University of Posts and Telecommunications*, vol. 29, no. 6, pp. 1–5, 2006.

[9] H. Khaleel, F. Penna, C. Pastrone, and R. Tomasi, "Frequency agile wireless sensor networks: design and implementation," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1599–1608, 2012.

[10] J. Aweya, M. Ouellette, D. Y. Montuno, and K. Felske, "Circuit emulation services over Ethernet—part 1: clock synchronization using timestamps," *International Journal of Network Management*, vol. 14, no. 1, pp. 29–44, 2004.

[11] J. Wan and Q. Liu, "Efficient data association in visual sensor networks with missing detection," *Eurasip Journal on Advances in Signal Processing*, vol. 2011, Article ID 176026, 2011.

[12] L. Xiaojuan, Y. Huiqun, and W. Xiang, "Energy-aware topology evolution model with link and node deletion in wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2012, Article ID 281465, pp. 1–14, 2012.

[13] M. Youssef, A. Youssef, and M. Younis, "Overlapping multihop clustering for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1844–1856, 2009.

[14] Z. Quan, A. Subramanian, and A. H. Sayed, "REACA: an efficient protocol architecture for large scale sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 2924–2933, 2007.

[15] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, p. 223, January 2000.

[16] A. Nayebi and H. Sarbazi-Azad, "Performance modeling of the LEACH protocol for mobile wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 812–821, 2011.

[17] J. Jinbao and Z. Lijun, "An advanced LEACH protocol for wireless sensor networks," in *Proceedings of the International Conference on Modeling, Simulation and Optimizaion*, pp. 346–349, December 2009.

[18] X. Gao, Y. Yang, and D. Zhou, "Coverage of communication-based sensor nodes deployed location and energy efficient clustering algorithm in WSN," *Journal of Systems Engineering and Electronics*, vol. 21, no. 4, pp. 698–704, 2010.

[19] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S8–S13, 2005.

[20] A. Manjeshwar and D. Agrawal, "TEEN: a routing protocol for enhance efficiency in wireless sensor networks," in *Proceedings of 15th International Parallel and Distributed Processing*, pp. 26–35, August 2001.

[21] G. Ragdakis, I. Matta, and A. Bestavros, "SEP: a stable election protocol for clustered heterogeneous wireless sensor network," in *Proceedings of The 1st International Workshop on SANPA*, pp. 107–115, March 2004.

[22] A. Papadopoulos, A. Navarra, J. A. McCann, and C. M. Pinotti, "VIBE: an energy efficient routing protocol for dense and mobile sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 4, pp. 1177–1190, 2012.

[23] Asaduzzaman and H. Y. Kong, "Energy efficient cooperative LEACH protocol for wireless sensor networks," *Journal of Communications and Networks*, vol. 12, no. 4, pp. 358–365, 2010.

[24] M. Horauer, K. Schossmaier, U. Schmid, and R. holler, "PSynUTC-evaluation of a high-precision time synchronization prototype system for Ethernet LANs," in *Proceedings of the 34th IEEE Precise Time and Time Interval Systems and Application Meeting*, pp. 263–277, December 2002.

[25] A. S. Hu and S. D. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2725–2748, 2006.

[26] T. Guodong, Z. Kougen, and Y. Ge, "A mobile-beacon-assisted sensor network localization based on RSS and connectivity observations," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 487209, 14 pages, 2011.

[27] X. Junwei, Z. Baohua, and Q. Yugui, "Research on IPv6 routing algorithm in WSN," *Journal of University of Science and Technology of China*, vol. 37, no. 2, pp. 1534–1538, 2007.

[28] H. Jiang, S. Jin, and C. Wang, "Framework for clustering-based data collection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 1064–1071, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 452686, 11 pages
doi:10.1155/2012/452686

Research Article

An Effective Find and Replicate Strategy for Data Communication in Intermittently Connected Wireless Ad Hoc and Sensor Networks

C. Poongodi[1] and A. M. Natarajan[2]

[1] *Department of Information Technology, Kongu Engineering College, Erode 638-052, India*
[2] *Department of Computer Science, Bannari Institute of Technology, Erode 638-401, India*

Correspondence should be addressed to C. Poongodi, poongodi321@yahoo.co.in

Received 12 July 2012; Revised 10 October 2012; Accepted 24 October 2012

Academic Editor: Deyun Gao

ICWASNs are a kind of wireless networks where, due to mobility of nodes and lack of connectivity, there may be frequent disconnections among the nodes. Hence, the routing path from the source to destination will not be available always. It is proven that, in these networks, messages are replicated multiple times in order to withstand the maximum delay and to achieve high throughput. But these multiple replication-based protocols result in an increase in network overhead and high resource consumption because of uncontrolled replication. Previous works in ICWASNs assume that networks will always have multiple partitions, and each message will be routed using store and forward mechanism. If source and destination are connected, then it is not needed to replicate multiple copies and waste the resources. We introduce a new simple scheme which applies single-copy routing if destination is available; else it switches to store and forward routing. The proposed system tries to reduce the average number of message replications while increasing the throughput.

1. Introduction

Basically, a mobile ad hoc network (MANET) is an infrastructureless and independent collection of various mobile nodes where the topology of network changes dynamically and unpredictably. MANET well complement infrastructure-based wireless networks and allows mobile node users to obtain access to interact directly with each other even when they are outside the coverage area of any cellular networks or Wi-Fi. Similarly, MANETs enable communication between vehicles, sensors, laptops, and other mobile equipment without the need to deploy a fixed infrastructure network. Plenty of dedicated routing protocols have been proposed to establish and maintain connectivity between communicating nodes in such dynamic environments.

Challenged networks or intermittently connected wireless ad hoc and sensor networks (ICWASNs) arise from MANETs primarily as a result of node mobility but may also come into being as a result of disconnections due to power management or interference. As a result, the network becomes partitioned. Unfortunately, with current ad hoc routing protocols, packets will not be routed if a disconnection exists between the source and the destination when a message is originated. Certain applications, such as real-time, constant bit rate communication may require a connected path for meaningful communication. However, a number of other application classes benefit from the eventual and timely delivery of messages, especially in the case where frequent and numerous network partitions would prevent messages from ever being delivered end to end [1]. Examples of such networks include terrestrial mobile networks, exotic media networks, military ad hoc networks, and other sensor networks. These challenged networks are characterized by high latency, bandwidth limitations, high error probability, node longevity, or path stability that are substantially worse than is typical of today's TCP/IP-based networks [2–4].

Existing TCP/IP-based ad hoc network protocols could not be utilized for these challenged networks since they operate on the basis of providing end-to-end inter-process communication with different link layer techniques. Numbers of

assumptions are made regarding the overall performance of the underlying links so as to achieve proper operation, there must be an association between a data source and its destination, the round trip time between any two node pairs in the network is not too large, and the link failure probability is small. Unfortunately, challenged networks which may violate one or more of these assumptions; hence, they may not be well served by the current end-to-end TCP/IP model.

When a direct contact routing approach is applied to this type of network, the message delivery delay is very high [5], and then they get dropped by the network if delay increases further. Since many of the wireless nodes are mobile nodes they may increase the delivery ratio in wireless environment [6]. Hence, by using the mobility property of the nodes, there are numbers of routing protocols proposed for these partially connected networks, and they are divided into replication based and knowledge based [7]. Replication-based protocols create multiple copies of a message. In order to maximize the probability of a message successfully transferred is to replicate as many copies of the message with the belief that at least one of the copies may reach its destination as said in [1]. In this flooding based protocol, each node tries to forward each message to every one of its nearby nodes except the source node. This results in every message being duplicated into all reachable parts of the network.

The other kind of protocols are knowledge-based [8]. Here, some knowledge about the network is required but they consume fewer resources compared to flooding strategies. In location-based routing stated in [9], a node requires location coordinates of its own, destination coordinates and the coordinates of the potential next hops. With these, a node can easily compute the distance function and determine where the message should be sent.

In this paper, we focus on studying effective find and replication (FIRE) strategy in realistic network environments with limited resources like bandwidth and buffer. The existing routing protocols are proposed under the assumption of always the source and destination are disconnected. But it is not true with challenged networks. Due to mobility and other reasons, the source and destination may be connected for some of the time and may go disconnected for some other time. Therefore, the proposed work makes use of the connections whenever possible. It will try to establish a connection from source to destination first. If the destination is not available for a specified delay then the source will switch to store and forward switching. Hence, at last the average number of copies replicated will be reduced based on the destinations availability.

2. Related Work

Ad hoc network nodes are often disconnected from the network because of their resource constraints like low battery or any link failure or due to mobility of nodes. Routing in these partially connected or disconnected networks will not succeed if it uses any of the table driven or on-demand ad hoc routing protocols since they expect the availability of the destination. But for the partially disconnected networks, the destination may be unreachable or not currently available by that time. If this is the case for the messages in MANET, then those messages will be dropped after some retransmissions. But, soon different links may come up and down owing to mobility of different nodes. It then indicates after some time interval the sequence of connectivity graphs indicating an end-to-end path may exist overtime. It implies that a message could be sent over an existing link, get stored at the next node until another node in the path comes up, and so forth, until it reaches its destination. This is referred as mobility-assisted routing which is different from that of the existing routing techniques.

The first routing method which was proposed for partially connected network is the epidemic routing is one of the mobility-assisted routing which was introduced as a different approach for partially disconnected ad hoc networks [1]. In this, random pairwise exchanges of messages occur among adjacent mobile nodes. The movement which is natural in the mobile nodes themselves made use of in assisting the distribution of the data when a network is connected partially. The epidemic algorithm is purely flooding-based, and it occupies more system bandwidth, node buffer space and energy consumption for the eventual delivery of a message. The hop count is to be fixed based on the message priority and the acknowledge mechanism is optional here.

The epidemic routing results in massive duplication of messages. Therefore, in order to avoid wasting of resources, the other one approach that has shown good potential in this context is the controlled flooding or optimized epidemic [7]. Here, the number of messages is limited by time to live, kill time, forward transmission count, and so forth. Efficient buffer management schemes are introduced in order to properly fill the buffers of all the nodes [10, 11]. Other approach is the spraying-based routing. In spraying method [12–14], a small preset amount of copies are generated and distributed to different relay nodes, each of which then buffers its copy until it meets the destination. By routing multiple numbers of copies independently, these protocols create possibility to travel around the network efficiently, while keeping resource consumption per message low. However, the shortcomings in these works are if mobility of nodes restricted to a small area, none of the messages might ever see the destination in case of spay and wait routing method. But spray and focus will overcome this difficulty.

In spray and focus, during spray phase, the preset amount of copies is thrown as like in spray and wait. During focus phase, a message will be transferred to next relay node when it has utility value which is greater than its own. Utility function can be taken as like the age of last encounter time of nodes. Another variation in spraying phase can be binary spraying or source spraying. In source spraying, source only sprays the copies. The relay node which got the copy from source will enter into wait phase or focus phase. But in binary spraying, source will copy half of the copies to a relay node. The relay node again copies the half of those copies to another relay node. When the copy becomes one, the relay nodes copy only to the destination and enter into wait or focus phase. Another kind of spraying called mutliperiod spraying will reduce again the fixed number of copies to some

extent [15]. In this, it will initially spray smaller number of copies which was very less than that of the existing spraying schemes. If acknowledgement comes from the destination, then it will stop spraying further. This periodical spraying can be done with two or three periods. But the disadvantage of spraying method is that it will find for optimal distribution strategy for fixed number of copies in spray phase.

In [16], it discusses the issue of disconnected nodes and gives the solution for sending messages to nodes in that environment. But the limitation here is that the movement information of nodes and trajectories should be known. Another one protocol that limits the flooding is the MaxProp [17]. MaxProp consists of an ordered-queue based on the destination of every message, ordered by the predicted likelihood of an upcoming transitive path to that destination. When any two nodes meet, they first exchange their predicted likelihood vectors of node meetings. Ideally, every node has the information of up-to-date vector for likelihood from every other node. The shortest path can be computed by a node through a depth-first search. Here the path weights indicate the probability that the link will not occur. Then these path weights are added to determine the entire cost of the path. Computation is done over all of the possible paths to the destinations desired and also for all the destinations of messages currently being held. The least total weight path is selected as the cost for that particular message destination. The messages are then listed by destination costs and then transmitted or dropped in that order for buffer management.

MaxProb has the advantages measuring the likelihood of the nodes and then transferring. But the overhead involved is that the path cost is updated for a destination based on its intermediary's likelihood also. In addition to the likelihood, the priority is given for the messages which are having a less hop count compared with some threshold value. But the hop count will not give a correct measure for a control of duplicated messages.

The basic idea in [18] is making each mobile node learn the random mobility parameters and similar pattern mobile nodes can join together with other nodes. The similar mobility nodes form a cluster. All the nodes in a cluster can then share their resources for overhead decrease and also for load balancing so as to improve the overall network performance. Contact probabilities are used to form a set of functions including Sync(), Leave(), and Join(). These are used for cluster formation and gateway selection. At last, the gateway nodes exchange network information and then routes.

Due to the discontinuous connections in intermittently connected networks, there are errors in the estimated contact probability of nodes; convergence and stability are the major challenges here. An exponentially weighted moving average scheme is employed for the contact probabilities for on-line updating with its mean shown for converging to the true contact probability.

In [19], they aim to store the data inside the network for maximum of the time, by spreading the data items from lower energy nodes to higher energy nodes. Another novel relaying scheme is that probabilistically determines a vehicle's suitability to carry messages [20]. Hence, messages are released to a current vehicle if and only if the current vehicle contributes to make mean transit delay. It utilizes the release probability which is quantifying the contribution of a vehicle in a present opportunity to the minimization of the overall mean message transit delay.

The paper [21] discussed the past history of nodes encounters and selected the best relay node based on the high encounter value. It will just look into the past history but did not consider the change of time history and buffer size. There are other protocols like RAPID which replicates packets which results locally in the highest increase in utility [22], and message ferries approach [23–26] discussing dedicated nodes to connect partially disconnected networks. In [27], the author has proposed look-ahead routing and message scheduling (ALARMS) scheme in which the ferry nodes inform the gateway nodes about their travel schedule so that the gateway nodes can decide which ferry to use for each message in advance and schedule the message in the queue accordingly. In [23], the cluster head nodes act as an intermediary between normal nodes and ferry. The nodes themselves have to register with the head of the cluster about their recent location information. Thereafter, the cluster head will then route the messages to the destination node by selecting the correct ferry which is going to that location. But the approaches using message ferry will fail in the case of absence of dedicated nodes.

The existing works assumes the disconnection among the nodes persist always. But the disconnections in ad hoc nodes will also not be consistent over a period of time. Therefore, whenever connectivity is available among the nodes that should be utilized. The proposed work, FIRE, discusses controlled replication by adapting single copy forwarding when connectivity exists, if not, later switch over to multiple copy or store and forward routing. Here, the multiple replications of messages are greatly reduced when it is not needed. Therefore, due to buffer size and energy constraints, the messages are selected for multiple replications only when a node could not find the destination within its proximity.

While going for multiple replication of the same message, it should result in minimum number of replicas in the network in order to increase the throughput and also to reduce the transmission delay. However, a node has no precise knowledge in controlling the replication of the messages to intermediate nodes. Therefore, it is very difficult to select the best intermediate node for transmission. Here, while replicating, the number of copies are reduced by replicating the message only to the relay nodes which have high probability to meet many number of nodes.

3. Find and Replicate Strategy

3.1. Basic Model. In intermittently connected networks, single copy routing fails when disconnections exist for a long time. Hence, multiple replications are needed for a message to reach the destination via store and forward switching. While replicating multiple messages of the same copy, the disadvantage is that the resources like buffers, battery are to be taken into account since multiple copies

engage the resources heavily. Therefore, while designing a multiple replication-based protocols, number of copies of a message should be reduced as much as possible at the same time increasing the throughput probability and reducing the average end-end delay.

The existing routing protocols assume that the destination is not accessible from the source or from any relay node while forwarding. As a result, whenever a source or a relay node has a message, it is forwarded with multiple copies to reach the destination soon. But this is not always the case with the wireless network. Based on the mobility of the nodes, the disconnections are not persistent. Also, it is not that every source and destination may not get connected always. Therefore, we are trying to utilize the connections whenever possible. The FIRE strategy tries to find for the destination initially, if it is not available then replicates multiple copies of the messages.

The proposed FIRE routing protocol is very simple. In the first phase, it initially searches for the destination. If the destination is available, then it transfers the copy to the destination via the routing information obtained by sending destination seek message and receiving destination found. If the destination is not available, then it switches to store and forward routing or controlled replication where the message is buffered for infinite delay which is the second phase.

3.2. Finite Delay Forwarding Strategy. If a source likes to send a message to any of the destinations then it initially searches the destination for its availability within the reach of the source. If it is available then, it forwards only a single copy to that destination with the route information obtained. When a destination is available, a source buffers the message for a finite time. But, if the destination is not available then the source gives the message to maximum delay buffer or infinite buffer. Infinite delay buffer forwarding is discussed in the next section.

The source finds a destination by broadcasting a destination seek message. The destination seek contains the information of the source node, destination node, sequence number, time to live, and hop count. The time to live value in the seek message limits the life time of that message. The destination seek messages need not to be propagated for a long time since the destination may not to be available now. The time to live for a destination seek message is fixed based on inter meeting times of nodes in a network. For a random way point mobility, the maximum delay in reaching the destination $\mathrm{ED}_{\mathrm{dt}}$ is derived and used in [12, 28] based on the meeting times of nodes. It is given by

$$\mathrm{ED}_{\mathrm{dt}} = 0.5N\left(0.34\log N - \frac{2^{K+1} - K - 2}{2^K - 1}\right), \tag{1}$$

where K is the transmission range of a node and N is the size of the network. Therefore, a destination seek message can propagate up to the maximum time limit which is the maximum expected delay in reaching the destination. Here, the maximum time limit is fixed for a destination seek, based on its average meeting times with other nodes. Therefore, the time to live for the destination seek is then given by

$$D\mathrm{seek}_{\mathrm{ttl}} = \frac{\mathrm{ED}_{\mathrm{dt}}}{t_2 - t_1}. \tag{2}$$

The $D\mathrm{seek}_{\mathrm{ttl}}$ is approximated directly with maximum delay in reaching the destination and inversely with difference in intermeeting times of a node with other node. The T_1, T_2 are the average times of nodes encounter with one node and with two different nodes, respectively. The empirical value of $\mathrm{ED}_{\mathrm{dt}}$ is calculated during the simulation of the scenario discussed in Section 4, and it comes around 200 s to 550 s. The average value of the $T_1 - T_2$ is calculated, and it changes about 5 s to 20 s depending upon the dynamic topology. Then the waiting time of the destination seeks also vary from 40 s to 110 s. If a node could not receive the destination reply within that time then all the messages destined for the same destination is moved to the infinite buffer.

Calculation of waiting time for the destination reply for a destination seek message depends upon the node intercontact time since the approximation of delay determines, waiting for the destination seek is reasonable or not. It is the round trip time of the delay in reaching the destination and then back to source. But, here we limited its value and it is assigned as

$$D\mathrm{seek}_{\mathrm{delay}} = D\mathrm{seek}_{\mathrm{ttl}}. \tag{3}$$

The delay calculated with the above equation is the maximum delay for the random way point mobility model. Therefore, it is reasonable for a node to wait for destination reply for the above maximum delay. Here, we relate the delay also with the meeting time since if meeting time is more than the probability to find the destination may be more. So, we make use of the opportunity to meet many nodes and then wait for the reply from the destination for finite delay.

Meeting times of nodes are considered for the delay and time to live since meeting time is small then the node may have chances to meet many numbers of nodes. If meeting time is high then passing the message for the destination seeks may go waste after some delay. Since, a network without much node movements may get disconnected frequently. Therefore, if delay goes beyond the expected maximum delay limit, messages need not to be dropped; instead the message can be replicated through multiple nodes for the destination so that the delay can be tolerated.

If the destination is found by the seek message then the destination node reply with the destination found message. The destination found is a unicast from the destination to source. When multiple seek messages reach the destination then the reply will be only the seek message which has shortest hop count. Like other MANET routing protocols, after receiving the destination found, the message will be unicasted with the same path of the destination found.

After establishing the route, while the transmission is going on, in between the destination may get disconnected from the network due to mobility or any link break down. This is the characteristics of intermittently connected network. The destination node or the next hop is not found then

the intermediate nodes sends the destination seek in order to search whether the node is within the reach of that node. If there is no reply for the seek message, then a destination not reachable message will be intimated to source, and the packet will be buffered in the intermediate as well as in the source node for infinite delay forwarding.

The destination seek can be differentiated from other protocols route request messages. Other MANET protocols like AODV and DSR use the route request messages for finding the destination. If route request is delayed then they use the expanding ring search technique to find the destination. But in destination seek, the destination is searched for a finite amount of delay and limited life time value. If it is delayed beyond that maximum expected value, then the message is shifted to infinite delay forwarding by assuming that the destination node is disconnected from the network.

3.3. Infinite Delay Forwarding Strategy. The destination seek packets are sent and the source is waiting for the reply from the destination. When the destination is not currently available then the destination seek packets may not be returned to the source. Other MANET routing protocols discard the message after some retransmission of route requests by assuming that the destination is unreachable.

But the unavailability does not mean that the destination node is permanently disconnected or shut down. It may be due to the destination node may be roaming outside but it may rejoin again or the destination may temporarily be shut down due to battery down or it is due to any link breakdown, and so forth. After some time delay, the destination node can get its messages instead of deleting its messages from the network.

In order to make a message reach its destination which is currently unavailable, the messages are put up in a maximum delay buffer or infinite delay buffer. These delayed messages are propagated into the network by the use of contact opportunity between the nodes. When a source node meets another node, the source makes use of this opportunity to send the infinite delay buffered message. Then this intermediate node again forwards it to other nodes during its encounter. At the last, after some delay, the messages reach the destination. When a destination gives an acknowledgement for the received message, it is flooded into the network. The flooding of acknowledgement deletes the duplicate copies of the messages which reached the destination.

In multiple replications, when two nodes meet, like node x and node y, they will exchange the summary vectors in order to know the messages that are to be transferred from x to y and vice versa. Uncontrolled replication transfers all the messages from node x to node y, those messages are not available in node y. But, without any knowledge about the destination of the messages and about the node y the messages are transferred. If the node y is not having a chance to meet the destination of any single message, then the resources of node y will be exhausted without any use. Thus, it is a must to derive a message replication based on some knowledge about the destination of messages and the relay node.

The messages are spread into the network from the infinite buffer in a controlled manner. Initially the source finds the relay node which is having a high intercontact or meeting update. This is to make the replication to be with only the relay nodes which are having a high chance to meet other nodes or even the destination. Since the replication with nodes which have constrained mobility need to be avoided.

The meeting update for a node is calculated based on its encounter history. Each node updates its own meeting updates when encountering a new node. When a node x meets new node y which is the nth node it meets, then the meeting update p_{xj} in node x is updated as

$$p_{xn} = \frac{\phi(1/2(n/n+1) + p_{xn-1})}{1 + \phi(t_2 - t_1)}, \tag{4}$$

where p_{xn-1} is the meeting update up to $n-1$ nodes, ϕ is the precedence given to history of encounters or the time difference $(t_2 - t_1)$ between current and previous node encounters. If first time a node meets other node then $(t_2 - t_1)$ is set as 1. If time difference elapses for a long time then the update value is very low. Therefore, if a node meets another node frequently the value will be higher. When a relay node x encounters another relay node y, it sprays its messages only when node y has $p_{yn} \geq p_{xn}$. The table below shows the simulated meeting updates calculated for a node which is meeting up to 15 new nodes in various time intervals.

The empirical values obtained for different number of nodes and during different time periods are given in Table 1. The table shows that when $t_2 - t_1$ is 2, and it is maintained in the subsequent meetings then the meeting update is increasing. Later, if $t_2 - t_1$ is increased to 6, it means that nodes are not meeting very frequently, then the meeting update is getting reduced. The different ϕ values makes the meeting update value to increase when ϕ is increased. At the same time, the $t_2 - t_1$ increase makes the meeting update to decrease dramatically even though the ϕ is increasing. The initial value of meeting update, when $n = 1$ is somewhat higher than the next few encounters. But, the initial value will go low in the next encounter.

If a message is forwarded using infinite buffer, then the expected delay in reaching the destination is given by

$$\mathrm{ED}_{\mathrm{inf}} = \mathrm{WT\ for\ finite\ delay} + \mathrm{delay\ during\ infinite\ buffer},$$

$$\mathrm{ED}_{\mathrm{inf}} = D\mathrm{seek}_{\mathrm{delay}} + \frac{D\mathrm{seek}_{\mathrm{delay}}}{N}, \tag{5}$$

where N indicates that the numbers of relay nodes which are having the good meeting time update during infinite delay forwarding. The value of N depends on the node movements and meeting with other nodes inside a network.

In a highly mobile environment, $N \approx M$, where M is the total number of nodes in a network. It is possible for

Table 1: Empirical Value Obtained for Meeting Updates for a node.

N	t_2 - t_1	Values of ϕ				
		0.5	1	2	5	10
1	1	0.083	0.125	0.167	0.208	0.227
2	5	0.06	0.076	0.091	0.104	0.11
3	5	0.062	0.075	0.085	0.092	0.095
4	2	0.116	0.158	0.194	0.224	0.236
5	2	0.133	0.192	0.244	0.291	0.311
6	2	0.14	0.207	0.269	0.327	0.352
7	2	0.144	0.215	0.283	0.348	0.376
8	2	0.147	0.22	0.291	0.36	0.391
9	6	0.075	0.096	0.114	0.131	0.138
10	6	0.066	0.079	0.087	0.094	0.097
11	10	0.044	0.049	0.052	0.054	0.055
12	10	0.042	0.046	0.049	0.051	0.051
13	5	0.072	0.085	0.093	0.099	0.101
14	5	0.077	0.092	0.102	0.109	0.111
15	5	0.078	0.093	0.104	0.111	0.114

a node to meet all the nodes inside a network. Hence, the delay approximately or closely equal to

$$\mathrm{ED}_{\mathrm{inf}} \approx D\mathrm{seek}_{\mathrm{delay}}. \quad (6)$$

In a constrained mobility environment $N \ll M$, hence, the delay becomes

$$\mathrm{ED}_{\mathrm{inf}} \approx 2D\mathrm{seek}_{\mathrm{delay}}. \quad (7)$$

Thus, the results show that in case of highly mobile environment regardless of the destination availability the delay is approximately equivalent to the expected value of directly reaching the destination. Since, in a highly mobile environment connectivity is maintained, and it makes all the messages to reach the destination via finite delay buffer itself. If mobility of nodes decreases then the delay is increased up to the extreme case of double the time of the expected delay. It actually happens during infinite delay forwarding.

Even though, the FIRE uses two phases of routing, the second phase is not increasing the delay when mobility of nodes is high. When nodes are moving slowly the delay may be increased. But the average delay is decreased since it depends on the availability of destinations. The average number of replications also gets reduced since the proposed system checks for the destination and then switch to infinite buffer forwarding. In the second phase of routing it forwards only limited copies based on the meeting probability. Therefore, the average number of replications of each message and average delay is reduced compared with existing protocols (Pseudocode 1).

4. Evaluation and Performance Analysis

We have used a discrete event-driven simulator called QualNet to evaluate the performance of different routing protocols under a large range of nodes, for different mobility and different number of messages. Although the intermittently connected or delay tolerant networks of interest are disconnected in general, they may range from extremely sparse to almost connected networks.

The proposed system is implemented and simulation results are compared with the following routing protocols: (i) epidemic routing; (ii) optimized epidemic; (iii) spray and wait; (iv) spray and focus; (v) multiperiod spraying. These protocols are explained in Section 2. Rather than other protocols discussed in Section 2, these protocols show some good metric in different scenarios in intermittently connected networks. Epidemic gives good performance when the buffer size is too large [1]. But when the buffer size is limited, then the epidemic slows down its performance due to heavy load but the spraying methods give better performance comparatively [12, 14, 15]. In spraying methods, the number of copies to be spread is limited based on the number of nodes in a network. Spraying will not exhaust the buffer of each node and hence the throughput is increased. But, spraying some limited copies also, unnecessary when destination is reachable. The nodes are resource constrained in ICWASN; hence, the performance of FIRE is checked with limited buffer size with all the protocols said above.

We need to define a meaningful connectivity metric for the network nodes, since here it is necessary to capture both disconnected and connected network environments. This kind of environment is implemented by forming some disconnected clusters. The cluster of nodes later may go around and connected to other nodes in the cluster. The connectivity is the measure of how many new nodes are encountered by a given node within some time interval and is important in situations where mobility is exploited to deliver traffic from source to destination. In the scenario, there are maximum of 100 nodes moving inside a 2500 $\times$ 2500 network. Also, 20 nodes are chosen randomly among the clusters and 25 messages are sent per pair per second throughout a run. The values obtained are the average of 15 runs with different seeds for 1000s of simulation time. Each node has a buffer space of 100 messages. The other parameters for the simulation analysis are given in Table 2.

The throughput, delay and number of messages duplicated parameters are checked for different scenarios and with different mobility too. Figure 1 shows the throughput of the different routing protocols for different numbers of nodes in the scenario. It is common in intermittently connected networks that the number of nodes raised the throughput is increased due to increase in relays. Hence, all the protocols are showing the improvement in throughput. But epidemic and optimized epidemic protocols are having very low throughput than others. Table 3 summarizes the performance of FIRE. It shows more than 100% improvement in FIRE compared to epidemic. This is due to the uncontrolled replications of each message. The nodes running with epidemic protocol transfer all their messages to nearby nodes that do not have a copy of it already. Therefore, soon the network becomes congested.

At last, a number of messages are dropped before reaching the destination. The optimized epidemic routing is used to control the message drops in a meaningful way. It utilizes

```
while (msg from node y to node x)
do
    case "data":
                do
                  if msgDstip == x
                                acceptDataAsDest();
                                floodAck(msg);
                  else
                    if routeEntry(msgDstip))
                              sendtoNexthop();
                    else
                              sendDestinationSeek(msgDstip);
                              FiniteBuffer.add(msg));
                    end if
                  end if
                  done
    case "hello":
                  do
                  if newNeighbour(y) == true
                    if InfiniteBuffer.hasMsgsForDest(y) == true
                              deliverMsgs(m)
                    end if
                    updateMeetingupdate(y);
                    requestMeetingupdate (y);
                  end if
                  done
    case "meetingupdate":
                  do
                  for InfiniteBuffer.Msgs
                    if Meetingupdate (y) >
                                          Meetingupdate (x)
                              do
                              transMsgs.add(InfiniteBuffer.getMsgs())
                              sendMsgs(y,transMsgs)
                              done
                    end if
                  end for
                  done
    case "timerExpired":
                  do
                    if routeEntry(m.timerExpired)
                      do
                        InfiniteBuffer.add(msgs(m));
                        DropfromFiniteBuffer(msgs(m));
                      done
                    end if
                  done
    case "ack":
                  do
                  if InfiniteBuffer.hasMsgsForDest(ack.d)
                              DropfromInFiniteBuffer(d);
                  end if
                  done
done
```

Pseudocode 1: Pseudo code for the FIRE forwarding.

Table 2: Simulation parameters.

Parameter	Default value
Network size	2500 m × 2500 m
Number of nodes	20 to 100
Transmission power	15 dBm
Battery model	Linear
Speed of a node	10 mbps to 50 mbps
Pause time	30 seconds
Infinite and finite buffer size	100 Messages/Node
Mobility model	Random Way point Mobility
Message size	512 Bytes
Simulation time	1000 s
Physical layer	802.11b Radio
MAC layer	802.11(Ad hoc)
Channel frequency	2.4 GHz

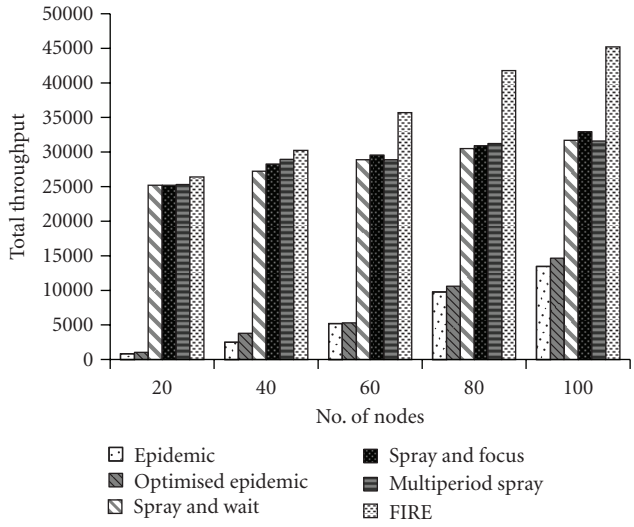


Figure 1: No. of nodes versus total throughput.

the age and hop count of the messages for deleting a message in the buffer. In this way, the message just entered into the network is propagated and the message which has duplicated in many numbers of nodes are dropped first. Therefore, controlled replication in optimized epidemic improved the performance of epidemic protocol in terms of throughput. But these protocols give very low throughput comparing with spraying methods. The nodes equipped with some moderate mobility makes the spraying methods have superior performance. The spraying methods distribute some fixed number of copies of each message; the messages are replicated only to certain number of nodes in spray phase. Owing to mobility, these copies distributed by the source may reach the destination during wait phase. The disadvantage of spray and wait is that it is not suitable for constrained mobility. Spray and focus slightly modify the Spray and Wait wherein, it finds for best relay in focus phase for further forwarding. The messages will be forwarded to best relay nodes which have recent encounter timers with the destination. But the results in Figure 1 show that the differences of those methods are smaller.

The multiperiod spraying [15] is another one version spraying methods. Here, fixed numbers of copies are distributed at some regular intervals rather than at the same time until receiving an acknowledgement from the destination. It is resulting with the same performance in terms of throughput with the above said two spraying methods. The proposed method FIRE outperforms both flooding-based and spraying-based routing protocols. Since, the FIRE will try to capture the destinations availability before replications begin. The simulation traffic is included with the combination of both destination availability and unavailability. Other protocols are going for multiple replications all the time. But the FIRE will look into the availability and make use of it whenever possible. Therefore, the buffer of each node is saved and on the average number of messages that reach the destination are higher. Unlike spray and focus, it does not have only the last encounter of destination with the relay node. Hence, the spray and focus have to wait for a relay node which has encountered the destination. But FIRE spray the copies to all the nodes which have high meeting update.

Figure 2 shows the effect of number of node increase in average end-end delay of the messages. It is also known that whenever we increase nodes in a network delay in reaching the destination is reduced. But the average end-end delay may vary based on the routing protocols. The delay measurement is shown in Figure 2 is different than that of the throughput results. Here, the delays of spraying methods are very highcompared to the epidemic and optimized epidemic.

The epidemic routing makes many copies to replicate; the messages are reaching the destination soon. But spraying limit the copies and the copies take longer time to reach the destination due to few numbers of nodes only receiving the copy. The FIRE is also lagging below about 20%-21% as in Table 3 when the nodes are minimal compared to epidemic. The spray and focus give higher delay, since the random way point mobility leads to encounter timers to go outdated soon. When the numbers of nodes are increased, in FIRE, the delay has been reduced up to 36% maximum even though it has two phases of routing.

In the first phase, if a source can reach the destination then the delay for all the messages of that source are greatly reduced than that of the multicopy forwarding. Then for a message to reach its destination, it is less than that of the delay of directly meeting the destination ED_{dt} or $Dseek_{ttl}$. On the average, it reduces the delay of all the messages. The messages will be forwarded to its destination without any drop in the network. Also in the second phase, that is if destination is not available, the messages are put up in an infinite buffer. The infinite buffered messages are again forwarded to nodes which have probability to meet many other nodes. This makes the messages to be propagated fast into the network in a controlled manner. The messages may reach the destination in second phase also without much loss and with reduced delay.

Figure 3 depicts the number of messages duplicated for different numbers of nodes in the scenario. Undoubtedly, we can see the results; it is very huge for Epidemic and optimized

Table 3: Summarization of FIRE performance versus others.

Performance of fire	Increase in nodes versus throughput	Increase in nodes versus delay	Increase in nodes versus duplications	Mobility versus delay	Traffic increase versus No. of drops
Epidemic	More than 100% ↑	20% ↓–8% ↑	84%–92% ↑	2% ↓–44% ↑	82%–96% ↑
Optimised epidemic	More than 100% ↑	21% ↓–1% ↑	83%–91% ↑	4% ↓–42% ↑	81%–95% ↑
Spray and Wait	4%–42% ↑	4%–36% ↑	59%–48% ↑	9%–41% ↑	15%–61% ↑
Spray and Focus	4%–37% ↑	5%–26% ↑	50%–60% ↑	8%–41% ↑	34%–63% ↑
Multiperiod Spray	4%–43% ↑	4%–31% ↑	14%–27% ↑	13%–33% ↑	9%–55% ↑

↑ Increase, ↓ Decrease

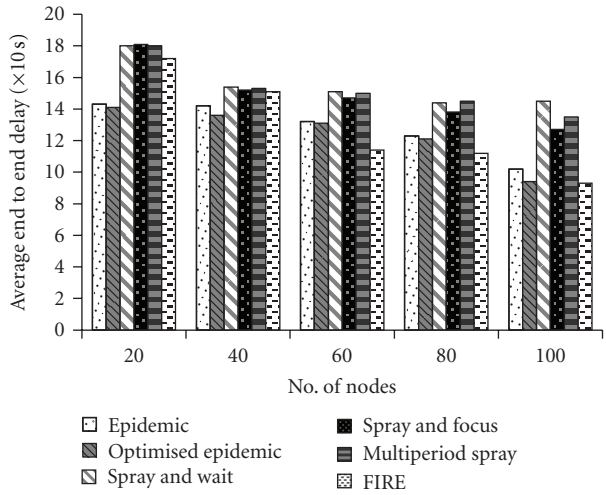


Figure 2: No. of nodes versus average end to end delay.

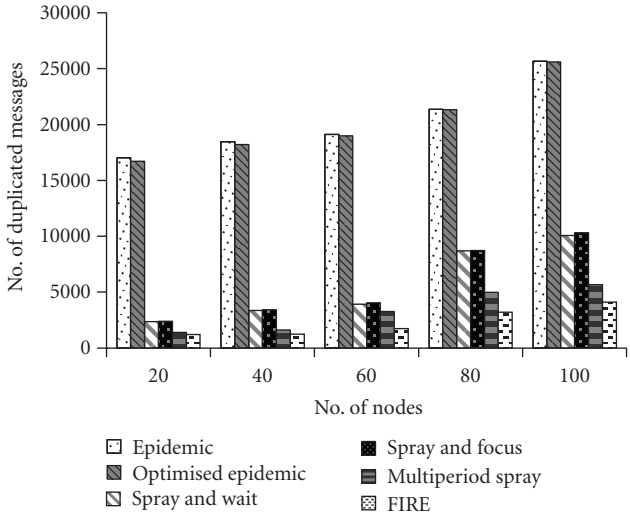


Figure 3: No. of nodes versus duplicated messages.

epidemic. Optimized epidemic also makes approximately equal copies since it limits the number of copies generated but delete the buffered messages with meaningful semantics as said above. Looking at spraying methods, they are greatly reduced the number of copies. In terms of number of copies, spray and focus and spray and wait have only slight variations. Spray and focus may have some few copies extra than that of the spray and wait. Multiperiod spraying limits the copies still to a greater extent since it sprays the copies in three different periods. If the message reached the destination, then it does not spray further. FIRE results in maximum number of reduced copies from 14% to 92%. This is due to the destination available opportunity, a single copy is forwarded to the destination and not it is duplicated. But the variation is not so much between FIRE and multiperiod, since the FIRE switch on to second phase of multicopy routing when destination is not available.

The mobility may change all the above results in a considerable manner. When node mobility speed increases, then the meeting opportunity among nodes increases and the performance of ad hoc routing protocols is improved. Therefore, we vary the speed of nodes and measure the delay and throughput. The delay measurement is shown in Figure 4 against the mobility. We have observed two results that depend upon the number of nodes in a network. When numbers of nodes are minimal then the node speed does not make the delay decrease. When it is doubled for the same scenario, then the delay is reduced to a certain extent. At the same time the changes reflect only to some limited speed. If speed is increased further then the results are the same or decreasing, due to contact times of nodes are reduced.

When nodes are slow, all the protocols have almost the highly delayed messages. Even though the spraying methods limit the number of copies to a greater extent, when nodes are restricted within some area they give minimal performance since the messages are copied to limited number of nodes only. But the remaining protocols have reduced delays, even though they have constrained mobility. When mobility increases, the spraying methods perform better in delivering the messages shortly; therefore, after 30 mps as in Figure 4 the spraying methods have reduced delay than the flooding based approaches. But when speed is increased further then all the protocols tend to work like the same. There is no further improvement in delays since the high speed makes the nodes to miss the contacts. The FIRE has an impact on the mobility. Initially, when nodes are at very low speed like 10 mps, it gives 2%–4% performance decrease with flooding based protocols. It is due to the relay nodes are not faster in meeting other nodes. When nodes are having increased speed, connectivity among the source and destination increases and the maximum of messages are delivered with finite delay forwarding itself. If we see the 50 mps delay values for FIRE, it is very lowcompared to all other protocols. Hence, the FIRE makes use of the dynamic

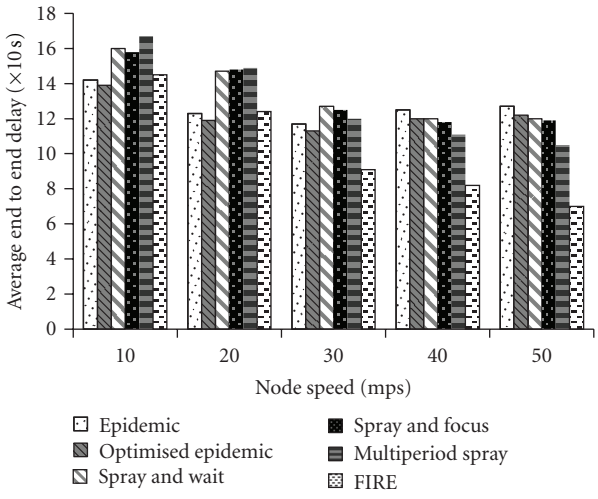


Figure 4: Node speed versus average end-end delay.

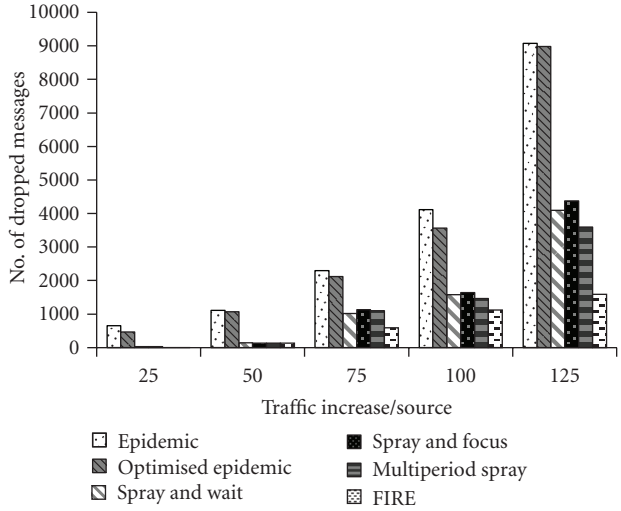


Figure 5: Increase in messages/source versus no. of messages dropped.

topology changes and adapt to that for having a better performance in terms of delay and also throughput.

The buffer sizes of nodes have a great impact on the performance of intermittently connected networks. The flooding based protocols work better if they have infinite buffer space. If buffer space is restricted then there are much number of packets drop. This is depicted in Figure 5. Here, the results are shown for the buffer size of each node is restricted to 200 messages. The x-axis is increased with CBR traffic per source node. When the traffic is low the message drops are really low in epidemic and optimized epidemic. Even it is zero for FIRE and other spraying methods. But when we increase the traffic the messages started to drop. But there are sudden increases in flooding-based schemes. Both epidemic and optimized epidemic have almost equal number of packets dropped when increasing the traffic. Since in flooding almost the buffer of all the nodes fully occupied the optimized epidemic cannot control the drop but drop the messages according to the criteria. The FIRE has smaller number of message drops, and it improves the results from 9% to 55% compared with multiperiod but increases up to 96% compared with epidemic protocols, since in the finite phase it transfer only a single copy for all the messages and in the infinite forwarding also it limits the copies only to the nodes which has good meeting update. Therefore, unnecessary flooding and unwanted fixed number of copies are avoided. Therefore, the buffer of the relay nodes is kept free for useful traffic.

5. Conclusion and Future Work

In this paper, we develop techniques to allow eventual message delivery in the case where connectivity from source to destination may or may not be available in wireless ad hoc networks. Existing routing techniques of mobile ad hoc protocols are unable to deliver packets in the occurrence of network separation between source node and destination node when partition exists for a long time. Therefore, multiple replication schemes are introduced. But these multiple replication schemes waste the resources like bandwidth, buffer, and battery of the wireless nodes most of the time. Since the resource consumption is very essential in wireless nodes, it is a must to go for controlled multiple replication schemes. We try to utilize the connections among the source and destination whenever possible by finding the destination first and then route. If destination is not available then we move to infinite buffer forwarding where messages are kept for long time. Thus, the FIRE makes use of the connectivity whenever possible and the results show that the delay is reducedin turn of increasing the throughput. The number of replications and message drops are greatly reduced compared with the existing controlled replication schemes.

As a future part, we are planning to analyze our protocol for energy awareness on relay nodes and then to route. Also, different mobility models are to be evaluated with the proposed method. If none of the destination is available, then all the traffic follows the infinite buffer forwarding. Therefore, different optimization can be tried during infinite buffer forwarding for further reducing the number of duplications.

References

[1] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University Technical Report CS-200006, 2000.

[2] S. Burleigh, A. Hooke, L. Torgerson et al., "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[3] V. Cerf et al., *Delay-Tolerant Network Architecture, IETF RFC, 4838, Informational*, 2007.

[4] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the Conference on Computer Communications (ACM SIGCOMM '03)*, pp. 27–34, August 2003.

[5] Wang and Y. Wu H, "DFT-MSN: the delay/fault-tolerant mobile sensor network for pervasive information gathering," in *Proceedings of the 25th IEEE International Conference on Computer Communications*, 2006.

[6] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[7] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of the Applications, Technologies, Architectures, and Protocols for Computer Communications (ACM SIGCOMM '4)*, vol. 34, pp. 145–158, ACM Press, 2004.

[8] J. LeBrun, C. N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proceedings of the IEEE 61st Vehicular Technology Conference (VTC '05)*, pp. 2289–2293, June 2005.

[9] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.

[10] A. Mathurapoj, C. Pornavalai, and G. Chakraborty, "Fuzzy-spray: efficient routing in delay tolerant ad-hoc network based on fuzzy decision mechanism," in *Proceedings of the IEEE 18th International Conference on Fuzzy Systems*, pp. 104–109, August 2009.

[11] C. Poongodi and A. M. Natarajan, "Optimized replication strategy for intermittently connected mobile etworks," in *Proceedings of the International Journal of Business Data Communications and Networking*, vol. 8, no. 1, 2012.

[12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the Conference on Computer Communications (ACM SIGCOMM '05)*, pp. 252–259, August 2005.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '06)*, pp. 49–60, May 2006.

[14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '07)*, pp. 79–85, March 2007.

[15] E. Bulut, Z. Wang, and B. K. Szymanski, "Cost-Effective multiperiod spraying for routing in delay-tolerant networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1530–1543, 2010.

[16] Q. Li and D. Rus, "Sending messages to mobile users in disconnected ad-hoc wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 44–55, August 2000.

[17] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: routing for vehicle-based disruption-tolerant networks," in *proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.

[18] H. Dang and H. Wu, "Clustering and cluster-based routing protocol for delay-tolerant mobile networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 1874–1881, 2010.

[19] M. Takahashi, B. Tang, and N. Jaggi, "Energy-efficient data preservation in intermittently connected sensor networks," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '11)*, pp. 590–595, April 2011.

[20] J. K. Maurice, F. F. Wissam, and M. A. Chadi, "Probabilistic bundle relaying schemes in two-hop vehicular delay tolerant networks," *IEEE Communications Letters*, vol. 15, no. 3, pp. 281–283, 2011.

[21] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *Proceedings of the IEEE 28th Conference on Computer Communications (INFOCOM '09)*, pp. 846–854, April 2009.

[22] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proceedings of the Conference on Computer Communications (ACM SIGCOMM '07)*, pp. 373–384, August 2007.

[23] C. Poongodi, A. M. Natarajan, and R. C. Suganthee, "Cluster head bundle delivery protocol for delay tolerant networks," in *Proceedings of the CiiT International Journal of Wireless Communication*, November 2010.

[24] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the Sensor Network Protocols and Applications*, pp. 30–41, 2003.

[25] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile Ad Hoc networks," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MoBiHoc '04)*, pp. 187–198, ACM Press, New York, NY, USA, May 2004.

[26] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 1407–1418, March 2005.

[27] Y. Xian, C. T. Huang, and J. Cobb, "Look-ahead routing and message scheduling in delay-tolerant networks," in *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN '10)*, pp. 40–47, October 2010.

[28] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Multiple-copy routing in intermittently connected mobile networks," Technical Report CENG-2004-12, 2004.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 813594, 12 pages
doi:10.1155/2012/813594

Research Article

A Group Key Distribution Scheme for Wireless Sensor Networks in the Internet of Things Scenario

Hong Yu,[1] Jingsha He,[2] Ting Zhang,[1] and Peng Xiao[1]

[1] *College of Computer Science and Technology, Beijing University of Technology, Beijing 100124, China*
[2] *School of Software Engineering, Beijing University of Technology, Beijing 100124, China*

Correspondence should be addressed to Hong Yu, yuhong_0826@emails.bjut.edu.cn

Received 5 July 2012; Revised 2 October 2012; Accepted 5 October 2012

Academic Editor: Deyun Gao

As an indispensable part of the Internet of Things (IoT), wireless sensor networks (WSNs) need to be completely integrated into the Internet. When an Internet user communicates with multiple sensor nodes in WSNs, secure group key management becomes necessary. However, most current group key management schemes developed for WSNs do not consider the Internet scenario while traditional group key management in the Internet is deemed to be not suitable for WSNs due to the resource constraint characteristics of WSNs. In this paper, we propose a group key distribution scheme for WSNs in the IoT scenario in which we organize sensor nodes into groups in a hierarchical structure. In the upper wired layer, an end-to-end secure communication protocol is used to distribute group keys for subgroups to the trusted head nodes and the head nodes then distribute the group keys through underlying tree-based topology and wireless multicast to minimize energy consumption. We also perform some quantitative analyses as well as experiments to show that our proposed scheme is secure and has t-revocation capability. The total cost of distributing and rekeying the group keys is also analyzed and compared to that in some other comparable schemes.

1. Introduction

As an indispensable part of the Internet of Things (IoT), wireless sensor networks (WSNs) need to adopt IP technologies to create a seamless, global network infrastructure together with the Internet. To achieve the above goal, many standardization organizations are actively pursuing standardization work for creating a global sensor network infrastructure. IPv6 over low power wireless personal area network (6LoWPAN) is one such technology that can enable complete integration of WSNs into the Internet. 6LoWPAN is aimed by the Internet Engineering Task Force (IETF) 6LoWPAN Working Group at enabling most capabilities of IPv6 on constrained nodes [1] and the transmission of IPv6 packets over low power wireless personal area networks based on IEEE 802.15.4 standards [2].

In the IoT scenario, any IP-enabled node in the Internet can communicate with any sensor node in a WSN remotely. It has thus become imperative that secure communication be supported between the remote entities. Recently, some secure communication schemes for creating an end-to-end secure channel between an Internet node and a sensor node have been researched in different layers of the Open System Interconnection (OSI) reference model. The research work by Granjal et al. [3–5] and Raza et al. [6, 7] is aimed at making the application of IPSec in 6LoWPAN sensor nodes a reality while the idea of providing transport level end-to-end security includes modified Secure Socket Layer (SSL) scheme based on Elliptic Curve Cryptography (ECC) [8], Transport Layer Security (TLS) scheme using IBC based on ECC or pairings [9] and the scheme that relies on asymmetric authentication and signcryption [10].

With the secure communication protocols mentioned above, a variety of applications for WSNs in the IoT scenario can be developed ranging from defense systems to health care, industrial monitoring, disaster management, home automation, and so forth [11]. As one of the basic requirements in the applications, an authorized user may need to issue queries and commands to multiple sensor nodes at the same time rather only to a single sensor node. Thus, secure group communication protocol could be more efficient than those mentioned above.

Secure group communication requires secure and robust distribution or negotiation of group keys. A single symmetric key known only to the group in which the authorized user in the Internet and the multiple targeted sensor nodes in WSNs are the members can effectively protect communication for multicast group. Current group key management schemes in WSNs that belong primarily to the broadcast fashion by making use of wireless channels and transmission ranges cannot be directly applied to the IoT context since the user is usually located in different physical locations, even in different networks. Meanwhile, traditional group key management protocols developed for the Internet, such as IP multicast [12] that needs the support of multicast routers or application layer multicast [13] that needs the support of endhosts, are deemed to be inadequate for WSNs due to the characteristics of limited energy, storage, network bandwidth, communication capabilities in most sensor nodes, as well as multihop characteristics of WSNs, that is, each sensor node can act both as a router and as an end host.

In this paper, we propose a group key distribution scheme for WSNs in the IoT scenario. In our proposed scheme, the sensor nodes in WSNs are organized into groups in a hierarchical structure. In the upper wired layer, an end-to-end secure communication protocol is used to distribute group keys for the subgroups to trusted head nodes. In the lower wireless layer, the head nodes distribute the group keys through underlying tree-based topology and wireless multicast to minimize energy consumption of the sensor nodes. The main contributions of this paper can be summarized as follows.

(1) We analyze the need for group key management for WSNs in the IoT scenario and propose a secure and efficient solution to overcome the limitations of existing mechanisms in the IoT scenario.

(2) We design a hierarchical group key distribution scheme without requiring any preshared keys between the user and the sensor nodes. The scheme has the t-revocation capability and can minimize communication cost of the WSNs.

(3) We demonstrate how our scheme can perform as a secure and efficient countermeasure against some attacks towards WSNs among which the cooperative compromised attack is analyzed emphatically, which is a capability that is absent in most existing mechanisms.

(4) We conduct mathematical analysis on our scheme as well as performance comparison between our scheme and the Topological Key Hierarchy (TKH) scheme which is considered to be the most efficient mechanism for group key management in traditional WSNs. The comparison results show that our scheme outperforms the TKH scheme for group key distribution and group key rekeying when a small number of nodes are deleted.

The remainder of this paper is organized as follows. Section 2, we review some related work on group key distribution, group key negotiation, the self-healing theory, and the TKH scheme with which we will compare our scheme in the analysis. In Section 3, we describe our proposed group key distribution scheme, which includes assumptions, initialization, group key distribution, in upper wired layer, group key distribution and rekeying in lower wireless layer. In Section 4, we analyze our proposed scheme in terms of security and performance and compare it to TKH. Finally, in Section 5, we conclude this paper in which we also discuss some future work.

2. Related Work

In both the Internet and the WSNs, group key management can be classified into group key distribution (centralized model) and group key negotiation (distributed model).

2.1. Centralized Model. Group key distribution schemes (e.g., Group Key Management Protocol (GKMP) [14]) usually rely on a group controller which shares a pairwise key with each member of the group and distributes group keys to group members on a point-to-point basis. This approach cannot scale to large groups, which generates $O(N)$ rekeying messages with a network of size N. In Logical Key Hierarchy (LKH) [15], individual and auxiliary keys are organized into a hierarchy and each group member is assigned to a leaf and holds all the keys from its leaf to the root. The root key as the group key is shared by all the group members. This scheme can reduce the number of rekeying messages to $O(\log N)$. One-Way Function Tree (OFT) [16] improves LKH by reducing the number of rekeying messages from $(2\log_2 N)$ to $(\log_2 N)$ in the binary key tree by using the local key computations. However, since in the multihop WSNs each sensor node can act both as a router and as an end host, rekeying messages generated from the logical key tree may be forwarded through one or more intermediate nodes to reach their final destination nodes, incurring heavy communication overheads. The Topological Key Hierarchy (TKH) [17] lowers the cost of rekeying messages by generating a key tree based on the underlying topology information of WSNs to minimize communication cost.

For group communication for WSNs in the IoT scenario, the authorized user in the Internet can act as the group controller, but all of the schemes mentioned above cannot be directly applied due to the following reasons. Firstly, the rekeying messages generated from the user must be transmitted over both the wired and wireless links, which may incur noticeable delay. Secondly, all of the centralized schemes rely on pairwise keys between the user and each sensor node, thus the user must authenticate and negotiate a shared secret key with each sensor node. Any group key distribution scheme should support end-to-end security communication that will make the cost of communication and computation grow linearly with the number of group members.

2.2. Distributed Model. In distributed group key negotiation, all group members are treated equally. Hence, group keys

should be negotiated among all group members through Diffie-Hellman (DH) key exchange or based on secret sharing theory to ensure fairness. In the CLIQUE scheme [18], group members can deliver their DH seeds orderly through insecure channels and the last member get all the DH seeds to compute the group key and then multicast the received DH seeds to other members so that all members can get the group key, thus generating $O(N^2)$ key messages and incurring $O(N^2)$ computation cost. Secret sharing theory [19] can enhance the robustness of group key generation in which each group member is issued a seed of the group key securely and any group member must collect $M(M < N)$ secret seeds from a subset of the group members to recover the group key. Therefore, an attacker who captures less than M members cannot recover the group key. This approach causes frequent interactions and incurs high computation cost, and the exchange of secret seeds must be protected using shared keys between the peers.

In the IoT scenario, the above group key negotiation schemes are not suitable for WSNs since the cost of communication and computation is more than that of group key distribution schemes. Moreover, the reasons for the infeasibility of group key distribution schemes are also exist.

Hierarchical group key management, for example, the Iolus scheme [20], is a tradeoff of the above two models in which group members are divided into many subgroups each of which has an independent group key. Thus, rekeying can be executed within the corresponding subgroup. Therefore, techniques based on hierarchical group are more suitable for WSNs in IoT scenario. In our proposed scheme, we separate the group key management into two layers. In the upper wired link layer, the Internet user distributes the group key to the head nodes in a WSN by using the end-to-end secure communication protocol. In the lower wireless link layer, each head node distributes the group key using self-healing theory [21] and TKH structure [17], which will be introduced in the next two subsections.

2.3. Self-Healing Theory. To be self-healing with the t-revocation capability, the group manager constructs and broadcasts a t-degree masking polynomial $\omega(x) = f(x)K + h(x)$ in which $h(x)$ is a t-degree shielded polynomial. For any normal group member i, $h(i)$ is preloaded, $f(x) = (x - r_1)(x - r_2) \cdots (x - r_j)$ is a t-degree revocation polynomial and $R = \{r_1, r_2, \ldots, r_j\}$ is the set of all deleted group members, $|R| \leq t$. For any normal group member i, it evaluates the polynomial at point i and gets $(i) = f(i)K + h(i)$. Because i knows $h(i)$ and $f(i) \neq 0$, it can compute $K = [\omega(i) - h(i)]/f(i)$. Since the coalition R gets at most t points over the t-degree polynomial $h(x)$, it is computationally infeasible for coalition R to learn $h(i)$ for $i \notin R$.

2.4. TKH Group Rekeying Policy. In the TKH scheme that is applied to WSNs, the nodes in the same subtree (ST) share the same tree key (TK). ST is a tree with nodes below each subroot node, and the subroot nodes are direct neighbors of a sink. The nodes sharing the same parent node in a tree, that is, the sibling nodes, share the same sibling key (SK). Every node shares its own individual key (IK) with the sink. The group key (GK) is used to encrypt all data traffic within a group. TKH offers an advantage that the depth of the key tree is bounded to "4" regardless of the size of the network. Therefore, each node is only required to save a maximum of four keys, which is highly suitable for storage-limited sensor nodes.

TKH takes the advantage of the wireless multicast. Since a message transmission can be heard by multiple neighbors, sibling nodes can efficiently receive a message by a single transmission from their parent. An example is shown in Figure 1.

When node 3 in ST_1 is revoked, the rekeying messages for ST_2 and ST_3 are $\{GK'\}_{TK_2}$ and $\{GK'\}_{TK_3}$, respectively. For ST_1, the rekeying messages (m) and the corresponding communication cost (C) are

$$\begin{aligned}
&m_{S\to\{1\}} : \{GK', TK_1'\}_{IK_1}, && C_{S\to\{1\}} : e_{tx} + e_{rx},\\
&m_{S\to\{2,4\}} : \{GK', TK_1'\}_{SK_1'}, && C_{S\to\{2,4\}} : 2e_{tx} + 3e_{rx},\\
&m_{S\to\{5,6\}} : \{GK', TK_1'\}_{SK_2}, && C_{S\to\{5,6\}} : 3e_{tx} + 4e_{rx},\\
&m_{S\to\{7,8\}} : \{GK', TK_1'\}_{SK_3}, && C_{S\to\{7,8\}} : 3e_{tx} + 4e_{rx},\\
&m_{S\to 2} : \{SK_1'\}_{IK_2}, && C_{S\to 2} : 2e_{tx} + 2e_{rx},\\
&m_{S\to 4} : \{SK_1'\}_{IK_4}, && C_{S\to 4} : 2e_{tx} + 2e_{rx},\\
&m_{S\to 7} : \{SK_3\}_{IK_7}, && C_{S\to 7} : 3e_{tx} + 3e_{rx},
\end{aligned} \tag{1}$$

where e_{tx}, e_{rx} are the energy consumption of transmitting, receiving one bit respectively. Note that the sibling sets that share SK_2 and SK_3 are slightly changed. However, TKH does not update SK_2 and SK_3 since none of the nodes sharing them are deleted. By maintaining the link from node 7 to SK_2 in the key tree, the sink can update both SK_2 and SK_3 later when node 7 is deleted. Thus, the total rekeying cost of ST_1 is

$$\begin{aligned}
\mathrm{TRC}_{ST1} &= 2|m| \times \left(C_{S\to\{1\}} + C_{S\to\{2,4\}} + C_{S\to\{5,6\}} + C_{S\to\{7,8\}}\right)\\
&\quad + |m| \times \left(C_{S\to 2} + C_{S\to 4} + C_{S\to 7} + C_{S\to 8}\right)\\
&= |m|(25e_{tx} + 31e_{rx}).
\end{aligned} \tag{2}$$

A new node should select a parent node to join the network and the existing nodes can change the corresponding GK, TK, and SK by using the preshared one-way function (Formula (3)) and the sink unicasts, $\{GK', TK', SK'\}_{IK_{new}}$ to the newly added node:

$$GK' = F(GK), \qquad TK' = F(TK), \qquad SK' = F(SK). \tag{3}$$

The TKH scheme does not mention the distribution of GK, TK, SK, and IK. Meanwhile, it is assumed that IK is preloaded into every sensor, which is feasible when the head node is a stationary sink node. However, for WSNs in IoT scenario, the head node may be an ordinary node directly or indirectly dynamically chosen by the Internet user. Therefore, it is impossible for every sensor node which may

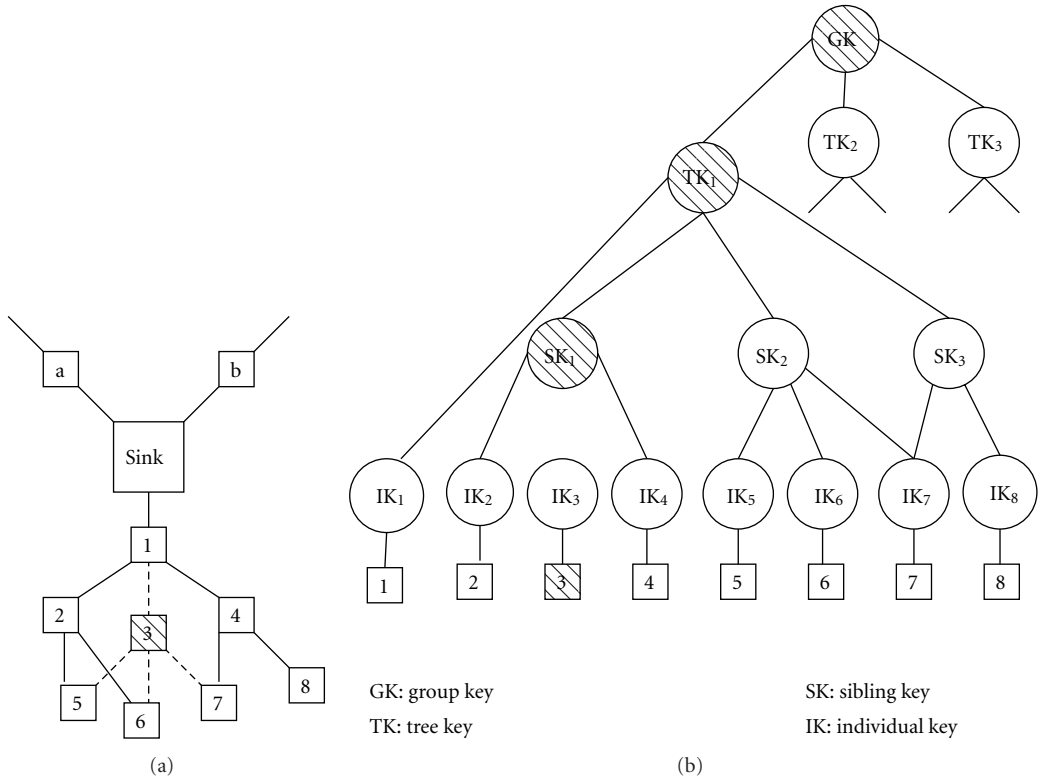


Figure 1: After node 3 is deleted, (a) shows the repaired tree topology while (b) shows the corresponding TKH structure. The keys that need to be updated when node 3 is deleted are shaded.

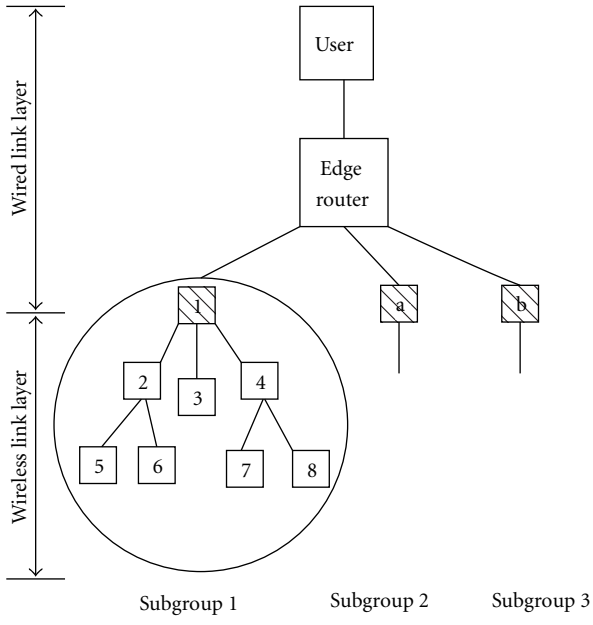


Figure 2: The hierarchical structure of our proposed scheme.

be the potential head node to store all shared IKs with each other.

Our method is motivated by the above analysis that the TKH scheme is suitable for WSNs due to taking use of the underlying sensor network topology to decrease the forwarding through intermediate nodes and the one-hop wireless multicast to save energy. However, in our proposed scheme, we further reduce the forwarding from intermediate nodes through the only once hop-by-hop wireless multicast along with the underlying topology. The legitimate sensor node in the group can recover the GK from the received information without the shared IK with the head node. Moreover, the adversary has to compromise at least t group nodes instead of only one to get the new GK, which is known as t-revocation property.

3. The Proposed Scheme

Our proposed scheme organizes sensor nodes in a WSN into groups in a hierarchical structure as shown in Figure 2. An end-to-end secure communication protocol is used to distribute group key to the head nodes in subgroups in the upper wired link layer, while the head nodes distribute group keys to subgroup members in the lower wireless link layer through the underlying tree-based topology and by the means of wireless multicast.

3.1. The Assumption. We assume that the edge router is deployed for a WSN by the service provider (SP) and the

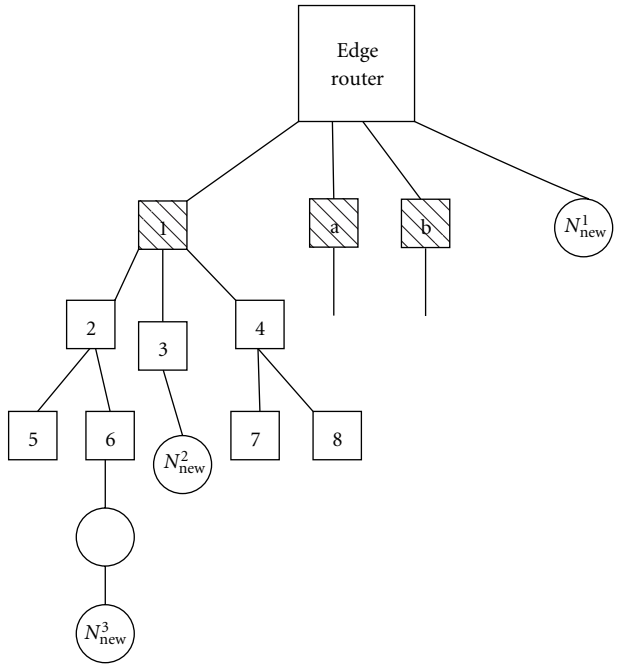


Figure 3: An example of node addition in our proposed scheme.

edge router is credible and has unlimited resources in terms of energy, computation, and storage.

3.2. Initialization. An SP should deploy and manage the WSN to provide services to the Internet users. The SP randomly picks a t-degree shielded polynomial $h(x, y) = \sum_{j=0}^{t} \sum_{i=0}^{t} a_{ij} x^i y^j$ from $F_p(x, y)$, where p is a prime number that is large enough to accommodate a cryptographic key, a_{ij} is a polynomial coefficient for the WSN which is provided to authorized Internet users, x is the current node, and y is the current head node. Each node in the WSN $N_r (r = 1, 2, \ldots, n)$ is issued a personal key $S_r = h(N_r, y)$ and the one-way function F.

After deployment, the secure bootstrapping process in the WSN could be used (referring to [22]) to establish pairwise keys between neighboring nodes and trust paths to the edge router so as to form a multihop cluster-tree hierarchical topology, which can be considered as the routing tree in our scheme. The edge router knows the whole network topology, and each node knows its descendants as well as parent nodes.

3.3. The Wired Link Layer. An authorized Internet user U can generate a group key GK for the set C of sensor nodes he/she want to communicate with, where $C = \{N_1, N_2, \ldots, N_c\}$, $|C| = c, c \leq n$. When U launches a group request attached C to the WSN, the edge router replies with a set of head nodes $H = \{N_1, N_2, \ldots, N_h\}$ $(|H| = h < c)$ by checking the topology, choosing the nodes which have the minimum hop count to the edge router in each ST and ensuring that all nodes in the set C could be included in all the STs of set H.

User U distributes GK, TK and a_{ij} along with the set C to all the nodes in H through established secure channels (referring to [10] for establishing such secure channels).

3.4. The Wireless Link Layer. Suppose that the number of STs in the WSN is h. We take the head node $N_g (N_g \in H)$, that is, the subtree ST_g as an example to describe the process executed in the wireless link layer.

3.4.1. Group Key Distribution. The head node N_g constructs a t-degree shielded polynomial $h(x, y) = \sum_{j=0}^{t} \sum_{i=0}^{t} a_{ij} x^i y^j$ using a_{ij} received from user U and computes $h(x, N_g)$. According to C and the descendant nodes, head node N_g will prepare three sets:

M: the malicious nodes in the WSN that have been deleted from the WSN and are not included in the routing tree;

I: the invited nodes in ST_g;

N: the normal nodes in ST_g that are not invited by U but can correctly and honestly execute the routing protocol.

N_g sends the message B along with the routing tree using wireless multicast fashion:

$$B = R \cup \left\{\omega(x) = f(x)\mathrm{TK}_g + h\left(x, N_g\right)\right\} \cup \mathrm{E}_{\mathrm{TK}_g}(\mathrm{GK}), \quad (4)$$

where $f(x) = (x - N_{r_1})(x - N_{r_2}) \cdots (x - N_{r_{\omega j}})$, a t-degree revocation polynomial, $R = M \cup N = \{N_{r_1}, N_{r_2}, \ldots, N_{r_{\omega j}}\}$, $|R| = \omega_j \leq t$, and TK_g is the tree key received from user U.

After receiving B, every remaining node N_r $(N_r \in I)$ will evaluate $f(x)$ and $\omega(x)$ at the point $x = N_r$, and $S_r = h(N_r, y)$ at the point $y = N_g$ to get TK_g:

$$\mathrm{TK}_g = \frac{\left[\omega(N_r) - h\left(N_r, N_g\right)\right]}{f(N_r)}, \quad (5)$$

where $f(N_r) \neq 0$. Then N_r decrypts GK with the computed TK_g.

3.4.2. Node Addition. The authorized Internet user U can add a new sensor node N_{new} to the set C. The edge route will inform U to which ST N_{new} should belong. N_{new} can be either in a new ST or a node in an existing ST.

In the former case, taking N^1_{new} in Figure 3 for example, U distributes $\mathrm{GK}' = F(\mathrm{GK})$, $\mathrm{TK}_{\mathrm{new}}$, and a_{ij} along with the set C' to N^1_{new} through secure channels as described in Section 3.3. All nodes in C will locally compute $\mathrm{GK}' = F(\mathrm{GK})$ by using the pre-shared one-way function F.

In the latter case, assuming that $N_{\mathrm{new}} \in \mathrm{ST}_g$, all nodes in C will locally compute $\mathrm{GK}' = F(\mathrm{GK})$ and all nodes in ST_g will locally compute $\mathrm{TK}' = F(\mathrm{TK})$. U informs N_g of the new set C', and N_g checks the location of N_{new} in the routing tree. If the immediate parent node of N_{new} is in C, that is N^2_{new} in Figure 3, N^2_{new} will receive $\{\mathrm{GK}', \mathrm{TK}'\}$ encrypted using the pairwise key shared between N^2_{new} and node 3. Otherwise, that is N^3_{new} in Figure 3, node 1 will unicast $R \cup \{\omega(x) = f(x)\mathrm{TK}'_g + h(x, N_g)\} \cup E_{\mathrm{TK}'_g}(\mathrm{GK}')$ along the routing tree to N^3_{new}which can then compute TK'_g and decrypt GK' as described in Section 3.4.1.

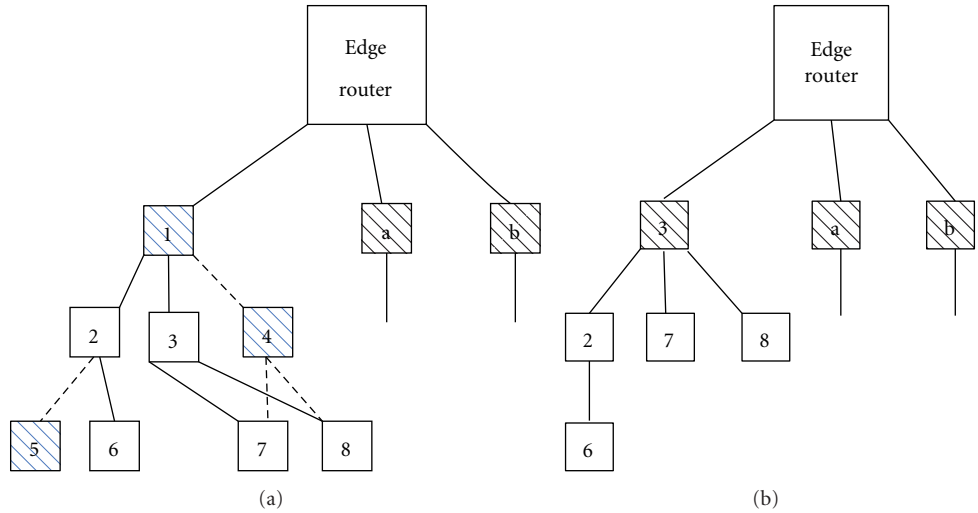


Figure 4: An example of node deletion in our proposed scheme. The repaired tree topology (a) after node 4 and 5 are deleted and (b) after node 1 is deleted, respectively. The deleted nodes are shaded in blue.

3.4.3. Node Deletion. The authorized Internet user U deletes a sensor node N_{rev} from the set C. The edge route will inform U of to which ST N_{rev} belongs. We classify the node deletion event into the cases of ordinary node deletion and head node deletion.

In the former case, assuming that N_{rev} is an ordinary node, $N_{\text{rev}} \in \text{ST}_g$, if N_{rev} is a leaf node, that is node 5 in Figure 4, the routing tree is not affected but if N_{rev} is a nonleaf node, that is node 4 in Figure 4, the network topology might be affected. Then, the routing tree should be repaired using a method in which a nonleaf ordinary node is replaced by one of its siblings in the routing tree [22]. U will then distribute the new GK'and R' ($R' = R \cup \{N_{\text{rev}}\}, |R'| \le t$) to all the head nodes in H and the new TK'_g to N_g. The head nodes in the other STs except ST_g will multicast $\{\text{GK}'\}_{\text{TK}} \cup R' \cdot N_g$ will send the message $B' = R' \cup \{\omega'(x) = f'(x)\text{TK}'_g + h(x, N_g)\} \cup E_{\text{TK}'_g}(\text{GK}')$ along with the routing tree. Every remaining node N_r ($N_r \in I$ and $N_r \notin R$) will get TK'_g and GK′ as described in Section 3.4.1.

In the latter case, assuming that U will delete node 1 in Figure 4, such a nonleaf head node can be replaced by one of its children which also has children in the routing tree, that is node 3 in Figure 4. The edge router will check the network topology and inform U of the new head node 3. U will implement the procedure as described in Section 3.3, node 3 will implement the procedure as described in Section 3.4.1, and all the other STs will multicast $\{\text{GK}'\}_{\text{TK}} \cup R'$.

The procedure of node addition and node deletion is illustrated in Figure 5.

3.4.4. Reduction of the Size of Set R. We can see that the multicast information B carries ω_j IDs in set R, which increases the packet length and therefore energy consumption for communication. This will become more serious since the size of R will increase dramatically when the number of revoked nodes increases. In the following, we use the *bloom filter* technique to reduce the size of set R.

Bloom filter is a well-known data structure that can be used for efficient membership checking. Using the method, we can find whether an element belongs to a predefined set. A bloom filter consists of a set $S = \{S_1, S_2, \ldots, S_n\}$, a string of m bits, and k-independent hash functions $h_1, h_2, \ldots, h_k$ [23].

Each hash function $h_i (i = 1, 2, \ldots, k)$ maps an element $S_j (j = 1, 2, \ldots, n)$ uniformly to range $[0, 1, \ldots, m-1]$, each of which corresponding to a bit in the m-bit string. The m-bit string is initially set to 0. For the element $S_j \in S$, we can obtain its k hash values $h_i(S_j)(i = 1, 2, \ldots, k)$. Thus, the bits corresponding to these values are set to 1 in the string. There may be more than one of the values mapped to the same bit in the string. In order to find whether element $S'_j \in S$, the k bits $h_i(S'_j)(i = 1, 2, \ldots, k)$ are checked. If all the bits are 1, then $S'_j \in S$. Otherwise, if at least one of the bits is 0, $S'_j \notin S$.

Bloom filter may yield false positives, that is, although an element is not in S, its bits $h_i(S_j)$ are collectively marked by other elements in S. If the hash is uniformly random over the m values, the probability that a bit is 0 after all the n elements are hashed and their bits marked is $(1 - (1/m))^{kn} \approx e^{-(kn/m)}$. Therefore, the probability for a false positive is:

$$\left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-kn/m}\right)^k. \tag{6}$$

An example is shown in Figure 6 in which we assume that $c = 128$ and $t = \lfloor 0.1c \rfloor = 12$. Hence, the length of node's ID is $|p| = \lg c = 7$ bits, and, consequently, the size of R is 84 bits. We should let $m < t|p| = 84$ in order to reduce the size of R and let $m > kt = 12k$ in order to minimize the probability of false positive. Therefore, we use the bloom filter technique with $k = 3$ hash functions, which will map 12 IDs to an $m = 63$ bit string. As a result, the total size of R can be reduced by 25% and the probability of false positive is

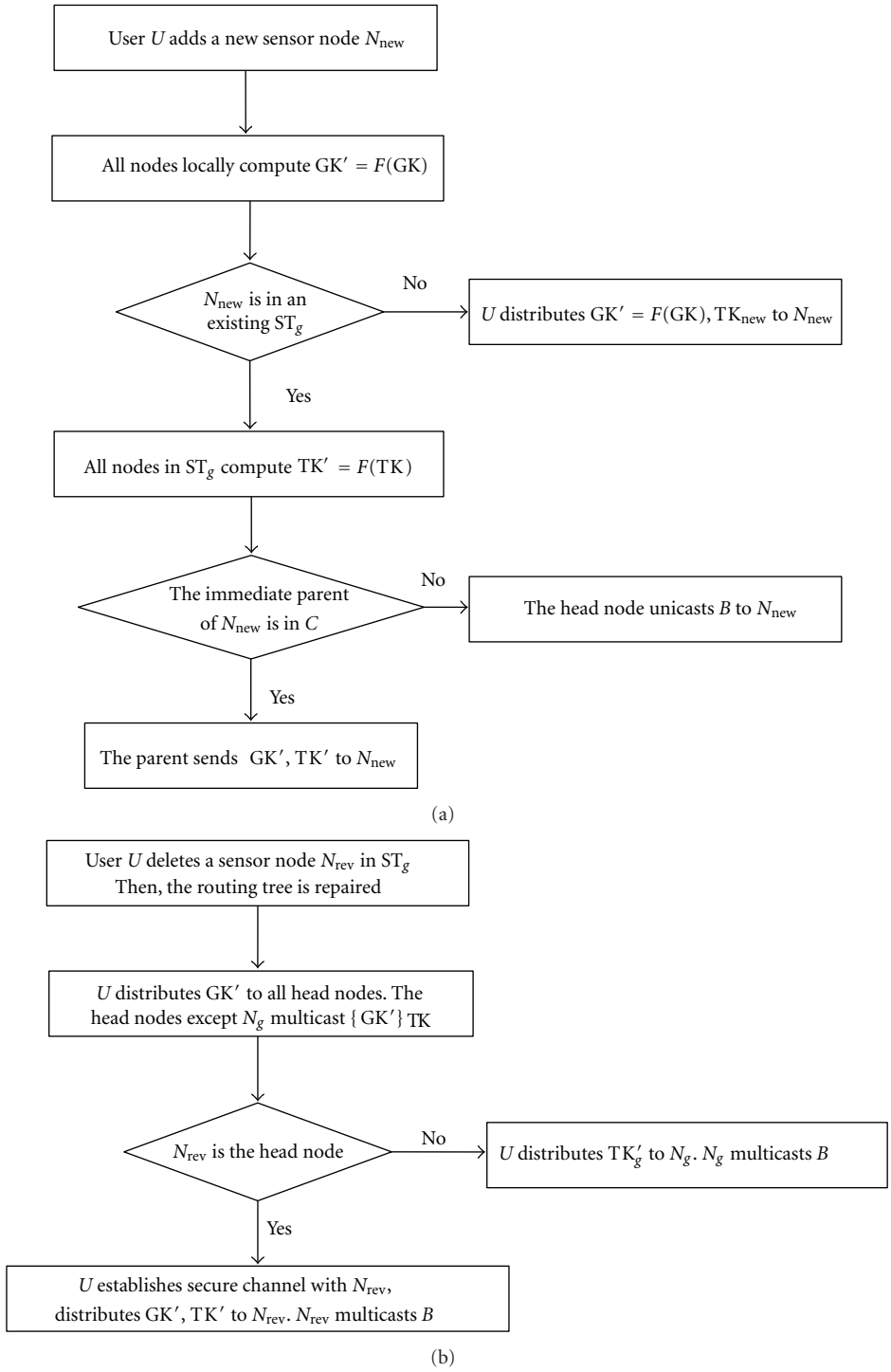


Figure 5: The procedure of (a) node addition and (b) node deletion in our proposed scheme.

about 8%. Fortunately, however, false positive has little effect on our scheme. On one hand, the illegitimate node that is mistakenly regarded as a legitimate node calculates $f(x) = 0$. On the other hand, the legitimate node which has been mistakenly revoked may also receive B and calculate TK due to communication in a multicast fashion.

4. Analysis

We analyze and show that our proposed scheme can provide confidentiality, forward secrecy, backward secrecy, and t-revocation capability for group keys. We also analyze the performance of our scheme and compare it with TKH

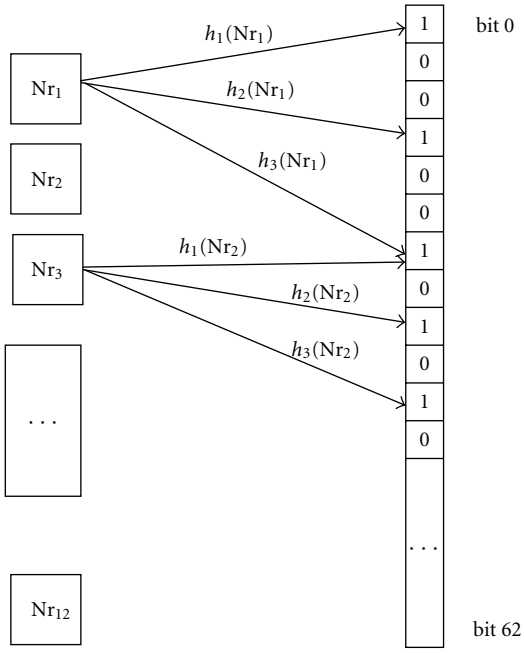


Figure 6: A bloom filter of $k = 3$ hash functions which map 12 IDs to the $m = 63$ bit string. $h_3(Nr_1)$ and $h_1(Nr_2)$ are mapped to bit 6.

scheme in terms of the cost of storage, computation, and communication.

4.1. Security Analysis. (1) Group key confidentiality: any sensor node out of the group defined by the Internet user cannot get the group key.

In our scheme, GK and TK are generated by the authorized user. In the wired link layer, GK and TK are protected by the end-to-end secure channel and transmitted to the authenticated head nodes. In the wireless link layer, firstly, the head node is responsible for distributing GK and TK to other nodes in its ST. Every node must compute TK from $\omega(x)$, $h(x, N_g)$ and $f(x)$ in order to decrypt GK. However, no attacker has the correct personal key $S_x = h(x, N_g)$, and any normal node which is not invited by U may calculate $f(x) = 0$, neither of which can generate the correct TK. Secondly, when a new node is added, the rekeying GK and TK are either locally computed by using the one-way function For transmitted after encryption using the pairwise key shared between the added node and its parent.

(2) Forward secrecy: any sensor node cannot get the group key after its deletion.

In our scheme, all the deleted nodes are added to the revocation set R and known by every node in the WSN. The head node constructs the new multicast message B' using the revocation polynomial $f'(x)$, but the deleted nodes compute $f'(x) = 0$ so that it cannot get the new GK$'$ and TK$'$.

(3) Backward secrecy: any sensor node added by the Internet user cannot get the group key before it is actually added.

In our scheme, the newly added node gets the $\mathrm{TK}_{\mathrm{new}}$ and GK$' = F$(GK) from U using an end-to-end secure channel, gets TK$' = F$(TK) and GK$' = F$(GK) from its immediate parent node, or computes TK'_g and decrypts GK$'$ from B' with its personal key. Because F is a one-way function, it is not possible to derive TK and GK from F(TK) and F(GK).

(4) t-revocation capability: the coalition R cannot get any information about the current GK.

In our scheme, in order to know TK_g, the coalition R needs to know $h(x, N_g)$ and at least $(t + 1)$ points on the polynomial $h(x, N_g)$. Since the size of the coalition R is at most t, the coalition R has at most t pieces of personal secrets $S_r = h(N_r, y)$, that is, t points on the polynomial $h(x, N_g)$. But at least $(t + 1)$ points are needed on the polynomial $h(x, N_g)$ to recover the current TK_g for any node in R.

4.2. Performance Analysis. For any group key management schemes for WSNs, even in the IoT scenario, storage, computation, and communication overhead as well as energy consumption of the sensor node are among the issues mostly concerned about. We therefore conduct performance analysis by comparing our proposed scheme with TKH in the following three aspects: storage, computation, communication. In the wireless link layer, (1) the WSN in our scheme is bootstrapped based on multihop hierarchical tree topology, (2) the group key distribution makes use of the underlying network topology with the consideration of ST-based group organization, and (3) wireless multicast advantage is taken to replace multicasting mechanisms, which are the same as TKH.

According to our scheme and TKH scheme, we do not consider the node addition event since the topology change and the corresponding rekeying cost is negligible.

In our analysis, we use the $\alpha\beta\gamma$-tree model in which α is the number of STs, β is the number of sibling sets in each ST and γ is the number of nodes in each sibling set. So, $|C| = c = \alpha(\beta\gamma + 1)$, and each ST has $(\beta\gamma + 1)$ nodes.

4.2.1. Storage Overhead. In our scheme, every node in WSN is preloaded with a personal key $S_r = h(N_r, y)$, which is a t-degree shielded polynomial, the two keys GK and TK, and $(\gamma + 1)$ pairwise keys shared with its neighboring nodes. Therefore, the storage space required is $(t+1)\lg p + (\gamma+3)\lg q$ assuming that the length of the node's ID is p and the length of the key is $\lg q$. In the TKH scheme, every ordinary node must store four keys: GK, TK, SK, IK, and the pairwise keys shared with every other node in the ST. Thus, the storage overhead is $(\beta\gamma + 4)\lg q$. In addition, every head node must store β SKs, making the storage overhead become $(\beta\gamma + \beta + 2)\lg q$.

4.2.2. Computation Overhead. In our scheme, for group key distribution, the head node must compute a point $h(x, N_g)$ on the polynomial $h(x, y) = \sum_{j=0}^{t} \sum_{i=0}^{t} a_{ij} x^i y^j$ which requires at most t multiplication operations and carry out one encryption operation. While the ordinary node must compute $f(x)$, $\omega(x)$ at the point $x = N_r$ and $S_r = h(N_r, y)$ at the point $y = N_g$. Since division can be regarded as multiplication, the total number of multiplication operations required to get TK_g is $(3t + 1)$. Meanwhile, the ordinary

node must carry out one decryption operation. In the TKH scheme, the group key distribution needs $\beta\gamma$ encryption operations for a head node and one decryption operation for each ordinary node. In addition, TKH needs $\beta\gamma$ pairwise key establishing operations between the head node and other ordinary nodes.

4.2.3. Communication Overhead. We define communication overhead as the total cost which reflects both the number of messages and the cost of message transmission.

(1) Total cost of distributing group key (TDC): in our scheme, the head node must multicast message $B(B = R \cup \{\omega(x) = f(x)\mathrm{TK}_g + h(x, N_g)\} \cup E_{\mathrm{TK}_g}(\mathrm{GK}))$ which is transmitted along the tree in every hop. Thus, the total cost of distributing group key is

$$\mathrm{TDC} = \alpha(\beta \cdot e_{tx} + \beta\gamma \cdot e_{rx}) \cdot |B|, \tag{7}$$

where $|B| = (\omega_j + t + 1)\lg p + \lg q$.

In the TKH scheme, the head node takes the responsibility of unicasting GK, TK and SK to every ordinary node encrypted using the pairwise key between the head node and an ordinary node. So the total cost of distributing group key is:

$$\mathrm{TDC}_{\mathrm{TKH}} = \alpha \cdot \beta\gamma \cdot \mathrm{avg}(1, \beta) \cdot (e_{tx} + e_{rx}) \cdot |\mathrm{keys}|, \tag{8}$$

where $\mathrm{avg}(1, n) = (1+2+\cdots+n)/n = (n+1)/2$ and $|\mathrm{keys}| = 3\lg q$.

(2) The total cost of rekeying group key (TRC): we only consider the node deletion event since the rekeying cost of node addition is negligible. When l nodes are deleted, including v head nodes and $(l - v)$ nonhead nodes, assuming that w STs have no deleting nodes, then the total cost of rekeying key is

$$\begin{aligned}\mathrm{TRC} = {} & [((\alpha - w) \cdot \beta - v) \cdot e_{tx} + ((\alpha - w) \cdot \beta \cdot \gamma - l) \cdot e_{rx}] \\ & \cdot |B'| + (\beta \cdot e_{tx} + \beta\gamma \cdot e_{rx}) \cdot w \cdot |B''|,\end{aligned} \tag{9}$$

where $|B'| = (l + t + 1)\lg p + \lg q$ and $|B''| = l \cdot \lg p + \lg q$.

$\mathrm{EXP}(w)$ calculates the average number of ST which has no deleting nodes when l nodes are deleted:

$$\mathrm{EXP}(w) = \sum_{i=\alpha-\min(l,\alpha)}^{\alpha-\lceil l/(\beta\gamma+1)\rceil} \Pr\{w = i\} \cdot i. \tag{10}$$

In the above equation,

$$\Pr\{w = i\} = \frac{C_\alpha^i \cdot N(\alpha - i, l, \beta\gamma + 1)}{C_c^l}, \tag{11}$$

where $\Pr\{w = i\}$ is the probability of $w = i$, $N(\alpha - i, l, \beta\gamma + 1)$ is the number of ways in which l nodes are to be deleted from the $(\alpha - i)$ STs, at least one has to come from each $(\alpha - i)$ ST that has $(\beta\gamma + 1)$ nodes. $N(\alpha - i, l, \beta\gamma + 1)$ can be calculated by using the recursive procedure and the result is as follows:

$$\begin{aligned}N(\alpha - i, l, \beta\gamma + 1) = {} & \sum_{k=1}^{l-(\alpha-i)+1} C_{\beta\gamma+1}^k \\ & \cdot N(\alpha - i - 1, l - k, \beta\gamma + 1).\end{aligned} \tag{12}$$

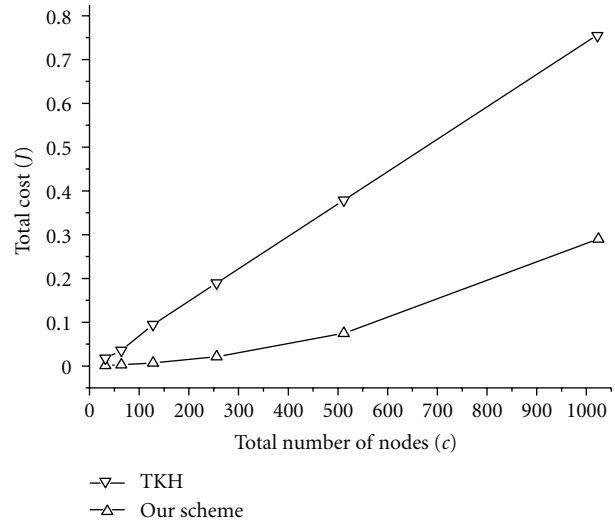


Figure 7: Total cost of group key distribution.

$\mathrm{EXP}(v)$ calculates the average number of head nodes that are deleted when l nodes are deleted:

$$\begin{aligned}\mathrm{EXP}(v) = {} & \sum_{i=\alpha-\min(l,\alpha)}^{\alpha-\lceil l/(\beta\gamma+1)\rceil} \left(\sum_{j=0}^{\min(l,\alpha-i)} \Pr\{v = j \mid w = i\} \cdot j \right) \\ & \cdot \Pr\{w = i\}.\end{aligned} \tag{13}$$

In the above equation,

$$\begin{aligned}&\Pr\{v = j \mid w = i\} \\ &\quad = \frac{\sum_{k=0}^{l-(\alpha-i)} C_{\alpha-i}^j \cdot N(\alpha - i - j, l - j - k, \beta\gamma) \cdot C_{j\cdot(\beta\gamma)}^k}{N(\alpha - i, l, \beta\gamma + 1)},\end{aligned} \tag{14}$$

where $\Pr\{v = j \mid w = i\}$ is the probability of $v = j$ under condition $w = i$.

In the TKH scheme, the total cost of rekeying key is $\mathrm{TRC}_{\mathrm{TKH}}$ which is also expressed by the function with α, β, γ, and l [17].

(3) Comparison between our scheme and the TKH scheme: we set the total number in set $C(c)$ to be 32, 64, 128, 256, 512, 1024 by designing (α, β, γ) as (2,5,3), (4,5,3), (2,7,9), (4,7,9), (8,7,9), and (16,7,9), respectively. Let $q = 2^{128}$, $t = \lfloor 0.1c \rfloor$, and $|p| = \lg c$. Each node can be identified by using $\lg c$ bits. From the characteristics of the CC2420 transceiver used in the Crossbow's MICAZ and Telos B sensor nodes [24], the unit communication cost is $e_{tx} = 0.209\,\mu J$ and $e_{rx} = 0.226\,\mu\mathrm{J}$.

Figure 7 shows the total cost of group key distribution for (2,5,3), (4,5,3), (2,7,9), (4,7,9), (8,7,9), and (16,7,9). We can see from Figure 3 that the total cost of group key distribution in our scheme is lower than that in TKH and the difference widens as the size of the network increases. This is because message B is transmitted only once in every hop along the hierarchical routing tree by using the wireless multicast advantage in our scheme. In TKH, however, key

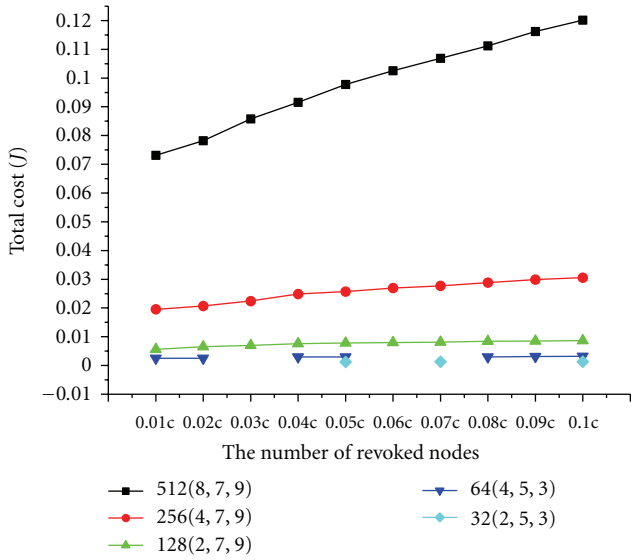


Figure 8: Total cost of rekeying group key in our scheme.

messages are different for each node, which are transmitted according to the structure of the TKH tree. Moreover, the length of B ($|B| = (\omega_j + t + 1)\lg p + \lg q$) is shorter than keys ($|\text{keys}| = 3\lg q$).

Figure 8 shows the total cost of rekeying group key when $l(1 \le l \le \lfloor 0.1c \rfloor)$ nodes are revoked for (2,5,3), (4,5,3), (2,7,9), (4,7,9), and (8,7,9) in our scheme. We can see that the total cost of rekeying group key increases as the total number of nodes increases. However, for each (α, β, γ), it increases slightly when l increases. This is because B must be transmitted by more nodes when the total number of nodes increases, which is the main source of the total cost of rekeying group key. However, when l increases in the same (α, β, γ), even though the cost on multicasting B in one ST decreases, the number of STs in which TK should be rekeyed increases, which leads to the slight increase in the total cost of rekeying group key.

The total cost of rekeying group key in both schemes is shown in Figure 9 when (a) one node is deleted and (b) $t = \lfloor 0.1c \rfloor$ nodes are deleted for (2,5,3), (4,5,3), (2,7,9), (4,7,9), and (8,7,9). When one node is deleted (i.e., Figure 9(a)), in the ST where the deleted node existed, the total cost in our scheme is smaller than that in the TKH scheme when the total number of nodes is smaller than about 400 while, in all other STs, the cost is almost the same in both schemes. This is because when the total number of nodes is more than about 400, the value t should be much larger to ensure t-revocation capabilities, causing $|B|$ to also become larger and longer than |keys| in the TKH scheme. Thus, the cost of multicasting is higher than that in the TKH scheme. However, when $t = \lfloor 0.1c \rfloor$ nodes are deleted (i.e., Figure 9(b)), the difference between the total costs seems to get smaller as the size of the network increases. This is because t gets larger as c increases and $|R|$ gets larger as l increases at the same time, causing $|B|$ to also become larger. Thus, the cost of multicasting is higher than that in the TKH scheme when the total number of nodes is more than about 300.

In conclusion, our scheme outperforms the TKH scheme for group key distribution and group key rekeying when fewer numbers of nodes are deleted but it may become less advantageous in group key rekeying as the size of the network increases and when a large number of nodes are deleted.

4.3. Experiment. We set up a real experimental environment in which the 34 Crossbow's MICAZ motes that are used as the sensor nodes each has 8-bit ATmegal 128L clocked at about 7.37-MHz microcontroller and complies with the IEEE 802.15.4 standards with data transmission rate of 250 kbps. As depicted in Figure 10, we only show half of the sensor nodes that are deployed on the fourth and the fifth floor in our lab (the other half are deployed on the second and the third floors in the same fashion). The Stargate NetBridge gateway (Base 0 of the light blue) has an Intel IXP420 XScale processor running at 266 MHz. The MIB 600 attached to the gateway can connect to a wired Ethernet and the 802.15.4. The Internet user uses a Pentium IV machine clocked at about 2.1 GHz CPU with data transmission rate of 100 mbps. Through the IP-enabled router, the remote Internet user can randomly choose $c = 32$ sensor nodes as the elements in the set C with which the user communicates. We set the unit communication cost to be the same as that in the analysis, that is, $e_{tx} = 0.209\,\mu$J and $e_{rx} = 0.226\,\mu$J. We obtain the total cost of group key distribution (TDC) and the total cost of rekeying group keys (TRC) when $l = 1$and $l = t = \lfloor 0.1c \rfloor = 3$ by executing the experiment 10 times and the experimental results are shown in Figure 11.

We can see from Figure 11 that the average TDC in our proposed scheme is 1.33 mJ, lower than 1.78 mJ in the TKH scheme. However, both of them are higher than the theoretical values of 1.3 mJ and 1.7 mJ, respectively. Meanwhile, the average TRC in our proposed scheme is 1.28 mJ and 1.35 mJ when $l = 1$ and $l = 3$, respectively, while the average TRC in the TKH scheme is 2.88 mJ and 3.08 mJ when $l = 1$ and $l = 3$, respectively, both of which also higher than the theoretical values. The reason is probably that messages could be retransmitted due to wireless channel errors, resulting in additional multicasting cost. Therefore, we consider the experiment results to be consistent with the theoretical analysis.

5. Conclusion

In this paper, we proposed a group key distribution scheme for WSNs in IoT scenario in which the sensor nodes are organized in two logic layers in a hierarchical structure. In the upper wired link layer, an end-to-end secure communication protocol is used to distribute group key to the head nodes in the subgroups. In the lower wireless link layer, each head node distributes the group key by using the underlying tree-based topology and wireless multicast advantage to minimize energy consumption. An analysis on the proposed scheme showed that our proposed scheme can achieve t-revocation security. We also performed some analyses to compare our

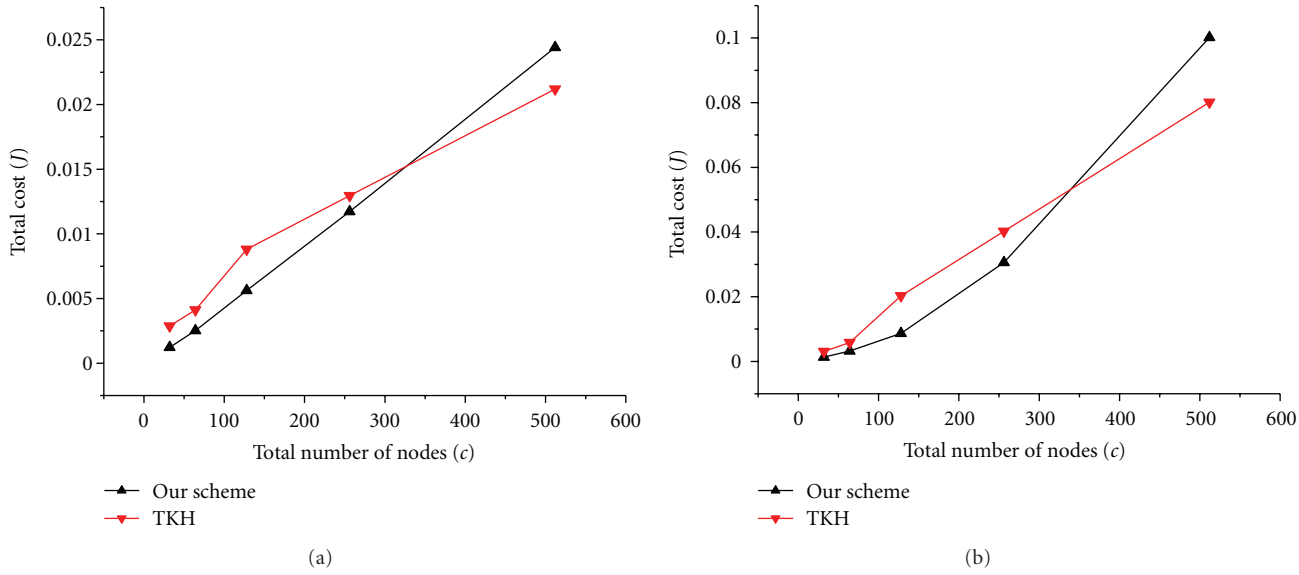


Figure 9: Total cost of rekeying group key when (a) one node is deleted and (b) ($t = \lfloor 0.1c \rfloor$) nodes are deleted.

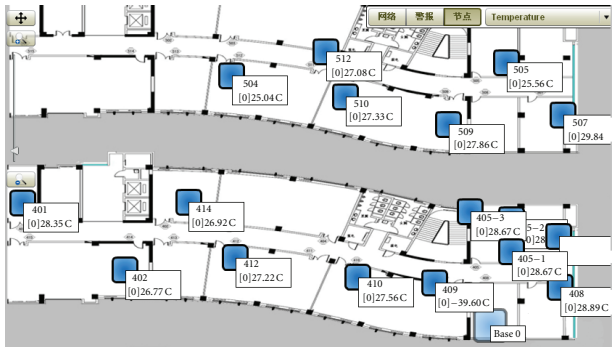


Figure 10: The real experimental environment with $c = 32$.

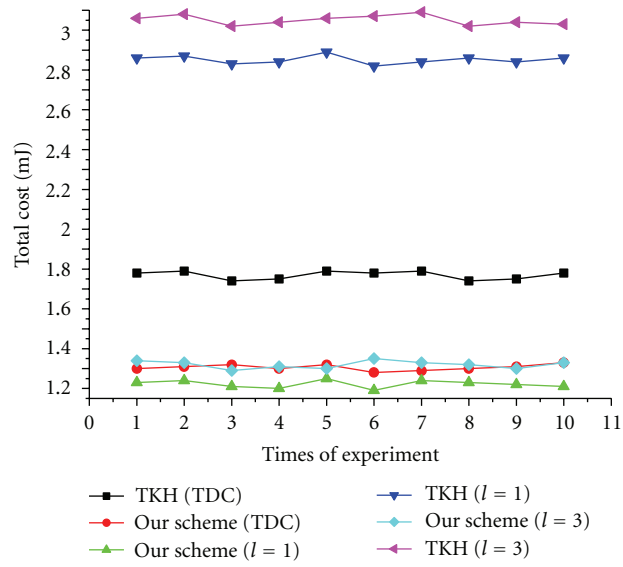


Figure 11: Experimental results for $c = 32$.

scheme with the TKH scheme in terms of the cost of storage, computation, and communication. The analyses showed that our scheme outperforms the TKH scheme in terms of total cost of communication for distributing group key and rekeying group key when fewer nodes are deleted but less advantageous when the size of the network increases and when a large number of nodes are deleted in rekeying group key. In the future, we will conduct more experiments to verify the results in real applications and further improve the performance in large-scale WSNs when a large number of nodes are deleted.

Acknowledgments

The work in this paper has been supported in part by National Natural Science Foundation of China (Grant no. 61272500) and in part by Beijing Education Commission Science and Technology Fund (Grant no. KM201010005027).

References

[1] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," *IETF RFC 4919*, 2007.

[2] G. Montenegro, N. Kushalnagar, and J. Hui, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," *IETF RFC 4944*, 2007.

[3] J. Granjal, R. Silva, E. Monteiro, J. S. Silva, and F. Boavida, "Why is ipsec a viable option for wireless sensor networks," in *Proceedings of the 5th IEEE International Conference on Mobile*

Ad-Hoc and Sensor Systems (MASS '08), pp. 802–807, October 2008.

[4] J. Granjal, E. Monteiro, and J. S. Silva, "A secure interconnection model for IPv6 enabled Wireless Sensor Networks," in *Proceedings of the IFIP Wireless Days (WD '10)*, pp. 1–6, October 2010.

[5] J. Granjal, E. Monteiro, and J. Sá Silva, "Enabling network-layer security on IPv6 wireless sensor networks," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, pp. 1–6, December 2010.

[6] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6LoWPAN with compressed IPsec," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems*, pp. 1–8, 2011.

[7] S. Raza, T. Voigt, and U. Roedig, "6LoWPAN Extension for IPsec," http://www.iab.org/wp-content/IAB-uploads/2011/03/Raza.pdf.

[8] W. Jung, S. Hong, M. Ha, Y. J. Kim, and D. Kim, "SSL-based lightweight security of ip-based wireless sensor networks," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA '09)*, pp. 1112–1117, May 2009.

[9] R. Mzid, M. Boujelben, H. Youssef, and M. Abid, "Adapting TLS handshake protocol for heterogeneous IP-based WSN using identity based cryptography," in *Proceedings of the International Conference on Wireless and Ubiquitous Systems*, pp. 1–8, 2010.

[10] H. Yu, J. He, T. Zhang, P. Xiao, and Y. Zhang, "Enabling End-to-End Secure Communication between Wireless Sensor Networks and the Internet," *World Wide Web Journal*.

[11] R. Riaz, K. H. Kim, and H. F. Ahmed, "Security analysis survey and framework design for IP connected LoWPANs," in *Proceedings of the International Symposium on Autonomous Decentralized Systems (ISADS '09)*, pp. 29–34, March 2009.

[12] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, 2000.

[13] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A survey of applicaiton-layer multicast protocols," *IEEE Communications Surveys and Tutorials*, vol. 9, no. 3, pp. 58–74, 2007.

[14] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture," *IETF RFC 2094*, 1997.

[15] D. Wailner, E. Harder, and R. Agee, "Key Management for Multicast: Issues and Architectures," *IETF RFC 2627*, 1997.

[16] G. Horng, "Cryptanalysis of a key management scheme for secure multicast communications," *IEICE Transactions on Communications*, vol. 85, no. 5, pp. 1050–1051, 2002.

[17] J. H. Son, J. S. Lee, and S. W. Seo, "Topological key hierarchy for energy-efficient group key management in wireless sensor networks," *Wireless Personal Communications*, vol. 52, no. 2, pp. 359–382, 2010.

[18] M. Setiner, G. Taudik, and M. Waidnet, "Cliques: a new approach to group key agreement," in *Proceedings of the 18th International Conference on Distributed Computing Systems*, pp. 380–387, 1998.

[19] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[20] S. Mittra, "Iolus: a framework for scalable secure multicast," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 4, pp. 277–288, 1997.

[21] R. Dutta, Y. D. Wu, and S. Mukhopadhyay, "Constant storage self-healing key distribution with revocation in wireless sensor network," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 1323–1328, June 2007.

[22] H. Yu and J. He, "Trust-based mutual authentication for bootstrapping in 6LoWPAN," *Journal of Communications*, vol. 7, no. 8, pp. 634–642, 2012.

[23] BLOOM BH, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[24] Texas Instruments Inc., "Single-Chip 2.4GHz IEEE 802.15.4 Compliant and ZigBee (TM) Ready RF Transceiver," http://www.ti.com/lit/ds/symlink/cc2420.pdf.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 469067, 13 pages
doi:10.1155/2012/469067

Research Article

Improving Driving Safety Based on Safe Distance Design in Vehicular Sensor Networks

Chen Chen,[1] Lei Liu,[1] Xiaobo Du,[1] Qingqi Pei,[1] and Xiangmo Zhao[2]

[1] *State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China*
[2] *The School of Information Engineering, Chang'an University, Xi'an 710064, China*

Correspondence should be addressed to Chen Chen, cc2000@mail.xidian.edu.cn

Received 13 June 2012; Revised 13 September 2012; Accepted 14 October 2012

Academic Editor: Oliver W. W. Yang

We proposed a Critical Safe Distance (CSD) model in V2V (Vehicle-to-Vehicle) communication systems. By elaborately analyzing the vehicular movement features for car following, we proposed a CSD definition which well balances the requirement between driving safety and traffic throughput efficiency. Different from the time-headway and traditional braking models, our model fully takes the relative movement status of the front and back cars into consideration and derives a CSD which is more practical and effective. We also explore the needed CSD during lane changing for multilane case. Numerical results show that our proposed model could provide reasonable safe distance settings under different movement scenarios, which has the capability of avoiding collisions and improving traffic throughput at the same time.

1. Introduction

Nowadays, traffic accidents have been happening uncountable times each year, taking thousands of lives or making them handicapped, outnumbering any deadly diseases or natural disasters in the world. Therefore, exploiting the new technologies, for example, wireless sensor networks, is required as a solution of reduction of these saddening and tragic events. As a consequence, Vehicular Sensor Networks (VSNs) have been a hot topic and received intensive research works in the recent years. To enhance driving safety, vehicles can be organized into Vehicle-to-Vehicle (V2V) and Vehicle-to-Roadside (V2R) networks based on communicating sensors, which can provide additional information to help to reduce accidents such as traffic density, road segment average velocity, weather or road condition,and drivers' emotion. If sensors could be further addressed by IP protocol, say an IP-enable VSNs, the interconnection to internet will bring lots of conveniences and merits such as GIS (Geographic Information System) information, traffic status report, weather, and road condition forecast) to make our driving more safety and comfortable.

Before applying VSNs technologies on road, we should first explore the deadly reasons which threat drivers' lives. Although there are many factors resulting in accidents such as weather condition, roadway status, and mechanical failure the dominating one is the reactive time of the drivers. By nature, humankinds suffer from perception limitations with a typical reactive range of 0.75 seconds to 1.5 seconds on emergency events. As illustrated in Figure 1, three vehicles A, B, and C are driving on a highway platoon. When vehicle A brakes suddenly, vehicle B can start to decelerate after a driver reaction time; that is, the duration when an event is observed and when the driver actually applies the brake, to avoid a collision. However, due to the line-of-sight limitation from B, vehicle C may not decelerate until its driver has seen the rear brake light of vehicle B. In such case, collision may be inevitable.

To solve the above problem, many studies have been done and the proposal to warning drivers based on V2V communications, such as VSNs, is attractive. In this research field, the Cooperative Collision Avoidance (CCA) strategy is an important category which aims at offering earlier warnings to drivers using V2V communications. As the example shows in Figure 2, once vehicle A encounters some accident and brakes suddenly, the warning messages will be directly sent out to B and C when they are in the radio range of A. In this case, C will have extra time in advance

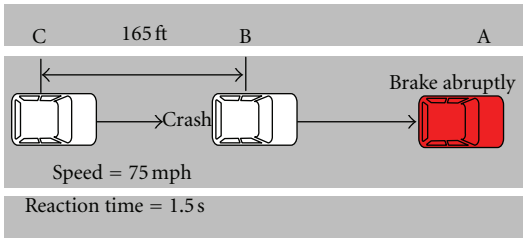


Figure 1: Collision happens due to human perception limitation.

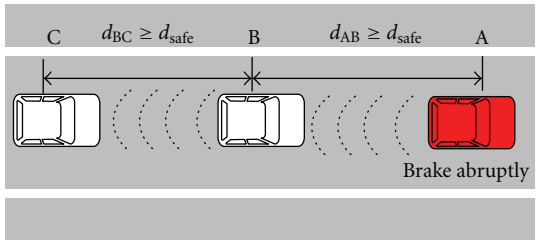


Figure 2: Collisions avoidance based on V2V communications.

to response to this emergent event and at the same time, both B and C could prebrake by detecting the wireless signal from A instead of watching the rear light with inborn perception and reaction limitation. Besides, in this discussed scenario, it is worth noting that vehicles should maintain a distance larger than Safe Distance (SD) between each other to guarantee the vehicle's stopping immediately and successfully to avoid collision. As shown in Figure 2, the distance between A and B, that is, d_{AB}, should be greater than the SD d_{safe}, and the same for d_{BC}. In fact, this SD is influenced by many factors such as driver's courage, delay spread of emergency notification packets, relative movement status of neighboring vehicles, and vehicular mechanical braking ability. In this paper, based on the analysis to classical time-headway and traditional braking models, an improved SD determination model is proposed in VSNs. This model takes the vehicles' relative movement status, traffic efficiency and driver's mental endurance into account to statistically analyze the SD under different scenarios. Our work could provide important information and measurements for the intelligent setting of beacon frequency and radio range for emergent notification messages and, at the same time, present reliable references for delay spread analysis which is curial to communication system design in VSNs.

The rest of this paper is organized as follows. In Section 2, we outline previous related works on SD determination. In Section 3, the theoretical basis on movement features of car following is introduced. Section 4 gives the assumptions and definition for CSD. The algorithmic models of our CSD in single lane and multi-lane case are presented in Sections 5 and 6 respectively. The method of notification zones' setting to implement multilevel warning scheme and provide references for beacon frequency determination is proposed in Section 7. Section 8 shows the numerical results and performance evaluation of our CSD model. Our paper is concluded in Section 9 followed by cited references.

2. Related Works

Lots of works have been done to reduce the collision possibilities with wireless communications help. In our work, we class these researches into two categories: one for avoiding collisions by designing proper protocols for warning messages dissemination; another focusing on alleviating collisions by wireless signal transmission but with research emphasis on vehicles' movement features, collision scenarios, road or weather conditions, and drivers' subjective reasons. Our work could be sorted in the latter class.

For studies with emphasis on vehicles moment features, an influential work is from Brackstone and McDonald [1] who had elaborately analyzed the car-following model especially for safety applications. Although they just discussed the movement characteristics in single-lane case during car-following process and did not refer to the term of SD, their work actually suggested that modeling SD is an important way to guarantee driving safety in car-following scenarios [2]. Based on SD, warning messages could be disseminated in advance to alert drivers for possible collisions. The latest published research results by Werling et al. [3] suggested an ICS (Inevitable Collision States) and PCS (Probabilistic Collision States) based driving safety assessment strategies for single road lanes. Their work considered the collision probability of the investigated vehicle and the collision avoidance behavior of the other vehicles at the same time. The authors also just discussed the case of single lane and their evaluation results showed the relevance of this safety assessment approach which allows to reduce computation time by using slower replanning cycles and to avoid unnecessary maneuvers caused by nonconservative safety assessment. Adell et al. [4] investigated the effectiveness of an assistance system for keeping safe speed and distance and showed positive effects of this system in terms of fewer alarm situations, shorter alarm lengths, shorter reaction times, increased headway, and better interactions with drivers. Even their work did not give the statistical analysis to SD and relied on feedbacks from drivers and logged driving data to output suggested results, they provided a powerful evidence for the necessity to introduce safety distance into driving assistance. Kannan et al. [5] proposed an Intelligent Driver Assistance System (I-DAS) which focuses on generating the alert messages based on the context aware parameters such as driving situations, vehicle dynamics, driver activity, and environment. In I-DAS, an adaptive cruise control system is embedded for alerting driver to reduce speed and maintain SD between vehicles and passengers. This system retrieves related driving and environment data by embedded sensors and returns the corresponding suggested actions based on previous collected relevant statistics. Therefore, their work is built on expert database and did not give the general statistical expressions to SD. Biral et al. [6] proposed a driver-support system that helps to maintain the correct speed and intervehicle distances with respect to lane curvature and other vehicles ahead. This work mainly focused on the warning maneuvers for drivers by comparing their behaviors to suggest system actions. Besides, they just took the different situations of the leading vehicles into account and did not consider the

relative movement status between the front and the back cars. Wu et al. [7] analyzed the SD for avoiding collisions when the leading car encounters obstacles or some emergency. However, they did not take the different movement status of the following car and traffic efficiency into consideration. Besides, the SD when the leading car drives normally, that is, without emergency or obstacles ahead, was not considered. Luo et al. [8] proposed the traditional braking model which derived the SD considering the influences from vehicular mechanical braking ability and relative velocity. But, they assumed the following car drives with a constant speed and did not discuss the scenario when the leading car suddenly stops. Ayres et al. [9] and Touran et al. [10] introduced the classical time-headway model. This model is based on the abstracted real test data from highway vehicles and uses the real-time distance between vehicles' heads as the metric to enhance driving safety. Although simulation results showed this work to be very effective in highway, its reliability is somehow weak because the authors mainly focused on traffic efficiency and did not consider the case where large velocity difference occurs between neighboring vehicles. Thereupon, the calculated SD is smaller and not enough to guarantee safety. Hiraoka et al. [11] presented a new metric, that is, Deceleration for Collision Avoidance (DCA), to evaluate the risk level to collide with the front obstacle. They derived the needed deceleration to avoid collision which could be used as the braking warning threshold for the Forward Obstacle Collision Warning System (FORCWS). This work just considered the case when obstacles or emergency is ahead of vehicles driving on single lane and did not take the normal acceleration/deceleration stages into account. Yan et al. [12] have given the mathematical expressions to the important parameters of cooperative collision warning systems (CCWS), such as inter-vehicle distance, reaction time, relative velocity, and conditional probability of collisions. They derived warning distance based on proposed Five-Second Rule, namely, the sum of three second for stopping plus the two Second reaction time, that is, five-second $\times$ velocity. Therefore, their obtained values were too realistic to guarantee safety under different mobility configurations. However, their work still built a better theoretical foundation for future researches on this field. Huang and Lin [13] designed a cooperative collision early-warning system (CCEW). They presented the early warning strategy by exchanging the relationship information (RI) between vehicles when they drive on the curve. Their work focused on deriving the collision probability by monitoring the distance between investigated vehicle and defined Closest Point of Approach (CPA). The discussed distance was not given by theoretical analysis but measured through wireless beacons. Garcia-Costa et al. [14] proposed a statistical model for evaluating the number of collided vehicles in a platoon. Their major contribution was providing a way to evaluate the influences of density, velocity, acceleration, and delay to collision probability. They also explored the impact of probability distribution of initial inter-vehicle distance to the resulted SD and collision possibility. Different from our work, they discussed the mobility scenario on single lane and did not consider traffic efficiency improvement.

Based on the research of previous strategies, we proposed a CSD model which has the following contributions.

(1) In single lane case, we analyzed the CSD considering different movement situations of the front and the back car. Besides, instead of assuming uniform or constant speed setting, we made full consideration on the impact from acceleration and deceleration on final computed CSD.

(2) Based on the obtained CSD, we introduced a multilevel warning mechanism by defining different notification zones corresponding to various safety requirements. This mechanism could consider the diverse mental tolerance of drivers and at the same time balance the demands between safety and traffic throughput.

(3) We also introduced the multi-lane case and gave a detailed analysis to the required SD when vehicles tend to change lane.

3. Theoretical Basis

To introduce our CSD model, we will first explore the movement features of vehicles especially when braking. Based on the analysis to the vehicular moving procedure for car following [1], a braking procedure could be divided into 4 stages as shown in Figure 3. t_1 indicates the reaction time of driver. Its average value is 0.9 s [15] in general. In this period, the driver finds that an emergency situation has happened ahead and is ready to take immediate actions. t_2 denotes the message propagation delay, which indicates the average needed time for warning messages is transmitted to the destination. Here, we also assume that it includes a little mechanical coordination time for braking preparation. t_3 denotes the process during which deceleration speed is still falling, which is about $0.1 \sim 0.2$ s and can be overlooked for calculation simplicity. t_4 expresses the period in which the vehicle maintains its maximum deceleration speed. Here, for simplicity, we set t_r as a braking operation response time, which is the sum of t_1 and t_2. During this period, the vehicle will still keep their original movement state. After t_r, its speed will linearly decrease due to the reception of braking triggering warning messages. Then, supposing the back car has an initial velocity v_0, that is, the velocity before t_1, and acceleration a, it will not brake until receiving the warning messages from the leading car. From [16], during $(0, t_r)$ the motion equation of the back car can be expressed as

$$v(t) = v_0 + at, \quad t < t_r. \tag{1}$$

After t_r, the vehicle's speed is linearly decreasing with the maximum deceleration value $a_{\max}$ and satisfying the following equation:

$$v(t) = v_0 + at_r - a_{\max}(t - t_r), \quad t > t_r. \tag{2}$$

4. Assumptions and Definition

In this section, we will give some assumptions and define CSD in our model based on the aforementioned movement features.

Assumption 1. Vehicles could know their own locations involving coordinates on map by GPS (Global Positioning System) and GIS (Geographic Information System).

Assumption 2. Vehicles could know neighbors' locations, speeds, and accelerations by periodical broadcasting beacons. The typical used localization methods are Receive Signal Strength Indicator (RSSI) under LOS (Line Of Sight) environment or hybrid TOA/AOA [17] algorithms under NLOS (Non Line Of Sight) circumstance. The needed localization parameters could be piggybacked or abstracted in beacons.

Assumption 3. All the vehicles have the same mechanical brake performance.

Assumption 1 is readily satisfied because GPS and GIS now are popular and cheap for vehicles equipment. Assumption 2 is also reasonable especially in vehicular networks where periodical beacons are necessary for vehicles to inform their neighbors of their current state (velocity, acceleration, position, direction) in order to improve safety conditions. Assumption 3 here makes our analysis simple but will not influence correctness of the final obtained results.

Based on the aforementioned assumptions, we define our CSD as follows.

Definition 1. Critical Safe Distance in our work is defined as the needed minimum distance between vehicles to avoid collisions based on V2V communications in VSNs.

By this definition, the CSD means that whatever the leading vehicle does, there always exists a feasible maneuver that leads to no collision for the back car. On the other hand "whatever" here has to be properly considered. Assuming a heavy vehicle intersects the leading car path in an instant, if here the safe distance is defined as the stopping distance between vehicles, then whatever actions are to be taken, a tragedy may not be avoided. For this reason, traditional safe distance usually supposes that leading vehicle will not crash and the back car only needs to stop before the relative distance becomes zero. In this sense, our definition seems to be riskful and not suitable. However, we think the above case is also a use case which should be included in the safe distance application range. In fact, by properly setting beacon frequency or introducing some forecasting algorithm, a crash to obstacle is also solvable. In our work, we discuss the probability to use safe distance to avoid collision and give references for beacon frequency setting based on CSD. In this way, there may exist chances to make the following car escaping from colliding with the leading vehicle when it crashes to obstacles ahead. Therefore, our definition is reasonable and universal in most of the cases.

Next, to give our CSD model, we will first discuss the scenario that all vehicles drive in one dimension highway and the results for the multi-lane case will be given later by introducing directions, vertical height of vehicles and the lane-changing maneuver.

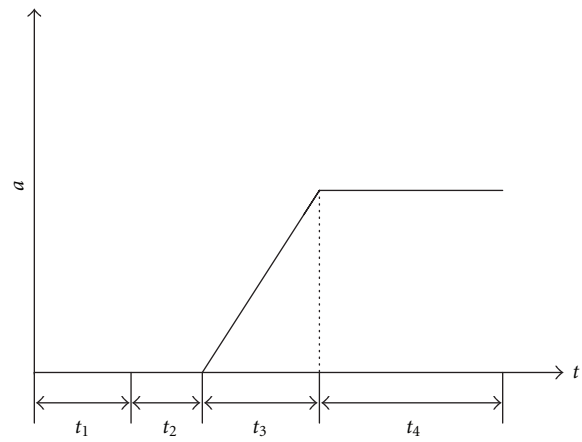


Figure 3: Vehicle braking procedure.

5. CSD Analysis in a Single Lane

In this section, we analyze the CSD under different movement configurations for the front and the back car in single lane case. The discussed scenario is shown in Figure 4. Supposing the displacement for A is s_A from sending out the collision warning messages to the moment that danger has disappeared. Here, A could be stationary; that is, s_A is zero, which will be discussed later. The displacement for B is s_B from the time it received the warning messages to the moment that collision has been successfully avoided. s_0 indicates the permitted minimum distance between A and B when collision has been successfully avoided [18]. For simplicity, we assign s_0 to zero in later analysis. It can also be set to an acceptable constant distance according to different safety requirements and drivers' courage. s_{csd} denotes the CSD between A and B and is expressed as

$$s_{csd} = s_B - s_A \tag{3}$$

when A has different movement states.

To deeply explore the CSD in this scenario, we further divide this case into two subcases: in subcase 1 (see Section 5.1), the front vehicle A travels without obstacles or an emergent events ahead, whereas in subcase 2 (see Section 5.2) A has encountered accidents or obstacles and has been forced to brake immediately. Here, we define v_a^0, v_b^0 as the initial speed of A and B, that is, the speed when A or B enters into the corresponding dotted line, and a_A, a_B as the corresponding acceleration/deceleration of them, respectively.

5.1. A Travels without Obstacle or an Emergent Event Ahead. According to A's movement status, this subcase can be further classified into 3 situations as follows.

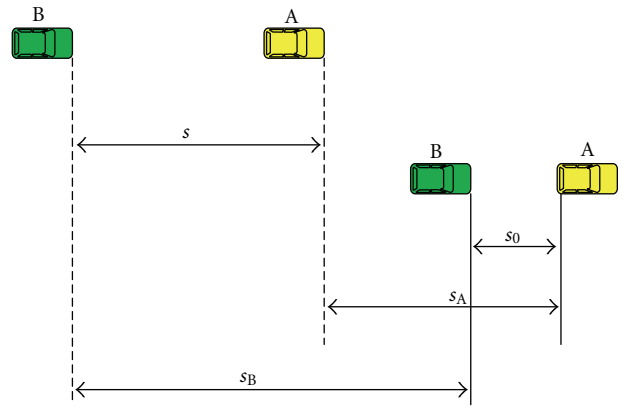


Figure 4: An example illustrating CSD.

5.1.1. The Leading Vehicle A Is Stationary

Corollary 2. *When A is stationary, if B can safely brake after receiving the warning messages from A, the needed CSD for B to avoid collision is*

$$s_{\text{csd1}} = \text{sign}\left(-\left(v_b^0 + a_{\text{B}}t_r\right)\right)\frac{\left(v_b^0\right)^2}{-2a_{\text{B}}} + \text{sign}\left(v_b^0 + a_{\text{B}}t_r\right)\left(v_b^0 t_r + \frac{1}{2}a_{\text{B}}{t_r}^2 + \frac{\left(v_b^0 + a_{\text{B}}t_r\right)^2}{2a_{b\max}}\right), \tag{4}$$

where $\text{sign}(x)$ *is the sign function, expressed as follows:*

$$\text{sign}(x) = \begin{cases} 1, & x > 0, \\ 0, & x < 0. \end{cases} \tag{5}$$

Proof. When B starts to take braking measure, it generally consists-two parts: the braking operation reaction time and the linearly decreasing progress.

If $v_b^0 + a_{\text{B}}t_r < 0$, it means B is still decelerating and has completely stopped during the braking operation response time. Then the total displacement for B to stop safely is

$$s_{bt} = \frac{\left(v_b^0\right)^2}{-2a_{\text{B}}}, \tag{6}$$

else, if $v_b^0 + a_{\text{B}}t_r > 0$, the distance travelled during $(0, t_r)$ is

$$s_{b1} = v_b^0 t_r + \frac{1}{2}a_{\text{B}}{t_r}^2. \tag{7}$$

Correspondingly, the displacement from t_r to the moment by which B has stopped safely is

$$s_{b2} = \frac{\left(v_b^0 + a_{\text{B}}t_r\right)^2}{2a_{b\max}}. \tag{8}$$

As a result, the total displacement for the back car B in this situation is

$$s_{bt} = s_{b1} + s_{b2}. \tag{9}$$

So, with (6) and (9), we have (4). □

5.1.2. The Leading Vehicle A Is Still Accelerating, That Is, $a_A \geq 0$. Here, according to the difference between two vehicles' initial speeds, we can discuss this subcases under two scenarios.

(1) A's Initial Speed is Less than B's. When the initial speed of A is less than that of B, there is a large probability to result in collision. Next, we discuss the needed CSD in this scenario.

Corollary 3. *In this situation, the needed CSD for* B *to avoid collision is*

$$\begin{aligned} s_{\text{csd2}} = {} & \text{sign}\left(v_a^0 + a_A t_r - v_b^0 - a_{\text{B}}t_r\right) \\ & \times \left(v_b^0 t_1 + \frac{1}{2}a_{\text{B}}{t_1}^2 - v_a^0 t_1 - \frac{1}{2}a_{\text{A}}{t_1}^2\right) \\ & + \text{sign}\left(v_b^0 + a_{\text{B}}t_r - v_a^0 - a_{\text{A}}t_r\right) \\ & \times \left\{v_b^0 t_r + \frac{1}{2}a_{\text{B}}{t_r}^2 + \left(v_b^0 + a_{\text{B}}t_r\right)(t_2 - t_r)\right. \\ & \left. - \frac{1}{2}a_{b\max}(t_2 - t_r)^2 - v_a^0 t_2 - \frac{1}{2}a_{\text{A}}{t_2}^2\right\}, \end{aligned} \tag{10}$$

where $t_1 = (v_b^0 - v_a^0)/(a_A - a_{\text{B}})$, *and* $t_2 = (v_b^0 - v_a^0 + (a_B + a_{b\max})t_r)/(a_{\text{A}} + a_{b\max})$.

Proof. If A's speed is greater than B's at t_r, that is, $v_a^0 + a_{\text{A}}t_r > v_b^0 + a_{\text{B}}t_r$, we define t_1 as the moment by which two vehicle velocities become the same. Then, we can express t_1 as

$$t_1 = \frac{v_b^0 - v_a^0}{a_{\text{A}} - a_{\text{B}}}. \tag{11}$$

Before t_1, A's speed is less than B's and the inter-vehicle distance is continuously decreasing. During (t_1, t_r), A's speed still increases and exceeds B. Then, B's braking action takes effect. Apparently, after t_1, the inter-vehicle distance starts to become bigger. So if the inter-vehicle distance is greater than zero at t_1, which is the moment that the inter-vehicle distance is the minimum, they are relatively safe.

Thereupon, the distance travelled for A by t_1 is

$$s_{a2} = v_a^0 t_1 + \frac{1}{2}a_{\text{A}}{t_1}^2 \tag{12}$$

and the corresponding displacement for B is

$$s_{b2} = v_b^0 t_1 + \frac{1}{2}a_{\text{B}}{t_1}^2. \tag{13}$$

Finally, the needed CSD for B to avoid collision in this situation is:

$$s_{\text{csd}} = s_{b2} - s_{a2}. \tag{14}$$

If A's speed is still less than B's at t_r, that is, $v_a^0 + a_{\text{A}}t_r < v_b^0 + a_{\text{B}}t_r$, then, when braking begins, A's speed will finally be equal to B's after t_r. We define the moment as t_2 by which the two vehicles speeds become the same, and it can be attained by the following equation:

$$t_2 = \frac{v_b^0 - v_a^0 + (a_{\text{B}} + a_{b\max})t_r}{a_{\text{A}} + a_{b\max}}. \tag{15}$$

Here, A's speed will not exceed B's until t_2. So during the time the inter-vehicle is always decreasing. After t_2, the inter-vehicle distance begins to gradually increase due to B's braking. Thereupon, if the inter-vehicle distance can be greater than zero at t_2, which is also the moment that the inter-vehicle distance is the minimum, the two vehicles are safe.

By t_2, the displacement for A is

$$s_{a3} = v_a^0 t_2 + \frac{1}{2} a_{\mathrm{A}} {t_2}^2, \tag{16}$$

and for B is

$$s_{b3} = v_b^0 t_r + \frac{1}{2} a_{\mathrm{B}} {t_r}^2 + \left(v_b^0 + a_{\mathrm{B}} t_r\right)(t_2 - t_r) - \frac{1}{2} a_{b\ \max}(t_2 e - t_r)^2. \tag{17}$$

Accordingly, the needed CSD for B to avoid collision in this situation is

$$s_{\mathrm{csd}} = s_{b3} - s_{a3}. \tag{18}$$

According to (14) and (18), (10) can be derived. □

(2) A's Initial Speed Is Greater than B's. If A's initial speed is greater than B's, the collision will only occur when B drastically accelerates before braking.

Corollary 4. *In this situation, the needed CSD for B to avoid collision is*

$$s_{\mathrm{csd3}} = \mathrm{sign}\left(v_b^0 + a_{\mathrm{B}} t_r - v_a^0 - a_A t_r\right) \times \left\{ v_b^0 t_r + \frac{1}{2} a_{\mathrm{B}} {t_r}^2 + \left(v_b^0 + a_{\mathrm{B}} t_r\right)(t_4 - t_r) - \frac{1}{2} a_{b\ \max}(t_4 - t_r)^2 - v_a^0 t_4 - \frac{1}{2} a_{\mathrm{A}} {t_4}^2 \right\}, \tag{19}$$

where, $t_4 = (v_b^0 - v_a^0 + (a_{\mathrm{B}} + a_{b\ \max})t_r)/(a_{\mathrm{A}} + a_{b\ \max})$.

Proof. If B's speed is still less than A's at t_r, that is, $v_b^0 + a_{\mathrm{B}} t_r < v_a^0 + a_{\mathrm{A}} t_r$, the inter-vehicle distance will continuously increase before t_r. After t_r, B begins to brake and the inter-vehicle distance further becomes bigger. Therefore, in such circumstance, there is no probability to result in collision. For simplicity, we set the needed CSD under this case to zero.

If B's speed is greater than A's at t_r, that is, $v_b^0 + a_{\mathrm{B}} t_r > v_a^0 + a_{\mathrm{A}} t_r$, we define t_3 as the moment by which their speeds are equal. During $(0, t_3)$, the inter-vehicle distance keeps increasing. During (t_3, t_r), the inter-vehicle distance begins to decrease because B's speed exceeds A's. After t_r, considering that A is accelerating and B is decelerating, there exists one occasion t_4 at which their speeds will be equal again, that is

$$t_4 = \frac{v_b^0 - v_a^0 + (a_{\mathrm{B}} + a_{b\ \max})t_r}{a_{\mathrm{A}} + a_{b\ \max}}. \tag{20}$$

By t_4, the distances travelled of the two vehicles are respectively:

$$s_{a4} = v_a^0 t_4 + \frac{1}{2} a_{\mathrm{A}} {t_4}^2,$$

$$s_{b4} = v_b^0 t_r + \frac{1}{2} a_{\mathrm{B}} {t_r}^2 + \left(v_b^0 + a_{\mathrm{B}} t_r\right)(t_4 - t_r) - \frac{1}{2} a_{b\ \max}(t_4 - t_r)^2. \tag{21}$$

During (t_r, t_4), the inter-vehicle distance continues to decrease, for B's speed is still greater than A's. After then, namely, B's barking actions come into play, the inter-vehicle distance starts to increase. So if the inter-vehicle distance could be greater than zero at t_4, they will be safe. Consequently, the needed CSD for B to avoid collision in this situation is

$$s_{\mathrm{csd}} = s_{b4} - s_{a4}. \tag{22}$$

Based on aforementioned analysis and (22), (19) can be derived. □

5.1.3. The Leading Vehicle A's Is Decelerating, That Is, $a_A < 0$. We assume A is decelerating with the maximum deceleration $a_{\max}$. Here, $a_{\mathrm{A}} = -a_{a\ \max} = -a_{\max}$.

(1) A's Initial Speed Is Less than B's. Consider the following.

Corollary 5. *In this situation, the needed CSD for B to avoid collision is*

$$s_{\mathrm{csd4}} = \mathrm{sign}(a_{\mathrm{B}}) \left(v_b^0 t_r + \frac{1}{2} a_{\mathrm{B}} {t_r}^2 + \frac{\left(v_b^0 + a_{\mathrm{B}} t_r\right)^2}{2a_{b\ \max}} - \frac{(v_a^0)^2}{2a_{a\ \max}} \right) + \mathrm{sign}(-a_{\mathrm{B}}) \left(\frac{\left(v_b^0\right)^2}{2a_{b\max}} - \frac{(v_a^0)^2}{2a_{a\max}} \right). \tag{23}$$

Proof. To proof this corollary, we need to divide these subcases into different situations based on B's movement status.

(1) B is still accelerating, that is, $a_{\mathrm{B}} \geq 0$. This situation is very dangerous. Because A is decelerating and B is accelerating, the inter-vehicle distance will continuously decrease even when B begins to brake based on Assumption 3. So the inter-vehicle distance reaches the minimum when both vehicles stop.

Define s_{at} as the displacement of A by the time it stops safely. It can be expressed as follows

$$s_{at} = \frac{(v_a^0)^2}{2a_{\mathrm{A}}}. \tag{24}$$

Correspondingly, s_{bt} for B can be attained from (9). Therefore, the needed CSD for B to avoid collision is:

$$s_{\text{csd}} = s_{bt} - s_{at}. \tag{25}$$

(2) B is still decelerating, that is, $a_{\text{B}} < 0$. Here, $a_{\text{B}} = -a_{b\max} = -a_{\max}$.

It can be noticed that the minimum inter-vehicle distance will also occur when both vehicles stop.

The generated displacement for A to stop safely is

$$s_{a5} = \frac{(v_a^0)^2}{2a_{a\max}}, \tag{26}$$

and for B is

$$s_{b5} = \frac{\left(v_b^0\right)^2}{2a_{b\ \max}}. \tag{27}$$

Therefore, the needed CSD for B to avoid collision is

$$s_{\text{csd}} = s_{b5} - s_{a5}. \tag{28}$$

With (25) and (28), we could proof (23). □

(2) A's Initial Speed Is Greater than B's. Consider the following.

Corollary 6. *In this situation, the needed CSD for B to avoid collision is*

$$s_{\text{csd5}} = \text{sign}(a_{\text{B}})\left(v_b^0 t_r + \frac{1}{2}a_{\text{B}}{t_r}^2 + \frac{\left(v_b^0 + a_{\text{B}}t_r\right)^2}{2a_{b\ \max}} - \frac{(v_a^0)^2}{2a_{a\ \max}}\right). \tag{29}$$

Proof. We also proof this corollary according to B's movement status.

(1) B is still accelerating, that is, $a_{\text{B}} > 0$. This situation is also dangerous. Similar to Section 5.1.2 (2), if B's speed at t_r is less than A's, the needed CSD for B to avoid to collision is zero. If B's speed at t_r is greater than A's, we define t_5 as the moment by which their speeds become equal. We can calculate t_5 with the following equation:

$$t_5 = \frac{v_b^0 - v_a^0}{a_{\text{A}} - a_{\text{B}}}. \tag{30}$$

During $(0, t_5)$, because A's speed is greater than B's, the inter-vehicle distance is becoming bigger. During (t_5, t_r), for B's speed exceeds A's, the inter-vehicle distance starts to decrease. After t_r, B's braking takes effect and the distance will continuously reduce until both vehicles stop. Then, the inter-vehicle distance reaches the minimum when both vehicles stop safely. As a result, the needed CSD for B to avoid collision in this situation is

$$s_{\text{csd}} = s_{bt} - s_{at}. \tag{31}$$

(2) B is still decelerating, that is, $a_{\text{B}} < 0$. This situation is safe and the needed CSD can be defined to zero.

Therefore, based on the above analysis and (31), (29) can be deduced. □

Corollary 7. *The general expression of the needed CSD to avoid collision can be written as*

$$\begin{aligned} s_{\text{acsd}} = {} & \text{sign}(a_{\text{A}})\{\text{sign}(v_r^0)s_{\text{csd2}} + \text{sign}(-v_r^0)s_{\text{csd3}}\} \\ & + \text{sign}(-a_{\text{A}})\{\text{sign}(v_r^0)s_{\text{csd4}} + \text{sign}(-v_r^0)s_{\text{csd5}}\}, \end{aligned} \tag{32}$$

where $v_r^0 = v_b^0 - v_a^0$.

Proof. Through our analysis, it is known that Section 5.1.1 is one special case of Section 5.1.2. Combining (10), (19), (23), and (29) leads to (32). □

5.2. A Encounters Obstacle or Emergency Ahead. In some cases, the leading vehicle may be forced to stop or even collide to the obstacle ahead. Based on [7], we know that there may generally three subcases for the leading vehicle A when it encounters obstacle or emergency ahead. In the first subcase, the speed of A becomes 0 through normal speed reduction; in the second subcase, after finding the emergency ahead, A begins to brake but finally collides with the obstacle before its speed approaches to 0; in the third subcase, after taking notice of the front obstacle, A will collide with the obstacle with its initial speed before the driver has response time to start braking. Now, we first explore the corresponding CSD values in these three subcases, respectively. Then, in the last part of the section, through the comparison of the three subcases, a reasonable CSD equation is given under the circumstance.

5.2.1. A Successfully Stops by Decelerating. In this situation, A finds the danger ahead and successfully stops through normal deceleration. Therefore, the needed CSD for B to avoid collision can also be obtained based on the analysis in Section 5.1.3.

5.2.2. A Collides with the Obstacle before Its Speed Reduces to Zero. When A discovers obstacle or emergency ahead, generally, it will try its best to brake. However, considering different reaction speeds and driving habits, it may collide with obstacle before its speed decreases to zero. Here, assuming A, B, and C are driving in the same lane as shown in Figure 5. When A and B received the warning messages from C notifying the accident ahead, they will immediately begin to brake. Assuming that A will inevitably collide to C, we next analyze the needed CSD for B to avoid to crash to A. First, we suppose the inter-vehicle distance following an exponential distribution [19] with a parameter λ: $f(x) = \lambda e^{-\lambda x}$.

Corollary 8. *In this situation, the needed CSD for B to avoid collision is*

$$s_{\text{csd}} = s_b - s_a(1 - F(s_a)) - \frac{1}{\lambda} + \left(s_a + \frac{1}{\lambda}\right)e^{-\lambda s_a}, \tag{33}$$

where

$$s_a = \operatorname{sign}\left(-\left(v_a^0 + a_A t_r\right)\right)\frac{\left(v_a^0\right)^2}{-2a_A} + \operatorname{sign}\left(v_a^0 + a_A t_r\right)\left(v_a^0 t_r + \frac{1}{2}a_A {t_r}^2 + \frac{\left(v_a^0 + a_A t_r\right)^2}{2a_{a\,\max}}\right)$$

$$s_b = s_{\text{csd1}}. \tag{34}$$

Proof. The distance s_a, s_b traveled by A, B until they completely stop without collision can be attained from (4), respectively.

For A, it will collide with C if and only if the needed distance for A to stop is greater than the distance between them plus the average distance $\bar{l}_c$ traveled by C. Thereupon, the collision probability for A is

$$p_{ac} = p\left(s_a > \bar{l}_c + s_{ac}\right) = F\left(s_a - \bar{l}_c\right), \tag{35}$$

where s_{ac} indicates the inter-vehicle distance between A and C. Then the average distance traveled by A in case of collision is

$$\begin{aligned}\bar{l}_{a\text{collide}} &= \frac{1}{p_{ac}}\int_0^{s_a-\bar{l}_c}\left(\bar{l}_c + x\right)\lambda e^{-\lambda x}dx \\ &= \frac{1}{p_{ac}}\left(\bar{l}_c + \frac{1}{\lambda} - \left(s_a + \frac{1}{\lambda}\right)e^{-\lambda\left(s_a-\bar{l}_c\right)}\right).\end{aligned} \tag{36}$$

Based on above analysis, the average distance traveled by A is

$$\bar{l}_a = s_a\left(1 - p_{ac}\right) + \bar{l}_{a\text{collide}}p_{ac}. \tag{37}$$

$\bar{l}_c$ can be obtained in the same way as $\bar{l}_a$, which is calculated with the assumption that vehicle C encounters an obstacle ahead. Here, for convenience, assume $\bar{l}_c = 0$. □

As a result, the need CSD for B to avoid to collide with A is

$$s_{\text{csd}} = s_b - \bar{l}_a. \tag{38}$$

From (38), we can further get (33).

5.2.3. A Collides with the Obstacle with Its Initial Speed. If the leading vehicle A is aware of the obstacle or emergency ahead before it has time to start braking, it will collide with the obstacle with its initial speed. Here, considering the few displacement, we can approximately regard A as stationary. After receiving the warning messages and realizing the accident ahead, B then begins to brake. Therefore, to avoid accident, the needed CSD between A and B can also be derived based on the analysis in Section 5.1.1 and from (4) as

$$s_{\text{csd}} = s_{\text{csd1}}. \tag{39}$$

Among the above three subcases, the front vehicle A has the largest displacement in the first subcase. Further, A in the third subcase has a shorter displacement than the second due to the absence of any speed reduction action. From (3), the needed CSD between A and B in the third subcase is the greatest among the three situations. However, because B can not predict the upcoming state of A, it is reasonable to assign the calculated CSD for the third subcase as the final critical safe inter-vehicle distance, which is defined as $s_{b\text{csd}}$ for all the three subcases.

6. Multilane Scene

To analyze the needed CSD in multi-lane, we introduce a discussed scenario as shown in Figure 6. c_0 indicates the vehicle which will change lane. c_1 and c_2 are two vehicles in the same lane with c_0. c_3 and c_4 are two vehicles in the adjacent lane. When c_0 begins to change lane, it will move to the adjacent lane with a given lateral acceleration and follows c_3. Introducing a Cartesian rectangular coordinate and define a_{xi}, v_{xi}^0, v_{xi}, x_i, a_{yi}, v_{yi}, and y_i as the longitudinal acceleration, longitudinal initial velocity, longitudinal velocity, longitudinal position, lateral acceleration, lateral velocity and lateral position, respectively, where $i \in \{c_0, c_1, c_2, c_3, c_4\}$ and (x_i, y_i) indicates the upper-right corner of the vehicle. For convenience, we assume $a_{yi} = 0$ and $v_{yi} = 0$, $i = c_1, c_2, c_3, c_4$. Without loss of generality, we suppose c_0 starts its lane-changing maneuver at $t = 0$ and this action consists of two stages, that is, the duration t_{adj} before taking lateral acceleration and the part t_{lat} after t_{adj} to complete lane changing. Here, based on [20], we introduce a simple but realistic model for the lateral acceleration during the lane changing maneuver, that is

$$a_{\text{lat}}(t) = \begin{cases}\dfrac{2\pi H}{t_{\text{lat}}^2}\sin\left(\dfrac{2\pi}{t_{\text{lat}}}\left(t - t_{\text{adj}}\right)\right), & t_{\text{adj}} \le t \le t_{\text{lat}} + t_{\text{adj}} \\ 0, & \text{otherwise,}\end{cases} \tag{40}$$

where H is the total lateral displacement.

Then the corresponding longitudinal velocity, longitudinal position can be expressed as follows:

$$v_{\text{lat}}(t) = \begin{cases}\dfrac{-H}{t_{\text{lat}}}\cos\left(\dfrac{2\pi}{t_{\text{lat}}}\left(t - t_{\text{adj}}\right)\right) + \dfrac{H}{t_{\text{lat}}}, & t_{\text{adj}} \le t \le t_{\text{lat}} + t_{\text{adj}}, \\ 0, & \text{otherwise,}\end{cases} \tag{41}$$

$$y_{\text{lat}} = \begin{cases} H, & t \ge t_{\text{lat}} + t_{\text{adj}}, \\ \dfrac{-H}{2\pi}\sin\left(\dfrac{2\pi}{t_{\text{lat}}}\left(t - t_{\text{adj}}\right)\right) & \\ \quad + \dfrac{H}{t_{\text{lat}}}\left(t - t_{\text{adj}}\right), & t_{\text{adj}} \le t \le t_{\text{lat}} + t_{\text{adj}}, \\ 0, & \text{otherwise.}\end{cases} \tag{42}$$

The needed CSD in the same lane, for example, between c_0 and c_1 or c_0 and c_2 in Figure 6 can be calculated according to Section 5. Next, we mainly analyze the needed CSD between c_0 and c_3 when c_0 tends to change its lane.

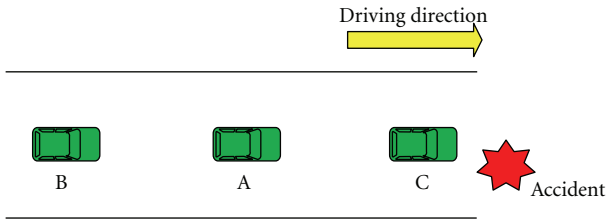


Figure 5: An example illustrating CSD analysis when the leading vehicle encounters an emergency.

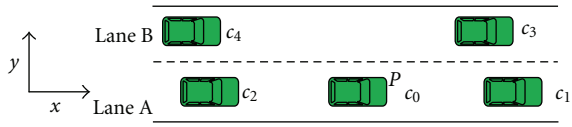


Figure 6: Changing lane scene in multilane.

Figure 7 shows the detailed lane-changing procedure discussed in our paper. Denote S as the initial lateral distance between c_0 and c_3. H is the total lateral displacement of c_0 after lane changing. LS is the tangent to the lower side of c_3. θ is the steering angle introduced in the lane-changing procedure. c_0 starts the lane-changing maneuver at $t = 0$ by adjusting its longitudinal position and velocity, and then applying lateral acceleration at $t = t_{\text{adj}}$ according to (40). The upper-right corner of c_0 will first pass the line at point C in Figure 7. Let $t_C + t_{\text{adj}}$ be the moment at which the upper-right corner of c_0 is at C. In other words, t_C indicates the time by which the vehicle will switch to the neighboring lane. According to (42), we have the following equation:

$$y_{\text{lat}}\left(t_C + t_{\text{adj}}\right) = S = y_{c_3} - w_{c_3}, \tag{43}$$

where w_{c_3} indicates the width of c_3. According to [21], the value of $t_C + t_{\text{adj}}$ can also be solved.

The type of collisions which may take place during or after $t_C + t_{\text{adj}}$ is the angle, side-wipe or rear-end collision. The angel or side-wipe, collision may occur during or after the moment when the upper-right corner of c_0 passes C. From [21], we know that the minimum initial longitudinal relative spacing between c_0 and c_3 to avoid angel or side-wipe collision is

$$d_{\min}(c_0, c_3) = \max\left(\int_0^t \int_0^\lambda \left(a_{c_0 y}(\tau) - a_{c_3 y}(\tau)\right) d\tau d\lambda + \left(v_{c_0 y}(0) - v_{c_3 y}(0)\right) t\right), \quad \forall t \in \left[0, t_c + t_{\text{adj}}\right]. \tag{44}$$

Then the corresponding longitudinal velocities of c_0 and c_3 are

$$v_{c_0} = v_{xc_0}^0 + \int_0^{t_C + t_{\text{adj}}} a_{xc_0}(t) dt, \tag{45}$$

$$v_{c_3} = v_{xc_3}^0 + \int_0^{t_C + t_{\text{adj}}} a_{xc_3}(t) dt, \tag{46}$$

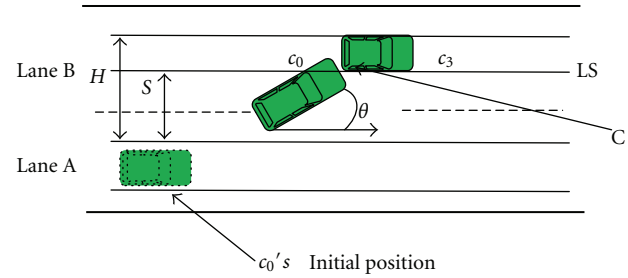


Figure 7: Lane changing procedure.

respectively. For simplicity, here, we assume that the longitudinal accelerations a_{xc_0} and a_{xc_3} are constant, and the spacing $d_{\min}(c_0, c_3)$ that could guarantee the inter-vehicle distance between c_0 and c_3 is greater than zero at $t_C + t_{\text{adj}}$. However, this minimum space does not take the rear-end collision into account which may occur after c_0 finishing the lane changing maneuver and following c_3 in the same lane. If c_3 encounters obstacle ahead at the time, there is possibility to happen collision between the two vehicles. In such case, according to the movement status of two vehicles and the needed CSD in the single lane, we can obtain the minimum longitudinal spacing $s_{c_0 c_3}$ between c_0 and c_3 after $t_C + t_{\text{adj}}$. It can also be expressed as a function of the mobility parameters of the two vehicles, that is

$$s_{c_0 c_3} = f\left(v_{c_0}, a_{xc_0}, v_{c_3}, a_{xc_3}\right). \tag{47}$$

Based on (44) and (47), we can get that the needed CSD for c_0 to avoid collision during and after the lane-changing maneuver is

$$s_{\text{csd}} = d_{\min}(c_0, c_3) + s_{c_0 c_3}. \tag{48}$$

7. Notification Zones Setting

A wise CSD definition could consider the successful collision avoidance and at the same time satisfy traffic throughput on a given road segment. To reach this goal, a strategy to define different warning messages notification zones based on obtained CSD is necessary. Here, we discuss the notification zones setting method in single-lane case and the policy for multi-lane case can be readily extended with this result. According to the above analysis (3), it can be noticed that the CSD $s_{a\text{csd}}$ in Section 5.1 is less than $s_{b\text{csd}}$ in Section 5.2. A smaller safe distance permits a larger number of vehicles driving on a given road segment. Therefore, the former, that is, $s_{a\text{csd}}$, can improve traffic throughput to some extent, although there is some probability of collision if obstacle or emergency exists ahead. On the contrary, the latter has a large probability to avoid collision whereas simultaneously decreases the traffic throughput. To balance the requirements from safety and traffic throughput, a multi-level warning strategy can be introduced with different notification zones' setting as shown in Figure 8. Here, $s_{\text{notification}}$, that is, $s_{b\text{csd}}$, is taken as the CSD for cautious drivers or high safety requirement applications whereas s_{critical}, that is, $s_{a\text{csd}}$, is for the daring drivers or scenarios with lower safety demands.

Therefore, we can introduce two emergency notification zones $|x - x_A| < s_{critical}$ and $s_{critical} < |x - x_A| < s_{notification}$. If B enters into the notification zones, it will decide to take corresponding measurements based on its warning level, that is, ignoring or braking.

Besides, our CSD model could also provide references for the interval or frequency setting for notification beacons. We assign t to indicate how long warning messages can be received. Let $t1$ be the time when Benters the low-level warning zone, then $SB(t1) + x_B \leq SA(t1) + x_A - s_{notification}$, where $SA(t1)$, $SB(t1)$ indicate the already generated displacement of A and B, respectively, at $t1$. Let $t2$ be the time when B enters the high-level warning area, then $SB(t2) + x_B \leq SA(t2) + x_A - s_{critical}$, where $SA(t2)$, $SB(t2)$ indicate the already generated displacement of A and B, respectively, at $t2$. Thereupon, we can get the required message transmission delay $t = t2 - t1$. With this result, we can flexibly control the beacon frequency so that messages could be successfully transmitted to B but at the same time avoided affecting the driver's mentality due to too often warnings.

8. Numerical Evaluation and Analysis

8.1. Compared Models' Description. In this section, to evaluate the performance of our proposed CSD model on driving safety guarantee and traffic throughput improvement, we implemented our algorithm on MATLAB and compared its results to the traditional braking model [8] and the time-headway model [9, 10]. The braking model also introduced SD conception and computed inter-vehicle distance based on vehicular movement features as well. Thereupon, we can naturally contrast its performance with ours. However, in time-headway model, the author assumed the host vehicle will always be in a safe state after the reaction time. The reason is, to calculate distance between vehicles' heads at different moments, the back car is supposed to be in the same position where was the leading vehicle. In this way, the elapsed time could be obtained between the front of the lead vehicle passing a point on the roadway and the front of the following vehicle passing the same point. As a result, it can be noticed that the definitions for SD in time-headway and our model are distinguishing and seemed not comparable. However, in fact, based on [22], we can note that our definition actually does not influence the safe distance derivation although assuming the leading vehicle may collide to obstacle or immediately stop. Accordingly, it is acceptable to compare the results between our model and the time headway. Next, the numerical results and performance evaluations will be given to show the correctness and effectiveness of our CSD model.

8.2. Numerical Results. The general simulation parameter is listed in Table 1.

First, we will investigate the CSD performance when the front or leading car is on different movement states. Figures 9 and 10 show the CSD performance when the front car is accelerating. Figure 9 shows the CSD performance among time headway, traditional braking, and our proposed

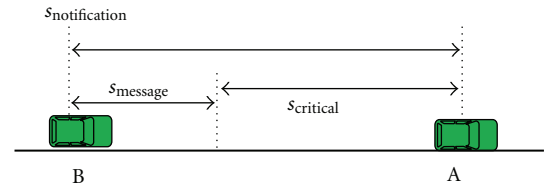


Figure 8: Communication zone.

model, when the initial speed of the leading car, that is, $v_A = 50$ km/h, is smaller than that of the rear car. The acceleration of the front car here is $a_A = 1\,m/s^2$. It can be noticed that the calculated CSDs for the three model will all increase when the initial velocity of the rear car grows. This result is reasonable for our common sense. The traditional braking model gives the largest CSD because a constant velocity is assumed for the back car by which the risk for collision will be overestimated in their work. For the time-headway model, regarding their design goal to increase traffic throughput on a highway, the needed inter-vehicle distance to avoid crash is underestimated due to few consideration on relative movement states of the investigated cars. For our proposed model, it can be noticed from Figure 9 that it has the smallest CSD than the other two in most of the situations. Although a small safe distance means a high risk for collision, our protocol could guarantee the vehicles' safety through considering not only the mobility situation of the leading car, but also the moving state of the back car and their relative movement status. As a result, our CSD model could ensure the safety, meantime increase the traffic efficiency by introducing more vehicles on a given length of road segment. Besides, it is shown by our model with different a_B that a larger acceleration will result in a bigger CSD when the initial speed of the back car is the same. In other words, a quick speeding up means more time needed for braking to avoid collision which is logical in our driving.

Figure 10 shows the CSD results when the leading car has a bigger initial speed than the rear car. Here, the two vehicles are both accelerating with $a_A = 1\,m/s^2$ and $a_B = 2\,m/s^2$. The other simulation parameters are listed in Table 1. It can be noted that our CSD model still shows a zero value before the initial velocity of the back car approaches 48 km/h, whereas the other two models output an increasing CSD with the initial speed grows during the whole simulation process. The reason is that, in this case, the speed of the back car is still smaller than the leading car before about 48 km/h though it has a bigger acceleration. In such situation, the velocity of the back car will never catch up that of the leading car before/after the braking action and there is no need to set safe distance. For simplicity, we set CSD to zero during this interval. After 48 km/h, the rear car could surpass the front car on speed before braking takes effect and there is probability to crash. Therefore, CSD is necessary and shows a positive value. From Figure 10, we can conclude that our proposed CSD model is more practical than the other two and can further improve the traffic efficiency.

Figures 11 and 12 give the CSD performance when the leading car has a negative acceleration or in an other words is

Table 1: General simulation parameters.

Name	Value	Name	Value
Drivers' physical response time t_1 (s)	0.9	Average propagation delay of warning messages t_2 (s)	0.1
Initial speed of the leading car v_A (km/h)	50	Maximum deceleration a_{max} (m/s)	7.0

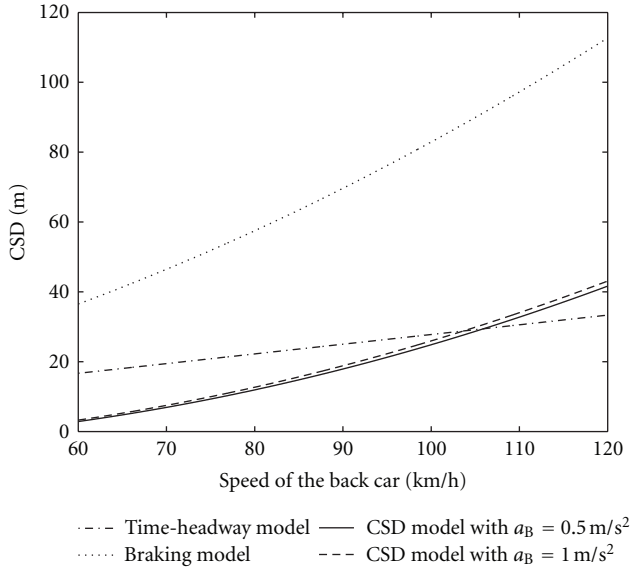


Figure 9: A is accelerating and its initial speed is less than B's.

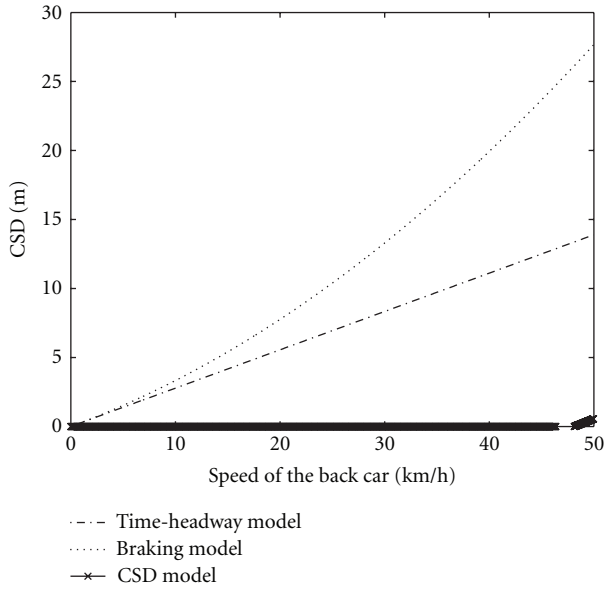


Figure 10: A is accelerating and its initial speed is greater than B's.

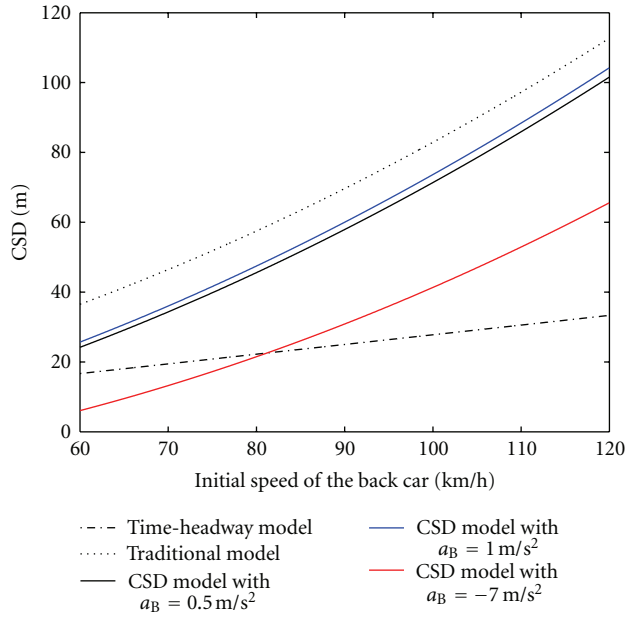


Figure 11: A is decelerating and its initial speed is less than B's.

decelerating. Figure 11 shows the case where the leading car's initial speed is less than the back car's. It can be noticed that our proposed model outputs the CSD in between the other two models. Here, traditional braking model overestimates the crash risk and gives a too large CSD, whereas the time-headway model underestimates the risk due to focusing on traffic efficiency enhancement and outputs a too small CSD. Therefore, our results are enough for safety guarantee and could increase traffic throughput to some extent. Besides, the CSD by our model decreases with the falling of acceleration of the back car, that is, a_B. It is reasonable because collision risk will continuously reduce when the acceleration drops in this scene.

The case that the leading car has an initial speed less than that of the rear car is plotted in Figure 12 with $a_B = 3\,m/s^2$. Similar to the results from Figure 10, the CSD is still zero before the initial speed of the back car is around 22 km/h. This is logical and the reason is the same as Figure 10. However, although there is no chance to result in collision, the traditional braking model and time-headway model both show a positive CSD which may decrease the traffic throughput before around 22 km/h. With the growth of the initial speed, the CSD, for the three model all increase considering the underlying risk.

In Figure 13, we also show the CSD performance comparison among the three models in multi-lane case. Here, we assume the speeds of the leading and rear car are both constant during simulation. Besides, due to the small steering angle as θ shown in Figure 8, we can overlook the vehicle's velocity vertical component and suppose its speed is still the same during lane changing [21]. The speed of the leading car is $v_A = 50$ km/h and the initial relative velocity of the two cars can be calculated by $v_r = v^0_{xc_0} - v^0_{xc_3}$. We also

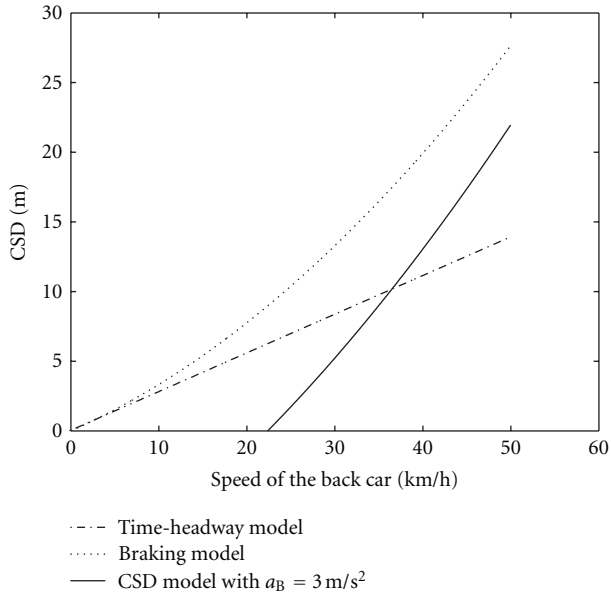


Figure 12: A is decelerating and its initial speed is greater than B's.

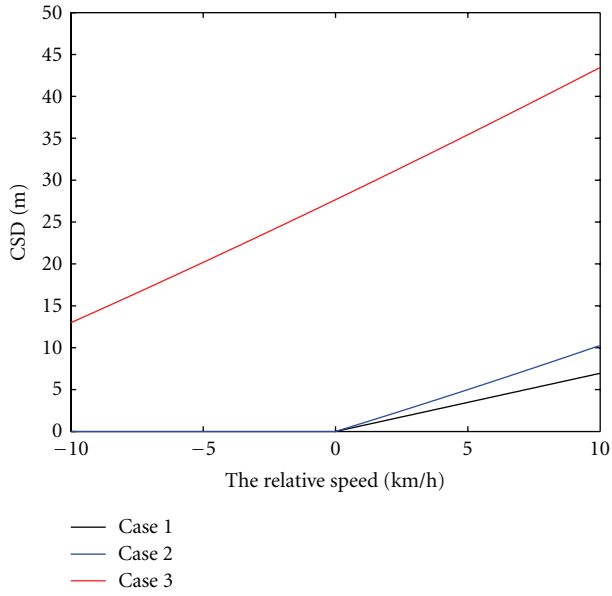


Figure 13: Multilane CSD.

set $t_0 + t_c = 2.5\,s$ to denote the needed duration for lane changing. To show the performance of our proposed CSD model, we introduce three scenarios and mark their results in different colors as shown in Figure 13. Here, the black curve indicates the needed CSD before t_c during lane-changing process. The blue one denotes the CSD when the vehicle has successfully switched to the neighboring lane and received the braking warning notification from the leading car. The green one also inspects the CSD after lane changing but assumes that the leading car has collided with some obstacles ahead or stopped immediately due to emergency. As shown in Figure 13, the final CSD will all increase with the growth of v_r for the three cases. The blue and green curves give a bigger CSD than the black due to their consideration of movement status changing, such as braking or immediate stop, of two cars. Further, the blue and black cases both show a length of zero when v_r is less than 0. It is reasonable because there is no possibility for the back car to collide with the leading one when its speed is smaller than the front vehicle.

9. Conclusion

In this paper, we define a Critical Safe Distance to balance the requirement between vehicle driving safety and traffic efficiency in VSNs. To accurate express this CSD, we elaborately analyze the movement features when braking and discuss its possible value in different mobility scenarios. Numerical results show that our proposed model could intelligently set the CSD according to the relative movement status of the leading and following vehicles in single and multi-lane cases. Further, our algorithm can better alleviate the performance degradation on safety and traffic throughput due to bigger or smaller safe distance settings by the time-headway and the introduced traditional braking models.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61201133; 61172055; 60832005; U0835004; 61072067, 50978030, and 51278058), the Postdoctoral Science Foundation of China (20100481323), the Program for New Century Excellent Talents (NCET-11-0691), the "111 Project" of China (B08038), and the Foundation of Guangxi Key Lab of Wireless Wideband Communication and Signal Processing (11105).

References

[1] M. Brackstone and M. McDonald, "Car-following: a historical review," *Transportation Research Part F*, vol. 2, no. 4, pp. 181–196, 1999.

[2] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B*, vol. 15, no. 2, pp. 105–111, 1981.

[3] M. Werling, N. Kaempchen, D. Wollherr, and M. Buss, "Lane-based safety assessment of road scenes using Inevitable Collision States," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '12)*, pp. 31–36, Alcala de Henares, Spain, 2012.

[4] E. Adell, A. Várhelyi, and M. D. Fontana, "The effects of a driver assistance system for safe speed and safe distance—a real-life field study," *Transportation Research Part C*, vol. 19, no. 1, pp. 145–155, 2011.

[5] S. Kannan, A. Thangavelu, and R. B. Kalivaradhan, "An intelligent Driver Assistance System (I-DAS) for vehicle safety modelling using ontology approach," *International Journal of UbiComp*, vol. 1, no. 3, pp. 15–29, 2010.

[6] F. Biral, M. da Lio, A. Saroldi, F. Tango, and E. Bertolazzi, "Supporting drivers in keeping safe speed and safe distance: the SASPENCE subproject within the European framework programme 6 integrating project PReVENT," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 525–538, 2010.

[7] Y. Wu, J. Xie, L. Du, and Z. Hou, "Analysis on traffic safety distance of considering the deceleration of the current vehicle," in *Proceedings of the 2nd International Conference on Intelligent Computing Technology and Automation (ICICTA '09)*, pp. 491–494, Changsha, China, October 2009.

[8] Q. Luo, L. Xun, Z. Cao, and Y. Huang, "Simulation analysis and study on car-following safety distance model based on braking process of leading vehicle," in *Proceedings of the of 9th World Congress onIntelligent Control and Automation (WCICA '11)*, pp. 740–743, Taipei, China, 2011.

[9] T. J. Ayres, L. Li, D. Schleuning, and D. Young, "Preferred time-headway of highway drivers," in *Proceedings of the IEEE Intelligent Transportation Systems Proceedings*, pp. 826–829, Oakland, Calif, USA, August 2001.

[10] A. Touran, M. A. Brackstone, and M. McDonald, "A collision model for safety evaluation of autonomous intelligent cruise control," *Accident Analysis and Prevention*, vol. 31, no. 5, pp. 567–578, 1999.

[11] T. Hiraoka, M. Tanaka, H. Kumamoto, T. Izumi, and K. Hatanaka, "Collision risk evaluation index based on deceleration for collision avoidance (first report)," *Review of Automotive Engineering*, vol. 30, pp. 429–437, 2009.

[12] G. Yan, W. Yang, M. C. Weigle, S. Olariu, and D. Rawat, "Cooperative collision warning through mobility and probability prediction," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '10)*, pp. 1172–1177, San Diego, Calif, USA, June 2010.

[13] C. M. Huang and S. Y. Lin, "A cooperative collision early-warning system for vehicles in the curve environment," in *Proceedings of the 15th IEEE Symposium on Computers and Communications (ISCC '10)*, pp. 14–19, Riccione, Italy, June 2010.

[14] C. Garcia-Costa, E. Egea-Lopez, J. B. Tomas-Gabarron, J. Garcia- Haro, and Z. J. Haas, "A stochastic model for chain collisions of vehicles equipped with vehicular communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 503–518, 2012.

[15] G. Johansson and K. Rumar, "Drivers' brake reaction times," *Human Factors*, vol. 13, no. 1, pp. 23–27, 1971.

[16] T. Hiraoka, M. Tanaka, S. Takeuchi, H. Kumamoto, T. Izumi, and K. Hatanaka, "Collision risk evaluation index based on deceleration for collision avoidance (second report)—forward obstacle collision warning system based on deceleration for collision avoidance," *Review of Automotive Engineering*, vol. 30, no. 4, pp. 439–447, 2009.

[17] S. Venkatraman and J. Caffery Jr., "Hybrid TOA/AOA techniques for mobile location in non-line-of-sight environments," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 1, pp. 274–278, Atlanta, Ga, USA, 2004.

[18] P. Seiler, B. Song, and J. K. Hedrick, "Development of a collision avoidance system," in *Proceedings of the SAE International Congress & Exposition*, pp. 97–103, February 1998.

[19] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Analytical model for connectivity in vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3341–3356, 2008.

[20] S. E. Shladover, C. A. Desoer, J. K. Hedrick et al., "Automatic vehicle control developments in the PATH program," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.

[21] H. Jula, E. B. Kosmatopoulos, and P. A. Ioannou, "Collision avoidance analysis for lane changing and merging," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2295–2308, 2000.

[22] Y. Tang, Z. Jiang, Y. Bai, and Y. Fang, "Model of vehicle safety distance for collision avoidance and simulation study," *Journal of Liaoning University of Technology*, vol. 28, no. 5, pp. 47–49, 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 567687, 15 pages
doi:10.1155/2012/567687

Research Article

Interconnection between IP Networks and Wireless Sensor Networks

Brandon Keith Maharrey, Alvin S. Lim, and Song Gao

Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA

Correspondence should be addressed to Alvin S. Lim, lim@eng.auburn.edu

Received 15 June 2012; Revised 16 October 2012; Accepted 17 October 2012

Academic Editor: Chin-Feng Lai

With applications ranging from environmental and health monitoring to military surveillance and inventory tracking, wireless sensor networks (WSNs) are changing the way we collect and use data and will be a major part of our technological future. The decreased manufacturing cost of these small devices has made it reasonable to deploy many sensor nodes—tens to thousands and more—over large and small indoor and outdoor areas for sensing tasks. With this increase in density data-gathering problems come. It would be useful if an IP-based host could collect information from multiple remote data-centric networks via transparent communication among sensor nodes and IP-based hosts, using a common application programming interface (API). Two APIs are presented for efficiently producing data in WSN and retrieving the data from an IP network. An associated WSN middleware layer called dynamic service is used to effectively interconnect these two APIs. These three components work together in harmony to enable IP-based hosts to gather sensed data from one or more remote WSNs through application-layer gateways which provide seamless interconnection between different remote WSNs and the IP network.

1. Introduction

A wireless sensor network (WSN) mainly consists of many independent, low-power, low-cost devices capable of sensing, processing, and wireless communication [1]. Their main purpose is to collect and disseminate environmental data and possibly perform some calculations [2]. There has been a push, especially in industry, in recent years to make real-time data collected from WSNs more readily available to consumers of this information. However, there are no convenient tools or specific frameworks in place to allow instant access to this sensed information in a programming environment. Thus, one of the main problems with deploying WSNs is gathering the data they produce and using it in flexible ways. This paper provides a solution to this problem by enabling seamless interconnection between IP networks and wireless sensor networks.

This paper proposes a middleware layer, called dynamic service, that provides transparent communication between IP-based hosts and sensor nodes via gateway nodes, which are employed as access points for the purpose of interoperable information retrieval from WSNs. It also eases the implementation of sensor network applications by providing a standard common interface to the data-centric WSN networking protocol and IP-based applications alike. Dynamic service (DS) is placed between application layer and direct diffusion (DD) on sensor nodes. By exposing neat and clean APIs, it allows the sensor node application programmer to ignore the details of the DD networking protocol, but only cares about data they are producing or processing. In addition to easing programming, DS allows tasking of nodes. Applications register the name of data they can provide with the DS service layer. An application sleeps until it receives an interest for this registered named data. This conserves energy in information or data production.

Other parts in the system include an implemented gateway application that bridges between IP networks and sensor networks, and a set of APIs for external agents (EAs), which are IP-based hosts that are not a part of the sensor network. EA API enables IP network applications to submit requests to and receive data from sensor networks. By using DS API and EA API, programmers can build efficient applications running across wireless sensor networks and IP networks without concerning complicated communication

details, such as translating between IP packets and DD packets, and managing DD protocol details.

The following sections of this paper will be organized as follows. Section 2 describes the problem and challenges in IP enabled WSN and outlays the motivation of this research. Section 3 discusses relevant approaches to gathering data from WSNs connected to IP network. In Section 4, an overall summary of the work is given, including architecture of the system, proposed dynamic service, and a discussion of directed diffusion. In Section 5, design details of dynamic services, external agent role and gateway role are given. Section 6.1 and Section 6.2 describe the experimental environment and platform used in this paper, and the applications implemented on the system, respectively. In Section 7, experimental results are given, in terms of application level result, application lines of code metric, and dynamic service performance. A more detailed presentation of this research can be found in [3].

2. Research Motivations

The area of sensor networking applications is exploding rapidly. In the recent past, many new sensor networking applications have surfaced in the literature, and most notably among them are wildlife habitat monitoring [4], forest fire detection [5], alarm systems [6], and monitoring of volcanic eruptions [7]. These scenarios involve many unique issues and challenges in addition to the problem of gathering this sensed data in real time for analysis, computation, or storage. IP-based application programmers are faced with the difficult problem of interconnecting sensor networks with IP-based hosts. Therefore, a main motivation of this research is to ease the data-gathering problems that IP-based programmers face when gathering data from one or more remote WSNs.

There are subtle problems when interconnecting IP networks and WSNs. First, there are major differences in the data retrieval paradigm used in WSNs and IP networks, where WSNs use data-centric paradigm for retrieving sensor data, whereas IP networks use host-centric paradigm for querying individual sensor node. WSNs use data-centric retrieval methods since they have been shown to be more energy-efficient and scalable than address-centric retrieval methods used in IP networks which are based on proactive routing algorithms that have been shown to be less energy-efficient. Second, for an IP network application programmer, it is relatively difficult to develop programs for querying remote sensor nodes for their data due to the differences in data retrieval paradigms. Third, a mechanism must be designed to enable IP-based hosts to actually retrieve data from remote WSNs. Named data coming from data-centric WSN needs to be properly translated to an address-centric IP network. Fourth, since the data-centric network discussed in this paper is based on the publish-subscribe paradigm, IP-based hosts must take this fact into account. Subscriptions must be sent from IP-based hosts to WSNs properly for data production to begin.

Solving the above problems will make it possible for us to design and implement IP-enabled WSNs whereby sensor data may be retrieved, and tasks in WSNs can be initiated from IP networks. The main advantage of using this approach is that WSNs can still use more energy-efficient and scalable data-centric retrieval methods for accessing sensor data.

3. Related Work

Many researchers have done previous work on interconnecting wireless sensor and IP networks and gathering data from WSNs. The majority of the techniques, like the one presented in this paper, treat WSNs as a separate entity from the Internet [8]. The techniques are divided into two main approaches: a gateway-based approach and an approach in which all sensor nodes are TCP/IP-enabled—that is, capable of direct, end-to-end communication with IP-based hosts.

3.1. Gateway or Proxy-Based Approaches. The most common approach to connecting a WSN with an IP network is through a gateway or proxy node. In this approach, the gateway node acts as a relay to translate and forward packets from one network to the other [9–15]. The authors of [12] describe two gateway-based approaches: using the gateway as a relay or as a front-end. When the gateway acts as a relay to the WSN, it simply relays any information from the WSN to any registered IP-based host that wants that information. This approach is taken in this paper. When the gateway node acts as a front-end to the WSN, it actively collects and stores data from the WSN in some kind of database that users can query with SQL-like query languages.

One of challenges in gateway-based approaches is that the gateway node can be a bottleneck to the flow of network traffic, especially if a surge of data needs to be transmitted from the gateway node to an IP-based host [14]. An advantage of gateway-based approaches is that the two communication networks are totally decoupled, allowing for specialized and more efficient protocols, such as directed diffusion, to be implemented in the WSN. The gateway node can also act as a mediator for WSN data transmission by implementing security features such as user and data authentication [12].

3.2. IP-Enabled Approaches. Besides gateway-based approaches to interconnecting wireless sensor and IP networks, there also exists IP-enabled WSNs. One of these approaches assumes a full TCP/IP stack on each sensor node. In this approach, the WSN is directly connected to the IP network to enable direct communication between WSN sensor nodes and IP-based hosts [16–20].

The main advantage of using TCP/IP in this way is that there is no need for protocol conversion or gateways. However, the overhead for the full networking stack on an energy-constrained sensing device may be prohibitive, especially when the end-to-end retransmissions incurred by the TCP protocol cause even more undue retransmissions at intermediate nodes. It has been shown that the majority of energy in a WSN is used for wireless communication [21, 22]. Therefore, if one considers the protocol overhead for TCP/IP networks in the context of WSNs, it can be seen

that this overhead is prohibitive. A further disadvantage of this approach is that just because each sensor node is addressable does not necessarily ease the task of gathering data the sensors can produce. In this case, IP hosts must be supplied with each individual WSN node IP address it wishes to query for data. There could potentially be many WSN nodes in possibly multiple remote WSNs so this may be an inefficient method of information retrieval, especially considering the wasted energy with TCP retransmission attempts. Moreover, this solution does not lend itself well to specialized and energy-efficient WSN protocols inside the WSN. This approach also uses IP routing algorithms which are proactive and less energy-efficient than reactive routing algorithms, such as directed diffusion, used in WSNs.

3.3. Overlay Approaches. In overlay approaches, gateway nodes are used to interconnect WSNs with IP networks and assign virtual identification information to either IP-based hosts, sensor nodes, or both [23–25]. According to [8], overlay approaches come in two basic forms: sensor network overlay IP network and IP network overlay sensor network. These two approaches employ application layer gateways through which the WSN is identified and information is passed.

In the sensor network overlay IP network approach, IP-based hosts are required to register with the WSN application-layer gateway node and be assigned a virtual sensor node ID by the gateway node. Once a packet from a sensor node destined for a virtual sensor node ID reaches the gateway node, the gateway node encapsulates the whole packet into a TCP or UDP and IP packet, while the IP-based host communicates with sensor nodes by supplying the sensor node ID to the gateway node.

In the IP network overlay sensor network, sensor nodes are required to register with the WSN gateway node and are assigned a virtual IP address. Individual sensors themselves do not actually possess an IP address in the WSN. Sensor nodes are instead assigned a WSN-wide unique standard 16-bit TCP/UDP port number by the gateway node. IP-based hosts communicate with individual WSN nodes by supplying the IP address of the gateway node and port of the sensor node with which it wishes to communicate.

This scheme has several issues. Firstly, if the standard 16-bit unassigned TCP/IP port numbers are used to identify individual sensor nodes, only around 16,000 nodes can be uniquely addressed. Secondly, it suffers from the protocol overhead attributable to TCP/IP. Thirdly, it may suffer from large routing tables due to the fact that the gateway node must keep track of two different mappings. Aside from these issues, neither of these two overlay approaches truly simplifies the task of gathering data from WSNs.

3.4. 6LoWPAN and IEEE 802.15.4 Standards. The IEEE 802.15.4 standard [26] defines the physical layer and media access control for the wireless personal area network. The 6LoWPAN standard [27] defines encapsulation and header compression mechanisms that allow seamless IP integration over IEEE 802.15.4 [28–30]. [31] presents uIPv6, an IPv6 stack for memory-constrained devices that can run over IEEE 802.15.4/6LoWPAN. In 6LoWPAN, individual sensor nodes are addressable with standard IPv6 IP addresses without the overhead of sending full IP addresses when routing messages inside the WSN. This is because a gateway node connected to an IP network maps full IP addresses into 16-bit node IDs for more efficient bandwidth usage along wireless hops.

This standard, however, is only in its preliminary stages and thus will probably undergo more changes before the final standard is widely available. Like other IP-enabled approaches to interconnecting WSNs, this approach also requires that IP-based hosts know the specific IP addresses of sensor nodes with whom they wish to gather data. Further, the IEEE 802.15.4 standard defines a maximum bandwidth that may be unsuitable for WSN applications requiring larger bandwidths. Moreover, these standards together or separately do not necessarily ease the task of gathering data from sensor nodes in one or more WSNs, although it does reduce the amount of wasted energy with respect to transmitting end-point identification information for each packet along every wireless hop.

4. System Overview

4.1. Architecture. Our system uses an approach for providing seamless interconnection and transparent interoperability between different sensor data dissemination paradigms of IP and WSNs via gateways which also decouple the IP networks and WSNs, allowing for specialized and more efficient protocols to be implemented in WSNs [12]. Figure 1 shows the network architecture. There are three node roles in the network.

(i) *External Agents*: An IP-based host that is not a part of the WSN is termed an *external agent* (EA). An EA node is a full-fledged computer with full TCP/IP networking stack and can access the Internet or IP network.

(ii) *Sensor Nodes*: Sensor nodes provide data that is requested by EA. Given the attributes of robustness, scalability, and energy-efficiency in multi-hop communication [32], *directed diffusion* (DD) is used as routing protocol within sensor nodes. A middleware layer, *dynamic service* (DS), is designed on top of DD. DS will be introduced in Section 4.2.2.

(iii) *Gateway Node*: As shown in Figure 2, gateway node is on the boundary of WSN and directs incoming and outgoing traffic of WSN. It has both IP network stack and DD sensor network capability. Any interest or subscription from an IP-based host is processed through the gateway node, translated into WSN interest, and disseminated using DD protocol. Data returned by sensor nodes is also processed and forwarded by gateway node to IP-based hosts.

4.2. Underlying Concepts

4.2.1. Direct Diffusion. Directed diffusion (DD) is a data centric ad hoc networking protocol capable of robust,

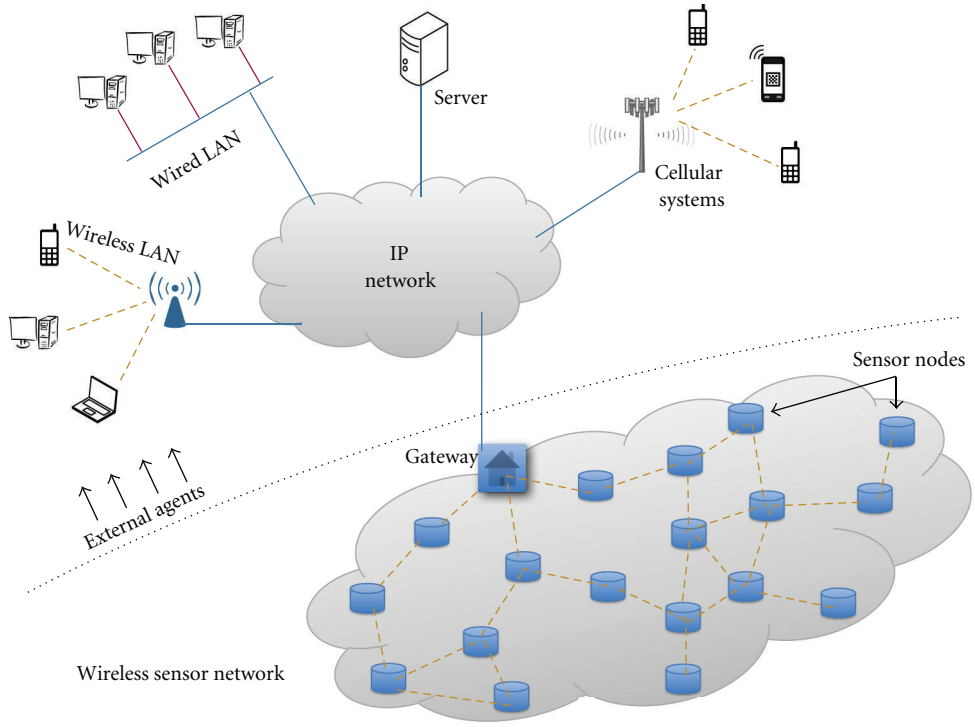


Figure 1: Network Architecture.

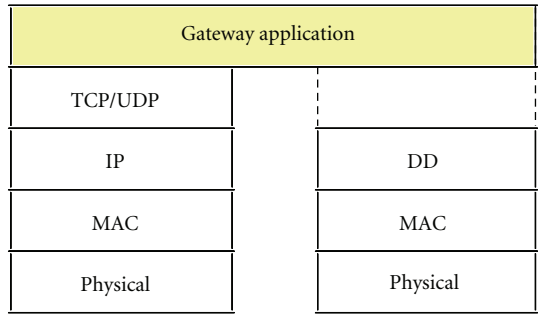
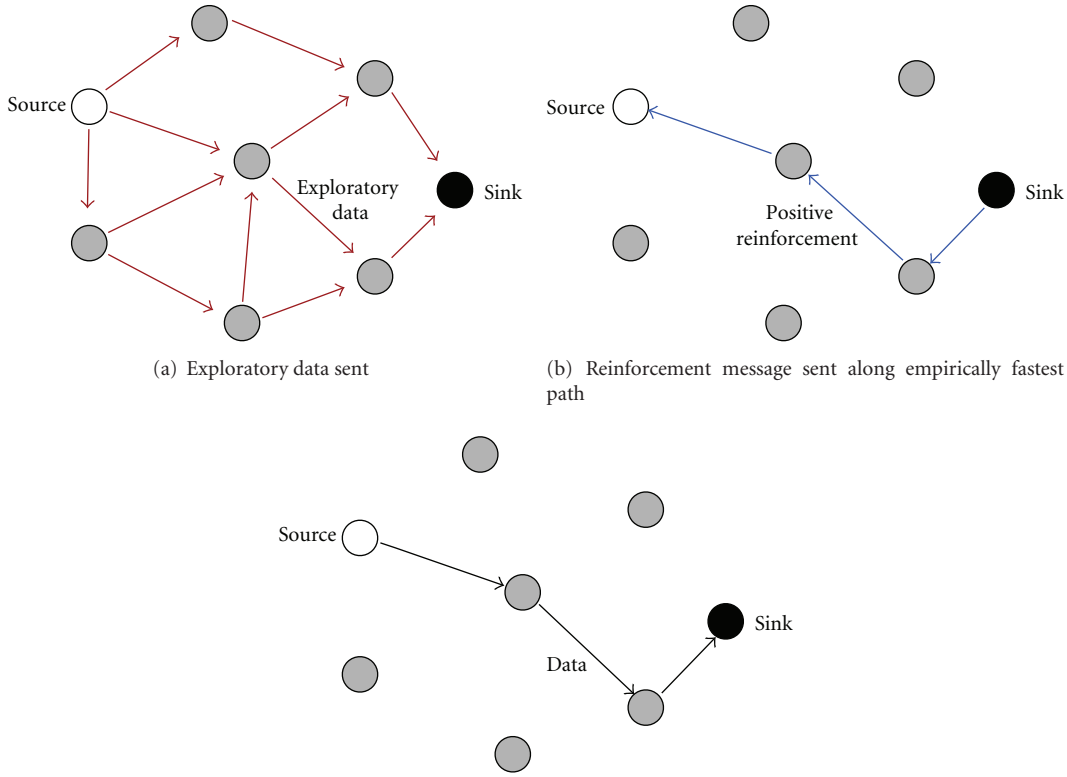


Figure 2: Gateway Networking Stacks.

scalable, and energy-efficient multi-hop communication [33, 34].

In DD, when a sensor node has the ability to produce named data, it specifies the name of this named data to the core DD routing algorithm. When the DD core receives an interest for named data that has previously been registered, a callback function is invoked to handle the production of data corresponding to this named interest. On the other hand, when the DD core inside the sensor node receives a named interest for named data that has not been previously registered, the DD core will either forward the interest message to its neighbors or drop the interest message altogether. When a node is interested in some data, it sends out an interest for that data. This node is referred to as *sink*. Interests are diffused throughout the network, and *gradients* are set up along the reverse path of travel of the interests.

Figure 3 shows what happens after the interest is diffused throughout the network. Once a sensor node receives an interest for data it can produce, this node, known as *source*, begins to produce that data. The data is forwarded hop-by-hop along multiple gradients and back toward the *sink*, establishing an empirically fastest path from the sink node to the source of the data, by way of exploratory data as shown in Figure 3(a). This exploratory data diffuses back across the network along the *gradients*. The empirically fastest path is chosen for reinforcement by the *sink* for fast data



(a) Exploratory data sent

(b) Reinforcement message sent along empirically fastest path

(c) Data delivered along reinforced path

Figure 3: Directed diffusion data diagram.

reception of future data packets, and the *sink* transmits a positive reinforcement message to the neighbor from which exploratory data was first received, as shown in Figure 3(b). The positive reinforcement message is sent along the gradient path until it reaches the source node. From this time onward, data is sent along the positively reinforced path, shown in Figure 3(c).

DD is chosen as the data-centric networking protocol in the WSN in this paper for several reasons.

(i) DD makes routing of the sensed data within the WSN more efficient compared to using TCP/IP on the sensor nodes. The responsibility of passing interests from the gateway node to individual sensor nodes, and subsequent passing of data from sensor nodes back to the gateway node, is given to DD. This makes the task of routing inside the network very simple, convenient, and efficient.

(ii) DD keeps routing tables inside the gateway node rather small. At the gateway node, it is only necessary to keep track of unique interests (subscriptions) and any IP address and port number of IP-based hosts interested in data provided by the WSN.

(iii) Because individual sensor nodes are not named in a DD network, there is no overhead in keeping track of or assigning sensor nodes unique IDs nor is there any sensor node energy wasted in transmitting endpoint identification information with each transmitted packet.

4.2.2. Dynamic Service. Dynamic service (DS) is a middleware layer built on top of DD protocol on sensor nodes, as shown in Figure 4. It provides the services necessary to facilitate IP-based information retrieval from sensor node applications built with this type of architecture. With DS inserted between DD and application layer, some WSN specific concepts are hidden from upper layer and communication details are made transparent, thus providing a more flexible and efficient way to develop WSN applications.

DS enables nodes tasking, specifically for data production. Applications register the name of data they can provide with the DS service layer. An application sleeps during normal status. When an interest for the registered named data is received by DS, DS awakens the node application to begin producing the corresponding data and send out through DS.

5. Design Details

Our design for seamless interconnection between IP networks and WSNs involves three main components: dynamic service, external agents, and gateways.

5.1. Dynamic Service. DS moves the complexity out of the individual sensor node applications and into the DS service

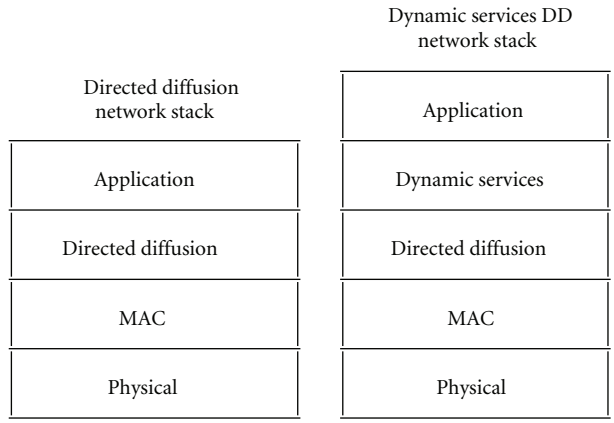


Figure 4: Network stack: pure DD versus DS enabled DD.

layer. DD and DS service layer must be running on the sensor node before applications are able to use the services provided by DS. The DS API functions available to the sensor node application are called within the sensor node application and information is passed from the sensor node application, through a message passing interface (MPI), to the DS service layer.

5.1.1. DS Architecture. As shown in Figure 4, DS is implemented on top of DD. It acts as a regular DD application; therefore, it contains a tasking thread and a main thread. However, since DS is a intermediate layer between DD and actual sensor node applications, the threads function differently from general DD applications.

(i) *Tasking Thread*: Instead of incrementing or decrementing value of shared variable, the tasking thread tasks sensor node applications through their respective message queues, when a tasking message is received from DD. In addition, it also passes data to the appropriate sensor node applications when a data message is received from DD.

(ii) *Main Thread*: Instead of polling shared variable for tasking, the main thread waits at message queue for requests or data from sensor node applications.

Figure 5 shows a diagram of the types of messages in which the DS service layer and the DS API communicate with each other. Although the messages are passed through MPI, the sensor nodes are unaware of communications details. They are encapsulated within DS API.

The *register* and *subscription* messages sent to DS through the DS API through DS's own message queue enter information into internal tables in which DS maintains to keep track of sensor node applications. These tables retain information regarding the data type each sensor node application produces and the data types to which the sensor node applications are subscribed as well as the message queue information of each sensor node application through which DS sends response or data to the sensor application.

5.1.2. Data Producers. There are two different classes of sensor node applications. One class of sensor node application, when tasked, simply produces the requested named data. These sensor node applications are called *simple producers*. The other class of sensor node applications, however, depends upon other named data types in the local WSN in order to produce its named data. These sensor node applications are called *complex producers*.

Simple Producer. When viewed from the network level, the simple producer messaging process looks like that shown in Figure 6. Figure 7 shows how messages are communicated between the DS API and the DS service layer for simple producers. The communication between DS API and DS is described in 4 phases as follows.

(1) *Registration*: Sensor applications register their intent to publish data. This registration message is passed to the DS service layer through the DS API, reaches the DD core, and a registration response message is eventually passed back to the sensor application through the DS service layer and through the DS API. If this registration response indicates a successful registration, the sensor application begins to await tasking.

(2) *Data Interest*: When an interest message flooding the DD network arrives at the sensor node at the DD core, the DD core realizes that a registration for this data type has been received in the past and invokes the tasking thread in DS. The tasking thread in DS then tasks the sensor application through the DS API.

(3) *Production Data*: The sensor application begins producing the named data it was programmed to produce. Once the sensor application has data to send out onto the network, it publishes this data through the DS API, through the DS service layer. The DD core, at last, actually sends the data out onto the network. A publish success message is propagated up to the sensor application through the DS service layer and through the DS API.

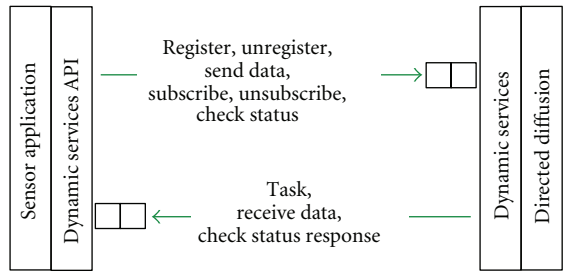
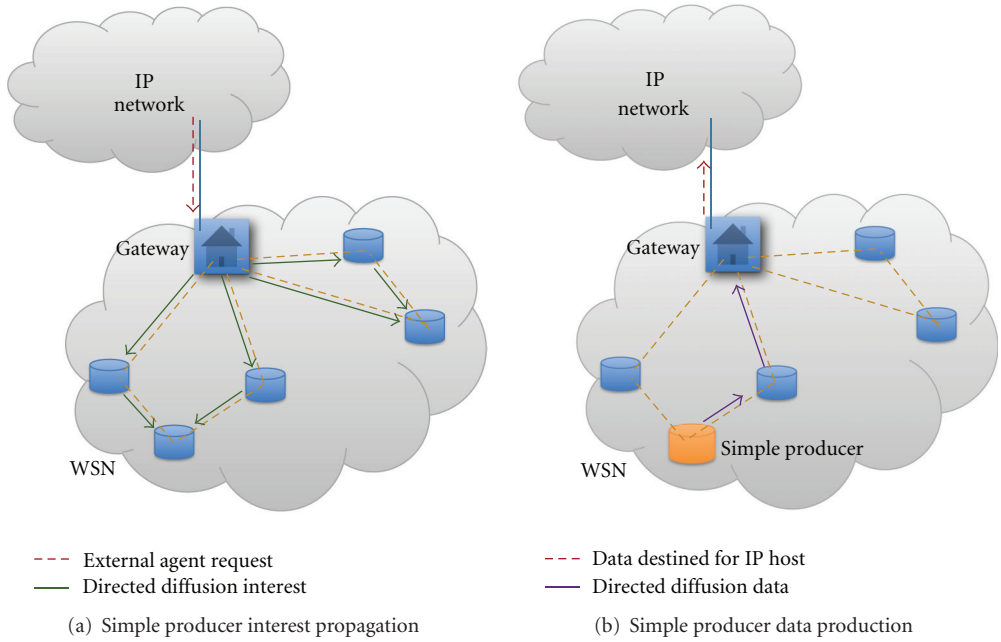


Figure 5: Types of messages that DS service layer and DS API communicate with each other.



(a) Simple producer interest propagation

(b) Simple producer data production

Figure 6: How data production takes place when a sensor node application known as a simple producer is tasked.

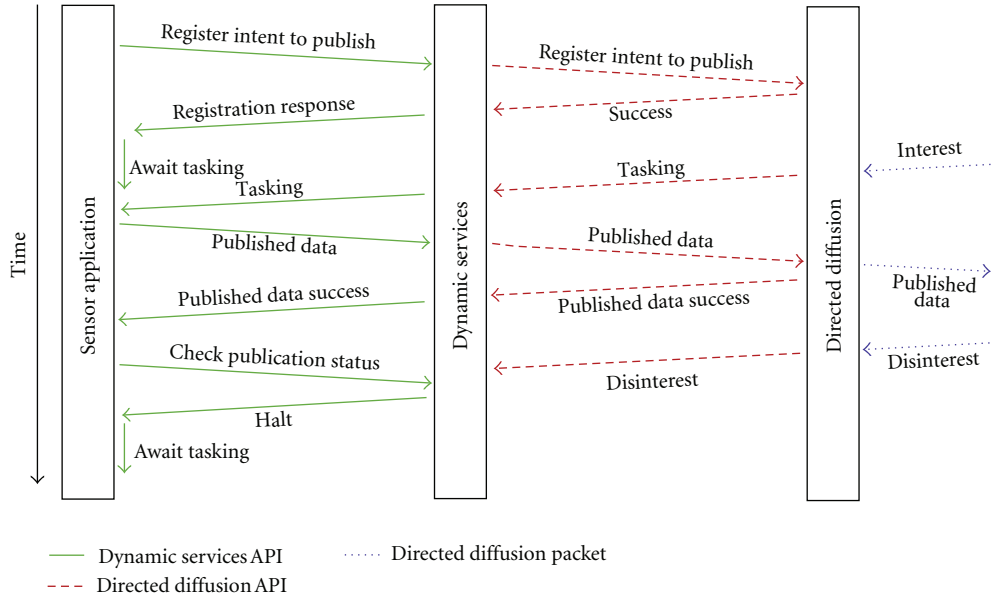


Figure 7: How messages are exchanged for simple producers.

(4) *Checking Status*: Ever so often, the sensor application should check whether it should continue producing the data it was once tasked to produce. The sensor application submits a message to the DS service layer requesting its status to continue producing data. If a disinterest message was received from the DD network before this request is made, the DS service layer replies to the check publication status message with a message indicating that data production should cease. The sensor application should then halt data production and continue to await tasking. The whole simple producer process repeats thus, henceforth.

Complex Producer. Unlike simple producers, when complex producers are tasked, they require named data produced by other sensor node applications within the local WSN in order to produce their named data.

Sensor applications first register with DS service layer their intent to publish data. When an interest message flooding the DD network arrives at the sensor node at DD core, DS layer tasking thread is invoked. DS then tasks the sensor application. Unlike a simple producer, the complex producer subscribes to other data types as well. The subscription is done through the DS API. When the required named data from other producer is received and passed up through DD core and DS service layer, the complex producer begins to produce its named data. Finally, a disinterest may be received to untask the complex producer.

5.2. External Agents. An external agent (EA) is any IP-based host not directly connected to the WSN. These devices are decoupled from the WSN protocol and are not a part of the WSN.

One of the purposes of the system is that sensor nodes do not expend energy on any sensing or transmission task until an EA submits an interest to the WSN. Therefore, EAs actually drive the activities of the WSN sensor nodes by sending requests to the gateway node. This is achieved by registration/subscription process described in Section 5.1. Sensor node applications register a data type it can produce through DS API and keep sleeping until they receive tasking information from some EA.

A simple API enables an EA to retrieve information from a DS enabled DD WSN. There are three functions provided by the API.

(1) *Subscribe*. Through this function, an EA initiates a subscription request to one or more WSNs by providing the address of gateway nodes as well as the types of data it is interested in. The EA API then interprets the subscription request and passes them to proper gateway nodes in WSNs, to be translated into WSN DD interests and disseminated within the WSNs. The function then returns a socket descriptor to EA application reference in the future.

(2) *Receive*. After *subscribe* is called, an EA can retrieve data from WSN through the *receive* function with the socket descriptor returned by *subscribe* function. It waits for the network until the requested data is available, interprets the received data, and stores them in an indicated buffer. When the function returns, the requested data is available in the buffer for EA applications to use.

(3) *Unsubscribe*. After the EA is no longer interested in a particular type of data from one or more WSNs, it can call this function with the socket descriptor and the data disinterested. The EA API passes the unsubscription to gateway nodes. Once no EAs are interested in a particular type of data, the gateway node (see Section 5.3) sends a disinterest into the WSN so that the named data is no longer being produced.

5.3. Gateway Role. The gateway role is the entity which physically enables communication between IP network and WSN. When the gateway node powers on, it waits for requests from EA in its main thread. When it receives a request from an EA, it translates this request into an interest packet that the DD networking protocol can understand. As this interest is diffused throughout the network, gradients are set up along the reverse path of this interest propagation. Named data later produced by a sensor node will traverse the network along these gradients, and the gateway will eventually positively reinforce gradients with empirically shortest delay.

In this way, named data is drawn towards the gateway node for which the gateway node previously sent out interest requests on behalf of EA. If the gateway node receives data for which there is no EA subscribed, the gateway node simply discards the packet. This could happen if the gateway node receives data from a sensor node that has not yet received the command to stop producing data.

The gateway node is a regular directed diffusion application which sits on the boundary of the wireless sensor and IP networks. The gateway nodes main thread, upon starting up, prepares an incoming socket on which to receive *requests* from EAs. The *request* structure is totally hidden from EAs therefore, the only thing that EAs need to know is the IP address and port number of the gateway node, the name of the named interest, the structure of the expected data, and how to use the EA API.

When a *request* is received from the an EA in the gateway nodes main thread, an entry is added to a local map structure. This map structure contains the IP address and port number on which the EA is awaiting named data. When named data arrives at the gateway node from the WSN, the DD core triggers the gateway nodes tasking thread. However, the tasking threads role in the gateway node has been redefined. Rather than using the tasking thread for tasking, the gateway nodes tasking thread looks into the IP address-port number map structure to determine to which EA(s) to forward this named data. If any EAs are found, the named data is forwarded to these EAs accordingly. If no EAs are found which has previously subscribed to this named data, the tasking thread completes dropping the received DD packet.

6. Implementation

6.1. System Implementation. The system is implemented on real devices including WSN, gateway role, and external

agent. Figure 10 shows a setup of the system implemented. Section 6.2 describes how applications are designed and implemented.

PC104 testbed, shown in Figure 8, is used as WSN nodes. Each PC104 has a 533 MHz VIA Mark processor, 256 MB RAM, and is equipped with a 1 GB flash card for external storage.

An Orinoco Gold wireless PCMCIA card is used on each PC104 for wireless communication. It works on 2.4 GHz and supports four speeds: 11 Mb/s, 5.5 Mb/s, 2 Mb/s, and 1 MB/s. An omnidirectional external antenna, as shown in Figure 9, is used to boost the wireless signal with +5 dBI gain.

A USB microphone is used on each PC104 for data production. It is a mono, high sensitivity, omnidirectional microphone with headphone amplifier and detects frequencies from 20 Hz to 20 KHz.

The PC104 sensor nodes use a Linux-based operating system Slax v6.0.7 to run all sensor node application software. Slax is chosen for its small size. Graphic user interface is removed to further save space on the compact flash card. All softwares used in the testbed, including the operating system, are around 75 MB.

A complete list of equipments used in the system is shown in Table 1.

6.2. Application Implementation. Previous theoretical research has developed algorithms for determining the location (at a particular point in time), speed, and direction of movement of a target which emits acoustic sound waves traveling through an array of acoustic sensors. The research was first mathematically formalized for flying airplanes in the more general three-dimensional scenario by Dommermuth in [35] and modified to the two-dimensional scenario by Yang et al. in [36–38] for use in tracking ground-based targets through an acoustic WSN.

A set of applications is implemented on top of the system, to determine the location, speed, and direction of movement of a target, meanwhile capturing video of the moving target with a pan-tilt-zoom camera. The target emits acoustic sound waves, and the sensor nodes monitor the sound waves to calculate location of the target. This is done through collaboration between a *simple producer* and a *complex producer*. An EA application can then utilize the data provided by WSN to drive the camera to track the moving target and capture video. Figure 10 shows the setup of the entire network. The rest of this section gives a detailed description of implemented applications.

CPA_INFO Simple Producer. The *CPA_INFO Simple Producer* senses the sound wave emitted by the target and produces closest-point-of-approach (CPA) information. Figure 11 shows a graph of the sound intensity as a target approaches the sensor node, moves through the CPA relative to the sensor node, and begins moving away from the node. The vertical line represents the time at which the target is at its CPA, and the timestamp of CPA is produced by the sensor application and forwarded to the cluster head, which runs *TARG_INFO complex producer* (see Section 6.2).

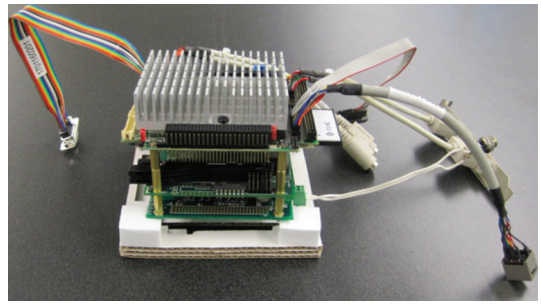


Figure 8: PC104 sensor node without casting.

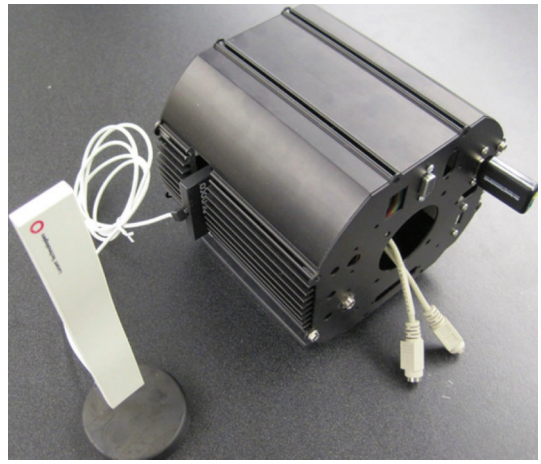


Figure 9: PC104 sensor node with casing and antenna.

Table 1: Equipments used in the system.

Device	Name/model
CPU module	PC104 with Aaeon PFM-550S 553 MHz
Power supply	Aaeon PFM-P13DW2
Wireless card	Orinoco Gold PCMCIA LAN
Antenna	2.4–2.5 GHz omnidirectional +5 dBi with 60″ cable
Storage	Type 1 compact flash 1 GB
RAM	Transcend 144 pin SDRAM 256 MB 133 MHz[a]
Microphone	Mono/omnidirectional SP-USB-MIC-1[b]
Battery	Power Sonic +12 V/5.0 amp hr
Camera	Sony EVI-D30 pan-tilt-zoom camera w/RCA-to-USB

[a] http://www.transcendusa.com/Support/DLCenter/Datasheet/TS32MSS64V6G_6755.pdf.
[b] http://www.soundprofessionals.com/cgi-bin/gold/item/SP-USB-MIC-1.

TARG_INFO Complex Producer. *TARG_INFO* is a *complex producer* that actually calculates the target's location, speed, and direction, using data obtained from other sensor nodes running *CPA_INFO simple producer*. It considers the GPS (Global Positioning System) coordinates of latitude and longitude with CPA data to calculate the target's location and use slope of the path of travel made by the moving target to calculate speed and direction of the target. The application

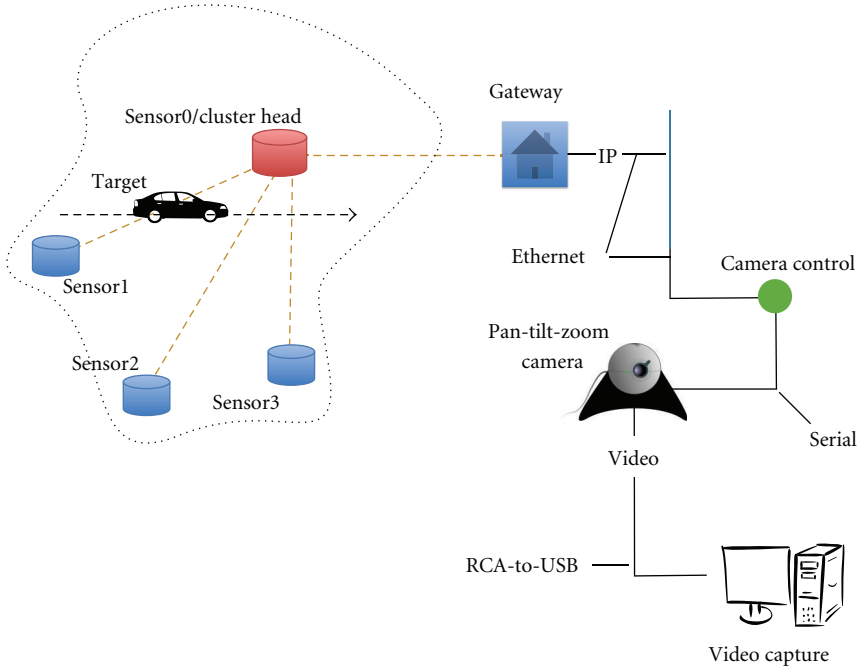


Figure 10: Setup of entire network.

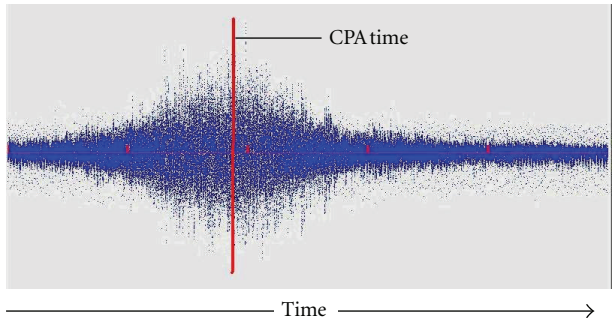


Figure 11: Sound Signature and CPA time of a target moving relative to a single acoustic sensor.

produces a named data TARG_INFO, which is forwarded to the IP network by gateway node.

Target Tracking Application. Target tracking application is implemented on an EA, the camera controller shown in Figure 10. The camera controller and the gateway node are both attached to an improvised IP Ethernet network. An EA application on camera controller sends subscriptions for named data TARG_INFO to the gateway and receives named data produced by *TARG_INFO complex producer* in WSN. It relies on the TARG_INFO data to drive the camera to track a moving target. The video capture by camera is stored in a separate PC in IP network ("video capture" in Figure 10).

Logging and Monitoring Application. Running on an IP network node, the logging and monitoring application implemented simply subscribes to all of the data types the WSN can produce. Upon receiving named data, the application checks the named data type and simply writes its contents to an appropriate log file for long-term storage or further offline analysis at a later time.

7. Experimental Results

7.1. Application Results. Several experiments are conducted, where a moving target travels through the acoustic WSN used for target tracking. Table 2 shows the results of eight experiment runs with the error for target position, speed, and angle of the target. The target position is the position of the target at the closest point of approach with respect to the cluster head. The angle of the target is the angle of the target trajectory from the latitude line. The minimum, maximum and average error for each of target position, speed, and angle is aggregated at the bottom of the table. As shown in the table, the average error of positioning is 1.78 meters, which is near the differential GPS's accuracy 1 meter [39].

Table 2: Target tracking experimental results.

Run	Target position (meters)		Speed (m/s)		Angle (degree)	
	Computed	Error	Computed	Error	Computed	Error
1st	(1099888.48, 195486.53)	1.01	12.52	0.89	90	3
2nd	(1099888.48, 195486.53)	1.01	12.07	1.34	90	3
3rd	(1099888.78, 195485.61)	2.23	18.78	5.36	84	3
4th	(1099886.65, 195486.22)	6.00	12.07	1.34	111	24
5th	(1099884.52, 195485.92)	1.01	12.96	0.45	127	1
6th	(1099884.21, 195486.22)	1.01	12.52	0.89	130	2
7th	(1099884.21, 195486.22)	1.01	12.96	0.45	131	3
8th	(1099884.52, 195486.53)	1.01	14.31	0.89	132	4
Min		1.01		0.45		1
Max		6.00		5.36		24
Average		1.78		1.45		5.375

7.2. LOC Metric. LOC, number of lines of code, is used as a metric to measure "easiness" of building sensor node applications. Table 3 shows the comparison of LOC of two different sensor node applications between with DS and without DS. There are two different sensor node applications: (1) CPA_INFO application which executes on each sensor node to produce the named data CPA_INFO and (2) TARG_INFO application which executes on the cluster head to produce its named data TARG_INFO using CPA_INFO received from each sensor node.

As shown in the table, DS helps to reduce the lines of code required for programming both simple producer application and complex producer application.

Table 3: Comparison of LOC required to write two different applications.

Sensor node application name	LOC without DS	LOC with DS
CPA_INFO application	>50	<10
TARG_INFO application	>65	<15

7.3. Dynamic Service Performance

7.3.1. Performance Metrics. To evaluate the performance of DS, it is necessary to evaluate the impact of DS on sensor node applications. A relatively simple timing analysis is employed to evaluate the impact of DS on sensor node applications' ability to be tasked and to publish data.

First, a consideration is made for the *tasking time* of pure DD and DS sensor node applications. For pure DD sensor applications, the *tasking time* is defined as the time between when the tasking thread is first entered and when the main thread realizes it has been tasked. For DS sensor applications, *tasking time* is defined as the time between when DSs tasking thread is first entered and when the application realizes it has been tasked. For DS, this will give an idea of the length of time it takes for this information to travel through DS, through the message queue, and into the sensor application.

Publishing time is defined as the time between when the sensor node application is tasked and when the data is handed over to DD for network transmission. For pure DD sensor applications, the *publishing time* is the time between when the application is tasked and when the application finished handing over the data to DD. For DS sensor applications, the *publishing time* is the time between when the application is tasked and when DS hands over the data to DD. For DS, this will give an idea of the length of time it takes for this information to travel through DSs message queue and through DS to the DD network.

Figures 12 and 13 illustrate *tasking time* and *publishing time* for sensor node applications with or without DS.

7.3.2. Results. *Tasking time* and *publishing time* are recorded in the experiments for different number of running sensor applications. Figures 14 and 15 are the plots of results.

In the case of tasking, DS has the ability to task the application in less than one millisecond, whereas the busy waiting, or polling, of pure DD applications increases the time to task the application dramatically as the number of running sensor applications increases.

For the publishing process, when with only a few applications running, DS publishes just as quickly as pure DD applications. As the number of running applications increases, however, it becomes clear that DS has an advantage. Since DS sensor applications are sleeping while awaiting tasking at their own message queues rather than busily waiting like pure DD sensor applications; the publishing time remains consistently better than the publishing time for pure DD sensor applications because it is not necessary for the tasked sensor node application to compete for CPU cycles.

8. Conclusions

This paper presents an approach for seamless interconnection between IP networks and WSNs, whereby IP-based hosts can access and manipulate IP-enabled WSNs. Our approach uses common dynamic services for transparent communication with IP-enabled WSNs that allows IP-based hosts to easily task and harvest data from remote dynamic services

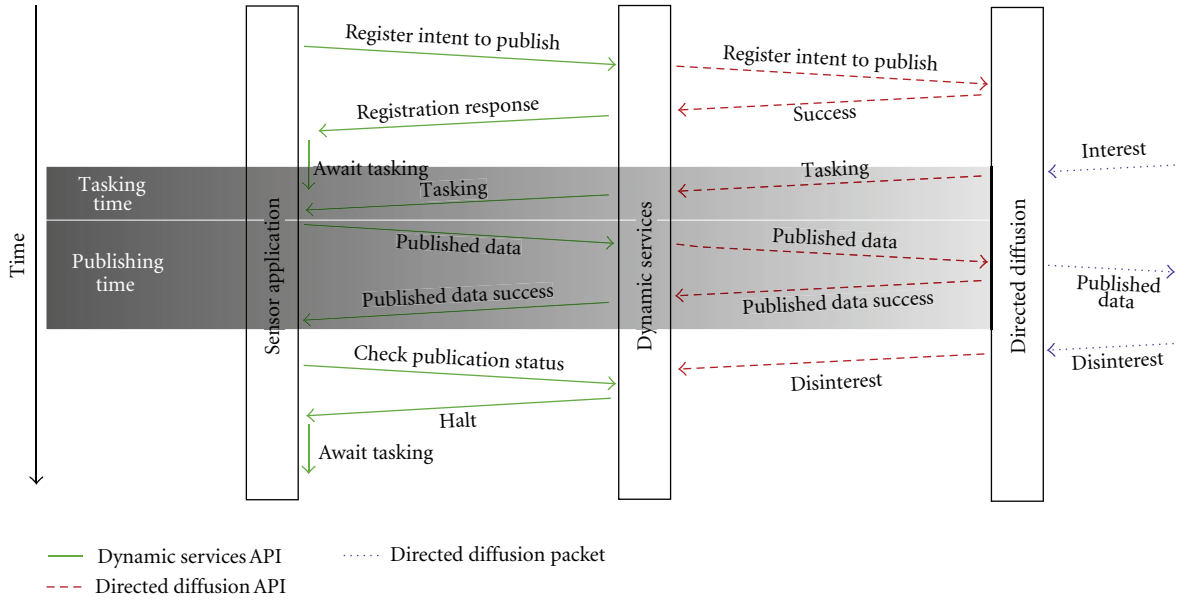


FIGURE 12: *Tasking time* and *publishing time* for sensor node applications with dynamic service.

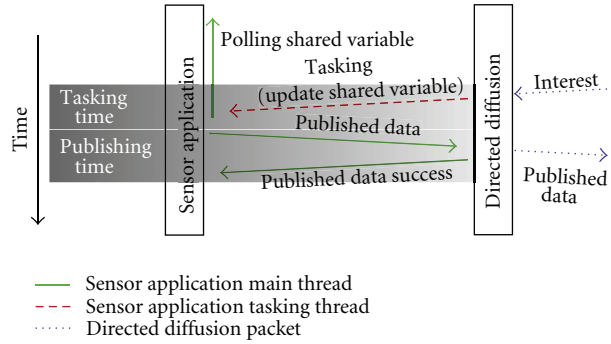


FIGURE 13: *Tasking time* and *publishing time* for sensor node applications without dynamic service.

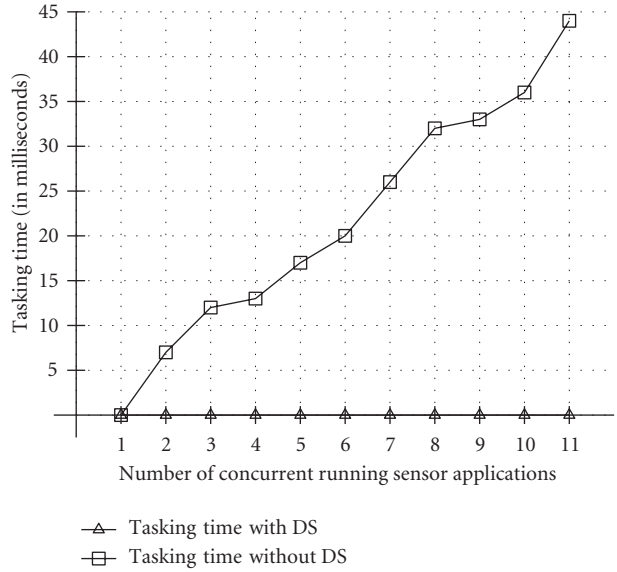


FIGURE 14: *Tasking time* for sensor applications both with and without Dynamic Services as the number of running sensor applications varies.

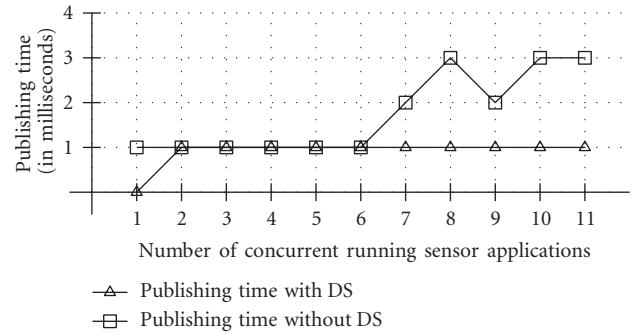


Figure 15: *Publishing time* for sensor applications both with and without dynamic services as the number of running sensor applications varies.

enabled directed diffusion wireless sensor networks. APIs for both IP-based and sensor node application programmers are presented. A description of an application-layer gateway has been given which is used to enable IP-based hosts to gather data from one or more remote WSNs.

A LOC count metric is used for comparing the pure DD API versus the DS API presented in this paper. In the two applications implemented using both the pure DD API and the DS API, it has been shown that the DS API significantly reduces the amount of programming work which must be done in implementing a sensor node application versus using the pure DD API. A performance analysis which clearly shows the value of using DS rather than relying on the polling of pure DD sensor applications is also shown. In short, the performance impact of DS, since it mostly relies on system V message queues for tasking and publishing, is negligible when compared to the performance of pure DD sensor applications for the same operations and actually improves performance due to the fact that pure DD applications busily wait to be tasked.

It is demonstrated that it is possible to, at least partially, bridge the gap between data-centric networks and host-centric networks like IP. Through using the gateway node's mapping service, it is possible to transfer data from a data-centric WSN to interested IP-based hosts.

Future work could follow many different paths.

(i) Currently only EAs submitting interests to WSNs are supported. Some work could be done in allowing the following:

  (1) sensor nodes in a WSN to submit interests for named data types to *other* remote WSNs;

  (2) sensor applications to submit interests for named data to EAs.

(ii) DS API could be extended to include all of the flexibility of the pure DD API.

(iii) DS could be ported to other data-centric networking protocols.

(iv) DD is a best effort service but essentially does not guarantee delivery of data. The DS API could be extended to ensure guaranteed delivery of data to the gateway node or to other sensor nodes.

(v) The current EA API only supports UDP. Extensions could be made to the API that also allow for TCP connections between the EA and the remote gateway node. By combining this and DS assisted guaranteed delivery described in the previous path, delivery of packets from individual sensor node to EAs could be guaranteed.

Acknowledgment

This research is supported in part by the US Army Night Vision Electronic Sensors Directorate (NVESD) under prime Contract no. DAAB07-03-D-C213-005, Subcontract no. SUB1170933RB.

References

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[2] A. Bharathidasan, V. Anand, and S. Ponduru, "Sensor networks: an overview," in *Proceedings of the IEEE Infocom*, 2004.

[3] B. Keith Maharrey, *A gateway-based approach for information retrieval from data-centric wireless sensor networks from IP hosts [M.S. thesis]*, Auburn University, December 2010.

[4] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, Atlanta, Ga, USA, September 2002.

[5] S. N. Simic and S. Sastry, "Distributed environmental monitoring using random sensor networks," in *In Proceedings of the 2nd International Workshohp on Information Processing in Sensor Networks*, pp. 582–592, Palo Alto, Calif, USA, 2003.

[6] A. Dunkels, T. Voigt, N. Bergman, and M. Jnsson, "An IP-based sensor network as a rapidly deployable building security system," in *Swedish National Computer Networking Workshop*, Karlstad, Sweden, November 2004.

[7] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proceedings of the 2nd European Workshop onWireless Sensor Networks (EWSN '05)*, pp. 108–120, February 2005.

[8] H. Dai and R. Han, "Unifying micro sensor networks with the internet via overlay networking," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 571–572, Tampa, Fla, USA, November 2004.

[9] M. El Barachi, A. Kadiwal, R. Glitho, F. Khendek, and R. Dssouli, "The design and implementation of architectural components for the integration of the IP multimedia subsystem and wireless sensor networks," *IEEE Communications Magazine*, vol. 48, no. 4, pp. 42–50, 2010.

[10] S. Hong, D. Kim, M. Ha et al., "SNAIL: an IP-based wireless sensor network approach to the Internet of things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34–42, 2010.

[11] B. Campos, J. Rodrigues, L. Mendes, E. Nakamura, and C. Figueiredo, "Design and construction of wireless sensor network gateway with IPv4/IPv6 support," in *Proceedings of IEEE International Conference on Communications (ICC '11)*, pp. 1–5, June 2011.

[12] A. Dunkels, T. Voigt, J. Alonso, H. Ritter, and J. Schiller, "Connecting wireless sensornets with TCP/IP networks," in *Proceedings of the 2nd International Conference on Wired/Wireless Internet Communications (WWIC '04)*, Frankfurt, Germany, February 2004.

[13] K. A. Emara, M. Abdeen, and M. Hashem, "A gateway-based framework for transparent interconnection between WSN and IP network," in *Proceedings of the IEEE EUROCON (EUROCON '09)*, pp. 1775–1780, May 2009.

[14] S. Ping, C. Chang, L. Kejie, and S. Li, "The design and realization of embedded gateway based on WSN," in *International Conference on Computer Science and Software Engineering (CSSE '08)*, pp. 32–36, December 2008.

[15] P. K. Mohanty, "A framework for interconnecting wireless sensor and IP networks," in *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*, September 2007.

[16] A. Leonardi, S. Palazzo, F. Scoto, and S. Signorello, "Design and construction of wireless sensor network gateway with IPv4/IPv6 support," in *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC '11)*, pp. 285–290, July 2011.

[17] R. Ding and H. Du, "Location-based IP addressing in IP-enable wireless sensor networks," in *Proceedings of the International Conference on Control, Automation and Systems Engineering (CASE '11)*, pp. 1–4, July 2011.

[18] A. Lewandowski, V. Köster, and C. Wietfeld, "Performance evaluation of AODV and OLSR-meshed IP-enabled IEEE802.15.4," in *Proceedings of the 3rd International Conference on Advances in Mesh Networks (MESH '10)*, pp. 7–12, July 2010.

[19] A. K. M. Azad, J. Kamruzzaman, B. Srinivasan, K. H. M. Alam, and S. Pervin, "Query processing over distributed heterogeneous sensor networks in future internet: scalable architecture and challenges," in *Proceedings of the 2nd International Conference on Advances in Future Internet (AFIN '10)*, pp. 75–81, July 2010.

[20] K. F. Navarro, E. Lawrence, and B. Lim, "Medical motecare: a distributed personal healthcare monitoring system," in *Proceedings of the International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED '09)*, pp. 25–30, February 2009.

[21] G. J. Pottie and W. J. Kaiser, "Embedding the Internet: wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[22] J. J. P. C. Rodrigues and P. A. C. S. Neves, "A survey on IP-based wireless sensor network solutions," *International Journal of Communication Systems*, vol. 23, no. 8, pp. 963–981, 2010.

[23] G. Wagenknecht, M. Anwander, and T. Braun, "SNOMC: an overlay multicast protocol for wireless sensor networks," in *Proceedings of the 9th Annual Conference on Wireless On-demand Network Systems and Services (WONS '12)*, pp. 75–78, January 2012.

[24] M. V. Pulgarin, R. Glitho, and A. Quintero, "An overlay gateway for the integration of IP multimedia subsystem and mobile sink based—wireless sensor networks," in *Proceedings of the 72nd IEEE Vehicular Technology Conference Fall (VTC '10-Fall)*, pp. 1–5, September 2010.

[25] N. Pollner, M. Daum, F. Dresslery, and K. Meyer-Wegener, "An overlay network for integration of WSNs in federated stream- processing environments," in *Proceedings of the10th IFIP Annual Mediterranean Ad Hoc Networking Workshop*, June 2011.

[26] S. Coleri Ergen, "ZigBEE/IEEE802.15.4 Summary," September 10, 2004.

[27] G. Mulligan, "The 6LoWPAN architecture," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets '07)*, pp. 78–82, June 2007.

[28] G. Moritz, F. Golatowski, and D. Timmermann, "A lightweight SOAP over CoAP transport binding for resource constraint networks," in *Proceedings of the IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS '11)*, pp. 861–866, October 2011.

[29] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6LoWPAN with compressed IPsec," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS '11)*, pp. 1–8, June 2011.

[30] A. A. Hasbollah, S. H. S. Ariffin, and M. I. A. Hamini, "Performance analysis for 6loWPAN IEEE 802.15.4 with IPv6 network," in *Proceedings of the IEEE Region 10 Conference (TENCON '09)*, pp. 1–5, January 2009.

[31] M. Durvy, J. Abeillé, P. Wetterwald et al., "Making sensor networks IPv6 ready," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 421–422, 2008.

[32] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 56–67, Boston, Mass, USA, August 2000.

[33] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 56–67, Boston, Mass, USA, August 2000.

[34] 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.

[35] F. M. Dommermuth, "The estimation of target motion parameters from CPA time measurements in a field of acoustic sensors," *The Journal of the Acoustical Society of America*, vol. 83, no. 4, pp. 1476–1480, 1988.

[36] Q. Yang, A. Lim, K. Casey, and R. K. Neelisetti, "An empirical study on real-time target tracking with enhanced CPAalgorithm in wireless sensor networks," *Ad-Hoc and Sensor Wireless Networks*, vol. 7, no. 3-4, pp. 225–249, 2009.

[37] Q. Yang, A. Lim, K. Casey, and R. K. Neelisetti, "An enhanced CPA algorithm for real-time target tracking in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 5, pp. 619–643, 2009.

[38] Q. Yang, A. Lim, K. Casey, and R. K. Neelisetti, "Real-time target tracking with CPA algorithm in wireless sensor networks," in *Proceedings of the 5th Annual IEEE Communications Society*

Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08), pp. 305–313, June 2008.

[39] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal, "GPS: location tracking technology," *Computer*, vol. 35, no. 4, pp. 92–94, 2002.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 957606, 11 pages
doi:10.1155/2012/957606

Research Article

Broadcasting with Least Redundancy in Wireless Sensor Networks

Ruiqin Zhao, Xiaohong Shen, Zhe Jiang, and Haiyan Wang

Department of Electronic and Communication Engineering, College of Marine, Northwestern Polytechnical University, 710072 Xi'an, China

Correspondence should be addressed to Ruiqin Zhao, rqinzhao@gmail.com

Received 4 July 2012; Revised 10 October 2012; Accepted 24 October 2012

Academic Editor: Deyun Gao

In wireless sensor networks (WSN), broadcasting could allow the nodes to share their data efficiently. Due to the limited energy supply of each sensor node, it has become a crucial issue to minimize energy consumption and maximize the network lifetime in the design of broadcast protocols. In this paper, we propose a Broadcast Algorithm with Least Redundancy (BALR) for WSN. By identifying the optimized number of induced forwarders as 2, BALR establishes a weighted sum model, taking both rebroadcast efficiency and residual energy into consideration, as a new metric to compute the self-delay of the nodes before rebroadcasting. BALR further incorporates both strategies based on distance and coverage degree which means the number of neighbors that have not yet received the broadcast packet, to optimize the rebroadcast node selections. To reveal the performance bounds, rebroadcast ratios in the ideal and worst case are theoretically analyzed, indicating that the rebroadcast ratio of BALR decreases with the increase of node density. BALR can significantly prolong the network lifetime of WSN and is scalable with respect to network size and node density, as demonstrated by simulations.

1. Introduction

Wireless sensor networks (WSN) are envisioned as consisting of a number of static sensor nodes that are densely deployed over a region of interest. A wide variety of applications of such networks include inventory managing, disaster areas monitoring, patient assisting, water quality monitoring, target tracking, and health monitoring of civil infrastructures. Recent advances in wireless communications and electronics have enabled the development of such low-cost sensor networks. Unfortunately, wireless sensor nodes, which are generally microelectronic devices, could only be equipped with limited power sources. Therefore, energy efficiency is of particular importance in WSN [1–4].

Broadcasting is a common means for nodes in WSN to efficiently share their data with each other. Broadcasting could be utilized to initialize the network configuration for network discovery, discover multiple routes between a given pair of nodes, and query for a piece of desired data in a network [5]. In WSN, broadcasting could also be served as an efficient approach for sensors to share their local measurements with each other. A straightforward way of broadcasting is the so-called flooding, under which each node will rebroadcast when it receives the broadcast packet for the first time. Although attractive for its simplicity, flooding will cause serious broadcast redundancy, packets collision, and bandwidth waste, referred to as broadcast storm problems [2]. An efficient broadcast strategy should be able to effectively reduce the broadcast redundancy, for both energy and bandwidth efficiency, especially in a band and power limited sensor networks.

With the aim of solving the broadcast storm problems and maximizing the network lifetime, we propose a Broadcast Algorithm with Least Redundancy (BALR) for wireless sensor networks, which possess the following properties.

Scalable Algorithm. Scalability is a critical issue for sensor networks which is composed of thousands of densely deployed nodes. BALR is designed in mind with the goal of obtaining satisfying broadcast performance in a high-density

and large-scale network. With BALR, the number of saved rebroadcasts increases with the increase of the network node density.

Localized Algorithm. Each node makes the decision of rebroadcast according to its one-hop local information. BALR needs not maintain any global topology information at each node, thus the overhead is small.

Energy-Efficient Approach. BALR cuts down the total energy consumption by reducing the redundancy of rebroadcast effectively which is also capable of relieving the broadcast storm problems significantly. To maximize network lifetime, BALR balances the energy consumption among all nodes when rebroadcast nodes are selected.

This paper is organized as follows. In Section 2, we analyze existing related works in the literature. In Section 3, we discuss system model and optimized number of forwarders in WSN. Based on the system model, our proposed Broadcast Algorithm with Least Redundancy (BALR) and its performance analysis are presented in Sections 4 and 5, respectively. We present computer simulation results in Section 6 for performance verification before concluding with Section 7.

2. Related Works

There have been a number of existing works on the broadcast storm problem of wireless multihop networks in the literature. In [3, 4, 6], each node computed a local cover set, consisting of as fewer neighbors as possible, to provide its whole 2-hop coverage area by exchanging connectivity information with its neighbors. However, each node in these works is required to update the information of its k-hop ($k \geq 2$) neighbors, resulting in a heavy overhead and prohibited energy consumption. Some other works [2, 6] concentrated on forward node selection based on probabilistic approach. Nonetheless, the reachability under such strategies may not be guaranteed.

Among related works in the literature, many proposed energy-efficient broadcast protocols are centralized, of which the topology of the whole network is required. Various protocols are proposed to search the minimizing energy cost of the broadcast tree. The authors of [7–9] accomplished the searching based on geometry or graph of the network. Alternatively, a connected dominating set (CDS) could be constructed, and only permitting nodes which belong to the CDS are allowed to rebroadcast packets. To minimize the overhead of broadcast, various strategies reducing the size of CDS were investigated in [10–16].

Since the centralized approach needs much more overhead in WSN, alternative localized algorithms have been proposed [17–21]. Under such protocols, each node establishes the network topology in a distributed way [18]. In [19, 20], each node should be aware of the geometry within its 2-hop neighborhood range. In order to ameliorate broadcasting, [10] utilize the information of all nodes that have been visited by the broadcast message. The authors of [16] proposed an algorithm suitable for a dynamic mobile Ad Hoc network, which did not require neither the k-hop neighbors information nor the entire network topology. To reduce the broadcast overhead, [21] proposed a Maximum Life-time Localized Broadcast (ML^2B) protocol, of which the information of only one-hop neighbors was required. ML^2B utilized the number of neighbors that have not received the broadcast message to reduce the rebroadcast redundancy.

Some broadcast mechanisms designed based on the features of WSN have been proposed recently [20–30]. In [20], two types of broadcasting protocols for WSN, called one-to-all and all-to-all broadcasting, were proposed. Both protocols are suitable for fixed and regular WSN topologies. An energy-efficient broadcasting strategy based on cooperative transmission was investigated in [23]. The cooperation was provided through a system, called Opportunistic Large Array (OLA), in which network broadcasting was accomplished by signal processing techniques at the physical layer. Some works [5, 24, 25] dealt with the query execution in large sensor networks. Their purpose is not to broadcast a packet to the whole network but to obtain or locate data or services for nodes within a large population, high-density WSN based on network partial broadcast. Several robust data delivery protocols [26–28] have been proposed for large sensor networks to disseminate data to interested sensors.

This paper focuses on the broadcasting strategy for WSN to efficiently forward a broadcasted packet from broadcast originator to all other nodes in the network. By identifying the optimized number of induced forwarders as 2, a broadcasting algorithm with least redundancy (BALR) is proposed, which optimizes broadcasting by reducing redundant rebroadcasts and balancing energy consumption among all nodes. In [29], a broadcast protocol for sensor networks (BPS) was proposed, which utilized an adaptive-geometric approach that enables a considerable reduction of retransmissions by maximizing each hop length. Simulation results regarding the rebroadcast ratio demonstrate the feasibility of applying adaptive-geometric approach to WSN broadcasting. Based on [29], the authors extended the ideas of BPS for broadcasting in the energy-constrained network consisting of nodes that sleep and wake up alternatively in [30]. And they proposed a protocol called Activecast to effectively transmit a packet to all active (awake) nodes in the network. However, this paper does not consider the WSN consisting of nodes that sleep and wake up alternatively but considers the homogeneous WSN consisting of nodes that have the identical transmission range and that are always active until their battery exhaustion. Furthermore, most of these literatures focus on rebroadcast ratio performances, without looking at the network lifetime which is one of the main design purposes for any broadcast schemes. Motivated by the optimized selection of induced forwarder, BALR establishes a weighted sum model, taking both rebroadcast efficiency and residual energy into consideration, as a new metric to compute the self-delay of the nodes before rebroadcasting. BALR further incorporates both strategies based on distance and coverage-degree, to optimize the rebroadcast node selections; thus, its scalability and high energy-efficiency being achieved. Besides, as each node

makes the decision of rebroadcast according to only its one-hop local information, BALR is a localized algorithm.

3. System Model

A wireless sensor network can be abstracted as a graph $G(V, E)$, in which V is the set of all the nodes in the network and E consists of edges in the graph. r is the radius of the coverage of each node. We assume that all links in the graph are bidirectional, the graph is in a connected state, and each node has a circular coverage area. Given a node i, time t is set as zero when it receives the broadcasted packet for the first time. The residual energy of node i is $e(i, t)$.

In particular, we use the radio transmission energy model as in [31]. To transmit a k bits packet over a distance d, the radio expends energy of

$$E_{Tx} = E_{\text{elect}} \cdot k + \varepsilon_{\text{amp}} \cdot k \cdot d^2 \tag{1}$$

and to receive this packet, the radio expends

$$E_{Rx} = E_{\text{elect}} \cdot k, \tag{2}$$

where E_{elect} is 50 nJ/b to run the transmitter circuitry, and ε_{amp} is 100 pJ/b/m^2 for the transmit amplifier to achieve an acceptable signal-to-noise ratio.

As location is more important than a specific node's ID in WSN, location awareness is necessary to make the sensor data meaningful. The proposed BALR utilizes geographic location to make localized broadcast decisions. Each node is required to be aware of only the positions of its one-hop neighbors.

3.1. Optimized Number of Induced Forwarders. Given a forward node that has done the rebroadcast, its neighbors that do the rebroadcast after hearing the rebroadcast from it are called its induced forwarders. To reduce the rebroadcast redundancy, the number of induced forwarders of each forward node should be minimized.

During the broadcast of a packet, as shown in Figure 1, a forward node S rebroadcasts a broadcast packet received from its preceding node U. Then its induced forwarders are chosen from its neighbors locally by themselves. Lastly the induced forwarders of S will do the rebroadcast as their preceding node S. Let n be the number of the induced forwarders of a forward node. We use $I_1, I_2, \ldots, I_n$ to represent n induced forwarders of S. The induced coverage region of node S consists of added coverage regions of the n induced forwarders and that of node U, which is the shadowed region in Figure 1.

To obtain a high coverage ratio of broadcast, large n is desired. But large n also leads to much redundant rebroadcasts. Therefore, an optimal n is required to obtain a satisfying delivery ratio and as fewer rebroadcasts as possible.

For $n = 1$, as shown in Figure 1(a), due to the limited number of induced forwarders, the induced coverage region of a node is geometrically unbalanced, resulting in an unbalanced coverage and incomplete delivery in the network. For $n \geq 2$, as shown in Figures 1(b) and 1(c), when the n induced forwarders and the preceding node of the forwarder are equally located at the edge of the coverage region of the forwarder node, the induced coverage region is symmetric and balanced [32]. As what will be seen later in Figure 2, full deliveries throughout the network could be guaranteed. The size of the superposition areas of the coverage areas of nodes' rebroadcast reflects the efficiency of the broadcast. The larger the size of the superposition areas, the lower the broadcast efficiency is. The size of the superposition areas in Figure 1(c) is much larger than that in Figure 1(b), resulting in a lower broadcast efficiency. For $n > 3$, the size of the superposition areas is further larger than that of $n = 3$. Thus, the broadcast efficiency with $n = 2$ is higher than that with $n \geq 3$.

When two induced forwarders are selected as in Figure 1(b), the network could obtain a satisfying delivery ratio with the highest broadcast efficiency [32]. Therefore, BALR optimizes a number of the induced forwarders by setting $n = 2$, as shown in Figure 1(b).

3.2. Ideal Coordinates of Induced Forwarders. Let (x_s, y_s) and (x_u, y_u) be the locations of node S and that of preceding node U in Figure 1(b). Then $l = \sqrt{(x_s - x_u)^2 + (y_s - y_u)^2}$ is the distance between node S and U. The coordinates of two ideal induced forwarders (I_1 and I_2 in of Figure 1(b)) for $n = 2$ can be obtained by solving the following two equations:

$$\begin{aligned} (x - x_s)^2 + (y - y_s)^2 &= r^2, \\ (x - x_u)^2 + (y - y_u)^2 &= \left(\frac{l + r}{2}\right)^2 + \frac{3r^2}{4}. \end{aligned} \tag{3}$$

When transmitting node S is the broadcast originator, there are three optimized (ideal) locations for $n = 2$, that is, $(x_s - r, y_s)$, $(x_s + r/2, y_s - r\sqrt{3}/2)$, and $(x_s + r/2, y_s + r\sqrt{3}/2)$. On the other hand, if the transmitting node S is a forward node, there are two ideal locations, that is, (x_1, y_1) and (x_2, y_2), which are calculated by solving (3).

4. Broadcasting with Least Redundancy (BALR)

As discussed in the previous section, nodes forward the broadcasted packet following patterns shown in Figure 1(b). Ideally, as shown in Figure 2, the broadcast packet is delivered along hexagons with edge length equal to the radius of the coverage of each node. And all rebroadcast nodes are located at vertices of the hexagons which are called ideal locations. To determine the ideal locations, each broadcasted packet contains a field A in its header. Whenever a node forwards a broadcast packet, it fills in A the position of the node from which it received the packet.

4.1. Weighted Sum Metric for Self-Delay. Nonetheless, in practical situations, nodes may not be located at ideal points. Naturally the nodes nearest to the ideal locations are selected as forwarders. However, this scheme would lead to the situation that nodes located at the ideal or quasi-ideal locations are exhausted rapidly with respect to energy. We hence propose to take energy metric into consideration, besides the location metric, for forwarder selections. BALR incorporates these two metrics together to form a new

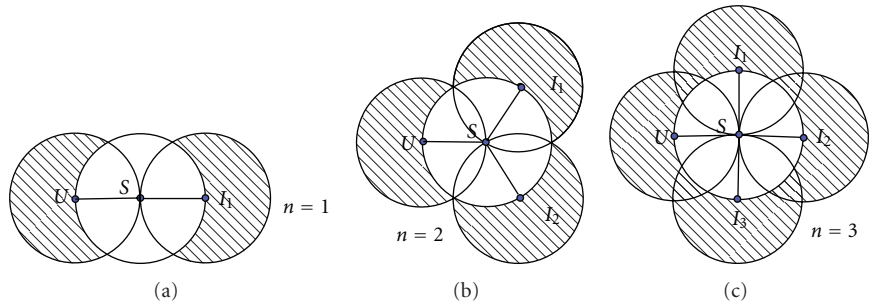


Figure 1: Selection of induced forwarders. The shadowed regions are the induced coverage region of node S.

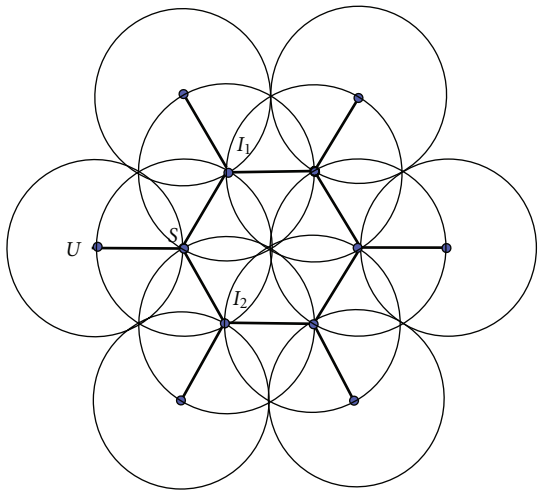


Figure 2: Path for delivering the broadcast packet in ideal conditions using BALR.

metric, via following a weighted sum model (SMD), and utilizes the self-delay mechanism to complete the rebroadcast node selection:

$$D(i) = D_m\left(\alpha \cdot \frac{l(i,t)}{r} + \beta \cdot \frac{[E' - e(i,t)]}{E' - E_T}\right), \quad (4)$$

where $D(i)$ denotes self-delay of node i before rebroadcasting, D_m is the permitted maximum delay, $l(i,t)$ is the distance from node i to the nearer ideal location at time t, E' is the initial maximum energy of each node, and E_T is the energy threshold which is used to prevent nodes with little energy from dying. If the residual energy at a node is smaller than E_T, it refuses to forward the broadcasted message. α and β denote the relative weights of importance of the location and residual energy of the node, respectively, that is, $\alpha + \beta = 1$, $0 \le \alpha \le 1$, and $0 \le \beta \le 1$.

When forall node i receives the broadcasted packet for the first time, it defers a period of $D(i)$ before its attempt to do the rebroadcast. During the interval of $0 \le t \le D(i)$, node i could abandon its attempt to rebroadcast as soon as either rebroadcast efficiency is low or residual energy is insufficient. The self-delay mechanism selects nodes with small $D(i)$ as rebroadcast ones with high priority. Based on the weighted sum metric for self-delay as shown by formula (4), the nodes nearest to the ideal locations with enough residual energy would be selected by BALR with the specified relative weights α and β, depending on specific applications.

Clearly there is a tradeoff between rebroadcast efficiency and network lifetime based on our new metric. In practice, we could regulate the tradeoff by adjusting the relative weights α and β. In addition, the tradeoff between broadcast reachability and network lifetime could be regulated by adjusting the threshold E_T.

Based on the new metric, our protocol further reduces the rebroadcast redundancy by incorporating both strategies based on distance and coverage degree as follows.

4.2. Strategy Based on Distance. We propose to reduce the rebroadcast redundancy by confining the location of quasi-forwarders. More specifically, only neighbors within a specified distance l_T from one of the ideal locations are allowed to rebroadcast. Intuitively, l_T would decrease as the node density increases, which corresponds to a high density and large WSN. The value of l_T could be determined by the surrounding node density. Let D denotes the node density, that is, the average number of nodes per region of $r \times r$. When nodes are placed in a grid pattern, the smallest distance between two neighbor nodes is $\sqrt{r^2/D}$. l_T could be computed as follows:

$$l_T = \begin{cases} r, & 0 \le D < a^2, \\ a\sqrt{\dfrac{r^2}{D}}, & a^2 \le D, \end{cases} \quad (5)$$

where a is a constant.

4.3. Strategy Based on Coverage Degree. It is noted that a node might receive a broadcast packet several times from different nodes in different directions, leading to redundant rebroadcasts. Define coverage degree as the number of neighbors that have not received the broadcast packet yet. Note that the coverage degree implies the rebroadcast efficiency of a node. To minimize rebroadcasts, we propose to have each node maintain its coverage degree, and rebroadcast only when its coverage degree is above the threshold d_T. Definitely there is a tradeoff between the rebroadcast redundancy and reachability via the threshold d_T. High threshold d_T may be superior for its rebroadcast efficiency, though probably lead to poor reachability. On the other hand, low threshold d_T will

affect the performance the other way round. For example, in large-scale WSN with nodes densely deployed, high threshold d_T should be selected to avoid highly redundant rebroadcasts.

4.4. Broadcasting Algorithm with Least Redundancy. Let s and P_s be the broadcast originator and the packet broadcasted from$s$, respectively. In addition, we define several variables for forall $i \in V$ as follows.

(i) *Neighbor* $nb(i)$ denotes a one-hop neighbor of node i.

(ii) *Neighbor set* $NB(i)$ denotes the set of all one-hop neighbors of node i.

(iii) *Uncovered set* $UC(i,t)$ consists of one-hop neighbors that have not been covered, at time t.

(iv) *Coverage degree* $d(i,t)$ is the number of nodes belonging to $UC(i,t)$ at t. $d(i,t)$ implies the rebroadcast efficiency of node i. If $d(i,t)$ is below a threshold before its attempt to do the rebroadcast, node i would not rebroadcast.

(v) *Preceding node* $u(i,t)$ is the $nb(i)$ that sent the broadcast packet to node i at time t $(0 \le t \le D(i))$ ($D(i)$ is the self-delay of node i). During the period of $0 < t \le D(i)$, node i may receive several copies of the same broadcast packet from different preceding nodes.

(vi) *Preceding node set* $U(i,t)$ is the set of all preceding nodes of node i before time t. If it has received the same broadcast packet for k times before time t $(t \le D(i))$, its preceding node set can be expressed as

$$U(i,t) = \{u(i,t_0), u(i,t_1), u(i,t_2), \ldots, u(i,t_{k-1})\}, \quad (6)$$

where $t_0, t_1, t_2, \ldots$, and t_{k-1} $(t_{k-1} \le t)$ record the time node and i receives the 1st, 2nd, 3rd, ..., and kth copy $(k \ge 1)$ of the same broadcast packet.

For $\forall u(i,t_j) \in U(i,t)$, $j \in \{0,1,\ldots,k-1\}$, node i will update its uncovered set $UC(i,t_j)$ when it receives the broadcasted packet P_s form $u(i,t_j)$. It is noted that the A field of P_s shows the position of $u(u(i,t_j),0)$, which is a preceding node of $u(i,t_j)$. Based on the A field of P_s and the locally obtained position of $u(i,t_j)$, node i updates $UC(i,t_j)$ by deleting nodes that are covered by $u(u(i,t_j),0)$ and $u(i,t_j)$ as shown by Figure 3. Therefore, node i could calculate its new coverage degree $d(i,t_j)$. Node i increases j by one each time it receives another repeated P_s, and $d(i,t_j)$ decreases with the increase of j. During the interval of self-delay, node i could abandon its attempt to rebroadcast as soon as $d(i,t_j) \le d_T$, thus reducing the rebroadcast redundancy and energy consumption efficiently.

Following the previous discussion, our proposed Broadcast Algorithm with Least Redundancy (BALR) for forall $i \in (V - \{s\})$ is summarized in Algorithm 1.

We remark that our BALR maximizes the lifetime of WSN by minimizing redundant rebroadcast and balancing the broadcast energy consumption of neighborhood. It utilizes the node self-delay scheme to reduce the redundancy of nodes' rebroadcast and energy consumption. This scheme guarantees that nodes with smallest distance from the ideal location and satisfying value of residual energy are self-selected as rebroadcast nodes. To further minimize the redundant rebroadcast, each node i tracks its coverage-degree $d(i,t)$ continually, which manifests accurately the rebroadcast efficiency of the node. $d(i,t)$ is determined as one of the main criterions for deciding whether to rebroadcast the packet. The other two criterions include distance from the nearest ideal location and residual energy. The cell-like hexagonal routes for broadcast packet delivery and three criterions for deciding whether to rebroadcast the packet constitute the rationales behind BALR.

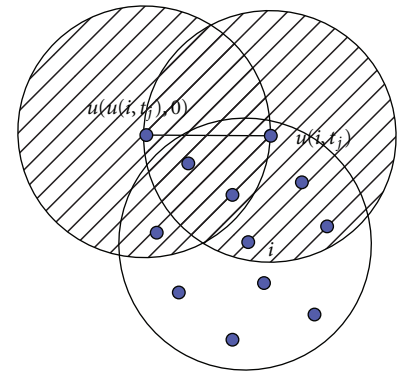


Figure 3: Node i obtains $UC(i,t_{j-1})$ by deleting the six neighbors covered by the shaded regions from $UC(i,t_{j-1})$.

5. Performance Analysis of BALR

5.1. Definitions

(i) C is the area of the entire network.

(ii) D is the node density of the network, which is the average number of nodes per region of $r \times r$.

(iii) g is the total number of all nodes in the network.

(iv) h is the number of nodes that have rebroadcasted the packet after their reception of the packet in the network.

(v) R is the rebroadcast ratio, which is the ratio of the number of nodes that have rebroadcasted the packet to the number of nodes in the entire network.

Based on the above definitions, we get

$$R = \frac{h}{g}, \qquad g = C\frac{D}{r^2}. \quad (7)$$

5.2. Efficiency of the Broadcast Protocol. Rebroadcast ratio R manifests the efficiency of the broadcast protocols. R is inversely proportional to the broadcast efficiency. Large R

(1) **Initialization**: $j = -1, U(i,0) = \varnothing, UC(i,0) = NB(i)$
(2) **if** node i has received packet P_s, go to step 3
else if $j \geq 0$, go to step 7
else the node is idle, stay in step 2
(3) **if** P_s is a new packet, go to step 4
else node i has received repeated P_s, let $j = j + 1$, and go to step 5
(4) node i has received the broadcast packet P_s for the first time. Let $t = 0$, $j = 0$ where j indicates the times of the repeated reception of P_s
if $e(i,t) < E_T$, node i abandons its attempt to rebroadcast, and go to step 8
else if $l(i,t) < l_T$, go to step 5
else node i abandons the rebroadcast, then go to step 8
(5) Let $t_j = t, u(i,t_j) = p_{t_j}, U(i,t) = U(i,t) \cup \{u(i,t_j)\}$, where p_{t_j} is the preceding node of node i at time t_j
(6) node i updates its uncovered set $UC(i,t_j)$ by deleting nodes that are covered by $u(u(i,t_j),0)$ and $u(i,t_j)$, and calculate its new coverage degree $d(i,t_j)$
if $d(i,t_j) \leq d_T$, it abandons its attempt to rebroadcast, and go to step 8
else if $j > 0$, go to step 7
else delay the rebroadcast by $D(i)$ based on formula (4)
(7) Check current time t:
if $t < D(i)$, go to step 2
else update the A field of P_s by the position of $u(i,0)$ and rebroadcasts P_s
(8) **end**

Algorithm 1: Algorithm description. Broadcast Algorithm with Least Redundancy (BALR).

results in a much redundant rebroadcast and low broadcast efficiency. Let R_i and R_w be the rebroadcast ratios of the proposed algorithm in the ideal case and in the worst case. The values of R_i and R_w could reveal the performance bounds of the broadcast algorithm.

Firstly we analyze the ideal efficiency of BALR, which is determined by the minimum rebroadcast ratio under the ideal conditions. Based on the formula (7), we get

$$R = \frac{hr^2}{CD}. \tag{8}$$

For a given sensor network, where values of C, D, and r are determinate, different broadcast protocols result in different values of h. From formula (8), we get that the rebroadcast ratio is determined by h, which is the number of nodes that have done the rebroadcast in the network. To obtain the minimum R in BALR, h should be minimized. Under ideal conditions, the network area is divided into many hexagons where, in each vertex, there is one node doing the rebroadcast. The side length of each hexagon in the network is equal to the radius of the coverage of each node. Then the number of hexagons in the entire network can be approximated as $\lceil 2C/(3\sqrt{3}r^2) \rceil$. Under ideal conditions, rebroadcast occurs at each vertex of hexagons, where each vertex locates at the ideal location. Each vertex belongs to three connected hexagons. Let h_i be the number of nodes that do the rebroadcast under ideal conditions, we have

$$h_i \approx 2 \cdot \left\lceil \frac{2C}{3\sqrt{3}r^2} \right\rceil. \tag{9}$$

Then the minimum R can be formulated as

$$R_i = \frac{2r^2 \cdot \lceil 2C/(3\sqrt{3}r^2) \rceil}{CD}. \tag{10}$$

When C is much larger than $3\sqrt{3}r^2/2$, R_i can be approximated as

$$R_i \approx \frac{4}{3\sqrt{3}D}. \tag{11}$$

From the above analyses, the minimum rebroadcast ratio R_i is obtained. R_i is dependent on the node density D. R_i decreases as the node density D increases. From formula (11), we get the ideal broadcast efficiency of BALR.

Then, we examine the efficiency and rebroadcast ratio of the proposed broadcast protocol under the worst conditions. As shown in Figure 4, after a forward node S firstly receives the broadcast packet from its preceding node U, it rebroadcasts the packet. Only $nb(S)$ located in the two limited regions C_1 and C_2, which are the shadow regions around I_1 and I_2 in Figure 4, may become the induced rebroadcast nodes of node S. The worst case occurs when the coverage degree threshold and energy threshold are set to zero. Under the worst conditions, each neighbor located in the two limited regions will rebroadcast the packet. C_3 is the limited region of node U in which node S is located. The regions where $nb(S)$ will do the rebroadcast under worst conditions can be given as $C_1 \cup C_2 \cup C_3$.

We use A_{sg}, $A_{\cup}$, $A_{\cap}^{1,2}$, $A_{\cap}^{1,3}$, and $A_{\cap}^{2,3}$ to represent the areas of C_1, $C_1 \cup C_2 \cup C_3$, $C_1 \cap C_2$, $C_1 \cap C_3$, and $C_2 \cap C_3$. With the assumption of the uniform node density in the network, the rebroadcast ratio in the worst case may be approximated as

$$R_w = \frac{A_{\cup}}{\pi r^2}. \tag{12}$$

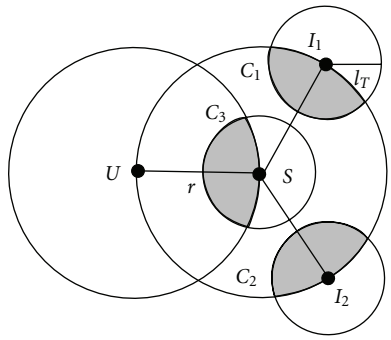


FIGURE 4: Limited shaded regions in which $nb(S)$ are permitted to do rebroadcast.

As the radius of the three limited sector regions are l_T, the areas of C_2 and C_3 are the same as that of C_1, and $A_{\cap}^{2,3} = A_{\cap}^{1,3}$. Then

$$\begin{aligned} A_{\cup} &= 3A - A_{\cap}^{1,2} - A_{\cap}^{1,3} - A_{\cap}^{2,3} \\ &= 3A - 2A_{\cap}^{1,3} - A_{\cap}^{1,2}. \end{aligned} \tag{13}$$

The area of C_1 can be formulated as

$$\begin{aligned} A_{sg} &= 2\int_0^{l_T} x\cos^{-1}\left(\frac{x}{2r}\right)dx \\ &= 2\left(\frac{x}{2} - \frac{(2r)^2}{4}\right)\cos^{-1}\left(\frac{x}{2r}\right) - \frac{x}{4}\sqrt{(2r)^2 - x^2}\Bigg|_0^{l_T}, \end{aligned} \tag{14}$$

that is,

$$\begin{aligned} A_{sg} &= \left({l_T}^2 - 2r^2\right)\cos^{-1}\left(\frac{l_T}{2r}\right) \\ &\quad - \frac{l_T}{2}\sqrt{4r^2 - {l_T}^2} + \pi r^2. \end{aligned} \tag{15}$$

The area of $C_1 \cap C_2$ and $C_1 \cap C_3$ are dependent on l_T which is the radius of the limited sector region. They can be formulated as follows:

$$A_{\cap}^{1,2} = \begin{cases} 0, & \left(0 \le l_T \le \dfrac{\sqrt{3}r}{2}\right), \\ 2\left[{l_T}^2\cos^{-1}\left(\dfrac{\sqrt{3}r}{2l_T}\right) - \dfrac{\sqrt{3}r}{2}\sqrt{{l_T}^2 - \dfrac{3r^2}{4}}\right], & \left(\dfrac{\sqrt{3}r}{2} < l_T \le r\right), \end{cases} \tag{16}$$

for $0 \le l_T \le (\sqrt{3}-1)r$,

$$A_{\cap}^{1,3} = 0 \tag{17}$$

and for $(\sqrt{3}-1)r < l_T \le r$,

$$\begin{aligned} A_{\cap}^{1,3} &= 2\int_{r\cos\phi}^{r} \sqrt{r^2 - x^2}dx \\ &\quad + 2\int_{l_T\cos\theta}^{l_T} \sqrt{{l_T}^2 - x^2}dx, \end{aligned} \tag{18}$$

where

$$\begin{aligned} \cos\phi &= \frac{4r^2 - {l_T}^2}{2\sqrt{3}r^2}, \\ \cos\theta &= \frac{2r^2 + {l_T}^2}{2\sqrt{3}r \cdot l_T}. \end{aligned} \tag{19}$$

Based on the above six formulas, the rebroadcast ratio in the worst case can be formulated as

$$R_w = \begin{cases} \dfrac{3A_{sg}}{\pi r^2}, & (0 \le l_T < (\sqrt{3}-1)r) \\ \dfrac{3A_{sg} - 2A_{\cap}^{1,3}}{\pi r^2}, & \left((\sqrt{3}-1)r \le l_T \le \dfrac{\sqrt{3}r}{2}\right) \\ \dfrac{3A_{sg} - 2A_{\cap}^{1,3} - A_{\cap}^{1,2}}{\pi r^2} \text{ and} & \left(\dfrac{\sqrt{3}r}{2} < l_T \le r\right). \end{cases} \tag{20}$$

It can be obtained from the above formulas that R_w is a function of l_T/r. R_w increases as l_T increases. When l_T is equal to r, $R_w = 1$. Therefore, l_T is usually not set as a value bigger than r.

The rebroadcast ratio of BALR in the worst case is achieved when $d_T = 0$ and $E_T = 0$. Formula (20) shows the worst performance bound of BALR. The rebroadcast ratio is inversely proportional to the efficiency of broadcast protocols. By properly choosing values of the three thresholds l_T, d_T, and E_T, satisfying efficiency of BALR will be achieved in the dense sensor networks.

6. Numerical Evaluation

We simulate BALR using OPNET and compare its performance with that of flooding and ML^{2}B [19] which could reduced redundant rebroadcast efficiently. As Broadcast Protocol for Sensor networks (BPS) [27] is one of the protocols that perform well in large-scale sensor networks, we also compare BALR with it. For physical (PHY) and medium access control (MAC) layers, we use the IEEE 802.11 wireless LAN (WLAN) model. And each node has the same transmission range of 250 m. The initial power of each node is 1.0 J. For all simulation results, Poisson streams are used. Each source sends out packets with an average rate of 5 packets per second. The data packet size is 1024 bits. The maximum delay D_m is set to 0.14 s. In the following simulations, the parameters are configured as $\alpha = 0.8$, $\beta = 0.2$, $d_T = 0.1d_m$, and $E_T = 0.2$ J, and for l_T formula (5) is used with $a = 2\sqrt{2}$. Each simulation is repeated until the 95-percent confidence intervals of all average results are within ± 5 percent.

6.1. Performance Metrics. We consider four performance metrics.

(1) *Rebroadcast ratio* (R): the ratio of the number of nodes that have rebroadcasted or broadcasted the packet to the number of nodes in the entire network. Therefore, R of flooding is 1 under all scenarios.

(2) *Reachability (RE)*: the ratio of the number of nodes that have received broadcasted packet to the number of all nodes in the simulated connected network. So RE also is known as the coverage rate.

(3) *Maximum end-to-end delay (MED)*: the interval from the time the broadcasted message is transmitted by the broadcast originator to the time the last node in the network receiving the message.

(4) *Lifetime (LT)*: the interval from the time the network is initiated to the time at which the first node dies in the network. We break the whole simulation time into many small time steps which are also called as rounds. The broadcast originator broadcasts each packet to all other nodes in the network in each round. To describe the network lifetime exactly, we use rounds to measure the network lifetime. LT is the round at which the first node dies in WSN.

6.2. Performance Comparisons

6.2.1. Performance Dependence on Network Scale. As wireless sensor networks consist of a large number of nodes, the broadcast protocol designed for WSN should adapt well to the large-scale network scenario. To study the influence of network scale on BALR, we simulate wireless sensor networks constituted by a different number of nodes. And nodes are randomly placed in the networks. As illustrated in Figures 5 and 6, compared with BPS, ML^{2}B, and flooding, BALR has the smallest rebroadcast ratio R without sacrificing the RE and MED for varying network sizes. When simulating flooding, we use a random delay for each node in the network before their rebroadcast to alleviate collisions, which enhance the performance of flooding. As shown in Figure 6, BALR has a smaller maximum end-to-end delay than BPS, and flooding has the smallest MED under most conditions. It is clear that the rebroadcast ratio of flooding is 1 under all conditions. Therefore, we do not show it in Figures 5 and 7.

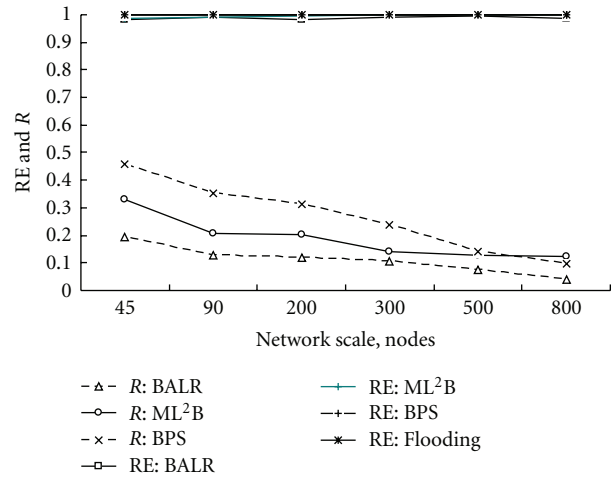


Figure 5: RE and R dependence on network scale.

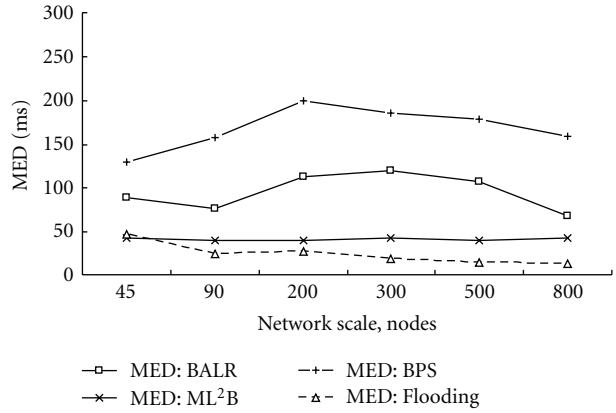


Figure 6: MED dependence on network scale.

6.2.2. Performance Dependence on Node Density. Nodes are randomly placed in the network region of 750 m $\times$ 750 m, with density varying from 5 nodes to 80 nodes per $r \times r$ region. As shown in Figure 7, BALR completes the broadcast with a satisfying coverage ratio using the least number of rebroadcasts among three protocols. In Figure 8, the maximum end-to-end delay of BALR is smaller than that of BPS. Figure 7 shows that the rebroadcast ratio R of BALR falls with the increase of node density, which guarantees the stability of BALR in high-density sensor networks.

Figure 9 shows a comparison of the rebroadcast ratio between simulation values of BALR and theoretical values in the ideal case and worst case. The rebroadcast ratio in the ideal case and worst case is computed based on formula (11) and formula (20). The influences of node density on the rebroadcast ratio of BALR under these three conditions are similar. As the rebroadcast ratio in the worst case is obtained under conditions of $d_T = 0$ and $E_T = 0$, its values are much larger than the simulation values and ideal case values. As shown in Figure 9, simulation results of the rebroadcast ratio in BALR are close to the ideal values, which prove that BALR is an efficient broadcast algorithm for wireless sensor networks.

6.2.3. Lifetime Evaluation. Each node's initial energy is 1.0 J. And the residual energy of each node decreases when it receives or transmits packets in the network. A node dies when its residual energy decreases to 0 J. As defined in Section 6.1, lifetime (LT) is the interval from the time the network is initiated to the time at which the first node dies in the network. Figure 10 shows the network lifetime of BALR, ML^{2}B, BPS, and flooding by rounds. As shown by Figure 10, due to the super redundant rebroadcast, flooding shortens network lifetime significantly. Though BPS could reduce the rebroadcast redundancy greatly by maximizing each hop length, the adopted adaptive-geometric mechanism causes that network lifetime to be independent of node density. For ML^{2}B, due to the lack of consideration of optimal induced forwarder selection, the number of nodes doing rebroadcast in the network increases when node density increases, which also can be calculated from Figure 8. Thus, when ML^{2}B is used in broadcasting, network lifetime falls slowly with node

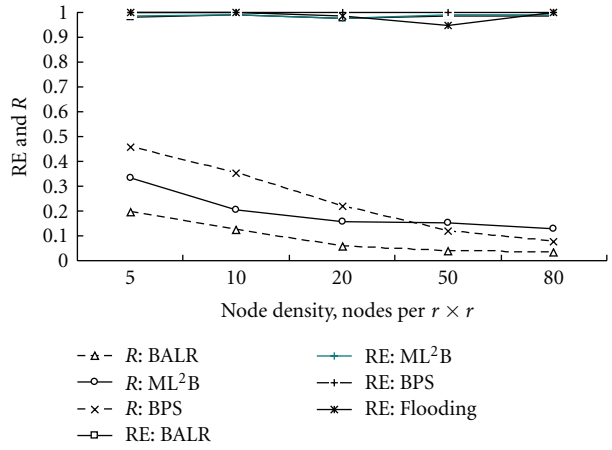


Figure 7: RE and R dependence on node density.

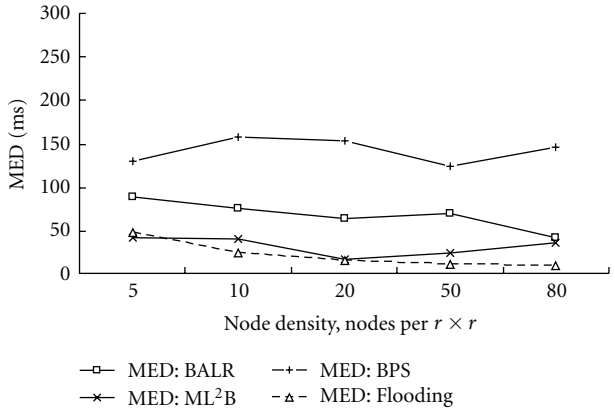


Figure 8: MED dependence on node density.

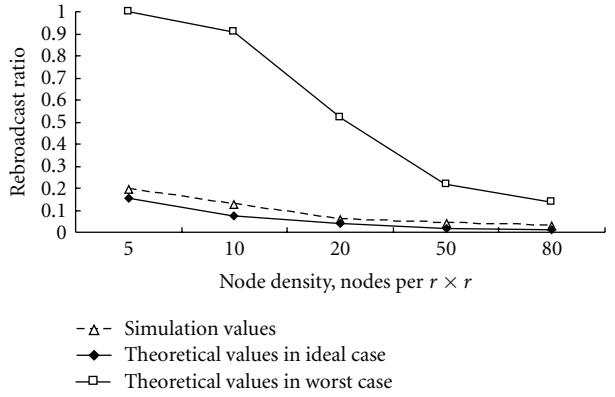


Figure 9: Simulation results versus analytical results.

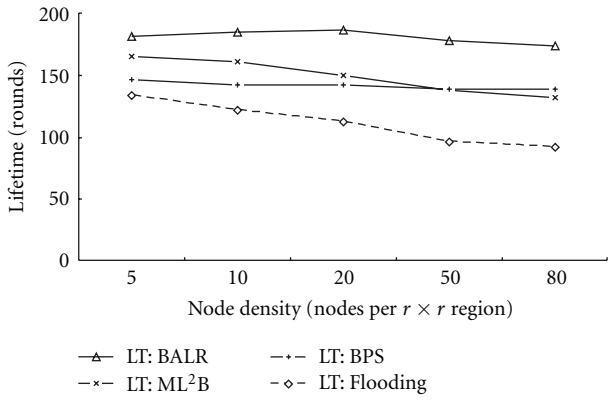


Figure 10: Lifetime of networks with different node densities.

density's increase as shown by Figure 10. Because of its least rebroadcast redundancy and energy balance consideration, BALR obtains longer network lifetime than ML^{2}B, BPS, and flooding. It prolongs the network lifetime of wireless sensor networks.

7. Conclusion

To broadcast packets efficiently and maximize the network lifetime in large-scale wireless sensor networks with densely deployed nodes, we propose a broadcast protocol BALR. It uses the coverage degree which is the number of neighbors that have not yet received the broadcasted packet of a node to measure its rebroadcast efficiency. It utilizes the geographical relationship between a rebroadcast node and its neighbors to choose as fewer new rebroadcast nodes as possible. Theoretical analysis and simulation results show that the rebroadcast ratio of BALR decreases with the increase of node density. BALR reduces the rebroadcast redundancy and prolongs the network lifetime effectively for wireless sensor networks, especially in large-scale networks with high node density. Simulation results show that the BALR strategy outperforms flooding, ML^{2}B, and BPS strategies.

Abbreviations

A: Field of packet header to contain the location of the preceding node
A_{sg}: Area of C_1
$A_{\cup}$: Area of $C_1 \cup C_2 \cup C_3$
$A_{\cap}^{i,j}$: Area of $C_i \cap C_j$ in Figure 4
a: A constant
C: Area of the entire network
C_1: Shadow region around I_1 in Figure 4
C_2: Shadow region around I_2 in Figure 4
C_3: The limited region of node U in which node S is located in Figure 4
D: Node density
$D(i)$: Self-delay of node i before rebroadcasting
D_m: Permitted maximum delay
d: Communication distance
$d(i,t)$: Coverage degree of node i at time t
d_T: Coverage degree threshold
E: Edge set of network graph
E_{Rx}: Energy expended by a node to receive a packet
E_{Tx}: Energy expended by a node to transmit a packet
E_{elect}: Energy expended by a node to run the transmitter circuitry

E': Initial maximum energy of nodes
E_T: Energy threshold
$e(i,t)$: Residual energy of node i at time t
$G(V,E)$: Network graph
g: Total number of all nodes in the network
h: Number of nodes that have rebroadcasted the packet after their reception of the packet in the network
I_b: An induced forwarder ($0 \le b \le n$)
j: Times of repeated reception of P_s
k: Data packet length in bits
LT: Network lifetime
l: Distance between node S and U
l_T: Distance threshold
$l(i,t)$: Distance from node i to the nearer ideal location at time t
MED: Maximum end-to-end delay
$NB(i)$: One-hop neighbor set of node i
n: Number of the induced forwarders of a forward node
$nb(i)$: One-hop neighbor of node i
P_s: Packet broadcasted from s
R: Rebroadcast ratio
R_i: Ideal rebroadcast ratio
R_w: Worst rebroadcast ratio
RE: Broadcast reachability
r: Radius of the coverage of each node
S: Forward node
s: Broadcast originator
t: Time
U: Preceding node of S
$U(i,t)$: Preceding node set of node i at time t
$UC(i,t)$: Uncovered set of node i at time t
$u(i,t)$: Preceding node of node i at time t
V: Node set of network graph
(x_i, y_i): Location of node i
ε_{amp}: Energy expended by transmit amplifier of a node to achieve an acceptable signal-to-noise ratio
α: Weight of importance of the location in self-delay
β: Weight of importance of residual energy in self-delay.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 51249005 and 60972153), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant no. 20106102120013; 20096102110038), and the Northwestern Polytechnical University Foundation for Fundamental Research (NPU-FFRJC201004).

References

[1] S. W. Ha, Y. K. Lee, T. H. N. Vu, Y. J. Jung, and K. H. Ryu, "An environmental monitoring system for managing spatiotemporal sensor data over sensor networks," *Sensors*, vol. 12, pp. 3997–4015, 2012.

[2] Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, "The broadcast storm problems in a mobile ad hoc network," in *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 151–162, Seattle, Wash, USA, August 1999.

[3] W. Peng and X. C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '00)*, pp. 129–130, 2000.

[4] M. T. Sun and T. H. Lai, "Location aided broadcast in wireless ad hoc network systems," in *Proceedings of the Wireless Communications and Networking Conference (WCNC '02)*, pp. 597–602, Orlando, Fla, USA, March 2002.

[5] N. B. Chang and M. Liu, "Controlled flooding search in a large network," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 436–449, 2007.

[6] J. Wu and F. Dai, "A generic distributed broadcast scheme in ad hoc wireless networks," *IEEE Transactions on Computers*, vol. 53, no. 10, pp. 1343–1354, 2004.

[7] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, pp. 585–594, Tel-Aviv, Israel, March 2000.

[8] P. J. Wan, G. Călinescu, X. Y. Li, and O. Frieder, "Minimum-energy broadcast routing in static ad hoc wireless networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1162–1171, Anchorage, Alaska, USA, April 2001.

[9] M. Cagalj, J. P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," in *Proceedings of The 8th Annual International Conference on Mobile Computing and Networking*, pp. 172–182, Atlanta, Ga, USA, September 2002.

[10] S. Agathos and E. Papapetrou, "Efficient broadcasting using packet history in mobile ad hoc networks," *IET Communications*, vol. 5, no. 15, pp. 2196–2205, 2011.

[11] R. Gandhi, A. Mishra, and S. Parthasarathy, "Minimizing broadcast latency and redundancy in ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 840–851, 2008.

[12] M. Khabbazian and V. K. Bhargava, "Efficient broadcasting in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 231–245, 2009.

[13] M. Khabbazian and V. K. Bhargava, "Localized broadcasting with guaranteed delivery and bounded transmission redundancy," *IEEE Transactions on Computers*, vol. 57, no. 8, pp. 1072–1086, 2008.

[14] K. A. M. Almahorg, S. Naik, and X. Shen, "Efficient localized protocols to compute connected dominating sets for ad hoc networks," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, usa, December 2010.

[15] M. T. Al Amin, S. Barua, S. Vhaduri, and A. Rahman, "Load aware broadcast in mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, Dresden, Germany, June 2009.

[16] S. Leu and R. S. Chang, "Simple algorithm for solving broadcast storm in mobile ad hoc network," *IET Communications*, vol. 5, no. 16, pp. 2356–2363, 2011.

[17] M. Agarwal, J. H. Cho, L. Gao, and J. Wu, "Energy efficient broadcast in wireless ad hoc networks with hitch-hiking," in

Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04), pp. 2096–2107, Hong Kong, China, March 2004.

[18] W. Z. Song, X. Y. Li, O. Frieder, and W. Z. Wang, "Localized topology control for unicast and broadcast in wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 4, pp. 321–334, 2006.

[19] F. Ingelrest and D. Simplot-Ryl, "Localized broadcast incremental power protocol for wireless ad hoc networks," in *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC '05)*, pp. 28–33, June 2005.

[20] M. Sheng, J. Li, and Y. Shi, "Relative degree adaptive flooding broadcast algorithm for Ad hoc networks," *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 216–222, 2005.

[21] R. Q. Zhao, A. J. Wen, Z. J. Liu, and P. Yue, "Maximum lifetime localized broadcast routing in MANET," in *Proceedings of the IFIP International Conference on Network and Parallel Computing (NPC '07)*, pp. 193–202, Dalian, China, August 2007.

[22] J. P. Sheu, C. S. Hsu, and Y. J. Chang, "Efficient broadcasting protocols for regular wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 6, no. 1, pp. 35–48, 2006.

[23] Y. W. Hong and A. Scaglione, "Energy-efficient broadcasting with cooperative transmissions in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 10, pp. 2844–2855, 2006.

[24] J. P. Sheu, S. C. Tu, and C. H. Yu, "A distributed query protocol in wireless sensor networks," *Wireless Personal Communications*, vol. 41, no. 4, pp. 449–464, 2007.

[25] C. R. Mann, R. O. Baldwin, J. P. Kharoufeh, and B. E. Mullins, "A trajectory-based selective broadcast query protocol for large-scale, high-density wireless sensor networks," *Telecommunication Systems*, vol. 35, no. 1-2, pp. 67–86, 2007.

[26] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient broadcast: a robust data delivery protocol for large scale sensor networks," *Wireless Networks*, vol. 11, no. 3, pp. 285–298, 2005.

[27] M. Maróti, "Directed flood-routing framework for wireless sensor networks.," in *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware (Middleware '04)*, vol. 3231 of *LNCS*, pp. 99–114, 2004.

[28] R. Q. Zhao, Z. J. Liu, and A. J. Wen, "An efficient energy-saving broadcast mechanisim for wireless sensor networks," *Chinese Journal of Electronics*', vol. 37, no. 11, pp. 2457–2462, 2009.

[29] A. Durresi, V. K. Paruchuri, S. S. Iyengar, and R. Kannan, "Optimized broadcast protocol for sensor networks," *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 1013–1024, 2005.

[30] A. Durresi and V. Paruchuri, "Broadcast protocol for energy-constrained networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 112–119, 2007.

[31] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS-33)*, p. 223, January 2000.

[32] R. Q. Zhao, "Optimized broadcast model for wireless sensor networks," *IEEE Communication Letters*. In press.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 286424, 9 pages
doi:10.1155/2012/286424

Research Article

Remote Industrial Sensor Network Monitoring Using M2M Based Ethical Sniffers

Syed Muhammad Asad Zaidi,[1] Jieun Jung,[1] Minsoo Kang,[2] Byunghun Song,[1] and Ki-Hyung Kim[3]

[1] *IoT Convergence Research Center, Korea Electronics Technology Institute (KETI), Seongnam 463-816, Republic of Korea*
[2] *RFID/USN Program, Korea Evaluation Institute of Industrial Technology, Seoul 135-080, Republic of Korea*
[3] *School of Information & Computer Engineering, Ajou University, Suwon 443-749, Republic of Korea*

Correspondence should be addressed to Syed Muhammad Asad Zaidi, zaidix@gmail.com

Received 11 May 2012; Revised 1 November 2012; Accepted 1 November 2012

Academic Editor: Chuan Foh

Diagnosing the deployed network efficiency and anomaly detection, which is an important research issue in traditional networking systems, has not been carefully addressed in industrial wireless sensor networks. Although recent wireless systems for industrial automation such as ISA100.11a employ device management protocols, these protocols generate and report a large amount of status information from individual sensor nodes. Also, these protocols do not capture influences on network performance from external sources such as malicious nodes or interference from other networks. We propose a latent network diagnosis system (LaNDS) for industrial sensor networks. LaNDS employs a packet sniffing method for efficiently evaluating network performance and instantly identifying degradation causes of networking performance. LaNDS adopts an efficient network evaluation approach for detecting abnormalities from both internal and external causes. In our proposed monitoring scenario, special sniffer devices having M2M capability (WiMAX interface) are used to monitor the industrial sensor network by employing ethical sniffing. Our approach does not incur additional traffic overhead for collecting desired information. For evaluation, we have tested LaNDS locally on an ISA100.11a based sensor network in a lab environment and have validated the efficiency of the system based on the possible erroneous cases of industrial sensor network.

1. Introduction

Wireless sensor networks (WSNs) enable users to interact with the physical environment at an unprecedented level. Their tiny, cheap, energy efficient, robust, and scalable properties have resulted in their deployment for a wide range of applications, such as military, health monitoring [1, 2], data acquisition in hazardous environments; and habitat monitoring [3–5]. Much of the research has been done on WSN in general issues like routing, reliability, QoS, energy consumption, and security, but very little work has been done targeting the in situ network diagnosis for testing operational sensor networks. Existing tools (debugger, testbed, simulation, and emulation) do not work for deployed networks, thus it is of great importance to provide network developers and administrators with some useful information on a system's working status. WSNs are by nature error prone and have unsatisfactory reliability, encountering various faults and failures during their operation. Diagnosis and evaluation of the deployed networks will enhance the applicability, reliability, and efficiency of WSNs.

While significant work on conventional network management tools exists [6], WSN counterparts have been slow to gain such efficient and useful tools. One of the main challenges for WSN anomaly detection is determining where to embed the intelligence for detecting and localizing anomalies. Another key requirement for any anomaly detection strategy is to cater for the needs and feedback of the human operator. A user-friendly detection strategy should provide several modes of notification, such as email and SMS alerts, and adapt its frequency of alerts to user feedback [7].

The next challenging problem for WSN management is not to generate a lot of control packets. For proactive approach, most of the current tools implant debugging

tools into the sensor nodes that periodically report the predefined parameters and internal status to the sink. For example, sympathy [8] actively collects run-time status from sensor nodes like routing table and flow information and detects possible faults by analyzing node status together with observed network exceptions, while, in the proposal by Zhao et al. [9], each node locally collects residual energy, link loss rate, and packet count and transmits it to sink for detailed analysis. These proactive techniques not only cause excessive computational operations on sensor nodes but also result in early depletion of the remaining energy level.

A recent review article on anomaly detection in WSNs [10] focuses on data anomalies, mainly due to security attacks, and the statistical approaches for detecting them. Because of their tight coupling to often harsh physical environments, WSNs and other networks used in extreme conditions (e.g., in space [11]) are more likely to experience anomalies related to connectivity or hardware failures than conventional networks. Recent work also focuses on devising detection strategies that target network level [8, 12], data level [13, 14], or node and data level [15, 16] anomalies.

One shortfall of the existing strategies is that none of them comprehensively addresses network, node, and data level anomalies in WSNs. Moreover, these problems are often not encountered during predeployment tests also, because the environmental conditions that trigger these problems are hard to simulate in the lab. One common difficulty is determining which metrics should be used to evaluate the health of the sensor network. Many of these sensor networks are used to report measurements about the environment, such as temperature, light level, and sound level. These nodes also have awareness of internal metrics such as processor utilization, current draw, and battery voltage level. When fielded to monitor the environment, these nodes are usually configured to sense and report on only a few of these data streams. The lack of comprehensive anomaly detection strategies for WSNs contributes to slower adoption and more frustration in deploying and maintaining these networks. Therefore, sensor networks have to be inspected in-situ on the deployment site to identify and locate failures and their causes. Determining the health of a sensor network is a difficult, yet important task. It is crucial that all sensor data reported by a sensor network is accurate so that it can be trusted by its user. Nodes may malfunction due to loss of power, extreme environmental conditions such as temperature or precipitation, or physical damage caused by falling debris or wildlife. Monitoring the health of a sensor network helps increase its trustworthiness by reporting nodes that may be malfunctioning.

In this paper, we propose a portable and user friendly diagnosis and monitoring tool "Latent Network Diagnosis System (LaNDS)" that observes, monitor and evaluates the ISA100.11a [17] based industrial sensor network using Freescale MC1322X USB dongle. Motivated from wireshark, a network protocol analyzer, our Java based application receives the ongoing transmission in the surrounding area of M2M enabled sniffer device through WiMAX interface and shows each OSI Layer parameters in a live view tab in a tree form. From this view, the user can perform deep packet inspection (DPI). Moreover there is a statistical view tab which, as the name suggests, outputs the live sniffed data in the shape of charts and graphs. From the graphs we can see the size and number of data and acknowledgment packets sent in the network, the composite packet rate, channel activity statistics and the number of packets each participating node has sent. A topology view tab shows virtual network topology showing nodes interconnected with other nodes based on their communication pattern (i.e., data packets sent to and from). Based on the output from the live view, the statistical view, and the topology view tab, we can get a quick and complete picture of the network environments. We can also identify the problematic node, inspect message contents down to the bit level, and share scenarios with vendors. Through the heedful analysis, we can keep an eye on network anomalies (loss of connectivity, intermittent connectivity, and broadcast storm) and hardware anomalies (node failure and node resets). Moreover, we have also devised ways not only to identify common security attacks like denial of service (DoS) and sinkhole attack [18, 19], but also to identify the source malicious node responsible for the attack.

The rest of this paper is organized as follows. Section 2 gives an abridged version of related work done in the past. Monitoring scenario proposed by us for efficient and large scale monitoring of industrial sensor network through LaNDS has been proposed in Section 3. Section 4 describes the design and description of our developed tool. ISA100.11a network deployed in a lab environment to test LaNDS has been explained in Section 5, while the final experimental evaluation has been done in Section 6 with special focus on common security threats and their detection scenarios. Finally we have concluded our paper in Section 7.

2. Related Work

Research on WSN monitoring and diagnosis of already deployed networks has not gained much attention while it is a critical issue as it has a direct influence on successful, efficient, and secure network operations. One of the reasons for this being neglected is that they are notoriously difficult to develop and debug. Most existing tools for WSN diagnosis are built on proactive approach, in which each sensor employs a debugging agent to collect the relevant status information and reports to the sink by periodically transmitting specific control messages which in turn decreases the network efficiency and can accelerate the energy depletion rate of sensor nodes. Some researchers propose to monitor sensor networks by scanning the residual energy [19] of each sensor node and collecting the aggregates of parameters of sensors where the in-network processing is leveraged. By collecting such information, the sink is aware of the network conditions. Periodic transmission of metrics from nodes to the sink is not a new idea. MintRoute [20] includes periodic transmission of neighbor tables to aid in debugging at the sink. However, it neither includes other metrics nor performs failure analysis at the sink.

Some debugging systems [21, 22] aim to detect and debug software failures in sensor nodes. For example, with

Clairvoyant [22], a source level debugger, a developer can wirelessly connect to a WSN and execute standard debugging commands including break, step, watch, and backtrace. But it is not enough since it does not cater for network and hardware issues. Sympathy [8] is an advanced debugging tool that detects and debugs the failures in a sensor network. Sympathy has selected metrics like neighbor list and traffic flow that enable efficient failure detection and includes an algorithm that root causes failures and localizes their sources in order to reduce overall failure notifications. It also applies an empirical decision tree to determine the most likely root causes for an observed exception.

The nucleus network management system (NMS) infrastructure helps sensor network applications export debugging and monitoring information [23]. Nucleus' support for exporting statistics and recording application metrics is not only easy to use but also to lightweight, but the limitation of NMS is that it does not provide infrastructure to analyze these metrics. Furthermore, these metrics consume more than double the RAM required for the rest of the stack.

A WSN, unlike an enterprise network, is featured by its hierarchical multilevel structures, which can hardly be approximated by the bipartite graph model used in most of the enterprise network monitoring tools, for example [24]. It is also impractical to maintain the network dependencies as stable inputs in highly dynamic and self-organized sensor networks.

3. Proposed Monitoring Mechanism

We have defined a new term called "Ethical Sniffing" in this paper. As the name suggests, we have incorporated a sniffing technique in order to monitor the network and in turn come up with detailed analysis, statistics, and graphs. At any given time, the coverage of our M2M enabled sniffer device will be limited to a small part of a sensor network due to the small radius of the sensor node radio signals; therefore our goal is to devise other ways to extend our monitoring range to a complete or partial (significant part) network area. We have suggested some of the ways how to implement these M2M based sniffer devices in order to get a wider monitoring scope of the network.

(a) *Crucial area coverage*: one approach is to put the sniffer devices only at particular areas in a network called crucial areas. We can place these devices at areas where there is high node density, where data delivery is crucial to delay and loss or where we anticipate an intermittent connectivity or data loss.

(b) *Network of sniffer devices*: we can also place multiple sniffing devices in the network haphazardly or separated by some distance and create a separate network of sniffer devices. Hence every device will keep an eye on its surrounding area and monitoring of the entire network can be done.

(c) *Mobile sniffing devices*: another approach can be employed using mobile sniffing devices that will sniff the data required for monitoring as it traverses through the network in a fixed predefined path or as directed by the network administrator through WiMAX downlink. The mobile sniffer devices can also be programmed to be active and monitor the network periodically.

(d) *Divide and monitor*: real-time active monitoring of entire network can be done efficiently by dividing the network into logical subnets such that each subnet has a dedicated mobile sniffing device. Figure 1 exhibits the scenario.

4. System Design and Description

In this section, we will briefly describe the software level system design and the user interface of our network monitoring tool "LaNDS". LaNDS is an all-in-one tool tailor made for WSN and its reliable, efficient, and user friendly output makes it quite different and unique from other existing tools of the same kind. M2M based sniffer devices exploit the wireless channel properties and send the sniffed packets to remote server having our LaNDS monitoring tool running. Multiple networks can be monitored remotely by a single server having multiple instances of LaNDS running.

LaNDS is a java based application and therefore can be run on a variety of operating systems including smart phones and tablets. Programmatic layout has been summarized in Figure 3. When the application runs, the interface is loaded by calling the "GUI" class. When the user clicks on the start button to initiate the sniffing process, "CaptureOptions" java class is triggered and a new window appears. In this window, the user is prompted to select the target channels from the radio button as shown in Figure 2. Since ISA100.11a runs on the principle of channel hopping, target channels here refer to the IEEE802.15.4 channels on which sniffing is to be performed. Upon pressing "Start," an object of sniffer class is made and "startSniffing()" process is executed. For every single channel to be sniffed, a separate Freescale MC1322x sniffer USB dongle is required and a separate agent (object) is created as "SnifferAgent". Each thread corresponding to its agent is parsed sequentially via the "initSerialReader()" function. This continues in a loop and every data or ACK packet received on each channel is read serially and the header parameters are arranged in a tree-like structure and later displayed in the live view tab of the main topology view tab. Within the loop boundary, another function "drawFigures()" is also called repeatedly whose sole purpose is to update the graphs and charts in the statistical view tab.

To be more specific, LaNDS consists of five major components: (i) serial reader, (ii) signal merger, (iii) channel manager, (iv) network manager, and (v) Security manager. Detailed system design summarizing the complete process from frame sniffing to data display stage has been shown in Figure 4.

At first, a serial reader reads sniffed packets from a single or multiple K1322-Sniffer interfaces and sends special commands to sniffing protocol for resetting the sniffer CPU, setting capturing channel and capturing mode, and so forth. After that, the signal merger receives multiple data streams

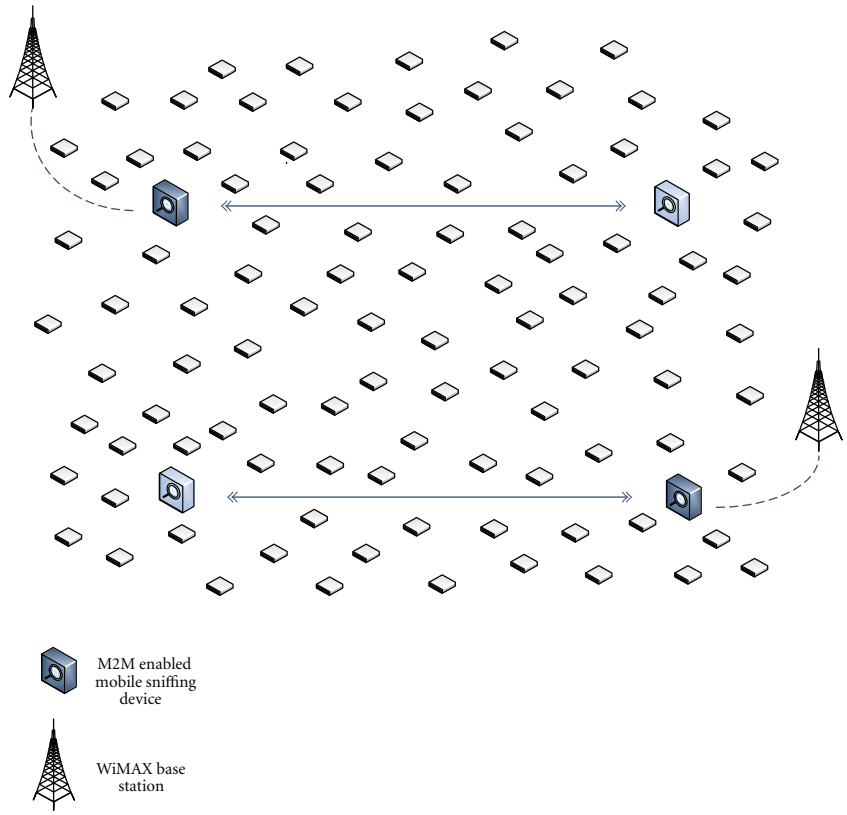


Figure 1: Divide and monitor—M2M enabled mobile sniffing devices for wide area network monitoring.

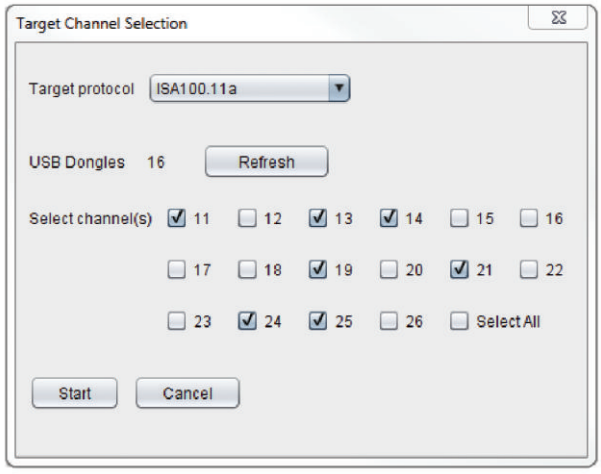


Figure 2: Target Channel Selection.

from the serial reader and combines them to serialized frames. Each frame contains captured packet from a single interface. Since K1322-Sniffer interfaces capture signals from multiple channels, correctly differentiating these signals into frames is an important task.

Network manager processes capture packets, extracts device and network information, and prepares statistical information related to network. This component also estimates efficiency of network based on collected statistical information. The parameters of network efficiency include TCP/UDP throughput, header compression efficiency, packet loss rate, packet delivery latency, and so forth. Channel manageris responsible for quantifying channel usage. ISA100.11a uses three types of hopping schemes which are (i) slotted hopping, (ii) slow hopping and (iii) hybrid hopping. Channel manager estimates how each channel is efficiently used. Channel manager also evaluates interference and intrusions, and suggests which channels should be avoided for efficient data exchange.

The purpose of Security manager is tomonitor the device abnormality and to detect various security threats. In order to provide industry-level wireless network system, ISA100.11a should guarantee robustness from security threats. Security manager in proposed framework detects denial of service attacks, channel jamming, node impersonation, and wormhole attacks and notifies the network administrator with special warning methods. After the following processes are complete, sniffed data is finally posted to the LaNDS GUI categorized as live, statistical, and network view tab. More detailed explanation along with figures is provided in Section 6.

5. Experimental Setup and Operations

For evaluation and testing purposes of our LaNDS tool, we carried out an experiment on an ISA100.11a network in

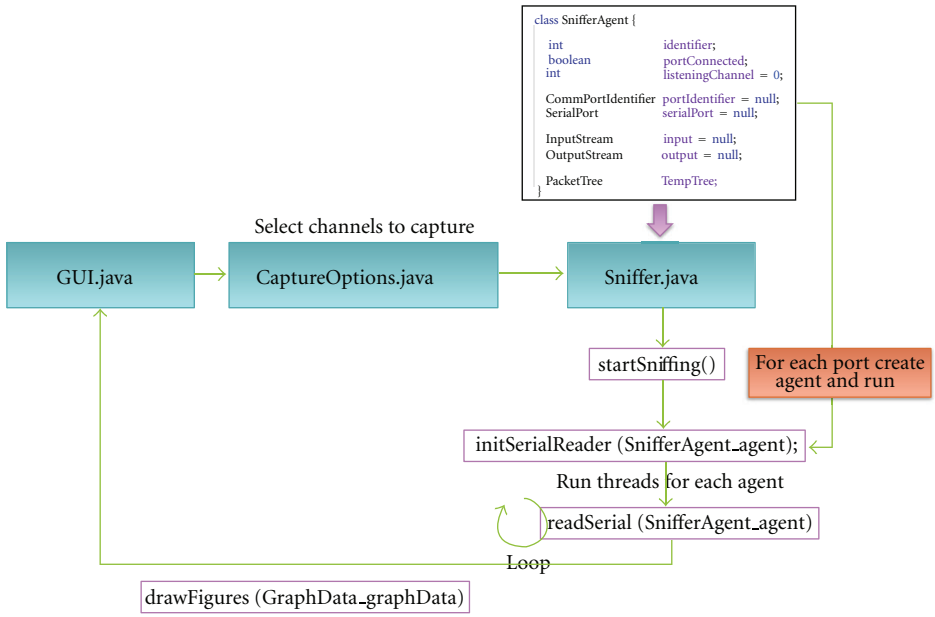


Figure 3: Sniffer class Java code Model.

a local lab environment. A group of USB dongle sniffers without M2M interface were used as sniffer device. Figure 5 shows the basic network and sniffing model. Multiple sensor nodes were connected to a gateway, while, on the other hand, a set of sniffing devices (USB dongles) each for a particular IEEE 802.15.4 channel sniffed the ongoing wireless communication between the neighboring ISA100.11a based sensor nodes. Laptop running our tool LaNDS interpreted and prepared packets after receiving the sniffed data sequentially from each sniffer data stream. Finally the network statistics were viewed on the laptop screen in three different tabs as explained in the previous section.

On the network side we used twelve ISA100.11a based sensor nodes and a gateway as shown in Figure 6. These gateway controlled nodes transmit data packets haphazardly to each other using the frequency hopping technique as per ISA100.11a standard. In our network scenario

(a) at IEEE802.15.4 layer, 16-bit short addressing has been used with PAN ID compression and security disabled;

(b) in ISA100.11a protocol;

(i) DHDR subheader specifies that slow hopping offset and DAUX subheader are included whereas signal quality in ACK is excluded.

(ii) ISA100.11a compress feature has been set to null as per DROUT sub-header;

(c) IPHC has been used in our network with IPNHC bit set, from which layer 3 header and layer 4 header are extracted.

On the sniffer side, in order to capture all channels of IEEE802.15.4, we used 16 Freescale MC1322x USB dongles mounted on a 16-port USB hub as shown in Figure 6. The USB hub was connected to a PC running LaNDS and it captured the live packets as they were being sent by the target nodes.

6. Network Diagnosis and Anomaly Detection

This section shows the output of our developed tool LaNDS when it is in the middle of the live traffic the monitoring stage. As discussed in the previous sections that, at an instance, the tool monitors a part of an already deployed WSN using an ethical sniffing technique by exploiting the intrinsic characteristics of wireless medium and, unlike other existing diagnosis and monitoring tool, it does not result in rapid energy depletion or additional control packet dissemination through the network.

The output of LaNDS is divided into three tabs each showing different representation of sniffed data. The first tab shows live packets sniffed in a tree form. Every packet is associated with the packet number and the channel number on which it was captured as shown in Figure 7. In this view, we can have a deep packet inspection by expanding the target layer header and can view the parameters and addresses present there.

The other two tabs shown in Figures 8 and 9 are more of a graphical representation of data. In the statistical view tab, there are four graphs showing network statistics (Figure 8). The first graph shows the channel utilization. A bar chart shows the number of packets sent through each channel. The second graph is a line graph showing the network efficiency; it shows the number of packets transmitted as the time progresses. Another graph on the bottom left classifies the packet according to the packet type. A pie chart shows the distribution of broadcast, multicast, and unicast packets.

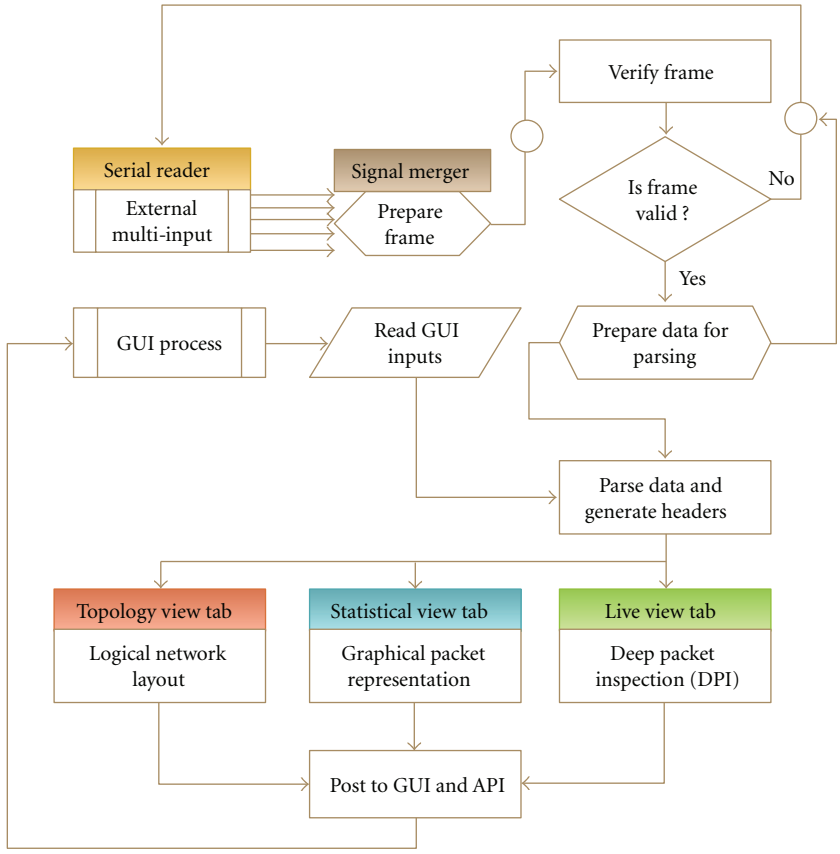


Figure 4: Flowchart of system design.

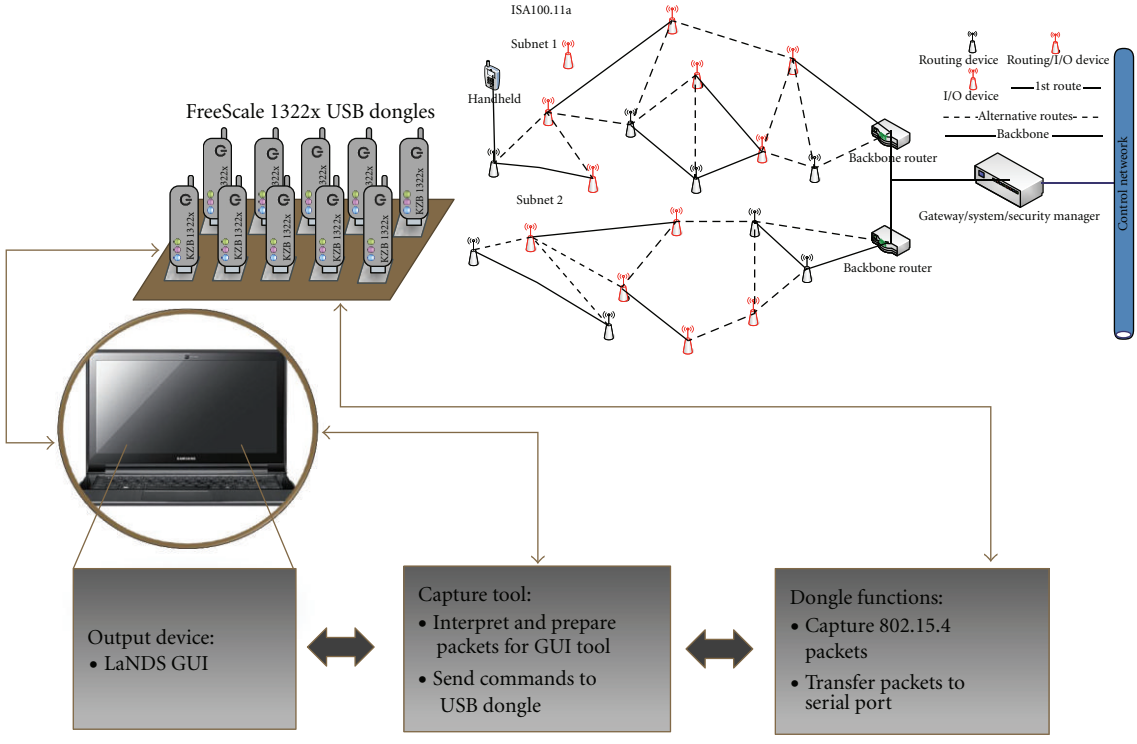


Figure 5: ISA100.11a network based monitoring scenario.

Figure 6: (a) Experimental network with eight ISA100.11a sensor nodes (b) Freescale MC1322X USB dongle hub.

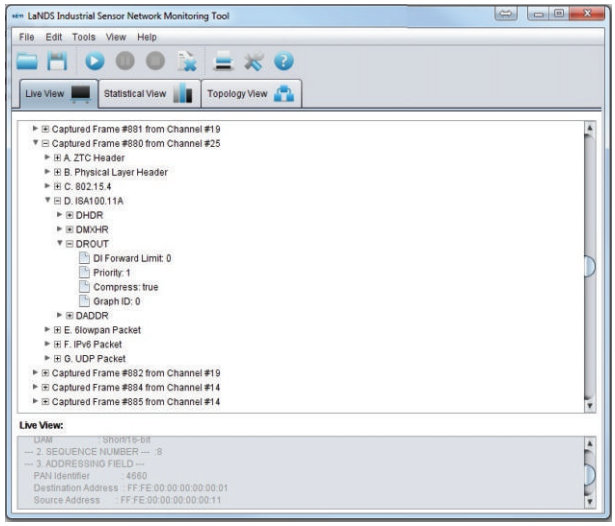


Figure 7: LaNDS Deep Packet Inspection.

Figure 8: LaNDS Statistical View Tab.

The last graph shows the packet size; each node has transmitted since the sniffing by LaNDS began.

The third tab, the network topology view tab, shows the topology view of the surrounding network. Sensor nodes sending data to each other will be interconnected with each other as shown in Figure 9. In this exemplary view, we can easily infer that the middle node is broadcasting packets to the surrounding nodes.

By studying typical protocols used in sensor networks, we found that a great deal of information about the state of the sensor network can be inferred from a message trace. For example, we can detect node failures and node reboots without modifying the protocols used in the sensor network. We can even infer routing topologies or detect the existence of network partitions without touching the sensor network.

LaNDS was originally designed for diagnosis and monitoring of WSN, but it also provides some additional benefits against detection and elimination of active security attack. Generally, if an attack is successful on a WSN, it is very difficult to identify and to eliminate the attack, but, through LaNDS, we cannot only identify the threat but also reach the source of the attack in a short time. From the security point of view, some of the protection scenarios are mentioned below.

6.1. Protection from External Malicious Nodes. Inspired by the approach of some cellular operators to maintain a list of legitimate users and implicitly prevent unauthoritative access, we have proposed to maintain a whitelist containing addresses of all devices deployed in our network, and if during sniffing, we encounter any external node (node

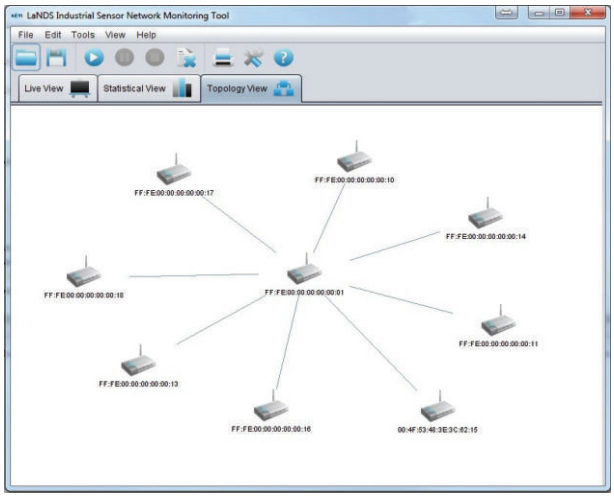


Figure 9: LaNDS Topological View Tab.

address not mentioned in the whitelist), it gets highlighted in the topology view and the next step would be to eliminate it from the network. Similarly, we have also maintained a blacklist through which nodes having a particular MAC address (known attackers) can be identified.

6.2. Protection against Internal Attack. The statistical view tab not only shows the graphical representation of the network, nodes, channels, and packet types, but we can also exploit it for detecting successful security attacks from inside nodes. Security based monitoring through LaNDS can provide the following:

(i) detecting a security attack,

(ii) identifying security attack type,

(iii) pin pointing source of attack. (graphically and via MAC address of malicious node).

Here we have shown the identification of the most common security attacks (Sinkhole attack and DoS attack). However other security attacks like selective forwarding attack, acknowledgment attack, and wormhole attack can also be identified after an acute study.

6.2.1. Blackhole/Sinkhole Attack Identification. Since Blackhole attack attracts all the traffic in the sensor network towards a single node, identification can be done in the following manner:

if number of packets with same destination/time period $\gg$ *normal packet rate*

OR

If number of packets with same destination/time period $\geq$ *threshold.*

threshold and time period can be set by the administrator and it depends on the type of network; threshold for multimedia based sensor network will be high than for temperature monitoring based sensor network.

6.2.2. DoS Attack Identification. DoS attack has severe impact on network as it tries to exhaust the resources available to the victim node by sending extra unnecessary packets and thus preventing legitimate network users from accessing services or resources. Its identification can be done through a simple approach:

if number of packets with same sender address/time period $\gg$ *normal packet rate*

OR

if number of packets with same sender address/time period $\geq$ *threshold.*

Here too, threshold and time period can be set by administrator and it depends on type of network; threshold for multimedia based sensor network will be high for temperature monitoring based sensor network.

7. Conclusions

Although much research has been done in various aspects of wireless sensor networks, little work has been done towards a diagnosis tool for monitoring the statuses of operational systems in the field. This paper proposes a new technique used for remote monitoring of already deployed industrial ISA100.11a based sensor network using LaNDS. LaNDS receives the data through the WiMAX link from M2M enabled sniffer devices and later processes it to give a detailed insight of the network. LaNDS system can be thought as a passive diagnosis tool that exploits the wireless medium characteristics, but it can be efficiently applied to any already deployed industrial sensor network system giving live traffic analysis and statistics in the form of text and figures.

What makes LaNDS system unique is that unlike other tools it does not periodically pull the status and other predefined parameters from the nodes; thus it does not result in early depletion of already scarce available energy of sensor nodes. Moreover, it does not disseminate additional control packet throughout the network, and hence it does not have adverse effects on network efficiency either. The only limitation is that it provides monitoring only to a part of a network at a time and, in order to get an overview of the entire network, we have to install multiple mobile devices which add to cost and complexity.

LaNDS is quite a user friendly tool and the output is presented in three different categories separated by tabs. We cannot only see parameters of each header in the form of tree, but different graphs in the statistical view tab tells us different aspects and hidden anomalies present in the network. Moreover it also shows a network topology view which can further assist us to monitor the live network quite efficiently.

Network monitoring using LaNDS can be further enhanced to detect active malicious attacks and to pin point the source at node level. As a result, the network once attacked can be quickly brought back to normal routine and the sensor network can be protected from prolonged internal and external attacks.

Acknowledgments

This work is supported in part by the IT R&D program of MKE/KETI (number 2011-10040909) and the Convergence ITRC (Convergence Information Technology Research Center) support program (NIPA-2012-H0401-12-1004) of MKE/NIPA.

References

[1] T. Gao, D. Greenspan, M. Welesh, R. R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *Proceedings of the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '05)*, September 2005.

[2] L. Gu, D. Jia, P. Vicaire et al., "Light weight detection and classification for wireless sensornetworks in realistic environments," in *Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems*, November 2005.

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

[4] J. Kahn, R. Katz, and K. Pister, "Next century challenges: mobile networking for smart dust," in *Proceedings of the Annual ACM International Conference on Mobile Computing MobiCom (MobiCom '99)*, August 1999.

[5] X. Zhang, M. Wei, P. Wang, and Y. Kim, "Research and implementation of security mechanism in ISA100.11a networks," in *Proceedings of the 9th International Conference on Electronic Measurement and Instruments (ICEMI '09)*, pp. 4716–4721, August 2009.

[6] M. Subramanian, *Network Management: An Introduction To Principles and Practice*, Addison-Wesley Longman Publishing Co., Inc., Boston. Mass, USA, 1999.

[7] R. Jurdak, R. Wang, O. Obst, and P. Valencia, "Wireless sensor network anomalies: diagnosis and detection strategies," in *Intelligence-Based Systems Engineering, ISRL*, vol. 10, pp. 309–325, Springer-Verlag, Berlin, Germany, 2011.

[8] N. Ramanathan, K. Chang, L. Girod, R. Kapur, E. Kohler, and D. Estrin, "Sympathy for the sensor network debugger," in *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys '05)*, pp. 255–267, 2005.

[9] J. Zhao, R. Govindan, and D. Estrin, "Residual energy scan for monitoring sensor networks," in *Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC '02)*, vol. 1, pp. 356–362, 2002.

[10] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 34–40, 2008.

[11] M. Prokopenko, P. Wang, M. Foreman, P. Valencia, D. Price, and G. Poulton, "On connectivity of reconfigurable impact networks in ageless aerospace vehicles," *Robotics and Autonomous Systems*, vol. 53, no. 1, pp. 36–58, 2005.

[12] S. Rost and H. Balakrishnan, "Memento: a health monitoring system for wireless sensor networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (Secon '06)*, pp. 575–584, Reston, Va, USA, September 2006.

[13] M. Wälchli and T. Braun, "Efficient signal processing and anomaly detection in wireless sensor networks," in *Proceedings of the European WorkShops on Applications of Evolutionary Computation (EvoWorkshops '09). LNCS*, M. Giacobini, A. Brabazon, S. Cagnoni et al., Eds., vol. 5484, pp. 81–86, Springer, Heidelberg, Germany, 2009.

[14] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[15] N. Ramanathan, L. Balzano, M. Burt et al., "Rapid deployment with confidence: calibration and fault detection in environmental sensor networks," Technical Report, UCLA CENS, Los Angeles, Calif, USA, 2006.

[16] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS '06)*, pp. 65–71, New York, NY, USA, September 2006.

[17] T. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: the format war hits the factory floor," *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, 2011.

[18] Y. Wang, G. Attebury, and B. Ramamurthy:, "A survey of security issues in wireless sensor networks," IEEE Communications Survey, 2006.

[19] J. Zhao, R. Govindan, and D. Estrin, "Residual energy scan for monitoring sensor networks," in *Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC '02)*, vol. 1, pp. 356–362, 2002.

[20] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Proceedings of the IEEE ICC Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, 2003.

[21] N. Ramanathan, K. Chang, L. Girod, R. Kapur, E. Kohler, and D. Estrin, "Sympathy for the sensor network debugger," in *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 255–267, 2005.

[22] J. Yang, M. L. Soffa, L. Selavo, and K. Whitehouse, "Clairvoyant: a comprehensive source-level debugger for wireless sensor networks," in *Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pp. 189–203, November 2007.

[23] G. Tolle and D. Culler, "SNMS: application-cooperative management for wireless sensor networks," in *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.

[24] S. Kandula, D. Katabi, and J. P. Vasseur, "Shrink: a tool for failure diagnosis in IP networks," in *Proceedings of the ACM Workshops: Conference on Computer Communications (SIGCOMM '05)*, pp. 173–178, August 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 750681, 9 pages
doi:10.1155/2012/750681

Research Article

A Routing Scheme for IPv6-Based All-IP Wireless Sensor Networks

Wang Xiaonan and Zhong Shan

Department of Computer Science and Engineering, Changshu Institute of Technology, Jiangsu, Changshu 215500, China

Correspondence should be addressed to Wang Xiaonan, wxn_2001@163.com

Received 23 March 2012; Revised 18 October 2012; Accepted 24 October 2012

Academic Editor: Deyun Gao

The paper proposes a routing scheme for IPv6-based all-IP wireless sensor networks. The paper creates the IPv6 address structure and the IPv6 address configuration algorithm for all-IP wireless sensor networks. Based on the IPv6 address structure, the paper proposes the routing algorithm in the link layer for all-IP wireless sensor networks. In the routing scheme, a sensor node stores the next channel-sampling time of its neighbor nodes. In this way, only during the next channel-sampling time, a sensor node keeps active. and during any other time it returns to sleep. Therefore, the routing performance is improved. and the power consumption is reduced. Finally, the paper discusses the reduced IPv6 protocol stack performing the routing scheme. The routing scheme's performance parameters are evaluated, including routing power consumption and delay, and the data results show the correctness and efficiency of the scheme.

1. Introduction

With the dramatic growth of the WSN (wireless sensor network) application space and the emergence of a variety of new applications, WSN is required urgently to achieve the point-to-point communication with the Internet [1]. IPv6 has both abundant address resources and steady communication performance, so it becomes the ideal solution for the point-to-point communication between WSN and the Internet. In the IPv6-based all-IP WSN, a sensor node has a globally unique IPv6 address and uses the IPv6 protocol to achieve the point-to-point communication with the Internet.

Compared to traditional WSN, all-IP WSN has more extensive applications space. For example, in the modern agriculture field, farm laborers can use the Internet to access all-IP WSN in the agricultural environment and acquire the real-time agricultural parameters for monitoring without geographical location constraint.

At present, the following key technologies on implementing IPv6-based all-IP WSN need further researches [2, 3].

Address Autoconfiguration. The IPv6 address auto-configuration is the important technical feature of IPv6, and it can configure an IPv6 address for each interface in the absence of intervention. The feature is consistent with WSN's design goals, such as self-organization and self-configuration. At present, there are still some problems in implementing the existing IPv6 address auto-configuration strategies in WSN. For example, the stateful address configuration causes a lot of control messages, and the stateless address configuration needs to perform DAD (duplicate address detection) to ensure an address's uniqueness. Therefore, it is necessary to propose an IPv6 address auto-configuration algorithm for all-IP WSN.

Routing Mechanism. The WSN architecture is different from the IPv6 one. For example, in WSN, a sensor node works as both a node and a router. Therefore, the existing WSN routing schemes are not suitable for all-IP WSN. Therefore, it needs to propose a routing algorithm for all-IP WSN.

IPv6 Protocol Stack Optimization. Due to resource constraints of WSN and the fact that IPv6 was not initially designed for embedded applications, it is necessary to reasonably reduce the IPv6 protocol stack so that the reduced IPv6 protocol stack can be implemented in all-IP WSN.

Therefore, the paper proposes a routing scheme for IPv6-based all-IP WSN, and the paper has the following contributions:

(1) the paper creates the IPv6 address structure and the IPv6 address auto-configuration algorithm for all-IP WSN,

(2) based on the proposed IPv6 address structure, the paper proposes a routing algorithm in the link layer for all-IP WSN,

(3) the IPv6 stack performing the proposed routing algorithm is optimized.

The remainder of the paper is organized as follows. In Section 2, we discuss the related work on all-IP WSN. We discuss the all-IP WSN architecture in Section 3 and the routing scheme in Section 4. The performance of the proposed scheme is analyzed in Section 5. We conclude the paper with a summary in Section 6.

2. Related Work

Reference [4] proposed an address configuration scheme for a sensor node based on location information, but the scheme was built on IPv4 and was not suitable for IPv6-based all-IP WSN. References [5, 6] proposed a reduced IPv6 stack which introduced an adaptation layer to achieve fragmentation and reassembly of an IPv6 packet. Reference [7] proposed a scheme for all-IP WSN. In the scheme, the network architecture was based on the logical grids, and the location information on sensor nodes was utilized to achieve both the address configuration and the routing discovery.

Reference [8] adopted the attributes of IEEE 802.15.4 [9] to enhance the routing performance of all-IP WSN. Border routers were used to store routing information on all sensor nodes, and the extension header, routing header, was included in each packet to achieve routing. Therefore, the fragment efficiency was reduced.

Reference [10] proposed a scheme on routing discovery and maintenance. However, the scheme only discussed how to establish routing paths reaching the destination subnet. In addition, the scheme did not analyze the relationship between the address hierarchical structure and the routing process. Reference [11] proposed a multipath routing scheme for 6LoWPAN to reduce the cost for reestablishing routing paths. In the scheme, the source node established multiple disjoint routing paths reaching the destination node through one routing discovery process and then ranked the multiple disjoint routing paths according to the link cost. The routing path with the minimum link cost was the primary path which was used to route the data. If the primary path failed, then the source node chose another routing path as the primary path to continue routing the data. The scheme effectively reduced the cost for reestablishing routing paths, but maintaining multiple disjoint routing paths increased the network resource consumption and reduced the routing performance.

The scheme [12] employed dispatch type in 6LoWPAN to determine the source/destination node of one packet, and intermediate nodes routed one packet to the next hop according to dispatch type encapsulated in the packet. Therefore, the routing delay was increased. In addition, the scheme added one header structure between the adaptation layer and the IP layer, so the transmission power was also increased. SPMIPv6 (Sensor Proxy Mobile IPv6) [13] presented the network architecture and the message formats and also evaluated the performance parameters, including the signaling cost and energy consumption. The data results showed that SPMIPv6 reduced the energy consumption significantly.

References [14–17] proposed the 6LoWPAN architecture where a sensor node only moved within a PAN (personal area network) with multiple gateway nodes. If a node's position changed, then it had to register the new position with all the gateway nodes. However, the architecture made no mention of IPv6 address configuration algorithm for all-IP WSN.

References [18, 19] proposed a scheme for all-IP WSN based on NEMO [20]. The scheme proposed the reduced IPv6 protocol stack, but it mentioned neither the IPv6 address auto-configuration algorithm nor the routing algorithm.

3. Architecture

The routing scheme is achieved in the link layer and adopts the 6LoWPAN architecture [6] where the MAC protocol is IEEE 802.15.4 [9]. IEEE 802.15.4 defines two kinds of nodes: FFD (full-function device) and RFD (reduced-function device). The scheme divides one all-IP WSN into multiple PANs, and from the routing perspective, one PAN includes three types of nodes: gateway nodes, cluster head nodes, and cluster members. Gateway nodes and cluster head nodes are FFD nodes with routing function, and cluster members are RFD nodes without routing function. Gateway nodes connect WSN to the IPv6 networks, and they communicate with each other in the multicast way through the IPv6 networks. One PAN includes only one gateway node with powerful hardware resources.

Taking into account the full integration of the WSN architecture and the IPv6 architecture, the scheme divides one PAN into multiple clusters, and each cluster has only one cluster head node. One gateway node and multiple cluster head nodes form a tree topology which is called a cluster tree, where the root node is the gateway node and intermediate/leaf nodes are the cluster head nodes. All the cluster trees in one WSN constitute the backbone routing system, as shown in Figure 1.

4. Routing Scheme

In the scheme, a node has an initial ID set by manufacturer, for example, the MAC address. The initial IDs of sensor nodes are independent of each other and unique in one all-IP WSN.

4.1. IPv6 Address Structure. According to the WSN characteristics, the following IPv6 address structure is proposed, as shown in Table 1.

Table 1: IPv6 address structure.

80 bits	16 bits	16 bits	16 bits
Global routing prefix	PAN ID	Cluster head ID	Cluster member ID
		Sensor node ID	

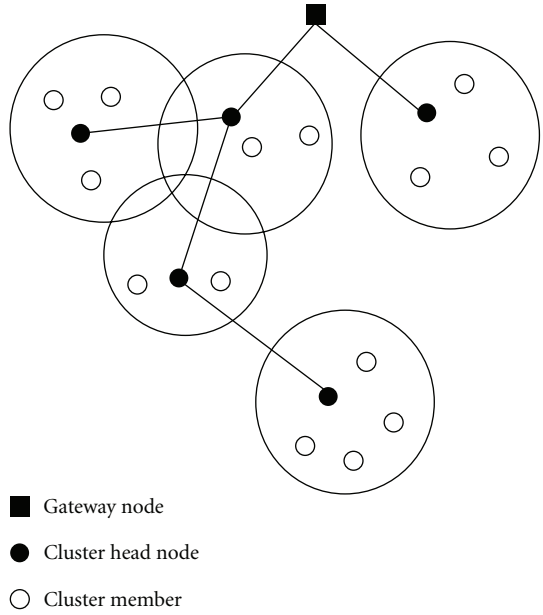


Figure 1: Cluster tree.

In Table 1, an IPv6 address is made up of two parts. The first part is global routing prefix, and global routing prefixes of all sensor nodes in one all-IP WSN are the same. The second part is sensor node ID, which is composed of PAN ID, cluster head ID and cluster member ID, and whose writing format is: PAN ID : cluster head ID: cluster member ID. PAN ID uniquely identifies one PAN which includes only one gateway node. PAN IDs of all sensor nodes within one PAN are the same, and its value is equivalent to the PAN ID of the gateway node in the same PAN, namely, the gateway node's initial ID. Cluster head ID uniquely identifies a cluster in a PAN, cluster head IDs of all cluster members in a cluster are identical, and its value is equivalent to the cluster head ID of the cluster head node in the same cluster, namely, the cluster head node's initial ID. Cluster member ID uniquely identifies a cluster member, and its value is equivalent to its initial ID.

In the scheme, the cluster head ID and cluster member ID of a gateway node are 0, and the cluster member ID of a cluster head node is 0.

Due to the resource constraints of sensor nodes, the routing scheme is achieved in the link layer. A sensor node's link address is its sensor node ID, that is, a gateway node's link address is its PAN ID, a cluster head node's link address is its cluster head ID, and a cluster member's link address is its cluster member ID.

4.2. Cluster Tree. IEEE 802.15.4 employs 3 mechanisms to achieve the data transmission [9]. These mechanisms include CSMA-CA, frame acknowledgment, and data verification. The scheme adopts the frame acknowledgment mechanism. In the frame acknowledgment mechanism, if a node is unable to handle the received frames, the frames are not acknowledged. If a node does not receive an acknowledgment frame within the specified time, it assumes that the transmission fails and retries the frame transmission. If an acknowledgment is yet not received after several retries, then the transaction is terminated.

4.2.1. Establishment of Cluster Tree. The routing scheme is built on cluster trees, and a cluster tree has the following characteristics:

(1) one cluster tree is made up of one gateway node and multiple cluster head nodes, where the root node is the gateway node, and the intermediate/leaf nodes are the cluster head nodes,

(2) one cluster tree forms one PAN which is uniquely identified by the PAN ID of the tree's root node, and PAN IDs of all cluster head nodes and cluster members in one PAN are identical,

(3) the gateway nodes are preset, and their IPv6 addresses are predetermined.

An FFD node becomes a cluster head node and acquires its sensor node ID through joining a cluster tree. Since a gateway node is preset, its neighbor FFD nodes first join the tree and become cluster head nodes through becoming the gateway node's child nodes.

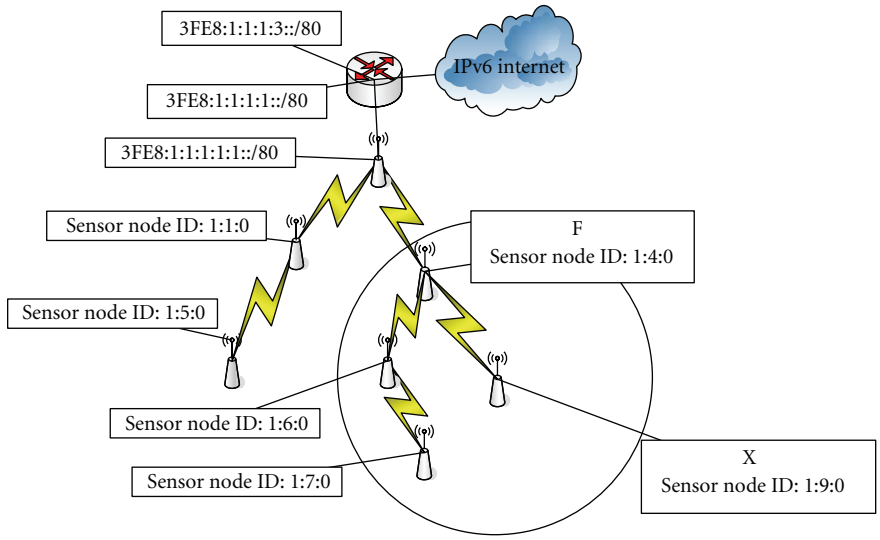


Figure 2: FFD becomes a cluster head node.

The process of an FFD sensor node X joining a cluster tree is as follows.

(1) X broadcasts a request beacon frame whose command identifier is 0×07.

(2) After X's neighbor gateway nodes/cluster head nodes receive the frame, they, respectively, return a response beacon frame whose payload includes the depth of the position in the corresponding cluster tree, the next channel-sampling time, and the PAN ID of the corresponding cluster tree.

(3) X selects the gateway node/cluster head node F with the minimum depth as its parent node and records F's cluster head ID, F's next channel-sampling time, and its own depth which is equal to F's depth plus 1. Then, X combines F's PAN ID with its initial ID to form a sensor node ID, marks itself as a cluster head node, and registers the sensor node ID with the gateway node of the cluster tree which it joins.

(4) X successfully joins a cluster tree to become a cluster head node and to obtain a sensor node ID, as shown in Figure 2.

In the scheme, a node's channel-sampling time means the time when a node awakens and can receive/send frames. In general, the number of nodes in a WSN is vast, so the topology of a cluster tree is maintained through a child node recording the sensor node ID of its parent node. In the scheme, the depth of a node's position in a tree is used as a metric to select a parent node in order to shorten the length of the routing path reaching the gateway node and reduce the routing cost and delay.

In Figure 2, the FFD node X with the initial ID 9 broadcasts a request beacon frame, and the cluster head nodes with the cluster head IDs 4, 6, and 7 return a response frame, respectively. X selects the cluster head node F with the cluster head ID 4 as its parent node and joins the cluster tree to become a cluster head node and obtain its sensor node ID.

4.2.2. Establishment of a Cluster. In the scheme, an RFD node joins a cluster to become a cluster member and acquire its sensor node ID.

After an FFD node H becomes a cluster head node, it periodically broadcasts a beacon frame whose payload includes the depth of its position in the cluster tree, the next channel-sampling time, and its PAN ID. After an RFD node X receives a beacon frame from H, it checks if it is marked as a cluster member. If not, then X marks itself as a cluster member, records the depth of H's position in the cluster tree, H's next channel-sampling time, and H's PAN ID, and combines H's PAN ID and cluster head ID with its own initial ID to obtain a sensor node ID.

From the previous cluster formation process, it can be inferred that a cluster is made up of only a cluster head node and multiple cluster members and has the star topology.

In the scheme, after a cluster member receives a beacon frame from its parent node, it updates the information on its cluster head node with the one in the frame's payload.

4.3. Failure of a Sensor Node. If a cluster head node X does not receive a beacon frame from its parent node within the specified time, then it considers that its parent node fails. In this situation, X chooses the neighbor cluster head node F with the minimum depth as its parent node, combines F's PAN ID with its initial ID to acquire a new sensor node ID, and records the depth of its position in the cluster tree. If X's sensor node ID/depth value is different from its original one, then it registers the sensor node ID with the gateway node, of the corresponding cluster tree.

A node acquires its sensor node ID from a neighbor node, as shown in Section 4.2. When a parent node fails, its child nodes choose the neighbor cluster head node with the minimum depth as a new parent node. If the velocity of convergence is not fast enough, then it means that the failed parent node is far from the leaf nodes which cannot be its child nodes' neighbor nodes. Therefore, the child nodes

cannot select the left node as its new parent node, and the routing loops are avoided.

If a cluster member X does not receive a beacon frame from its cluster head node within the specified time, then it considers that its cluster head node fails. Therefore, X chooses the neighbor cluster head node H with the minimum depth as its cluster head node and combines H's PAN ID and cluster head ID with its initial ID to acquire a new sensor node ID.

4.4. Registration. A gateway node maintains a cluster head node table which records the information on the cluster head nodes in the same cluster tree, and the information includes a cluster head node's cluster node ID and the depth of its position in the cluster tree.

The process of a cluster head node X registering a sensor node ID with the corresponding gateway node G is as follows.

(1) X sends G a Reg command frame whose payload includes its sensor node ID and the depth of its position in the cluster tree.

(2) After G receives the Reg frame, it checks if there is X's record in its cluster head node table. If there is, then G updates the depth value with the depth value in the frame. Otherwise, G adds a record into the table to store X's sensor node ID and depth value.

4.5. Routing

4.5.1. Routing Table. In the scheme, a gateway node/cluster head node stores a temporary routing table, and each routing entry consists of four fields: destination cluster, next hop, next channel-sampling time, and life time. Among them, the destination cluster records the sensor node ID of the destination cluster head node, the next hop field records the sensor node ID of the next hop reaching the destination cluster, the next channel-sampling time records the next channel-sampling time of the next hop, and its value is periodically updated with the one in the beacon frame from the next hop, and the life time field records the life survival time of the corresponding routing entry and automatically attenuates with the machine clock. When the life time is equivalent to 0, the corresponding entry is automatically removed from the routing table. Each time one routing table entry is used, its life time is set to the initial value, namely, the maximum life time.

4.5.2. Routing Process. An IPv6 node N uses the IPv6 address of a cluster member X to request the collected data, and the routing process is as follows.

(1) N sends a request packet to X.

(2) Through the IPv6 networks, the packet is routed to the gateway node G which identifies the cluster tree where X locates.

(3) G checks if there is a routing entry reaching the destination cluster in its routing table. If not, G establishes a routing path reaching the destination cluster head node H.

(4) G performs the fragmentation of the request packet, encapsulates each fragment with the mesh delivery header, where the source address is G's PAN ID and the destination address is X's sensor node ID, and the IEEE802.15.4 header, where the source address is G's PAN ID and the destination address is the cluster head ID of the next hop in the corresponding routing entry, and sends the frames.

(5) After the next hop receives the frames, it sends the frames to the next hop in the corresponding routing entry. In this way, the frames finally reach H which then forwards the frames to X.

(6) After X receives all the frames, it reassembles the fragments into the request packet, performs the fragmentation of the response packet, encapsulates each fragment with the IEEE802.15.4 header, where the source address is X's cluster member ID and the destination address is H's cluster head ID, and then sends the frames at H's next channel-sampling time. After H receives the frames, it sends the frames to its parent node. In this way, the frames finally reach G.

(7) G reassembles the fragments into the response packet and then sends the response packet to the IPv6 network, where the response packet reaches N in the IPv6 routing way, as shown in Figures 3(a) and 3(b).

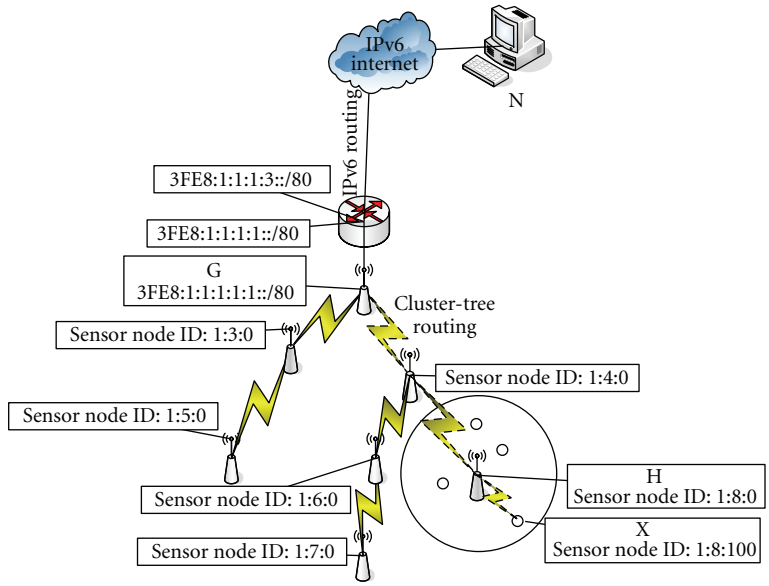
In Figures 3(a) and 3(b), the IPv6 node N sends a request packet whose destination address is 3fe8 : 1 : 1 : 1 : 1 : 1 : 8 : 100/80. The packet first reaches the gateway node G. G establishes the routing path reaching the destination cluster head node H and then routes the corresponding frames to H through the routing path. Finally, H forwards the frames to the destination cluster member X. After X deals with the request packet, it sends the response frames to G which routes the corresponding response packet to N.

In the scheme, a cluster head node periodically sends a beacon frame. If a cluster head node receives a beacon frame from the next hop, it updates the next channel-sampling time field of the next hop. Since a node only communicates with the next hop, only during the next channel-sampling time of the next hop it keeps active in order to send/receive the frames, and during any other time, it returns to sleep. In this way, the power consumption is reduced.

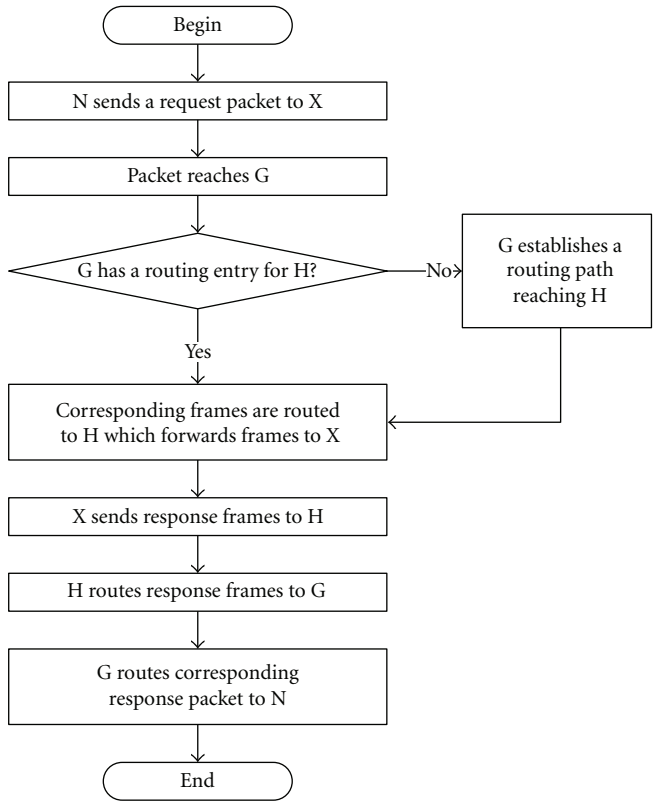
4.6. Data Frame. In IEEE 802.15.4, the command frame and data frame are defined by the frame type of the frame control. For example, if the frame type is 001/011, then the frame is the data frame/command frame. The IEEE802.15.4 data frame adopted by the scheme is shown in Table 2.

Routing of a data frame is performed through a cluster tree. The PAN ID of the source address of a data frame is the same as the one of the destination address of the frame, so the link address is 16-bit short address. If the next hop of a data frame is a cluster head node, then the destination address is the cluster head ID of the next hop. If the next hop of a data frame is a cluster member, then the destination address is the cluster member ID of the next hop.

The MAC payload format adopted by the scheme is shown in Table 3.



(a)



(b)

Figure 3: Routing process.

Table 2: Data frame.

2 bytes	1 byte	2 bytes	2 bytes	*n* bytes	2 bytes
Frame control	Sequence number	Destination address	Source address	MAC payload	Frame check sequence

Table 3: MAC payload format.

Mesh delivery header	Fragment header	Compression control header	IPv6 header and payload

Table 4: Mesh delivery header format.

10	O	F	Hop limit	Source address	Destination address
2 bits	1 bit	1 bit	4 bits	2 bytes	8 bytes

Table 5: First fragment header.

11000	Datagram size	Datagram tag
5 bits	11 bits	16 bits

Table 6: Subsequent fragment header.

11100	Datagram size	Datagram tag	Datagram offset
5 bits	11 bits	16 bits	8 bits

Table 7: Compression control header.

HC1	HC2
4 bits	4 bits

Table 8: HC1.

TF	Next header	HC2
1 bit	2 bits(01)	1 bit

Table 9: HC2.

Length	RSV
1 bit	3 bit

The mesh delivery header format is shown in Table 4.

In Table 4, O/F defines the type of the source/destination address. If O/F is 0, then the source/destination address is an EUI-64 address, or it is a short 16-bit address. The hop limit is the depth of the destination cluster head node's position in its cluster tree, and it decreases by 1 with one hop. The source address is the PAN ID of the gateway node of the tree where the destination cluster locates, and the destination address is the sensor node ID of the destination cluster member.

The fragment header includes the first fragment header format and the subsequent fragment header format, as shown in Tables 5 and 6 [6].

In Tables 5 and 6, the datagram size is the total size of an IPv6 packet. The datagram tag uniquely identifies an IPv6 packet, and the datagram tags of all fragments of an IPv6 packet are identical. The datagram offset is the offset value from the first fragment. The datagram offset of the first fragment is 0, so it is omitted from the first fragment format.

The compression control header is shown in Table 7.

In Table 7, HC1 specifies the way the IPv6 header is compressed, as shown in Table 8, and HC2 specifies the way the IPv6 payload is compressed, as shown in Table 9.

In Table 8, TF defines the compression method for the traffic class and flow label. Next header specifies the type of the next header, and 01 means UDP. HC2 determines the type of HC2 followed.

In Table 9, length determines the computing method for the compressed UDP length.

To sum up, the differences between the structure of the adaptation layer in the scheme and the one in 6LoWPAN are as follows.

(1) In the scheme, only the first fragment includes the IPv6 header and other fragments do not include the IPv6 header.

(2) The size of the control information in the scheme is reduced.

Therefore, the scheme reduces the transmission power consumption.

4.7. Security Consideration. The proposed scheme adopts the public-key infrastructure for 6LoWPAN [21] to achieve the security.

5. Performance Evaluation

5.1. Routing Analysis. The existing routing schemes [18, 19] are achieved in the network layer, while the proposed routing scheme is performed in the link layer. Therefore, in the existing routing schemes, the IPv6 header in each frame includes the 128-bit source/destination IPv6 address. In the proposed routing scheme, each frame contains 16-bit source link address and 64-bit destination link address. Therefore, for each frame the size of the redundant data in the existing schemes is 178 bits. It is assumed that the power consumed by transmitting one 127-byte frame between two neighbor nodes is e, and the distance from the destination cluster head to the root node of the corresponding tree is d. The power consumed by processing a frame is less by several orders of magnitude than the one by transmitting a frame [22], so it is negligible. Thus, the redundant power E consumed by transmitting one frame is shown in

$$E = \frac{(64 + 112) \cdot e \cdot d}{127 \times 8} = 0.175 \cdot e \cdot d. \tag{1}$$

In formula (1), e can be calculated from formulas (2), where $E_T(k, r)$ is the total power consumed by sending k bits, E_{tx} is the power consumed by sending 1 bit, ε is the magnification of the signal amplifier, r is the distance from the sending node to the receiving node, $E_R(k)$ is the total power consumed by receiving k bits, and E_{rx} is the power consumed by receiving 1 bit. Consider,

$$\begin{aligned} e &= E_T(k, r) + E_R(k), \\ E_T(k, r) &= k(E_{tx} + \varepsilon r^2), \\ E_R(k) &= kE_{rx}. \end{aligned} \tag{2}$$

According to references [23, 24], E_{tx} and E_{rx} are set to 50 nJ/bit, ε is set to 10 pJ/b/m^2, and k is set to 1016. It is assumed that FFD nodes are distributed evenly and r is set to 20 m. Then, Figure 4 is acquired.

It is assumed that the delay taken by transmitting 1 bit between two neighbor nodes is t'. Then, the redundant delay D taken by transmitting one frame is show in

$$D = \sum_{d}(64 + 112) \cdot t' = 178 \cdot \sum_{d} t'. \tag{3}$$

When t' is set to 4 μs, Figure 5 is acquired.

Figure 4: Redundant power consumption.

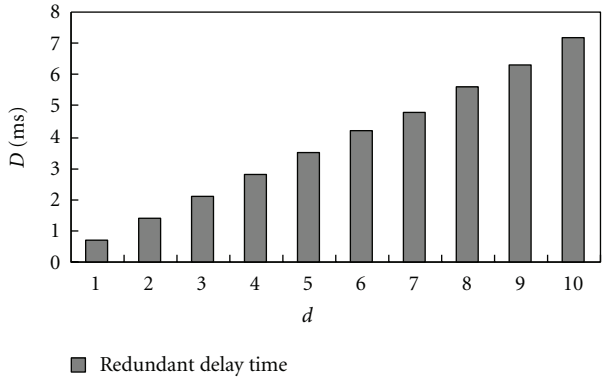


Figure 5: Redundant delay.

5.2. Routing Simulation. In ns-2, the simulation region is $100\times100\ m^2$, and the region includes 4 IPv6 ingress gateways, 20 FFD nodes, and 100 RFD nodes. The MAC protocol adopts IEEE 802.15.4, an FFD/RFD node's communication range is 20 m, and the bandwidth of WSN is 250Kbps. The FFD nodes are distributed uniformly around the simulation area, and the RFD nodes are distributed randomly around the simulation region.

We select the existing scheme [18] to compare with the proposed scheme due to the following reasons:

(1) the existing scheme is a routing scheme for IPv6-based all-IP WSN,

(2) the existing scheme has better performance than the typical routing protocols, such as AODV and LOAD.

The goal of the scheme is to reduce the routing cost and delay, so we evaluate the routing cost and delay, as shown in Figures 6 and 7.

The results of Figures 6 and 7 are analyzed as follows.

(1) The proposed routing scheme is achieved in the link layer, so the routing performance is better and the packet loss rate is lower.

(2) Only the first fragment includes the IPv6 header, and other fragments do not include the IPv6 header, so the fragment utilization is improved. As a result, the routing performance is improved.

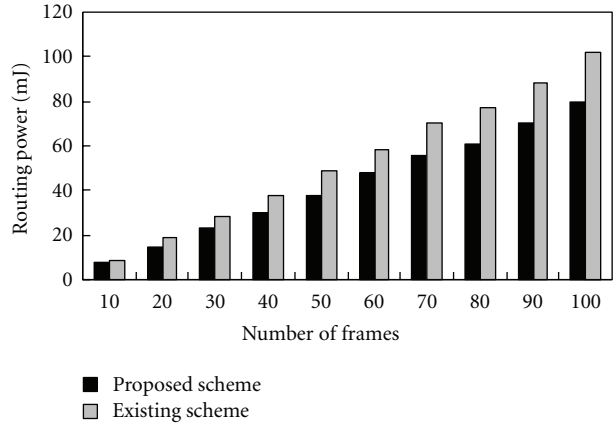


Figure 6: Routing power.

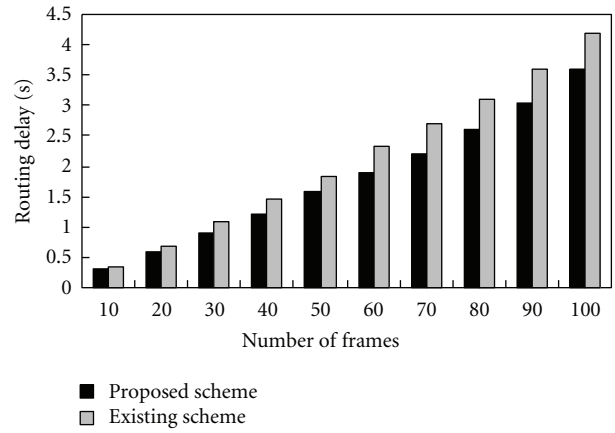


Figure 7: Routing delay.

(3) In the proposed scheme, the IPv6 stack is optimized and the size of the control information is reduced, so the transmission power consumption is reduced.

6. Conclusion

This paper proposes a routing scheme in the link layer for all-IP WSN. The paper evaluated the proposed scheme's performance parameters, including the routing power consumption and the routing delay, and the data results show the efficiency of the proposed scheme.

In the scheme, if a cluster head node near a gateway node fails, then the sensor nodes in the corresponding branch have to acquire new sensor node IDs and register them with the gateway. Therefore, the scalability is limited to some extent. In our future works, we plan to overcome the deficiency and improve the scalability.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61202440).

References

[1] A. Dunkels and J. P. Vasseur, "IP for smart objects," IPSO Alliance White Paper no. 1, september 2008.

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

[3] S. Dai, X. Jing, and L. Li, "Research and analysis on routing protocols for wireless sensor networks," in *Proceedings of the International Conference on Communications, Circuits and Systems*, pp. 27–30, New York, NY, USA, May 2005.

[4] A. Dunkels J Alonso T Voigt, "Making TCP/IP viable for wireless sensor networks," in *Proceedings of the 1st European Workshop on Wireless Sensor Networks*, Swedish Institute of Computer Science, Sweden, 2004.

[5] N. Kushalnagar, G. Montenegro, and C. Schumacher, "6LoWPAN: overview, assumptions, problem statement, and goals," IETF RFC, 4919, 2007.

[6] G. Montenegro, N. Kushalnagar, and J. Hui, "Transmission of IPv6 packets over IEEE802," 15.4 Networks. IETF RFC, 4944, 2007.

[7] W. Xiaonan and Z. Shan, "All-IP communication between wireless sensor networks and IPv6 networks based on location information," *Computer Standards & Interfaces*, vol. 35, no. 1, pp. 65–77, 2013.

[8] J. W. Hui and D. E. Culler, "IP is dead, Long Live IP for wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 15–28, ACM Press, New York, NY, USA, 2008.

[9] IEEE Computer Society, "Part 15. 4: wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for low-rate Wireless Personal Area Networks (WPANs)," IEEE Standard 802.15.4, 2007.

[10] T. Winter and P. Thubert, "RPL: IPv6 routing protocol for low power and lossy networks: draft-ietf-roll-rpl-07," IETF, 2010.

[11] J.-M. Chang, H.-Y. Yang, and H.-C. Chao, "Multipath design for 6LoWPAN ad hoc on-demand distance vector routing," *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 1, pp. 24–40, 2010.

[12] G. Bag, M. T. Raza, K. H. Kim, and S. W. Yoo, "LoWMob: Intra-PAN mobility support schemes for 6LoWPAN," *Sensors*, vol. 9, no. 7, pp. 5844–5877, 2009.

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

[14] G. Bag, M. T. Raza, H. Mukhtar et al., "Energy-aware and bandwidth-efficient mobility architecture for 6L0WPAN," in *Proceedings of the IEEE Military Communications Conference, (MILCOM '08)*, pp. 1–7, San Diego, Calif, USA, November 2008.

[15] G. Bag, S. M. S. Shams, A. H. Akhbar, M. T. Raza, K. H. Kim, and S. W. Yoo, "Network assisted mobility support for 6LoWPAN," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference, (CCNC '09)*, Las Vegas, Nev, USA, January 2009.

[16] G. Bag, S. M. S. Shams, H. Mukhtar, K. H. Kim, and S. W. Yoo, "Inter-PAN mobility support for 6L0WPAN," in *Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology, (ICCIT '08)*, pp. 787–792, Busan, Korea, November 2008.

[17] M.-K. Shin and H.-J. Kim, "L3 mobility support in large-scale IP-based sensor networks (6LoWPAN)," in *Proceedings of the 11th International Conference on Advanced Communication Technology*, pp. 941–945, IEEE press, New York, NY, USA.

[18] H. Kim and C. hong, "A routing scheme for supporting network mobility of sensor network based on 6LoWPAN," in *Proceedings of the Asia-Pacific Network Operations and Management Symposium*, pp. 155–164, Springer Press, Berlin, Germany, 2007.

[19] H. Kim, C. hong, and T. Shon, "A lightweight NEMO protocol to support 6LoWPAN," *ETRI Journal*, vol. 30, no. 5, pp. 685–695, 2008.

[20] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility(NEMO) basic support protocol," IETF RFC, 3963, 2005.

[21] M. Hasan, A. H. Akbar, R. Riaz et al., "Key management in IP-based ubiquitous sensor networks: issues, challenges and solutions," in *Proceedings of the 1st International Conference of Ubiquitous Information Technology (ICUT '07)*, Dubai, United Arab Emirates, February 2007.

[22] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[23] R. Zhang, L. Zhang, and Y. Feng, "Very low energy consumption wireless sensor localization for danger environments with single mobile anchor node," *Wireless Personal Communications*, vol. 47, no. 4, pp. 497–521, 2008.

[24] E. M. Royer and C. K. Toh, "Review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, pp. 46–55, 1999.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 268478, 10 pages
doi:10.1155/2012/268478

Review Article

Security and Vulnerability of SCADA Systems over IP-Based Wireless Sensor Networks

HyungJun Kim

Division of Information Technology, Department of Computer Engineering, Hansei University, 604-5 Dangjung-dong, Gunpo, Republic of Korea

Correspondence should be addressed to HyungJun Kim, harry@hansei.ac.kr

Received 15 June 2012; Revised 9 October 2012; Accepted 24 October 2012

Academic Editor: Deyun Gao

An overview of supervisory control and data acquisition (SCADA) systems is presented and relevant security concerns are addressed. To secure IP-based SCADA systems, it is vital to implement secure architectures which prevent access to the SCADA from corporate IT, in addition to excellent management practices in managing IP-based networks. We give an overall explanation of SCADA systems along with their security concerns and vulnerability. In the case of legacy SCADA systems, the concerns regarding security were minimal since it used proprietary networks; however, since the move of SCADA systems to the Internet, security problems have become an imperative issue, especially problems regarding IP-based systems. We also discuss various forms of threats and vulnerabilities on future SCADA systems applied with IPv6 over low-power, wireless personal area networks (6LoWPANs) and suggest security management methods to overcome these concerns. In order to identify and remove various vulnerabilities and threats to SCADA systems over IP-based wireless sensor networks, especially for 6LoWPAN applications, we examine possible threats and propose proper and precise security management methods.

1. Introduction

A supervisory control and data acquisition (SCADA) system refers to an industrial control system (ICS); it is a common process automation system which is used to gather data from sensors and instruments located at remote sites and to transmit data at a central site for either controlling or monitoring purposes [1]. The collected data is usually viewed on one or more SCADA host computers located at a central or master site. Based on information received from remote stations, automated or operator-driven supervisory commands can be transmitted to remote station control devices, which are often referred to as field devices. SCADA systems are used to monitor and control industrial, infrastructure, or facility-based processes such as telecommunications, water control, oil and gas refining, and transportation. SCADA systems can be relatively simple, such as the one that monitors environmental conditions of a small office building, or incredibly complex, such as a system that monitors all the activity in a nuclear power plant.

Generally, a SCADA system consists of the following components. First, analytical instruments that sense process variables and operating equipment connected to instruments. Second, one or more field data interface devices, usually remote terminal units (RTUs), or programmable logic controllers (PLCs). RTUs connect to sensors in the process, converting sensor signals to digital data and sending digital data to the supervisory system. PLCs are used as field devices because they are more economical, versatile, flexible, and configurable than special-purpose RTUs. Third, communication infrastructure used to transfer data between field data interface devices and control units and the supervisory system. The infrastructure can be radio, telephones, cables, satellites, and so forth, or any combination of these. Fourth, host computers are the center of human monitoring and control of the processe—storing databases and displaying statistical control charts and reports. Host computers are also known as master terminal units (MTUs) or the SCADA server. Fifth, a human-machine interface (HMI) is the apparatus or device which presents process data to a human

operator, and through this, the human operator monitors and controls the process. Nowadays, the distinctions between SCADA and the distributed control system (DCS) have mostly disappeared since each has adopted the strengths of the other as networking infrastructure with higher capacity has become available. Figure 1 shows an example of an integrated SCADA system.

SCADA systems were designed when concerns for information security measures and protocols were minimal in 1960s. In the past, the systems operated in an isolated environment and relied mainly on proprietary software, hardware, and communication technology. Current SCADA systems are distributed, networked, and dependent on open protocols for the internet, which make them vulnerable to remote cyber terrorism. They are particularly vulnerable to unauthorized access, and we have examined a typical vulnerability and corresponding measures for security and present an example of concrete measures for the security of mass transportation as a critical infrastructure [2]. In the following section, we discuss the key security issues during the development of SCADA systems. A review of SCADA systems evolution allows us to better comprehend various security concerns that exist.

2. Vulnerability of SCADA Systems

In the past, when SCADA systems were independent and vendor-controlled systems with no connections to other systems and when the network protocol was proprietary, only a few people, such as developers and hackers, knew of the existence of SCADA installations. However, the present SCADA systems are widely distributed and networked. Since the systems are dependent on open protocols for the internet, they are vulnerable to external remote cyber threats as discussed in [2]. SCADA systems are different from general information systems in terms of security management. In the risk and security management of general information systems, after analyzing the assets, threats, and vulnerabilities of information systems and calculating the degrees of a risk, security measures are prioritized for calculating the remaining risk. In contrast, for SCADA systems, the analysis of the assets is performed not from the viewpoint of systems but from the viewpoint of target facilities managed and operated.

There are two distinct threats that can affect modern SCADA systems. The first one is the threat of unauthorized access to the control software, whether it is human access or changes made deliberately or unintentionally by virus infections and other software threats existing on the control host machine. The second is the threat of packet access to the network segments hosting SCADA devices. In particular, security researchers are concerned about security and authentication in the design, deployment, and operation of some existing SCADA networks. Moreover, they need to also take into consideration whether the SCADA networks are secured just because they are physically disconnected from the internet. In addition, security researchers are also concerned about the existing security and authentication protocols in the design, deployment, and operation of SCADA networks, with the belief that SCADA systems have the benefit of security through obscurity through the use of specialized protocols and proprietary interfaces.

The following list suggests ways to help protect the SCADA network in conjunction with the corporate network as discussed in [3]. Security measures of SCADA systems in terms of technology can be presented as follows: (1) strict limitations and authority control are needed for external connections, (2) reinforced security for the systems in demilitarized zones (DMZs) as well as for the internal network is recommended, (3) enhancing security using virtual private networks (VPNs) in addition to integrity tools of servers, (4) Minimization of access paths to the internal network and enhanced concentration of monitoring, (5) encryption of emails and locking of files and directories, (6) regular and thorough inspection of security and vulnerability, and (7) developing control and monitoring methods to cope with any contingencies in the SCADA equipment.

3. IP-Based SCADA Systems

One of the main reasons why the Internet Protocol (IP) is enormously successful is that it can be used over virtually any physical media. In complex SCADA architectures, there is a variety of both wired and wireless media and protocols involved in getting data back to the central monitoring site. This allows implementation of strong IP-based SCADA networks over mixed cellular, satellite, and landline systems. SCADA communications can employ various ranges of both wired (telephone lines, optical fibers, ADSL, cables) and wireless media (radio, spread spectrum, cellular, WLAN, or satellite). The choice depends on a number of factors that characterize the existing communication infrastructure. Factors such as existing equipment, connections, available communications at isolated sites, data rates and polling frequency, remoteness of site, installation budget, and ability to accommodate future needs all impact the final decision for SCADA architecture.

A major enhancement in new SCADA systems comes from the use of WAN protocols such as the Internet Protocol for communication between the central station and communications equipment. RTUs can communicate with the master station using an Ethernet connection [1]. Figure 2 shows a networked SCADA system. Another advantage brought about by the distribution of SCADA functionality over a WAN is that of disaster survivability. By distributing the processing across physically separated locations, it becomes feasible to build a SCADA system that can survive a total loss at any one location. Many of the traditional utility devices such as RTUs or even relays are today equipped with Ethernet interfaces. This, however, does not imply that all services can be migrated immediately in a plug-and-play manner to an IP-based communication infrastructure. Differential protection services are known as one of the most delicate applications.

Legacy SCADA system components may still work as initially designed. However, new operational and business processes often require new, higher-level functionality not

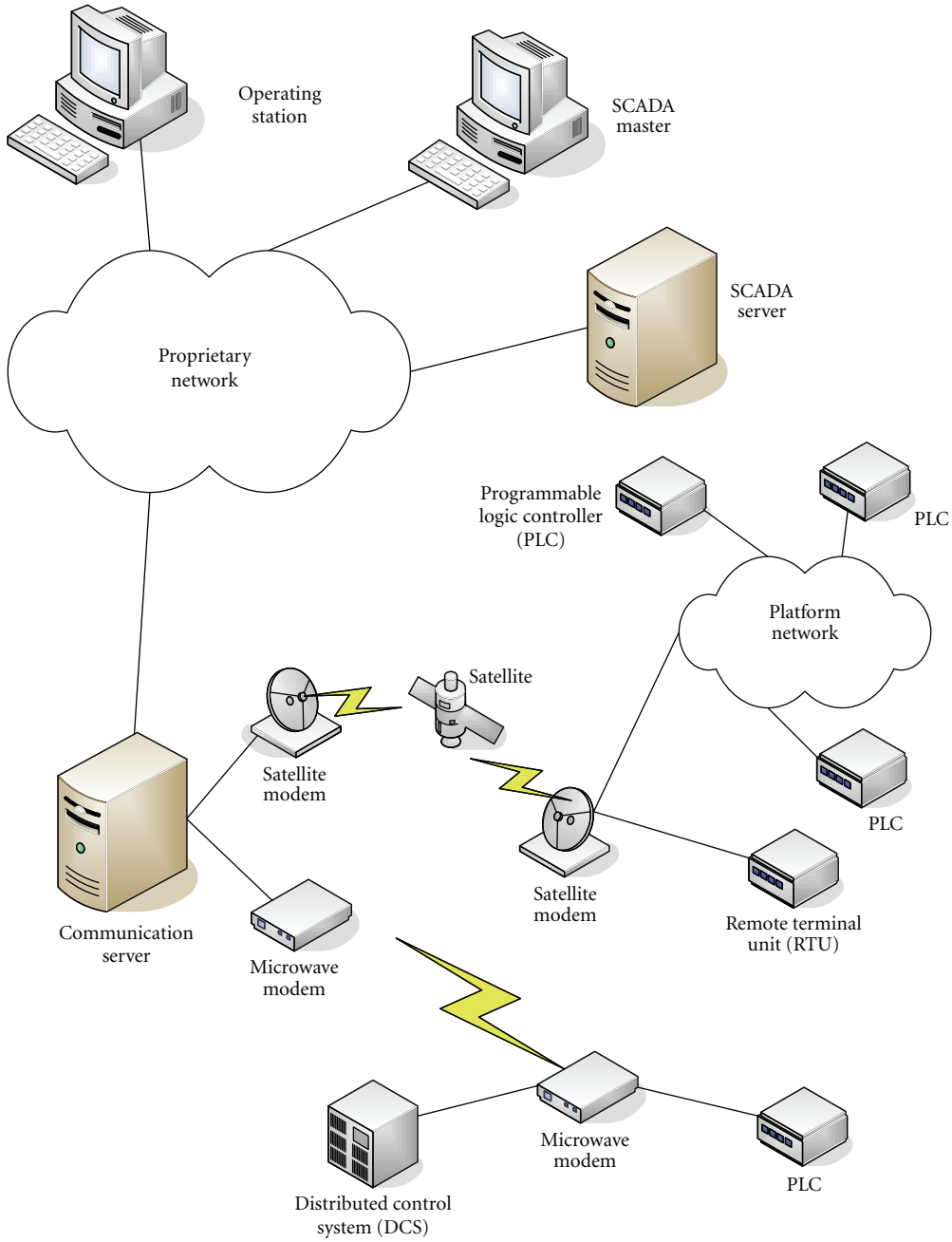


Figure 1: Integrated SCADA system.

included in the original components. Such extensions, including new physical and logical communication network connections, bear additional risks in terms of cyber security. SCADA systems were traditionally walled off from business systems and operated independently via the operational network only. Prior to the awareness of the risk of possible attacks, this seemed to provide all the protection the SCADA system needed. Their often proprietary character (operating systems, protocols, etc.) were often seen as additional safety assurance.

To run SCADA information over an IP network, various issues have to be considered such as operating equipment types, bandwidth used for SCADA center communication, network redundancy criteria and protection schemes, restoration times in case of failures, and other IP services within the network. There are several relevant advantages brought by IP technology. These advantages include the efficient use of bandwidth to avoid the allocation of capacity where it is not necessary, widely accepted standards based on proven technologies and a high degree of interoperability. Also, reliability is enhanced because in IP networks, packets are instantly rerouted if a node or link fails. Other related advantages are scalability to cope with growth, high degree of freedom to evolve network performance according to

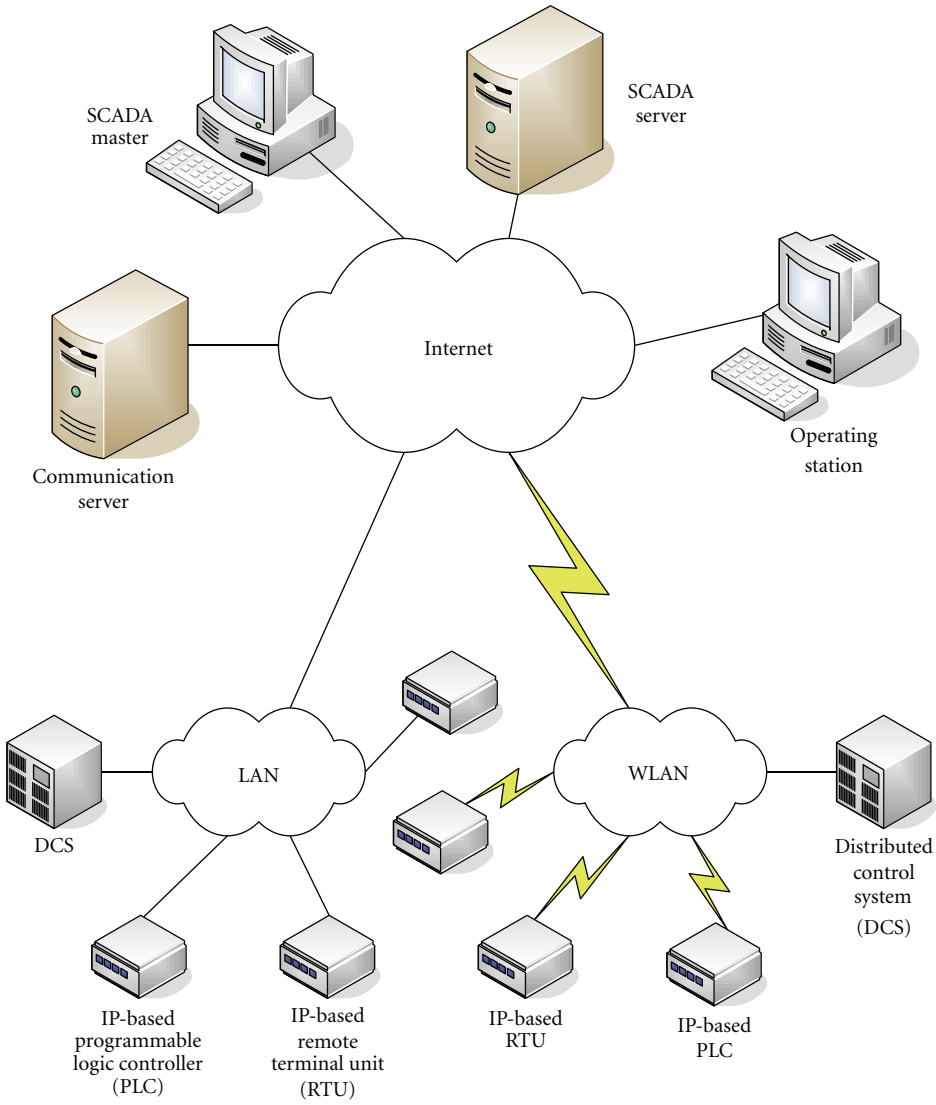


Figure 2: IP-based SCADA system.

the strategic needs of use, optimization of the total cost of ownership, and taking into account initial investments and later costs of operation. Lastly, upgrades, maintenance, and related personnel cost and protection of the investment are secured by the integration of Ethernet/IP over existing transport networks.

Along with advancements in IP technology, IP-based SCADA systems have incorporated various beneficial features as well. These features include unlimited locations for servers and clients where users can install and move their SCADA servers, RTUs, and terminal servers to any site, which gives high flexibility in terms of redundancy and security and in the case of failure in SCADA servers where servers connected to the IP network provide mutual backup for optimized availability. Also, other benefits we can consider are service takeover and remote support as the control centers are not manned during the night. During this period, other regions can either take over the control or supervise log-ins via VPN in case of emergencies. Lastly, savings are obtained through IP-enabled RTUs; many front-end devices are no longer required since a lot of hardware, spares, and cabling can be saved and maintenance costs reduced.

When considering migration from a serial system that uses analog communications to an Ethernet networked environment, it is not as straightforward as simply configuring the PLC, RTU, or field equipment to acknowledge communications from IP networks. Changes will probably need to be made at the front-end processor (FEP) server to switch the SCADA driver from a serial protocol to an Ethernet protocol.

Telecommunications media and hardware may need to be upgraded or replaced, and finally changes will need to be made in the field controller hardware as well. With the shift to IP communications, the risk to cyber threats is much greater since packets can be routed into field devices from other external networks, and attacks no longer require

physical access to the analog circuits. Security risks should be considered when migrating to IP-based communications, together with practical steps for ensuring the reliability and security of SCADA systems when leveraging the Ethernet TCP/IP protocols and communications links.

4. Security of IP-Based SCADA Systems

SCADA systems were originally designed to control and monitor industrial processes using proprietary serial protocols. They were normally located away and secluded from other computer systems. However, in recent years, SCADA systems have been connected to corporate networks and the internet. This can enable businesses to monitor line processes and to support and enhance the process of making correct and beneficial decisions. However, the downside of this is that SCADA systems were never designed with security. With IP-based communications, unexpected threats that did not exist with legacy serial communications can occur at any point and anywhere. It is imperative that we understand how to securely design and to manage SCADA systems in internet-based settings and environments [4].

We need to understand why a conventional SCADA system is more vulnerable when connected to an IP-network. All SCADA systems involve sensors, instrumentation, or other metering devices in order to acquire information about the physical processes that they control and monitor. The sensors, instrumentation, and metering devices are connected to field control devices such as PLCs or RTUs. These devices are used to receive physical input signals, voltages, and currents from sensors and instrumentation. They then transmit the signals to digital data and determine the next decision based on predesignated programmed commands or logic of the system operators. The decisions can terminate or initiate other equipment or change system control parameters. These field devices used to be separated and isolated from cyber threats that faced computer systems, as they were initially deployed over serial analog circuits, and the attacker had to make physical contact with the analog circuits to clamp onto the channel and inject serial data into the circuit or otherwise disturb or capture the serial protocol. Unfortunately, cyber threat risks are now much greater with entry into IP communications [2]. With IP communications, packets can be routed into the field devices from anywhere and attackers are no longer required to gain physical access to analog circuits. Therefore, SCADA systems that are networked via IP communications are more vulnerable than serial communications.

To protect SCADA systems from cyber threats, we have to perform the following tasks [4]. (1) The SCADA IP network should be located physically separate from corporate networks and other untrustworthy networks. When physical separation is not possible, logical separation must be applied. Logical separation is more complicated to implement effectively and runs the risk of ineffective configuration. One should avoid the use of the virtual LAN technology for keeping SCADA IP communications logically separated from corporate IP communications, as VLAN technology is not designed as a security measure but as a bandwidth-shaping tool. (2) IP communications that originate from untrustworthy networks from outside the SCADA system networks should terminate in a buffer network. They should not be allowed direct connections with components in the SCADA system networks; devices inside the SCADA system networks should not be able to communicate directly with the internet. Occasionally, existing corporate IT network infrastructure such as switches, routers, and WAN links must be used as a transport method for portions of the SCADA communications. If that is the case, then the SCADA communications should be encrypted and routed through a VPN tunnel that runs through corporate IT or other noncritical networks. Avoid SCADA devices that are dual-homed to two or more networks at different security zones or trust zones. (3) Additionally, when building a complete end-to-end IP network, avoid using devices that use layer 3 separation between SCADA and other noncritical networks. For proper network isolation, operate equipment that can provide a layer 2 separation. Lastly, a solid cyber defense must offer active blocking devices such as firewalls, IPS, and in-line network antivirus appliances. (4) Designs and procedures are another crucial component. Develop quality insurance techniques to ensure that all security requirements are recognized during the design phase and then executed and tested within the final product. In addition, consider using the ISA S99 security levels as a model when constructing SCADA systems based on IP protocols. If remote access to the SCADA system is permitted over an IP-based network, do not allow users to undergo a similar authentication process used to log into the corporate network. Instead, a different authentication procedure should be applied.

Once a unique SCADA IP-based network is designed and constructed; here are eight recommendations to follow to manage security. First, disable unnecessary services which apply to IP-enabled telecommunication devices, network equipment, PLCs, RTUs, protocol gateway converters, and any other embedded device. Second, limit the utilization of clear text protocols such as telnet, ftp, and http. Instead, force the use of encrypted protocols where technically possible. Third, ensure that the latest version of the Simple Network Management Protocol (SNMP) is up to date, since most IP-enabled telecommunication devices are supported for monitoring the health and performance of the devices. Fourth, keep an event log resident on the device and have a copy sent down to the centralized Syslog server. Fifth, consider deploying in-line network appliances at the choke points that perform network intrusion prevention and antivirus functions. In this way one can filter and drop packets and traffic known to be malevolent based on heuristics and signature matches. Sixth, firmware for IP-enabled telecommunications equipment and control devices should be kept up to date with the latest version. Seventh, control devices such as PLCs, RTUs, smart meters, Ethernet I/O, and IP-enabled instrumentation should be employed with an encryption of PIN code. Eighth, any network devices in front of control devices should be given rate-limiting commands to restrict and limit data from flooding the device.

5. Threats to SCADA Systems over 6LoWPAN

One major advantage of the Internet Protocol version 6 (IPv6) is that it provides larger address space. Being a more recent protocol, IPv6 has a few design enhancements over IPv4, particularly in the areas of autoconfiguration, mobility, and extensibility. However, from a security standpoint, IPv6 implementations are much less mature than their IPv4 counterparts, making it likely that a number of vulnerabilities will be discovered and mitigated before their robustness matches that of existing IPv4 implementations [5]. In [6], Hui and Culler presented the design of a complete IPv6-based network architecture for wireless sensor networks and validated the architecture with a production-quality implementation that incorporates many techniques pioneered in the sensor network community. They claimed that IPv6 is better suited to the needs of WSNs than IPv4 in every aspect. Recent efforts within the IETF now make IP over low-power communication links feasible, including IEEE 802.15.4.

RFC 4944 gives a complete description of IPv6 over low-power wireless personal area networks (6LoWPANs), the packet format standardized by the IETF to enable IPv6 communication over LoWPANs [7]. The main 6LoWPAN features are summarized as follows: the compatibility with respect to stateless address autoconfiguration and neighbor discovery, IPv6 header compression and fragmentation, and support for IP-based routing [8]. The internet of the future is an IPv6 network connecting traditional computers and a large number of smart objects. IPv6 enables the Internet of Things (IoT). This IoT will be the foundation of many services and our daily life will depend on its availability and reliable operation. Billions of devices will use direct, secure, always on, and ubiquitous connections [9].

The challenge of implementing secure communication in the IoT must be taken into consideration for a future model of SCADA systems. On the traditional internet, IP security (IPsec) is the established and tested way of securing networks. IPv6 includes optional support for IPsec authentication and encryption, and web services typically make use of secure sockets or transport layer security mechanisms. These techniques may be too complex, especially for simple embedded devices. It is therefore sensible to explore the option of using IPsec as a security mechanism for the IoT. Smart objects are generally added to the internet using 6LoWPAN, which defines IP communication for resource-constrained networks. Therefore, to present security for the IoT based on the trusted and tested IPsec mechanisms, it is essential to define an IPsec extension of 6LoWPAN.

Various possible complications can be found and encountered in WSNs as described in previous sections. Consequently, determining the vulnerability level estimation and the possible countermeasures can be classified as imperative and substantial. Throughout this section, we provide an overview of layer-based security threats for 6LoWPAN and introduce countermeasures regarding attacks for each layer. Threats are possible which target a specific layer in the protocol stack, that is, the physical, data link, network, transport, application layers, or a combination of any of the mentioned. Eavesdropping, tampering, or jamming radio signal are also possible attacks on the physical layer. For this specific situation, enforcing spectrum techniques for radio communication can be used to prevent attacks. Attacks can also target the data link layer by interfering with the cooperation of the layer's protocols. Simply by inducing a collision or contention or by deliberately fragmenting packets to bypass the Intrusion detection system (IDS), an attacker is able to disrupt an entire message. Fortunately, the induction can be prevented by ignoring excessive requests without identifying authenticity or by adopting proper admission control mechanisms. Network layer protocols can extend their connectivity from neighboring nodes to all other nodes within the wireless sensor network range. To prevent incoming attacks such as message interruption, fabrication, and modification, commanding encryption mechanisms should be applied at the network layer. Additional attacks may be brought against routing protocols such as Wormholes, Sinkholes, or Sybil attacks. These attacks are prevented by building a durable key management and securing the routing protocols. At the transport layer, flooding attacks or session hijacking attacks are very possible. They can be inhibited by controlling the number of connections which a node can make or by using proper authentication mechanisms. Attacks including message interception, fabrication and modification, subversion, and malicious code can be aimed at the application. Simple malicious code detection, isolation, and enforcing a strong encryption mechanism are sufficient to prevent these types of attacks.

SCADA systems will utilize 6LoWPAN in the near future. Figure 3 shows an example of the 6LoWPAN architecture model for SCADA systems. In Figure 3, a border router connects a 6LoWPAN to other IP networks. A 6LoWPAN router provides a packet forwarding service to other nodes. 6LoWPAN remote assets (RAs) do not participate in networking and can only assume the role of leaf sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or pollutants at different locations. By communicating natively with IP, 6LoWPAN networks are connected to other IP networks simply by using IP routers. As shown in Figure 3, 6LoWPANs will normally function on the edge, acting as stub networks. The 6LoWPAN may be connected to other IP networks through one or more border routers that forward IP datagrams between diverse media. Connectivity to other IP networks may be provided through any arbitrary link, including Ethernet, Wi-Fi, GPRS, or satellite. Because 6LoWPAN only specifies operation of IPv6 over IEEE 802.15.4, border routers may also employ stateless IP/ICMP translation [10] or other IPv6 transition mechanisms to connect 6LoWPAN networks to IPv4 networks [11].

6. Security Analysis of SCADA Systems over 6LoWPAN

6LoWPAN networks cannot be protected using traditional network security techniques, because the sensor nodes have limited resources and often operate unattended in publicly

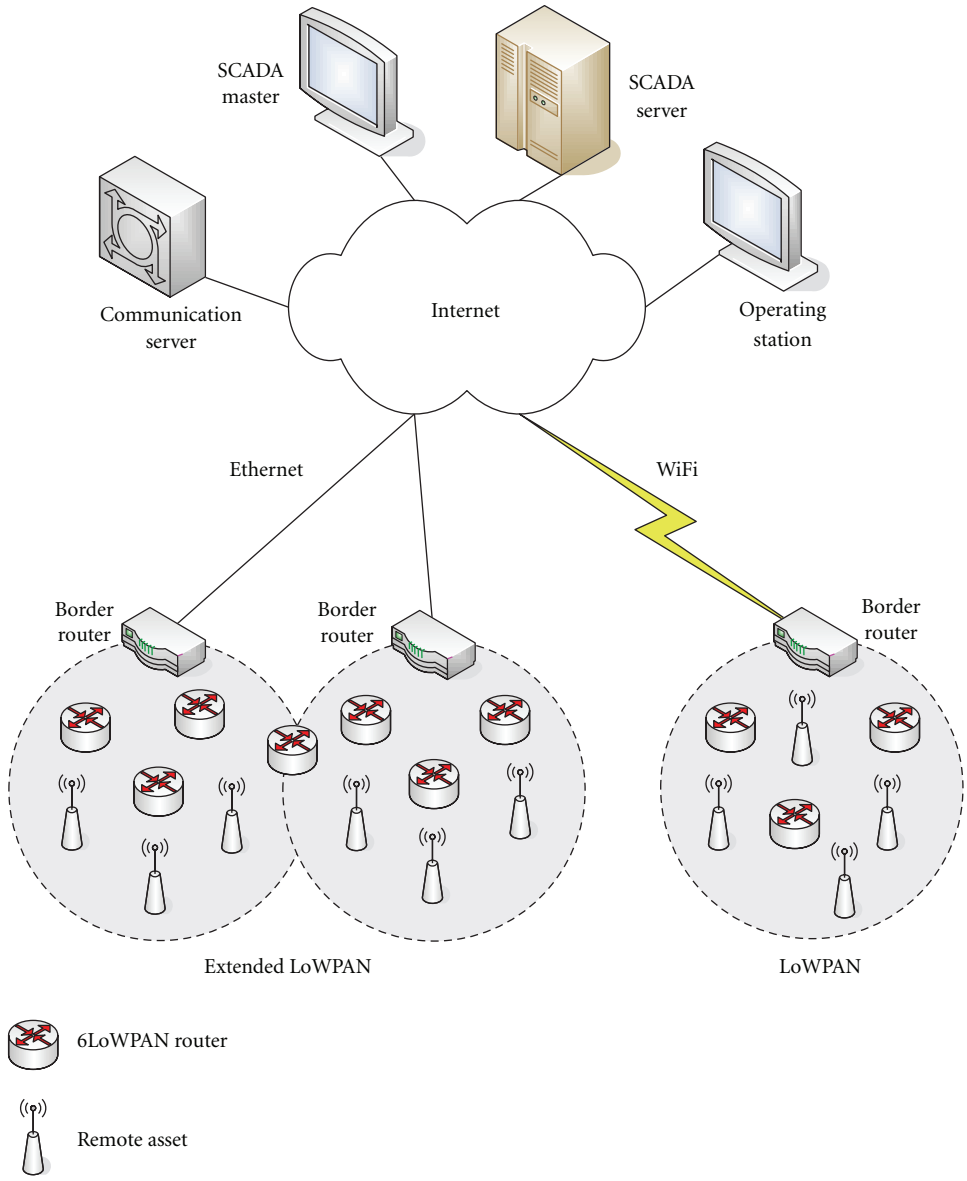


Figure 3: 6LoWPAN architecture model for SCADA systems.

accessible areas. In fact, some security issues are still to be addressed. Resource scarcity is the main constraint of 6LoWPAN technology and also affects the selection of the most appropriate security countermeasures [12]. To raise security to an acceptable level, appropriate risk management and security planning are needed. Such an approach allows for comparison between different configurations of the system, that is, with or without security countermeasures such that performance cost versus security improvements can be properly considered. In addition, existing IP security technologies have to be simplified to be implemented on 6LoWPAN small devices.

Figure 4 depicts a general overview of 6LoWPAN security model for SCADA systems. The interaction between a SCADA server and a SCADA RA can be done in a specific sequence as shown in Figure 5. The network joining process is the required initiation step to form a 6LoWPAN network. The RA node performs gateway discovery by broadcasting gateway solicitation messages. This process will let the gateway acknowledge the message so that the sensor nodes get bound to the gateway. The autoconfiguration is performed in the neighbor discovery (ND) process during bootstrapping. To protect the network against different forms of attacks, it is critical to discover a secured node. During this phase, sensor networks are vulnerable and exposed to a variety of attacks as noted previously. Providentially, SCADA RAs can utilize predeployed keys at which point an authentication is required upon access to the data, allowing only those who possess authorization permission to access the information. After the network establishes the connection, key management processes can initiate. Once the SCADA server authorizes the network, the server can get data from the node or send

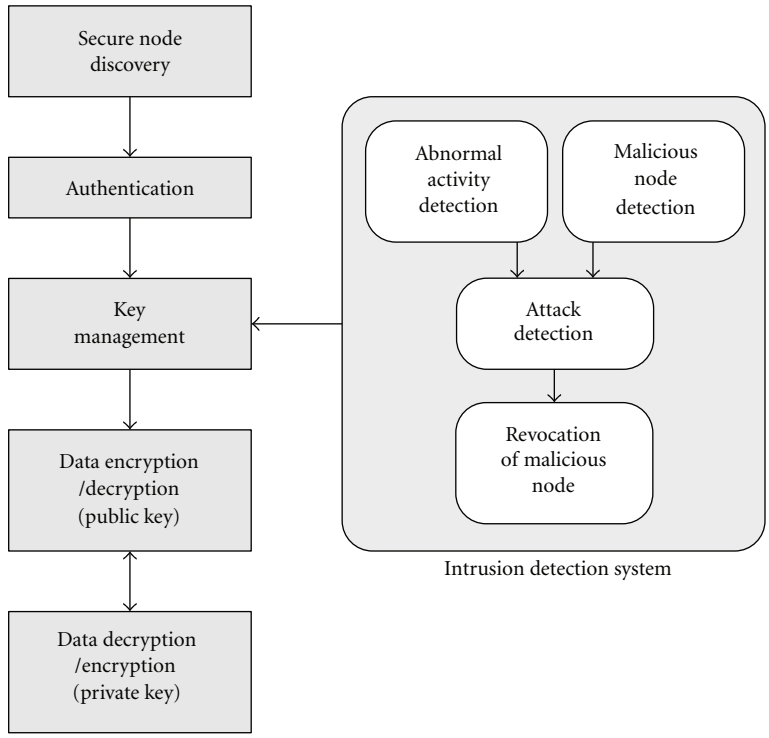


Figure 4: General overview of 6LoWPAN security model for SCADA systems.

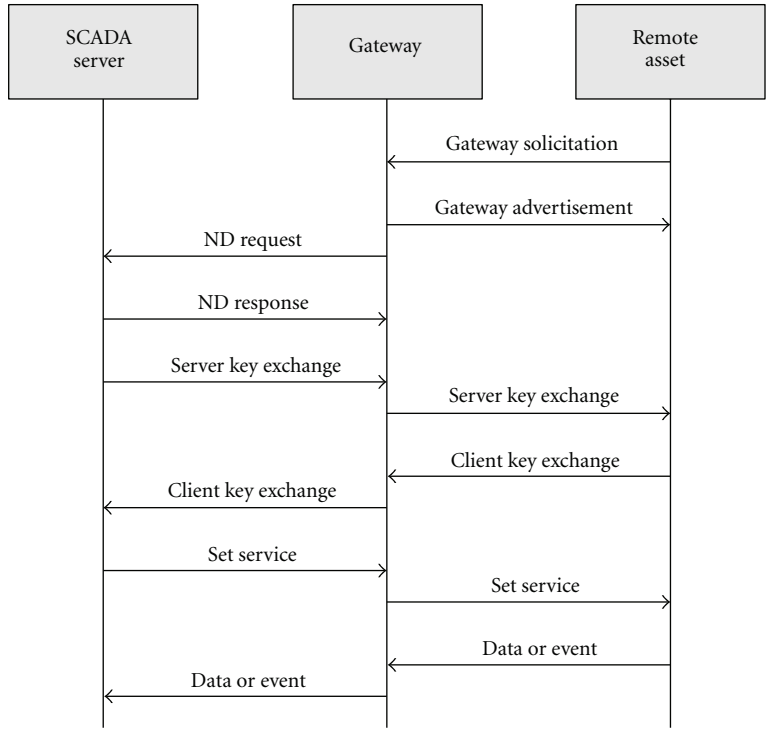


Figure 5: Simplified request-response communication scheme.

actions to it. The node performs the action or sends data to the SCADA server.

In addition to feasible security threats due to connection to the IP, 6LoWPANs share all the same security challenges of WSNs. For 6LoWPAN, IEEE 802.15.4 uses IPsec (IP security protocol) as a default security protocol. Internet key exchange (IKE), the key management solution for IPsec, is considered to be too large due to great number of signaling messages. Having a smaller packet size and low communication overhead without compromising the security level would be considered as the key management solutions for 6LoWPAN. 6LoWPANs that require medium to low security authentications can use symmetric keys for tightening their security communication. In contrast, in an environment which may require an extreme security protection, like a nuclear power plant monitoring system, one may establish an asymmetric key-based system as it provides higher-level security. Asymmetric key encryption uses two different keys, also known as public/private key pairs, for encryption and decryption. When one of the keys from the key pair is used to encrypt a message, the other key is required to decrypt the message. The private key member of the pair must be kept private and secured. The public key, however, can be distributed to anyone who requests it. The public key of a key pair is often distributed by means of a digital certificate. Thus, if a SCADA master's public key is utilized to encrypt a data, then only SCADA RAs can decrypt the data. If a SCADA RA's private key is used to encrypt data, only a SACADA master's public key will be able to decrypt the data, thus indicating that SCADA RA was successful in encryption. Unfortunately, since public key algorithms are slower compared to symmetric algorithms, it is impractical to use them to encrypt large amounts of data. In practice, public key algorithms are mainly used to encrypt session keys. In the real world, symmetric algorithms are used for encryption and decryption of most data. Riaz et al. presented and evaluated key management schemes against an abroad range of metrics such as energy, resource utilization, scalability, and resilience to node compromises [13]. SCADA security experts may choose appropriate key management schemes depending on the desired outcome.

The intrusion detection system (IDS) must continuously monitor the network for abnormal activities and malicious nodes once the bootstrapping phase between the SCADA master and RAs is established. Once a malicious node has been detected, there exists an additional mechanism requirement to remove such a node. After malicious node isolation or abnormal activity detection, a key management process can be applied like the bootstrapping phase. All these requirements can be fulfilled by using a solid authentication and encryption mechanism.

6LoWPAN security architecture will be augmented with IP security technologies whenever available. IEEE 802.15.4 AES (Advanced Encryption Standard) should be used for 6LoWPAN security architecture in conjunction with IP security whenever available. Modified IPsec may be a feasible option for securing the IoT in terms of packet size, energy consumption, memory usage, and processing time [14]. Currently, AES is the default encryption algorithm as mentioned in 6LoWPAN specifications. IEEE 802.15.4 provides built-in encryption based on the 128-bit AES. A comprehensive but light weight key management system is also required to securely generate, distribute, and update keys to be used by AES. A result of utilizing 6LoWPAN is the proficient extension of IPv6 into wireless embedded domains, thus enabling end-to-end IP networking and features for a wide range of embedded applications, such as IP-based SCADA systems over wireless sensor networks.

7. Conclusions

The supervisory control and data acquisition (SCADA) industry's move into the TCP/IP world has been accelerated with business demand for more open and interoperable systems. Presently, operator consoles, SCADA servers, and control room system components are already most likely connected to an internet network. The last components to move to IP are embedded devices which include field controllers, meters, instrumentation, and telecommunications systems linking the control room with embedded devices in the field. Moving to IP-based communications opens up these devices in the field to other networks and systems. Unfortunately, the risk from cyber threat is therefore much greater, as these devices do not have the ability to support typical security features that most computing systems require (such as antivirus, authentication, encryption, and end-point security) and many do not support network monitoring and logging. Thus, to secure IP-based SCADA systems, it is vital to implement secure architectures which prevent access to the SCADA from corporate IT and other third-party networks and to enforce excellent management practices to manage IP-based networks.

It is worth noting that 6LoWPAN architecture for SCADA systems will be used in several areas such as industrial, military, and environmental applications. In such sensitive and critical environments, acceptable delay, high responsiveness, and reliable results and measurements as well as data and services availabilities will often be required. Disaster survivability is another major advantage brought about by the distribution of SCADA functionality over 6LoWPANs. By distributing processing across physically separate locations, it becomes possible to build a SCADA system that can survive a total loss at any one location. It is imperative that critical infrastructure such as power grids and water processing systems be monitored and protected. SCADA architectures, protocols, typical deployments, and security vulnerability concerns have been addressed in this paper. When considering the use of WSN technology with SCADA and control systems, network redundancy and isolation must be taken into consideration. Depending on the level of risk, various encryption standards can be implemented over telecommunication links, such as the Advanced Encryption Standard (AES). In addition, 6LoWPAN is the efficient extension of IPv6 to make an end-to-end IP networking for IP-based SCADA systems over WSN. We have provided an overview of IP-based SCADA systems as well as SCADA systems over 6LoWPANs and addressed their relevant security

concerns. We have also proposed use of public key algorithms to encrypt session keys and symmetric algorithms for encryption and decryption of data for SCADA systems over 6LoWPAN. In order to identify and remove various threats and vulnerabilities to SCADA systems over IP-based WSNs, especially for 6LoWPAN applications, additional and supplementary research and studies on security management methods must be conducted at the national security policy level.

Acknowledgment

This work was supported by the Hansei University.

References

[1] National Communications System, "Supervisory control and data acquisition (SCADA) systems," Technical Information Bulletin 04-1, 2004.

[2] D. H. Ryu, H. Kim, and K. Um, "Reducing security vulnerabilities for critical infrastructure," *Journal of Loss Prevention in the Process Industries*, vol. 22, no. 6, pp. 1020–1024, 2009.

[3] "Botnets, cybercrime, and cyberterrorism: vulnerabilities and policy issues for congress," CRC Report RL32114, 2008.

[4] Centre for the Protection and National Infrastructure, *Good Practice Guide: Securing the Move to IP-Based SCADA/PLC Networks*, 2011.

[5] Centre for the Protection and National Infrastructure, *CPNI Viewpoint: Security Implications of IPv6*, 2011.

[6] J. W. Hui and D. E. Culler, "IP is dead, long live IP for wireless sensor network," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*, pp. 15–28, 2008.

[7] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE802.15.4 Networks," RFC, 4944 (Proposed Standard), 2007.

[8] J. W. Hui, D. E. Culler, and S. Chakrabarti, "6LoWPAN: Incorporating IEEE 802. 15. 4 into the IP Architecture," Internet Protocol for Smart Objects (IPSO) Alliance, White Paper #3, 2009.

[9] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the internet of things—a comparison of link-layer security and IPsec for 6LoWPAN," *Security and Communications Network*. In press.

[10] E. Nordmark, "Stateless IP/ICMP Translation Algorithm (SIIT)," RFC, 2765 (Proposed Standard), 2000.

[11] E. Nordmark and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers," RFC, 4213 (Proposed Standard), 2005.

[12] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," RFC4919 (Proposed Standard), 2007.

[13] R. Riaz, A. Naureen, A. Akram, A. H. Akbar, K. H. Kim, and H. Farooq Ahmed, "A unified security framework with three key management schemes for wireless sensor networks," *Computer Communications*, vol. 31, no. 18, pp. 4269–4280, 2008.

[14] E. Kim, D. Kaspar, and J. P. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," RFC6568 (Proposed Standard), 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 685107, 12 pages
doi:10.1155/2012/685107

Research Article

Wireless Sensor Networks for Vital Signs Monitoring: Application in a Nursing Home

Yuan-Jen Chang,[1,2] Chin-Hsing Chen,[1] Li-Feng Lin,[3] Ruo-Ping Han,[1] Wen-Tzeng Huang,[4] and Guey-Chuen Lee[1]

[1] *Department of Management Information Systems, Central Taiwan University of Science and Technology, Taichung 40601, Taiwan*
[2] *Institute of Biomedical Engineering and Materials Science, Central Taiwan University of Science and Technology, Taichung 40601, Taiwan*
[3] *Department of Nursing, Central Taiwan University of Science and Technology, Taichung 40601, Taiwan*
[4] *Department of Computer Science and Information Engineering, Minghsin University of Science and Technology, Hsinchu 30401, Taiwan*

Correspondence should be addressed to Yuan-Jen Chang, ronchang@ctust.edu.tw

Received 14 June 2012; Accepted 23 September 2012

Academic Editor: Xusheng Sun

This study evaluated the application of a wireless sensor network (WSN) on a web-based vital signs monitoring system to nursing homes in Taiwan. The applicability assessment focused on the timely provision of information, information accuracy, system usability, and system accessibility of healthcare systems using a wireless sensor network. Experiments were performed under Internet-based network conditions to verify the timely information provision, especially for a web-based system, including Ajax technology. The accuracy of the information was verified from statistical analyses of the residents' daily vital sign measurements. A comparison was performed between having and not having a healthcare monitoring system in nursing homes for system usability, system accessibility, and system efficacy. The results indicate that the successful application of a WSN healthcare monitoring system is feasible for use in nursing homes in Taiwan.

1. Introduction

As the number of elderly people in Taiwan has continued to increase in recent decades, the percentage of persons aged 65 and older has increased from 7.1% in 1993 to 9.9% in 2006 [1]. It is estimated that Taiwan will become an aged society in 2017, when elderly people will account for more than 14% of the entire Taiwanese population. Thus, health-related issues are becoming increasingly important for this segment of the population. The needs for medical care and long-term care, as well as their linkage with the National Health Insurance (NHI) program for elderly people, should be considered. It is also expected that the costs of long-term care for older people will increase [1].

Besides, one of the severe problems for long-term patient care is the shortage of human resources and the growing needs of health workers [2]. The adoption of information communication technology (ICT) may help to alleviate some parts of this problem. Additionally, a higher quality of nursing service may be achieved. A vital signs monitoring system with wireless sensor networks (WSN) may be a key solution for long-term care. A wireless sensor network monitoring system not only avoids the high cost of devices and installation but also preserves the resident's comfort and privacy [3]. These sensor networks are designed to be self-management and self-healing and are installed to gather data in places where the use of cabled sensors is undesired [4, 5]. Many advantages of healthcare monitoring systems based on wireless sensor networks (WSNs) have been reported [6, 7]. To improve the throughput of WSN, a DSP-based WSN platform was proposed [8]. To enhance the performance of communication, an integrated wireless health service system with 3G/3.5G and mobile devices was revealed in some researches [9–13]. Impacts of monitoring system to

elderly people were also evaluated in some researches [14–17]. The results showed that most of elderly people have positive attitude to wireless health monitoring systems. These applications help medical caretakers in hospitals to obtain patients' electronic records anytime and anywhere. Further, as mentioned in a previous study [16, 17], a ZigBee-based healthcare service system can provide significant satisfaction for the community care. The results also suggest that a telecare system will not be accepted without an auxiliary medical service. This may be especially true for a new technology, where users have to trust the system by seeing that it meets their expectations and delivers satisfaction [18]. Additionally, people may lack confidence in such a healthcare monitoring system if it includes no medical service provided by physicians or nurses. Because of Chinese traditional concepts, when they feel sick, many older people hope to talk to doctors face-to-face to meet their psychological needs. A successful line of business (LOB) application for home healthcare monitoring systems in Taiwan should take this into consideration. In this study, we suggest that a successful healthcare monitoring system should be combined with a nursing/medical service. In the present environment in Taiwan, such a system would probably be implemented in nursing homes or clinics because those facilities can provide nursing/medical services. When all nursing homes and clinics use healthcare monitoring systems in their daily work, they can ask their residents/patients to use healthcare monitoring systems at home. Then, the nursing service provided by nursing homes and medical services provided by clinics can be extended to home. However, it is obvious that a feasible strategy for replacing the old system with a new one should provide a seamless transition and should not change the workflow too much. This development should provide caregivers more time and reduce the routine loading. After trust in the new healthcare monitoring system has been established, it will be possible for the users, including residents and caregivers, to accept the change. Because of limitations in medical law, telemedicine remains impermissible in Taiwan presently. However, the healthcare monitoring system we propose can be used as part of a care service, combined with traditional nursing service in nursing homes and long-term healthcare institutions. Thus, this study focused on assessing the applicability of a wireless sensor network (WSN) healthcare monitoring system for nursing homes, from the viewpoint of software development and deployment [19]. The results of our 3-month clinical experiment may lead to a better understanding of the feasibility of a telecare business model in Taiwan. The objectives of this study were as follows:

(i) to evaluate the timely provision of information, information accuracy, system usability, and system accessibility of a WSN healthcare monitoring system in daily healthcare measurements,

(ii) to provide WSN healthcare monitoring system implementation and assessment in nursing homes,

(iii) to provide results indicating user satisfaction with a WSN healthcare monitoring system in nursing homes.

This paper describes the design and implementation of a WSN healthcare monitoring system for the care of elderly people living in nursing homes, using a clinical experiment to evaluate the system.

2. Research Methodology

2.1. System Architecture of Healthcare Platform. The design and implementation of a WSN healthcare monitoring system was required for this study because no web-based healthcare monitoring system with WSN offering daily vital signs measurement is currently commercially available. The system architecture we designed is shown in Figure 1. The wireless sensor network consists of a Personal Area Network (PAN) coordinator, routers, and end devices. The end devices of the wireless sensor network include a temperature sensor and a blood pressure sensor. The vital signs detected by the sensors are transmitted from the devices. When the router receives data from the associated device, it immediately sends data to the associated PAN coordinator. The PAN coordinator then sends it to the UART connected by a gateway. Java program was developed to transmit vital signs data from the gateway to the remote database server via the Internet using a client-server access method [20]. These WSN components and gateway were set up at the Haechun Nursing Home, located in Taichung, Taiwan, and were connected to the Internet with an ADSL broadband connection at a connection speed of 2 M/256 kbps (downstream/upstream). An independent global web and database server was constructed and set up at the Central Taiwan University of Science and Technology (CTUST), about 10 km from the Haechun Nursing Home. The web-based healthcare service management platform was hosted on a web server. Client-side users, for example, the caregiver, physician, or a resident's family member, can access the vital signs data through the Internet. Three web-based modules were provided to client users: the Daily Vital Signs Recording Module (DVSRM), the Caregiver Signature Form Printing Module (CSFPM), and the Historical Record Plotting Module (HRPM). Each resident had to have basic physiological signals measured once a day, including temperature, heart rate, and blood pressure. The DVSRM can receive vital signs data from the sensor and store data on the database server. Caregivers can use the CSFPM to print out and sign the daily vital signs' records of the residents. The physician and the resident's family members can use the HRPM to view the resident's vital signs record history and know his health status.

2.2. A Wireless Sensor Network Layout in a Nursing Home. The wireless sensor network was built at the Haechun Nursing Home. At the time of the study, 94 beds were housed in the six-floor building. A total of 10 residents on the fifth floor cared by one were chosen by convenience sampling to participate in this experiment. The complete WSN layout is shown in Figure 2. The wireless sensor network consists of a Personal Area Network (PAN) coordinator, routers, and end devices. A microcontroller unit built in the device with a ZigBee RF chip selected one of the routers that had the shortest distance between the router and the end device.

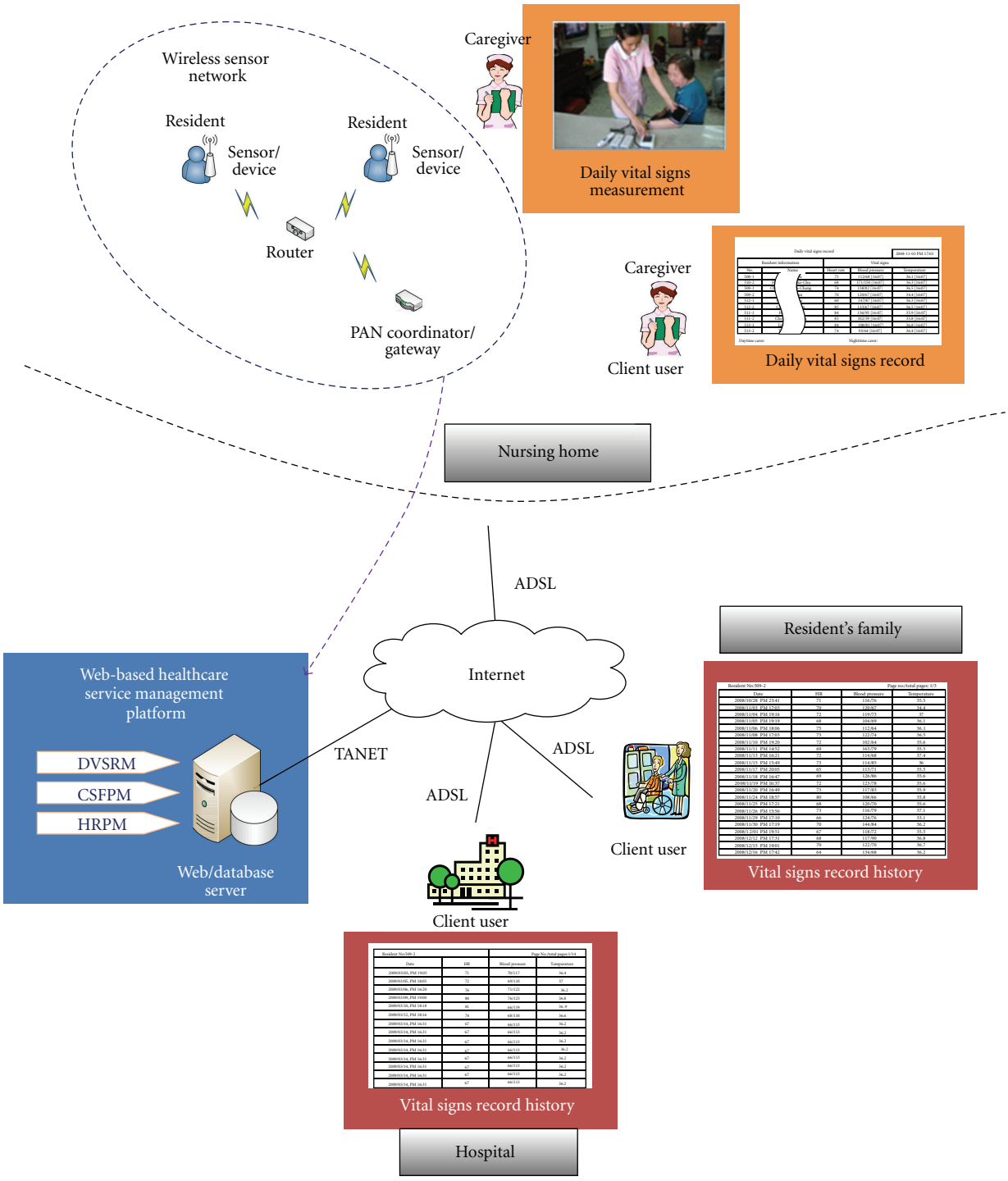


Figure 1: System architecture of the WSN healthcare monitoring system designed in this study.

When the device joined a router, it began retrieving vital signs data from the sensor and sent the data to the router. When the router received data from the associated device, it immediately sent data to the associated PAN coordinator. The PAN coordinator then sent the data to the UART connected with a gateway. The gateway then sent data to the remote database server by a client-server program built in a gateway via the Internet.

The detailed description of the components is as follows.

(1) Internet connection: Haechun Nursing Home was connected to the ISP HINET by an ADSL broadband network at a connection speed of 2 M/256 kbps (downstream/upstream). The caregiver can also access the healthcare platform via this broadband network connection.

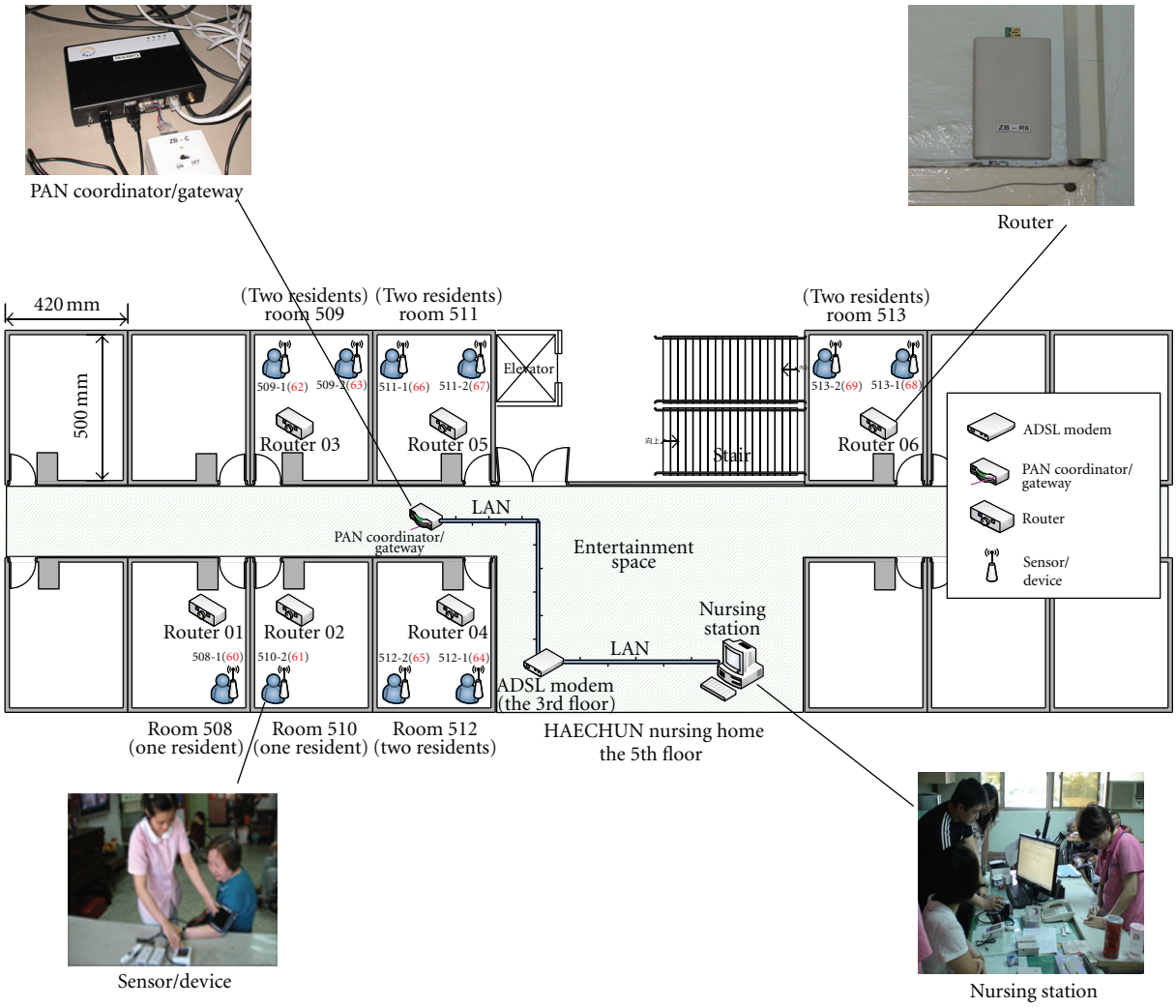
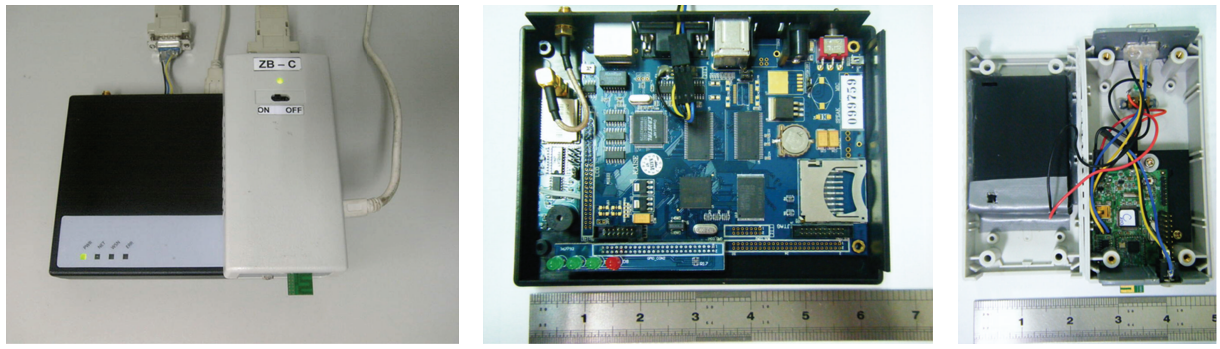


Figure 2: Complete wireless sensor network layout in the nursing home.

(2) Gateway and PAN coordinator: the gateway and the PAN coordinator ZB-C were placed in the ceiling of a corridor (Figure 2). Only one gateway and a PAN coordinator were needed on each floor. The gateway and the PAN coordinator are shown in Figure 3(a). The circuit board in the gateway and PAN coordinator is shown in Figures 3(b) and 3(c), respectively. An embedded platform, provided by the Rosonix Technology (Rosonix Technology Inc., Taipei, Taiwan), was used to develop the gateway. The platform uses the ARM9 processor (Samsung 2410) with 64 Mbyte NAND Flash and two 128 Mbyte memory modules on board. Embedded Linux 2.4 was installed, and C/C++ language was used to develop the gateway client module. This sensor gateway was connected to the PAN coordinator through a UART. The WSN components, including the PAN coordinator, router, and device, were homemade. The Texas Instruments TI-MSP430F1611 (Texas Instruments, Dallas, TX, USA) was chosen as the core control unit, because of its ultra-low-power consumption, and the UBEC ZigBee module UZ2400 (Uniband Electronic Corp., Hsinchu, Taiwan) was chosen as the RF module for its stability and design flexibility. Different firmware versions were used with the PAN coordinator, router, and device, depending on the application. The design of the WSN hardware platform was described in detail previously [21, 22].

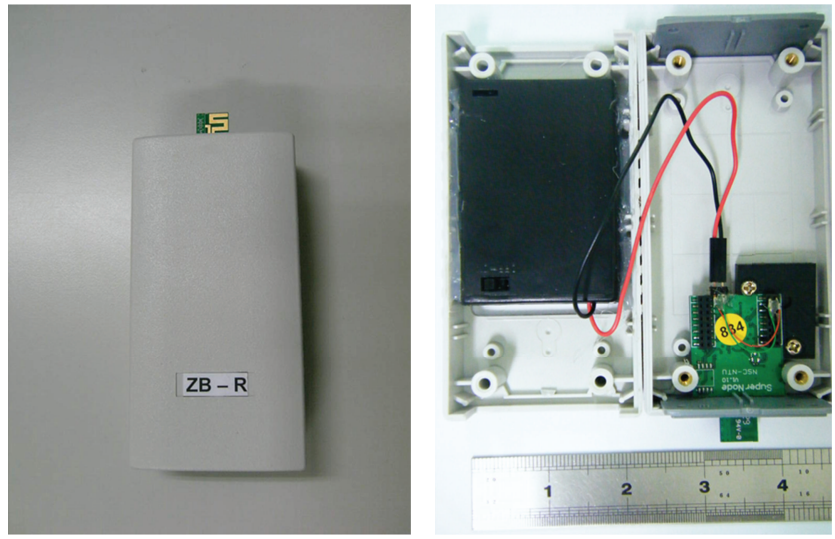
(3) Router: one router, ZB-R, was installed in each room on top of the door (Figure 2). The router received vital signs data from the device and transmitted the data to the PAN coordinator. The router and circuit board are shown in Figures 4(a) and 4(b), respectively.

(4) Device: the ZigBee Device ZB-T and ZB-P can retrieve vital signs data from the thermometer and sphygmomanometer, respectively, and transmit data to the router ZB-R. Each resident had his/her own device for the different sensors in which a specified



(a) (b) (c)

Figure 3: Photographs of (a) gateway and PAN coordinator ZB-C, (b) circuit board in gateway, and (c) circuit board in PAN coordinator.



(a) (b)

Figure 4: Photographs of (a) router ZB-R and (b) circuit board in router.

sensor ID was recorded in the firmware. Each device had a unique resident ID in this wireless sensor network. An LED light was designed to indicate data transmission status. Photographs of the devices for temperature and blood pressure are shown in Figures 5(a) and 5(b), respectively. A photograph of the circuit board in the device is shown in Figure 5(c).

(5) Sensor: two sensors, a thermometer and a sphygmomanometer, were used to measure three vital signs, temperature, heart rate, and blood pressure. The thermometer used in the WSN healthcare monitoring system was a TD1261A (TaiDoc Corporation, New Taipei City, Taiwan), as shown in Figure 5(a). The sphygmomanometer used in the WSN healthcare monitoring system was the TD3250B (TaiDoc Corporation), as shown in Figure 5(b).

(6) Web/Database Server: A "WAMP" (Windows, Apache, MySQL, PHP) system was set up to manage the vital signs data. The web-based healthcare service management platform was hosted on a web server, and it provided the aforementioned three modules to client users. The web/database server was located at the CTUST and connected to the TANET at a connection speed of up to 1,000 Mbps.

2.3. Evaluation Methodology. To evaluate the system in clinical practice, a clinical paired-comparison experiment was designed [23]. Ten residents cared by one caregiver were selected by convenience sampling to participate with their agreement. For convenience, they were cared for by the same caregiver and were located on the same floor. In the nursing home, every resident has to have his/her vital signs measured at least once a day. The daytime caregiver and the nighttime caregiver have to write down the results of the measurement on a signature form and sign the form (Figure 6). The caregivers who take care of these residents in the nursing home were most concerned about whether the vital signs data could be recorded automatically, and the most important issues were that the vital signs data were

(a) (b) (c)

Figure 5: Photographs of the sensor and device. (a) Thermometer and device, (b) sphygmomanometer and device, and (c) the circuit board in device.

Daily vital signs record

2008-11-03 PM 17:03

Resident information		Vital signs		
Number	Name	Heart rate	Blood pressure	Temperature
508-1		75	112/68 (16:07)	36.1 (16:07)
510-2		68	171/150 (16:07)	36.3 (16:07)
509-1		74	158/82 (16:07)	36.5 (16:07)
509-2		70	120/67 (16:07)	34.4 (16:07)
512-1		60	147/87 (16:07)	36.3 (16:07)
512-2		85	133/67 (16:07)	36.5 (16:07)
511-1		84	136/95 (16:07)	35.9 (16:07)
511-2		83	102/59 (16:07)	35.8 (16:07)
513-1		84	108/81 (16:07)	36.8 (16:07)
513-2		74	93/64 (16:07)	36.4 (16:07)

Daytime carer: Nighttime carer:

Figure 6: Signature form for the daily vital signs record.

always correct and that they were not lost through wireless transmission. In addition, the nurses in the nursing home had to collate each resident's daily vital signs record into a vital signs history record (Figure 7). Periodic routine visits by physicians were scheduled once a month. The physicians needed to review the resident's historical vital signs record during each visit. This WSN healthcare monitoring system can help nurses generate reports with vital signs history records and can enhance the efficiency of the nurses' work. Thus, the aim of the experiment was to evaluate the timely provision of information, information accuracy, system usability, and system accessibility of a WSN healthcare monitoring system in daily vital signs measurement. One caregiver measured vital signs of these 10 residents using conventional equipment and without the WSN healthcare monitoring system. Another caregiver measured the vital signs of the same 10 residents using the WSN healthcare monitoring system. The experiment lasted 3 months. The detailed evaluation method is described as follows.

(1) Timely provision of information: the total time it took to display the residents' sensor-measured vital signs on the user's screen was calculated. Time1, Time2, and Time3 were measured to obtain the mean and standard deviation for each experiment (Figure 8) [24]. Thus, the total time is equal to the sum of Time1, Time2, and Time3. This represented the timely provision of information. Time1 included device processing time, transmission time from device to router, router processing time, and transmission time from router to gateway. Time2 included gateway processing time and transmission time from gateway to database server. Time3 included web page processing time and data transmission time from database server to a web page browsed by client users. An Internet broadband network connection with a connection speed of 2 Mbps/256 kbps (downstream/upstream) was chosen for client users because such an Internet broadband network connection is popular, with its usage exceeding 66% by families in Taiwan Network Information Centre.

Resident number: 509-2 Page number/total pages: 1/3

Date	HR	Blood pressure	Temperature
2008/10/28 PM 23:41	71	116/76	35.5
2008/11/03 PM 17:03	70	120/67	34.4
2008/11/04 PM 19:16	72	119/73	37
2008/11/05 PM 19:19	68	104/69	36.1
2008/11/06 PM 18:06	75	112/64	36.1
2008/11/08 PM 17:03	73	122/74	36.5
2008/11/10 PM 19:20	72	102/64	35.6
2008/11/11 PM 14:52	69	163/79	35.5
2008/11/13 PM 16:21	72	114/68	37.4
2008/11/15 PM 15:49	73	114/85	36
2008/11/17 PM 20:05	65	113/71	35.5
2008/11/18 PM 16:47	69	126/86	35.6
2008/11/19 PM 16:37	72	123/78	35.6
2008/11/20 PM 16:49	73	117/83	35.9
2008/11/24 PM 18:57	80	108/66	35.8
2008/11/25 PM 17:21	68	120/70	35.6
2008/11/26 PM 15:56	73	116/79	37.1
2008/11/29 PM 17:10	66	124/76	33.1
2008/11/30 PM 17:19	70	144/84	36.2
2008/12/01 PM 19:51	67	118/72	35.5
2008/12/12 PM 17:31	68	117/90	36.8
2008/12/15 PM 19:01	70	122/70	36.7
2008/12/16 PM 17:42	64	134/68	36.2

1 2 3

Figure 7: Vital signs record history of a resident (only some records were shown here).

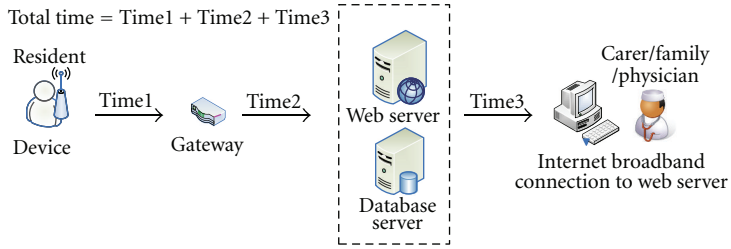


Figure 8: Diagram depicting the experimental design for timely delivery of information.

(2) Information accuracy: when a new technology is adopted, users generally ask whether they can trust the technology. Thus, we wanted to compare the two methods for measuring residents' vital signs. A paired experiment was designed to test for differences between the two measurement methods. One caregiver used conventional sensors to measure 10 residents' vital signs. The thermometer used was an OMRON MC-110B (OMRON Healthcare Inc., Kyoto, Japan) and the sphygmomanometer used was the Spirit ck-101 (Hsing Sheng Medical Goods Inc., Taipei, Taiwan). Another caregiver repeated the measurements on the same 10 residents' vital signs using the WSN healthcare monitoring system. The sensors used with the WSN system were a thermometer, TD1261A, and a sphygmomanometer, TD3250B, both from the TaiDoc Corporation (TaiDoc Corporation, New Taipei, Taiwan). The two values, the paired data, were recorded and a paired t-test statistical analysis was conducted to compare the two measurement methods.

(3) System usability: the main reason for applying usability in the development of a healthcare monitoring system is to increase user satisfaction and acceptance. Even when our system was deployed at the Haechun Nursing Home, we continued with our system modifications. Some system usability issues were considered:

(i) provide obvious visual indicators when the system is retrieving and transmitting data,

(ii) provide a standard operational procedure (SOP) that users can follow,

(iii) reform the user interface for the sensor device to make it easy for the user to understand the status of the sensor device,

(iv) redesign the workflow for daily measurements to save the user time [19].

Table 1: Results of experiments on the timely attainment of information.

Time1*1 (second)	Time2*2 (second)	Time3*3 (second)	Total time (second)
0.006 ± 0.000	0.228 ± 0.030	0.586 ± 0.040	0.82 ± 0.128

*1 Average time of 30 times experiments, and each experiment 10,000 row data: average time ± standard deviation.
*2 The connection speed from the nursing home to Internet is 2 M/256 k.
*3 The connection speed from the resident's family to Internet is 2 M/256 k.

Table 2: Data for different measurement techniques.

Residents			Conventional method				Sensors with a WSN healthcare monitoring system			
Number	Age	Gender	HR	Systolic BP	Diastolic BP	Temperature	HR	Systolic BP	Diastolic BP	Temperature
508-1	50	M	80	110	60	36.0	75	109	74	36.5
509-1	85	F	80	130	90	36.0	67	115	71	36.4
509-2	74	F	68	110	80	36.0	68	112	76	36.1
510-2	54	M	88	160	90	36.2	80	147	67	36.4
512-1	76	F	66	130	60	36.7	59	121	67	36.2
512-2	88	F	98	130	80	36.6	97	96	63	36.3
511-1	52	F	80	110	60	35.4	73	122	87	35.6
511-2	74	F	70	90	70	35.5	89	96	71	35.7
513-1	61	M	80	100	60	36.2	74	108	73	36.6
513-2	87	M	84	120	86	37.2	90	117	87	36.6

(4) System accessibility: some features were customized to caregivers' and client users' needs to promote system accessibility. A web-based graphic user interface (GUI) was designed to make the system easy to use. The user simply opens the browser and each of the processes was performed, including the printing of the daily vital signs record, monitoring of the physiological status of the residents, and viewing of the vital signs history records of residents.

After the experiment, user satisfaction was investigated to understand the intended use of the WSN healthcare monitoring system.

3. Results and Discussions

3.1. Attainment of Timely Information. The results of the measurements for the timely provision of information in this experiment are shown in Table 1. The average duration of Time1 was 0.006 ± 0.000 seconds. Time1 was primarily affected by two factors. First, the number of sensor nodes transmitting data concurrently affected transmission time. As a complication, an increase in the number of sensor nodes increased the possibility of a collision of packets. Second, the priority of each packet affected the waiting time for transmission if two or more sensor nodes were transmitting data concurrently. In this study, only one sensor node transmitted data at any time.

The average duration of Time2 was 0.028 ± 0.030 seconds. Time2 was primarily affected by two factors [25]. First, the network connection speed from the gateway to the database server affected the transmission time. In this study, the Haechun Nursing Home used an ADSL connection with a connection speed of 2 M/256 kbps (downstream/upstream). Second, the access method from the gateway to the database server affected transmission performance. According to a previous study [25], we adopted the client-server access method to achieve better transmission performance.

Time3 is the time needed for the client users to inspect the vital signs' records of the residents. A web-based interface was designed for client users in which Ajax was used to retrieve data from the remote web server. However, the asynchronous JavaScript function did not display the results on a web page until a response was received from the server. This response time affected Time3. Due to the low data transmission rate needed by Ajax technology, a different network connection speed for the client side did not affect Time3. The Internet broadband network connection with a connection speed of 2 Mbps/256 kbps (downstream/upstream) was chosen for client users. In a total of 30 experiments, with each experiment involving transmission of 10,000 data records, the average total time for transmission from device to client users was 0.82 ± 0.128 seconds (Table 1). The results showed that whenever the vital signs were measured, the data were immediately recorded into the database. This means that any client user can inspect the data at once. This provides physicians the opportunity to monitor residents' health status remotely via the Internet.

3.2. Information Accuracy. To convince caregivers and residents to trust the vital signs' values measured by the sensors of the WSN healthcare monitoring system, a paired t-test statistics analysis was conducted. Table 2 shows the paired data measured by the two different methods. A total of 107 data records were used in the calculations, but only one day's record is presented. Table 3 shows the paired t-test results for the two different measurement techniques. For all vital signs

Table 3: Paired t-test results for different measurement techniques.

		Conventional method		Sensors with WSN healthcare monitoring system		t	P
		M	SD	M	SD		
Vital signs	HR	78.07	9.321	79.64	10.229	−1.788	0.077
	Systolic BP	119.81	13.666	120.92	17.888	−0.861	0.391
	Diastolic BP	77.16	8.306	75.55	9.550	1.802	0.074
	Temperature	36.18	0.508	36.10	0.5	1.199	0.233

Table 4: Questionnaire responses ($n = 162$).

Question	User score					Mean
	A1 n (%)	A2 n (%)	A3 n (%)	A4 n (%)	A5 n (%)	
(1) To use WSN healthcare monitoring system in your job would make it easier to accomplish care work.	20 (12.3)	75 (46.3)	50 (30.9)	14 (8.6)	3 (1.9)	3.59
(2) To use WSN healthcare monitoring system would improve care.	19 (11.7)	80 (49.4)	45 (27.8)	15 (9.3)	3 (1.9)	3.60
(3) To use WSN healthcare monitoring system would make it easier to conduct care work.	20 (12.3)	76(46.9)	52 (32.1)	11 (6.8)	3 (1.9)	3.61
(4) You would find WSN healthcare monitoring system useful to conducting care work.	17 (10.5)	87 (53.7)	45 (27.8)	10 (6.2)	3 (1.9)	3.65
(5) You would be clear and understandable with WSN healthcare monitoring system.	16 (9.9)	83 (51.2)	49 (30.2)	11 (6.8)	3 (1.9)	3.60
(6) To learn how to operate WSN healthcare monitoring system is easy for you.	19 (11.7)	77 (47.5)	52 (32.1)	10 (6.2)	4 (2.5)	3.60
(7) I think it is easy for me to become familiar with WSN healthcare monitoring system after learning and practice.	19 (11.7)	83 (51.2)	51 (31.5)	7 (4.3)	2 (1.2)	3.68
(8) I think WSN healthcare monitoring system is easy to use.	18 (11.1)	75 (46.3)	58 (35.8)	9 (536)	2 (1.2)	3.60
(9) I am willing to use the WSN healthcare monitoring system in my work.	YES: 135 (83.3)			NO: 27 (16.7)		

A1: strongly agree; A2: agree; A3: neutral; A4: disagree; A5: strongly disagree.

measured in this study, heart rate, blood pressure, and temperature, the P values were greater than 0.05. Thus, we are confident that there was no statistically significant difference between the two measurement methods. Additionally, these data were reviewed by physicians and nurses, who claimed that these data were routine. Thus, caregivers can trust the WSN healthcare monitoring system to perform daily vital signs measurements.

3.3. System Usability. Caregivers rely on technology to perform their daily vital signs measurement. Information accuracy affects their confidence in using the technology [18, 19]. For the wireless sensor network, the transmission distance between sensor nodes affected the packet loss rate (PLR), which consequently affected the accuracy of our data. This study included an experiment on PLR in the indoor space. The experiments used a transmitter node and a receiver node; the transmitter node continuously sent one hundred full packets to the receiver node. The receiver node checked how many packets were received after varying the distance between the transmitter node and the receiver node and repeating the experiment 10 times at a given distance, as follows:

$$\text{Packet Lost Rate (PLR)} = \frac{100 - \text{Received Packets}}{100} \times 100\%. \quad (1)$$

Figure 9 shows that the packet loss rate was less than 2.5% when the transmission distance was less than 18 m. Serious data errors may occur when the packet loss rate increased. We are not able to control the packet loss rate because the wireless environment differs, case by case. To avoid data transmission errors, a specially designed indicator of the data transmission status was guaranteed to correct data storage on the database server. An LED light was used to display the data transmission status. When the device was turned on, initial test data were sent to the gateway, and the LED light flashed continuously until the device received a message from the gateway to ensure that vital signs data were stored in the remote database. If the response revealed the

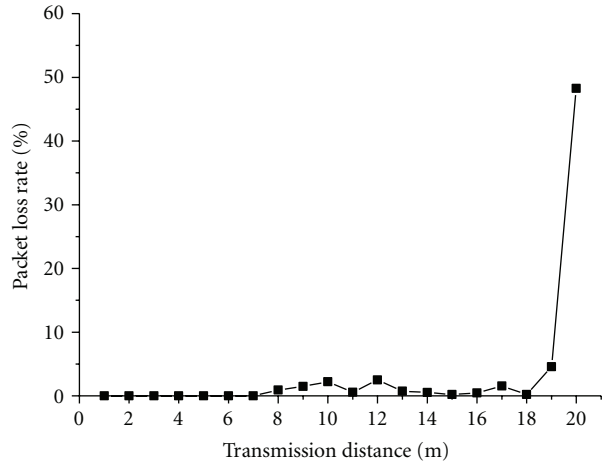


Figure 9: Packet loss rate versus transmission distance (meter).

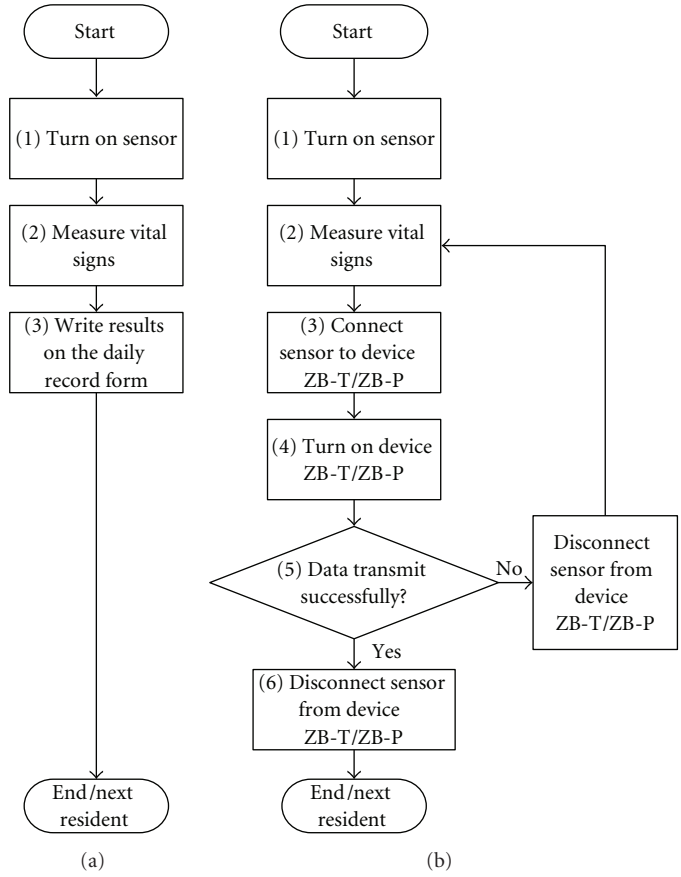


Figure 10: (a) Conventional workflow and (b) new workflow for vital signs measurement procedure.

transmission was correct, the LED light turned off and waited for the caregiver's immediate measurement. When the next vital signs were measured, the LED light again lit up and the process was repeated. The data were correctly measured and stored if the caregiver followed the SOP. This LED indicator made users more confident in performing their daily vital signs measurement and enhanced the usability of the WSN healthcare monitoring system.

3.4. System Accessibility. A similar workflow process for caregivers to measure vital signs of residents using the WSN healthcare monitoring system enabled the caregivers to transfer seamlessly from the conventional measurement technique to a new one. Figure 10 depicts the conventional and new workflow procedures for measuring vital signs. Compared with the conventional workflow, three extra steps (steps 4–6) were needed for the new workflow; however,

these added steps took less than 30 s. Moreover, the caregivers did not need to write down their retrieved results on the signature form of the daily vital signs record. Following the SOP also reduced typing errors. From the physician's point of view, it is possible to review any resident's health status every day via the Internet using only a browser, thus reducing scheduled clinic visits for residents. Additionally, unexpected health alarms can be sent immediately, if needed.

3.5. Costs. The cost for developing the system designed in this study for nursing home was minimal, because the workflow already existed. The training course can also be accomplished easily. The calculated costs included an equipment cost of one gateway (US $514), one PAN coordinator (US $29), six routers (US $174), and 20 devices (US $580), for a total cost of US $1,297, or an average of US $130 per resident. The director of the nursing home states that this cost would be acceptable if the WNS healthcare monitoring system could increase the residents' satisfaction. Of course, such costs can be reduced when the equipment is mass produced and purchased in bulk.

3.6. User Satisfaction. This WSN healthcare monitoring system was demonstrated to 130 nurses and 32 physicians; all 162 users responded to the user satisfaction questionnaire. Table 4 shows the percentage of the nurses and physicians who were satisfied and dissatisfied. Overall, the nurses and physicians were satisfied with the WSN healthcare monitoring system. About 61.1% of the users agreed that the WSN healthcare monitoring system can improve care. 57.4% of the users agreed that the WSN healthcare monitoring system was easy to use. However, 62.9% of the users agreed that it was easy to become familiar with the WSN healthcare monitoring system only after learning and practice. Finally, 135 (83.3%) users were willing to use the WSN healthcare monitoring system.

4. Conclusion

Based on the seamless transfer from a conventional workflow to a new one, the WSN healthcare monitoring system was found acceptable by the nursing home caregivers in this study. As opposed to the conventional procedure in which a caregiver has to spend time collecting daily vital signs data from residents, the new WSN healthcare monitoring system collected all the needed vital signs data automatically, emphasizing that caregivers do not need to change their work habits to use the new technology. This is a key point in whether a new system can be successfully introduced. Additionally, it was demonstrated that the system could provide near real time performance and was sufficient for nursing home use. Adequate distance in the deployment between sensor nodes and specially designed indicators guaranteed the data to be correct in its transmission and storage in the database server. The study also convinced users to trust the new WSN healthcare monitoring system, especially following a comparison between the new measurement method and the conventional one. Moreover, a clear SOP for the measurement of vital signs gave the users confidence in conducting care work, which made the system more accessible. With the use of the WSN healthcare monitoring system, the extent of nursing services for residents actually remained the same. However, the residents felt that higher quality services were provided. In conclusion, with the WSN healthcare monitoring system, the caregivers' load of routine work could be reduced and quality of care could be improved. The results showed that the WSN healthcare monitoring system was applicable for use in a nursing home. After the healthcare monitoring system has been implemented successfully in nursing homes, elderly people and their families are likely to accept the healthcare monitoring system for use at home. The most important issue is to retain the linkage of nursing services provided by nursing homes and medical services provided by clinics or hospitals.

Acknowledgments

The authors would like to thank the National Science Council of China for financially supporting this research under Contract no. NSC 100-2221-E-166-008- and NSC 101-2221-E-166-009-. In addition, the authors thank Lin Tsai-Lien, the Director of the Haechun Nursing Home, for her help and suggestions throughout this study.

References

[1] Statistics from Statistical Bureau, Ministry of Interior, Executive Yuan, 2012, http://www.moi.gov.tw/stat/index.asp.

[2] "The World Health Report 2006," World Health Organization, 2006, http://www.who.int/whr/2006/en/index.html.

[3] G. Virone, A. Wood, L. Selavo et al., "An advanced wireless sensor network for health monitoring," in *Proceedings of the Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare*, April 2006.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[5] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, 2004.

[6] X. Wang, "Wireless mesh networks," *Journal of Telemedicine and Telecare*, vol. 14, no. 8, pp. 401–403, 2008.

[7] H. Cao, V. Leung, C. Chow, and H. Chan, "Enabling technologies for wireless body area networks: a survey and outlook," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 84–93, 2009.

[8] S. J. Hsu, C. H. Chen, S. H. Chen, W. T. Huang, Y. J. Chang, and Y. Y. Chen, "Conserving bandwidth in a wireless sensor network for telemedicine application," *Intelligent Automation and Soft Computing*, vol. 16, no. 4, pp. 537–551, 2010.

[9] C. C. Lin, P. Y. Lin, P. K. Lu, G. Y. Hsieh, W. L. Lee, and R. G. Lee, "A healthcare integration system for disease assessment and safety monitoring of dementia patients," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 5, pp. 579–586, 2008.

[10] H. J. Lee, S. H. Lee, K. S. Ha et al., "Ubiquitous healthcare service using Zigbee and mobile phone for elderly patients," *International Journal of Medical Informatics*, vol. 78, no. 3, pp. 193–198, 2009.

[11] C. C. Lin, R. G. Lee, and C. C. Hsiao, "A pervasive health monitoring service system based on ubiquitous network technology," *International Journal of Medical Informatics*, vol. 77, no. 7, pp. 461–469, 2008.

[12] C. C. Lin, M. J. Chiu, C. C. Hsiao, R. G. Lee, and Y. S. Tsai, "Wireless health care service system for elderly with dementia," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 696–704, 2006.

[13] S. Fischer, T. E. Stewart, S. Mehta, R. Wax, and S. E. Lapinsky, "Handheld computing in medicine," *Journal of the American Medical Informatics Association*, vol. 10, no. 2, pp. 139–149, 2003.

[14] H. Cao, V. Leung, C. Chow, and H. Chan, "Evaluation of the practical feasibility and acceptability of home monitoring in residential homes," *Journal of Telemedicine and Telecare*, vol. 11, no. 1, supplement, pp. 29–31, 2005.

[15] M. Alwan, S. Dalal, D. Mack et al., "Psychosocial impact of monitoring technology in assisted living: a pilot study," *Assisted Living Consult*, vol. 2, no. 6, pp. 21–25, 2006.

[16] R. Steele, A. Lo, C. Secombe, and Y. K. Wong, "Elderly persons' perception and acceptance of using wireless sensor networks to assist healthcare," *International Journal of Medical Informatics*, vol. 78, no. 12, pp. 788–801, 2009.

[17] Y. J. Chang, L. F. Lin, C. H. Chen, and W. T. Huang, "An integrated service model for community cre: designing a healthy life for the elderly in Taiwan," *Public Health Frontier*, vol. 1, no. 2, pp. 1–12, 2012.

[18] B. M. Muir, "Trust between humans and machines, and the design of decision aids," *International Journal of Human-Computer Studies*, vol. 27, no. 5-6, pp. 527–539, 1987.

[19] X. Ferré, N. Juristo, H. Windl, and L. Constantine, "Usability basics for software developers," *IEEE Software*, vol. 18, no. 1, pp. 22–29, 2001.

[20] Y. J. Chang and W. T. Huang, "A novel design of data-driven architecture for remote monitoring and remote control of sensors over a wireless sensor network and the Internet," *Journal of Internet Technology*, vol. 12, no. 1, pp. 129–138, 2011.

[21] C. C. Tuan, C. H. Chang, Y. J. Chang, C. H. Chen, H. D. Jeong, and W. T. Huang, "A highly reliable platform with a serpentine antenna for IEEE 802.15.4 over a wireless sensor network," *WSEAS Transactions on Circuits and Systems*, vol. 11, no. 6, pp. 182–197, 2012.

[22] W. T. Huang, J. T. Lin, C. H. Chen, Y. J. Chang, and Y. Y. Chen, "Accurate analysis of IEEE 802.15.4 slotted CSMA/CA over a real-time wireless sensor network," *Lecture Notes of the Institute for Computer Sciences*, vol. 24, pp. 98–110, 2010.

[23] D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, 1991.

[24] C. C. Lin, M. J. Chiu, C. C. Hsiao, R. G. Lee, and Y. S. Tsai, "Wireless health care service system for elderly with dementia," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 696–704, 2006.

[25] K. I. Hwang, J. In, N. K. Park, and D. S. Eom, "A design and implementation of wireless sensor gateway for efficient querying and managing through world wide web," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1090–1097, 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 381726, 12 pages
doi:10.1155/2012/381726

Research Article

Environment-Aware CMT for Efficient Video Delivery in Wireless Multimedia Sensor Networks

Yuanlong Cao,[1] Changqiao Xu,[1,2] Jianfeng Guan,[1] Fei Song,[3] and Hongke Zhang[1,3]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Wuxi 214028, China*
[3] *National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Changqiao Xu, cqxu@bupt.edu.cn

Received 12 June 2012; Accepted 30 September 2012

Academic Editor: Chin-Feng Lai

Wireless Multimedia Sensor Networks (WMSNs) have gained significant attention with capabilities of retrieving video and audio streams, still images, and scalar sensor data but with challenge of high data loss. As more and more wireless devices are equipped with multiple network interfaces, multimedia delivery over multipath has been cognized as a more promising approach in wireless transmission. In this paper, based on the Concurrent Multipath Transfer (CMT) extension for Stream Control Transport Protocol (SCTP), we propose a novel environment-aware CMT (*e*-CMT) to overcome the high data loss challenge, as well as "a hot potato" congestion problem in wireless transmission. The *e*-CMT provides environment-aware cognitive ability for efficient video delivery with three modules, which are Path Quality-aware Model (PQM) that devotes to estimate path quality and select candidate path using for retransmission, Adaptive Retransmission Trigger (ART) that contributes to cognize packet loss and trigger efficient retransmission behaviors, and Congestion-avoid Data Distributor (CDD) that serves to enable *Partial-Reliable* retransmission to mitigate congestion condition. We design a close realistic topology to present how the *e*-CMT outperforms the original CMT for efficient video delivery over multihomed WMSNs.

1. Introduction

Due to the advances in low-cost and low power consumption, Wireless Sensor Networks (WSNs) have gained variety of attentions and resulted in thousands of peer-reviewed publications. Significant results in this area have enabled multiple applications used in military and civilian. Most of researches and deployments on WSNs are concerned with scalar sensor networks that measure physical phenomena, such as temperature, pressure, humidity, or location of objects that can be transferred through low-bandwidth and delay-tolerant data streams. In general, WSNs are designed with purpose of data-only delay-tolerant applications without high bandwidth requirements [1].

The growing availability of low-power wireless networking technologies and low-cost multimedia devices such as Complementary Metal-Oxide Semiconductor (CMOS) cameras and microphones, which can acquire rich-content media from the environment like images and videos, provides the opportunity for development and deployment of distributed Wireless Multimedia Sensor Networks (WMSNs) which has capabilities of retrieving video and audio streams, still images, and scalar sensor data [2]; these capabilities make WMSNs can be applied to many fields such as video surveillance, traffic avoidance and so on. However, most researches on WMSNs mainly focus on the design of network layer and Medium Access Control (MAC) layer [3], they seldom consider the design of multipath transfer way in transport layer to enhance the performance of video content delivery over multihomed WMSNs.

As more and more wireless multimedia devices are equipped with multiple network interfaces, it becomes

increasingly common for a wireless video device to be connected to more than one access networks employing either a homogenous technology or heterogeneous, the multihomed technology is becoming an important technology in wireless transmission. Due its feature of multihoming, Stream Control Transport Protocol (SCTP) [4] shows its advantages on the serious data loss nature of wireless networks, and its performance adopted in multimedia streaming services over-multihomed wireless networks has been studied widely [5]. Thus, SCTP will become a promising transport protocol for video delivery over multihomed WMSNs. Figure 1 illustrates the multihomed SCTP based wireless multimedia sensor networks.

Video transport usually has stringent bandwidth, delay, and loss requirements due to its nature of real time [6]. To improve the performance of video content delivery over multihomed WMSNs, it is important to distribute data across all available paths to achieve high users' of quality of experience. Concurrent Multipath Transfer (CMT) [7] extension for SCTP (CMT-SCTP), further referred to as CMT, uses the SCTP's multihoming feature to distribute data across multiple end-to-end paths in a multihomed SCTP association and maintains more accurate information (such as available bandwidth and RTT) of all the paths [8]. These features make CMT attract more and more studies for the video delivery under stringent bandwidth, delay, and loss wireless transmission. However, the original CMT lacks the capability of cognizing wireless link condition; this disadvantage makes CMT cannot enable an adaptive retransmission trigger mechanism to provide a best services for the high data loss nature of WMSNs.

Motivated by fact that the design of efficient video content delivery over multipath will be an urgent needs in the future WMSNs, this paper proposes a novel environment-aware CMT (*e*-CMT) with considering network condition and the causes of the condition change to provide an efficient video delivery approach over multihomed WMSNs. The *e*-CMT is constructed by three modules, which are Path Quality-aware Model (PQM) that devotes to estimate path quality and select candidate path using for retransmission, Adaptive Retransmission Trigger (ART) that contributes to cognize packet loss and trigger efficient retransmission behaviors, and the further proposed Congestion-avoid Data Distributor (CDD) that serves to enable *Partial-Reliable* retransmission to overcome "a hot potato" congestion problem in wireless transmission.

The rest of the paper is organized as follows. In Section 2 a brief description of related work is given. Section 3 gives an overview of the *e*-CMT. Section 4 details the *e*-CMT design. Section 5 evaluates and analyzes the performance of the *e*-CMT. Section 6 concludes the paper and gives our future work.

2. Related Work and Contributions

Multimedia data delivery over multipath has been cognized as a more promising resolution to overcome the challenges such as high-rate required and time-sensitive delivery in WMSNs, especially for video streams due to its real-time nature and usually has stringent bandwidth, delay, and loss requirements. Felemban et al. [9] proposed a novel packet delivery mechanism called Multi-Path and Multi-SPEED Routing Protocol (MMSPEED) for probabilistic Quality of Service (QoS) guarantee in WSNs. Politis et al. [10] proposed a power efficient multipath video packet scheduling scheme for minimum video distortion transmission over WMSNs, they proved that the transmission of video packets over multiple paths in WSNs can improve the aggregate data rate of the network and minimizes the traffic load handled by each node.

Video content transfer over multihomed SCTP-based WSNs is becoming an attractive research topic. Stephan et al. [11] proposed a novel approach that consists of bundling SCTP-based multiple connections at the transport layer on the gateway to improve the reliability by employing redundancy in WSNs. Qiao et al. [12] proposed a multihomed SCTP-based Body Sensor Networks (BSNs) framework and investigate handover strategies during sensor nodes' movement to increase data reliability for BSNs. Lu and Wu [13] studied the performance of SIP based attractive services using SCTP over WSNs. However, these researches mentioned above cannot utilize the capabilities of the multihomed technology because of SCTP's single-path transfer way.

As an extension of SCTP, CMT has been recognized as a good protocol for multimedia content delivery over multihomed wireless networks with ability of flows across multiple interfaces. Our previous work [8, 14, 15] investigated the performance of multimedia data transfer using CMT over multihomed wireless networks with the designed Evalvid-CMT platform. Huang and Lin [16] proposed PR-CMT to provide a timed reliable service for multimedia data transfer. All previous work showed that CMT can achieve a high users' experience of quality for multimedia streaming services in wireless transmission.

One of the most challenges in wireless transmission is its nature of high data loss. In order to cope with lost packets to enhance the performance of CMT, Iyengar et al. [7] proposed five retransmission policies for the original CMT. Huang and Lin [17] proposed RG-CMT with goal of providing a fast retransmission for concurrent multipath data transfer over wireless vehicular networks. With RG-CMT, lost packets can be fast retransmitted from the relay gateway to the vehicle. Cui et al. [18] proposed a Fast SACK (FSACK) scheme which can be applied to both SCTP and CMT. With FSACK, the sender can select the optimal return path which serves the data delivery or retransmission. Shailendra et al. [19] proposed an additional 32-bit Path Sequence Number (PSN) attached in the payload chunk header and SACK to provide the flexibility to retransmit the lost chunk and achieve a more benefit multimedia delivery in CMT.

Combing aforementioned, we note that current CMT researches only focus on how to improve the performance of CMT based on the original CMT's fast retransmission trigger regardless of the network condition change and the causes of change. This disadvantage makes CMT cannot really achieve the desired performance in wireless transmission.

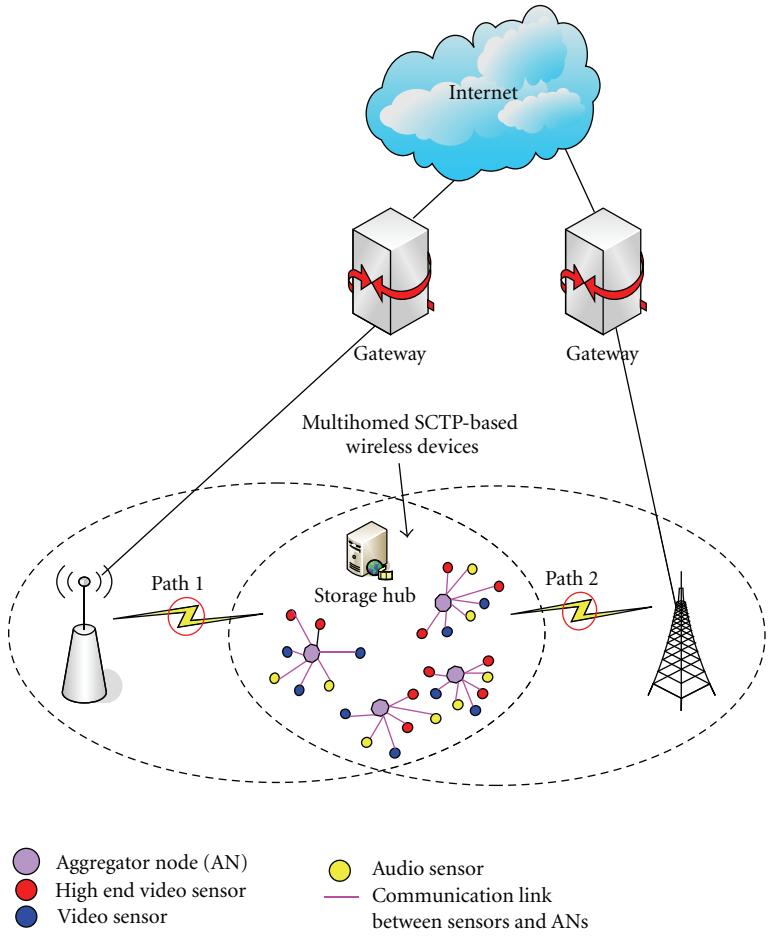


Figure 1: Multihomed SCTP-based video delivery over wireless multimedia sensor network.

This paper makes important contributions against the state of art of the literature in the following three stages:

(i) introducing an accurate path quality-aware model to sense each path's current transmission status and the causes of transmission condition change;

(ii) designing an efficient packet loss sensor and further proposes an adaptive retransmission trigger mechanism to handle the high packet loss challenge in WMSNs;

(iii) introducing an intelligent data distribution strategy to deliver data accordance with each path's current transmission quality and mitigate network congestion if any.

3. *e*-CMT Overview

As mentioned in the related work section above, current researches on CMT do not consider network condition and the causes of condition change. Moreover, they seldom consider the characteristics of multimedia data during protocol design. These shortages make CMT cannot provide an efficient transmission behavior to achieve the desired performance in wireless transmission.

To make CMT support a best service for efficient multimedia content transfer over multihomed WMSNs, this paper pays attention on following three research problems.

(i) How to design a newly path quality-aware sensor to distinguish the transmission quality of each path and the causes of transmission condition change.

(ii) How to design an adaptive retransmission trigger in compliance with real-time condition to handle the high packet loss challenge of WMSNs.

(iii) How to make CMT support a self-aware cognitive capability for efficient multimedia content transfer over multihomed WMSNs.

With purposes mentioned above, we propose *e*-CMT, a novel environment-aware concurrent multipath data transfer approach for efficient multimedia delivery in WMSNs. Figure 2 shows an overview of the *e*-CMT. The *e*-CMT consists of three modules, which are dubbed as Path Quality-aware Model (PQM), Adaptive Retransmission Trigger (ART), and Congestion-avoid Data Distributor (CDD), respectively. All the three modules are implemented at the CMT sender side. Following descriptions introduce the responsibilities of the three modules simply, as well as how to make adequate collaboration with each other.

PQM: PQM is in charge of providing a path quality-aware model, selecting candidate path for retransmission, and providing path condition information for ART and CDD.

ART: ART is responsible for analyzing the causes of path condition change, sensing packet loss, and triggering adaptive retransmission behaviors accordance with network condition.

CDD: CDD aims at cognize the condition of candidate path, determinates available data chunk and decides whether to enable Partial-Reliable retransmission to mitigate congestion condition.

Once a received Selective Acknowledgment (SACK) indicates that a data chunk is missing, the *e*-CMT starts efficient multimedia content retransmission complying with the following below steps.

(1) The *e*-CMT sender enables the ART to cognize the path condition where the unacknowledged chunk comes from and trigger a proper retransmission behavior.

(2) The *e*-CMT sender starts the PQM to evaluate the quality of all available path within an SCTP association, then selects candidate path for retransmission.

(3) Before retransmission, the *e*-CMT sender uses the CDD to decide whether to enable *Partial-Reliable* or not accordance with cognizing the condition of candidate path.

By adequate communication and collaboration among PQM, ART and CDD, the *e*-CMT can act as a well-transport protocol for video data transfer in multihomed WMSNs with capabilities of cognizing path condition, triggering efficient retransmission behavior, and reducing deteriorated congestion.

4. *e*-CMT Details Design

This section details how the proposed *e*-CMT enables environment-aware cognitive feature for efficient video delivery over multihomed WMSNs. We first introduce the PQM which is used to sense paths' quality and select candidate path for retransmission. We then address how the ART working for analyzing the causes of path condition change and triggering adaptive retransmission for the lost packets. Finally we examine how the CDD devotes to congestion mitigation by making intelligent decision whether give up retransmission or not.

4.1. Path Quality-Aware Model (PQM). Most works on multimedia content transfer use *Available Bandwidth* (AB) [20–22] to evaluate the path quality. A well-known AB model [20] consists of *Round Trip Time* (RTT) and *Packet Loss Rate* (PLR) which can be expressed as

$$\mathrm{AB} = \frac{1.22 \times \mathrm{MTU}}{\mathrm{RTT} \times \sqrt{\mathrm{PLR}}}, \tag{1}$$

where MTU is the Maximum Transmission Unit. PLR can be evaluated by the two-state discrete Markov Chain known as Gibert's Model detailed in [20–22], while RTT can be estimated by

$$\mathrm{RTT} = \alpha \times \overline{\mathrm{RTT}} + (1 - \alpha) \times (t - T_{\mathrm{send}} - \Delta T), \tag{2}$$

where $\overline{\mathrm{RTT}}$ denotes the current round trip time of path, t is the timestamp on behalf of the time at which the packet ACK is received at the sender, α is a weighting parameter with a common value of 0.875, T_{send} is for the timestamp the packet sending time, and ΔT is the time interval of a packet handling time at the receiver.

Since RTT is a good parameter to reflect the end-to-end path congestion condition and PLR can work accurately for packet loss evaluation occurred by both congestion and link error. Thus, we just consider RTT and PLR to define a Path Quality-aware Model (PQM) for multihomed SCTP hosts which can be expressed by

$$M_i = \frac{1}{\mathrm{RTT}_i \times \sqrt{\mathrm{PLR}_i}}, \tag{3}$$

which M_i works for sensing the quality of the ith end-to-end path, RTT_i is used to reflect the congestion condition, and PLR_i for estimating packet loss of the ith end-to-end path. Path with the greatest value of M denotes that it is the one that has the best path quality.

For (3), there are three variables, namely, ΔM_i, $\Delta\mathrm{RTT}_i$ and $\Delta\mathrm{PLR}_i$. So, we have

$$\begin{aligned}
\Delta M_i^{\mathrm{RTT}_i} &= \frac{\partial M_i}{\partial \mathrm{RTT}_i} \times \Delta\mathrm{RTT}_i = \frac{-\Delta\mathrm{RTT}_i}{(\mathrm{RTT}_i)^2 \times \sqrt{\mathrm{PLR}_i}}, \\
\Delta M_i^{\mathrm{PLR}_i} &= \frac{\partial M_i}{\partial \mathrm{PLR}_i} \times \Delta\mathrm{PLR}_i = \frac{-\Delta\mathrm{PLR}_i}{2\mathrm{RTT}_i \times \left(\sqrt{\mathrm{PLR}_i}\right)^3}, \\
\Delta M_i &= \Delta M_i^{\mathrm{RTT}_i} + \Delta M_i^{\mathrm{PLR}_i},
\end{aligned} \tag{4}$$

where $\Delta M_i^{\mathrm{RTT}_i}$ is the fluctuation on ΔM_i occurred by $\Delta\mathrm{RTT}_i$, and $\Delta M_i^{\mathrm{PLR}_i}$ is the fluctuation on ΔM_i occurred by $\Delta\mathrm{PLR}_i$.

Combining RTT and PLR, the PQM cannot only select candidate path served for efficiently retransmission but also enable an important capability of distinguishing the causes of the path quality change. This feature is very useful for the intelligent retransmission scheme design.

4.2. Adaptive Retransmission Trigger (ART). An efficient retransmission trigger in CMT would significantly reduce the issue of packet loss in the high packet loss nature of wireless networks. In the original CMT, when a Transmission Sequence Number (TSN) is reported missing indications τ ($\tau = 4$) times, the original CMT sender marks it for fast retransmission, then the data chunk will be retransmitted over an available destination that has the largest cwnd value (selected by the default RTX-CWND retransmission policy). As it is shown in Figure 3, the received blue chunks will wait, while the red one is missing and waiting for four missing reports to trigger its retransmission. Actually, it would achieve a more benefit if a lost packet can be cognized

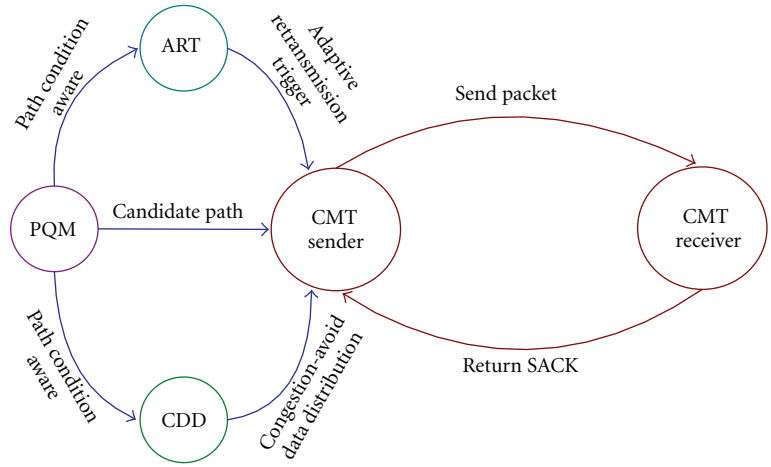


Figure 2: An overview of the *e*-CMT.

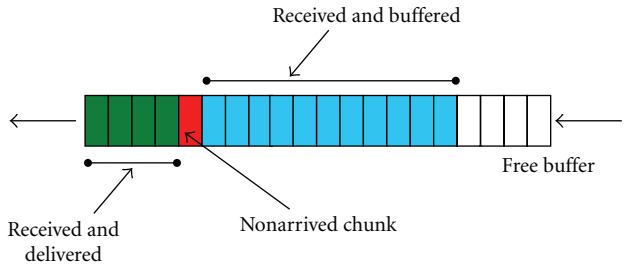


Figure 3: Receiver buffer contains a unacknowledged chunk.

timely and further be retransmitted rapidly, especially for chunks which are time sensitive such as video streaming.

To enable the cognitive feature for sensing data loss accurately and efficiently, once having a missing reported by the Selective Acknowledgment (SACK), the *e*-CMT sender further cognizes the network condition for corresponding path (denoted as *LossOccurPath*) that the missing chunk comes from. Thereby, the *e*-CMT sender can make a more accurate decision if trigger retransmission behavior or not in compliance with the detected network condition. For example, if a deteriorated unreliable network condition (with high packet loss) is detected for the *LossOccurPath*, the *e*-CMT sender will mark the data chunk with τ (τ is an integer and $1 \leq \tau < 4$) missing indications for triggering fast retransmission. Therefore, as long as the data chunk's missing report reaches τ times, the *e*-CMT sender will trigger retransmission behavior to resend the chunk instantly over a candidate path selected by PQM, rather than the four missing reports the original CMT does.

To make the *e*-CMT sender support an environment-aware capability, four network conditions are defined based on (4) for the transmission paths within the SCTP association, which can be expressed as the following.

(i) Network Condition-1 (NC-1): if ΔM_i decreases while $|\Delta M_i^{\mathrm{RTT}_i}| \geq |\Delta M_i^{\mathrm{PLR}_i}|$, it indicates that the transmission quality of the ith link is deteriorated mostly associated with congestion condition.

(ii) Network Condition-2 (NC-2): if ΔM_i decreases while $|\Delta M_i^{\mathrm{RTT}_i}| < |\Delta M_i^{\mathrm{PLR}_i}|$, it denotes that the transmission quality of the ith link is deteriorated mostly associated with unreliable condition, which occurred by wireless link error.

(iii) Network Condition-3 (NC-3): if ΔM_i increases while $|\Delta M_i^{\mathrm{RTT}_i}| \leq |\Delta M_i^{\mathrm{PLR}_i}|$, it means that the transmission quality of the ith link becomes better mainly benefits from congestion relief.

(iv) Network Condition-4 (NC-4): if if ΔM_i increases while $|\Delta M_i^{\mathrm{RTT}_i}| > \Delta M_i^{\mathrm{PLR}_i}|$, it signifies that the transmission quality of the ith link becomes better mainly benefits from relieved unreliable condition.

To make sure the *e*-CMT do not perform worse than the original CMT, for NC-1, NC-3, and NC-4 condition, the *e*-CMT sender marks the DATA chunk with τ ($\tau = 4$) missing indications for retransmission like the original CMT. But for NC-2 case, the *e*-CMT sender will enable fast retransmission once τ ($1 \leq \tau < 4$) times missing indication(s) is (are) reached. Figure 4 portrays the state model of the cognitive retransmission trigger used in the *e*-CMT.

Once the *e*-CMT sender receives a SACK that indicates that a TSN (e.g., TSN_k) is missing, it will enable following workflows to support efficient retransmission.

(1) The *e*-CMT sender inquires the destination (denoted as d_l) where the TSN_k comes from.

(2) The *e*-CMT sender calculates M_{d_l}, $|\Delta M_{M_{d_l}}^{\mathrm{RTT}_{d_l}}|$ and $|\Delta M_{M_{d_l}}^{\mathrm{PLR}_{d_l}}|$ by (3) and (4) to estimate current network state of the d_l.

(3) If the network condition of d_l is NC-2, the *e*-CMT sender marks the DATA chunk with τ ($1 \leq \tau < 4$) missing indication(s) for retransmission, else, it marks the DATA chunk with τ ($\tau = 4$) missing indications for retransmission.

Definition:
d_i: the ith destination within the SCTP association
d_{list}: the active destination list of core node
M_i: the quality metric value of the ith destination
d_{send}: the candidate path selected to resend data chunk

Once a SACK contained Gap Ack Reports is received by the e-CMT sender, the sender will loop each destination d_i then

(1) **for** each TSN TSN_k, set count $\text{MC}_k = 0$;
(2) **if** TSN_k belongs to d_l, **then**
(3) MC_k++;
(4) analyzes the condition of d_l by (3) and (4);
(5) **if** the d_l is in NC-2 condition **then**
(6) marks TSN_k with τ $(1 \le \tau < 4)$ miss indication(s) for retransmission;
(7) **else**
(8) marks TSN_k with $\tau = 4$ miss indication(s) for retransmission;
(9) **end if**
(10) **if** $(\tau == \text{MC}_k)$ **then**
(11) **for** all destination d_i:
(12) **if** status of d_i == ACTIVE, puts d_i into d_{list};
(13) **for** $(i = 1, i \le \text{count}(d_{\text{list}}), i++)$, calculates M_i;
(14) sorts d_i in an descending order according to M_i;
(15) sets $d_{\text{send}} = d_{\text{list}(0)}$;
(16) **while** cwnd of d_{send} is full, sets $d_{\text{send}} = d_{\text{send}} \rightarrow$ next;
(17) the e-CMT sender distributes the chunk over d_{send}.
(18) **end if**
(19) **end if**

Algorithm 1: Adaptive retransmission trigger in the e-CMT.

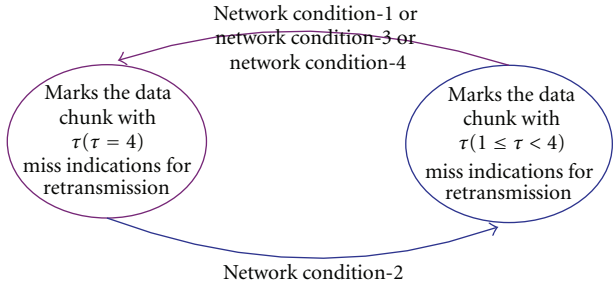


Figure 4: The state model of the cognitive retransmission trigger designed in the e-CMT.

(4) If the TSN_k reaches τ times missing indication(s), the e-CMT sender starts PQM to evaluate the quality of available paths corresponding to each destination i within an SCTP association.

(5) The e-CMT sender sorts the destinations in a descending order of its measured M_i then selects the first destination, namely with the largest M_i in the list, as *candidate destination* (denoted as d_{send}).

(6) Before retransmission, the e-CMT sender verifies the cwnd of this path. If the cwnd is full, the next destination in the list will be selected as the d_{send}.

(7) The e-CMT sender distributes the TSN_k chunk over the d_{send}.

The Algorithm 1 details the adaptive retransmission trigger in the e-CMT.

4.3. Congestion-Avoid Data Distributor (CDD). How to provide a best service for the real-time characteristic of video streaming services in congestion condition is still a challenge. This subsection addresses a Congestion-aware Data Distributor (CDD) to overcome "a hot potato" congestion problem in wireless transmission and to enhance the quality of multimedia content delivery. With the CDD, the e-CMT can provide *Partial-Reliable* transmission when severe network congestion occurs. That is, Once having a DATA chunk is distributed over the d_{send}, the e-CMT sender will sense transmission condition of the d_{send}. If the congestion condition NC-1 is detected, the e-CMT sender will not retransmit the video data chunk when their playing deadline is overdue. With this feature, the e-CMT sender can mitigate the deteriorated network congestion condition to achieve high users experience of quality for multimedia streaming services.

To make an efficient congestion-avoid data distribution strategy happen, in frame level, we define that only *available frame* can be retransmitted in the CDD if severe congestion condition is detected. A retransmission-required multimedia frame is an *available frame* if it can be received by the receiver before its playout time. Vice versa, a retransmission-required multimedia frame is an *unavailable frame* if it cannot be received by the receiver before its playout time.

An *unavailable frame* will be given up its retransmission by the *e*-CMT sender for congestion mitigation.

Below expression (5) can be used to verify whether the retransmission-required multimedia frame is an *available frame* or not before its delivery on the selected candidate path d_{send}:

$$T_{\text{leftTime}} = T_{\text{lifeCycle}} - \text{Current Time},$$
$$T_{\text{leftTime}} > \frac{\text{RTT}_{d_{\text{send}}}}{2}, \tag{5}$$

which T_{leftTime} represents the left time of multimedia frame for retransmission. $T_{\text{lifeCycle}}$ stands for the lifetime of the retransmission-required multimedia frame. The multimedia frame is an available frame and can be transferred over the d_{send} as long as its T_{leftTime} value is greater than $\text{RTT}_{d_{\text{send}}}/2$.

With the CDD, the *e*-CMT will enable following workflows to reduce congestion once a retransmission is distributed over the d_{send}:

(1) the *e*-CMT sender estimates network condition of the d_{send} using (3) and (4);

(2) if the network condition is recognized as NC-1, namely, $M_{d_{\text{send}}}$ decreases while $|\Delta M_{d_{\text{send}}}^{\text{RTT}_{d_{\text{send}}}}| \geq |\Delta M_{d_{\text{send}}}^{\text{PLR}_{d_{\text{send}}}}|$, the *e*-CMT sender further verifies if the multimedia frame is an *available frame* or not using (5);

(3) if the frame is verified as an *available frame*, the *e*-CMT sender retransmits it over the d_{send};

(4) if the frame's left time is less than $\text{RTT}_{d_{\text{send}}}/2$, it can be recognized as an *unavailable frame*. Correspondingly, the *e*-CMT sender gives up its retransmission.

The Algorithm 2 details the congestion-avoid data distributor in the *e*-CMT.

5. Simulation and Analysis

Internet measurement studies showed complex behaviors of Internet traffic [22, 23] that are necessary for performance evaluation of network protocols. Therefore, without background traffic, the performance evaluation of CMT cannot fully investigate the CMT's behaviors that are likely to be observed when it is deployed in wireless environments such as WANs and WMSNs. We consider a more close realistic simulation topology including reasonable background traffic, and then we present sufficient performance evaluations for the *e*-CMT based on the designed simulation topology.

5.1. Background Traffic Design. Accordance with the Internet survey [24], TCP traffic on the Internet is about 80%–83%, and UDP traffic is about 17%–20%. In addition, the content-rich multimedia streaming will be the most attractive services in the future networks, more and more multimedia content encoded by VBR will be deployed in the Internet. Thus, a more reasonable background traffic that consists of TCP traffic and UDP traffic (TCP : UDP is 4 : 1) should be taken into account to evaluate the performance of data distribution over concurrent multipath.

Table 1: Parameters used in VBR traffic generator.

Variables	Values
Application/Traffic/VBR set rate_	448 Kb
Application/Traffic/VBR set random_	0
Application/Traffic/VBR set maxpkts_	268435456
Application/Traffic/VBR set maxSize_	200
Application/Traffic/VBR set minSize_	100
Application/Traffic/VBR set intervaltime_	200

Since NS2 [25] still cannot support VBR traffic, we enable VBR traffic generator into NS2 by adding *PT_VBR* as packet enumeration and then setting VBR for *PT_VBR*'s value in packet information function. The parameters used for VBR traffic generator are set as Table 1. Parameters of TCP traffic generator and UDP traffic generator used in our experiments just adopt the default values which are provided by NS2.

5.2. Simulation Topology Setup. Nowadays the most frequently used standard for variety of attractive services such as Video-on-Demand (VoD) and Internet Protocol Television (IPTV) over multihomed wireless networks is IEEE 802.11 [26, 27]. As it is shown in Figure 1, we assume that both Aggregator Node (AN) and Storage Hub (SH) in WMSNs are equipped with two IEEE 802.11 interfaces (which have also been discussed by Misra et al. [28]). And the AN acts as the SCTP sender to transfer multimedia content to the SH (the SCTP receiver), we compare and analyze the performance between the *e*-CMT and the original CMT in detail. We believe that the *e*-CMT can present the same behaviors in other proposed standards such as IEEE 802.15.4 [29].

The simulation topology is shown in Figure 5 and includes the SCTP sender and receiver. Both CMT endpoints have two wireless 802.11b interfaces of 11 Mbits/s at 2.4 GHz. To avoid interferences occurance between the two 802.11b interfaces, both of them are assigned to tow different channels. The Maximum Transmit Unit (MTU) of each path is 1500B. The queue length in both paths is 50 packets. Default receive buffer (rbuf) values in commonly used in operating systems today vary from 32 KB to 64 KB and beyond. So, we compare the performance of the two approaches with an rbuf value of 32 KB, 64 KB, 128 KB, and 256 KB, respectively.

We introduce a realistic Internet traffic gathered from BUPT consisted of FTP traffic, VBR traffic, and CBR traffic. All FTP/TCP, VBR/UDP, traffic and CBR/UDP traffic generators connect to the two APs (AP1 and AP2), respectively with a reasonable bandwidth value of 100 Mb, and the propagation delay is 5 ms in accordance with [30]. We perform two experimental scenarios dubbed Cases 1 and 2 to study the performance of *e*-CMT, respectively.

Case 1. The loss rate on Path 1 (with FTP/TCP traffic and CBR/UDP traffic, the ratio is 4 : 1) is always kept at 1%, and

Definition:
d_{send}: the selected candidate path for retransmission
M_i: the quality metric value of the ith destination
T_{leftTime}: the left time used for transmission
$T_{\text{lifeCycle}}$: the specified lifetime for the multimedia frame
Mf_k: a retransmission-required multimedia frame

Once having a multimedia retransmission (Mf_k) is distributed over d_{send}, at the e-CMT sender side,

(1) The sender calculates values of $M_{d_{\text{send}}}$, $|\Delta M_{d_{\text{send}}}^{\text{RTT}_{d_{\text{send}}}}|$ and $|\Delta M_{d_{\text{send}}}^{\text{PLR}_{d_{\text{send}}}}|$ by (3) and (4);
(2) **if** $M_{d_{\text{send}}}$ decreases while $|\Delta M_{d_{\text{send}}}^{\text{RTT}_{d_{\text{send}}}}| \geq |\Delta M_{d_{\text{send}}}^{\text{PLR}_{d_{\text{send}}}}|$ **then**
//evaluates Mf_k's left time
(3) sets $T_{\text{leftTime}} = (T_{\text{lifeCycle}} - \text{CurrentTime})$;
(4) **if** $T_{\text{leftTime}} > \text{RTT}_{d_{\text{send}}}/2$ **then**
(5) retransmits the Mf_k;
(6) **else**
(7) gives up Mf_k's retransmission;
(8) **end if**
(9) **end if**

Algorithm 2: Congestion-avoid Data Distributor in the e-CMT.

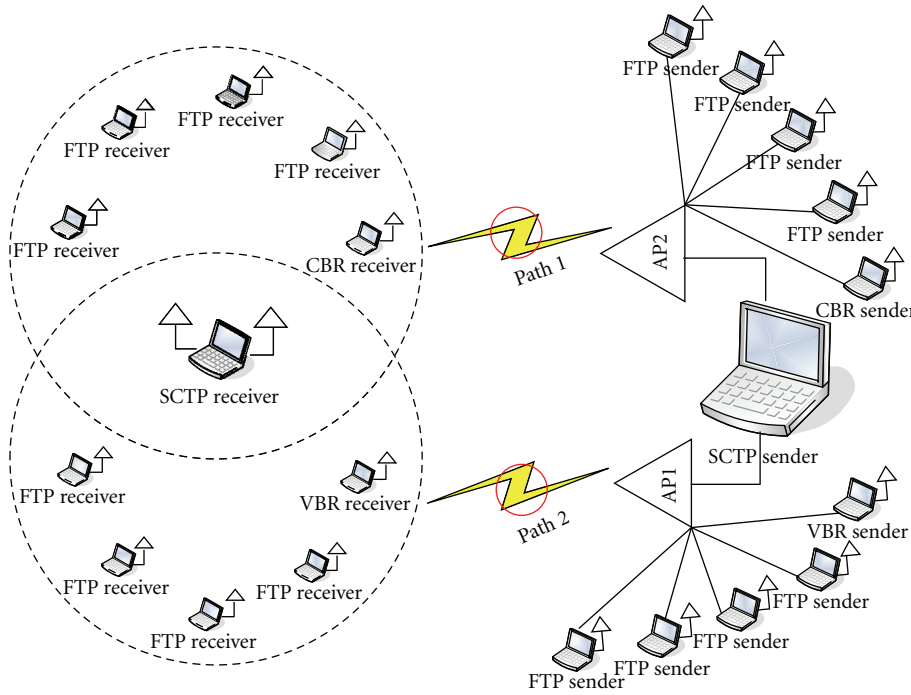


Figure 5: Simulation topology.

on Path 2 (with FTP/TCP traffic and VBR/UDP traffic, the ratio is 4 : 1), it is with varied value at 5%, 10%, 15%, 20%, 25%, and 30%, respectively.

Case 2. The loss rate on Path 2 is always kept at 1%, and on Path 1, it is with varied value at 5%, 10%, 15%, 20%, 25%, and 30%, respectively.

To avoid the false missing report, a well-known challenge in CMT, we just set $\tau = 2$ for the e-CMT when the measured path is detected in NC-2 condition with goal of addressing how advantage of the feature of cognizing and retransmitting a lost packet timely. Our future work will investigate which τ can reach the largest advantage in terms of performance in realistic networks. Our simulation will stop at 60s. Testing

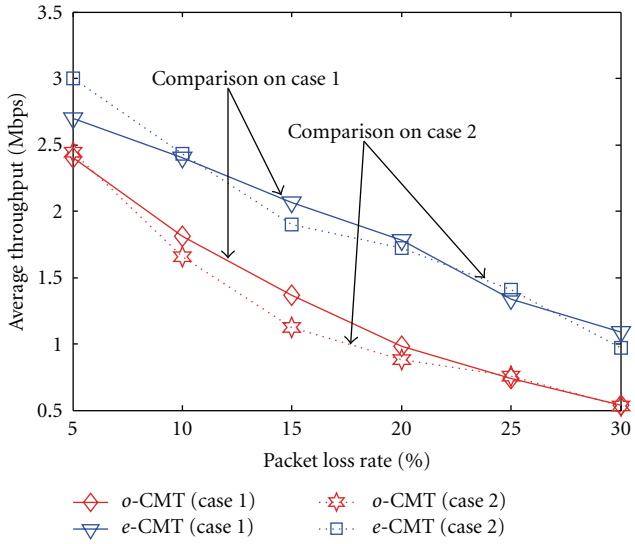


Figure 6: Comparison with an rbuf value of 32 KB.

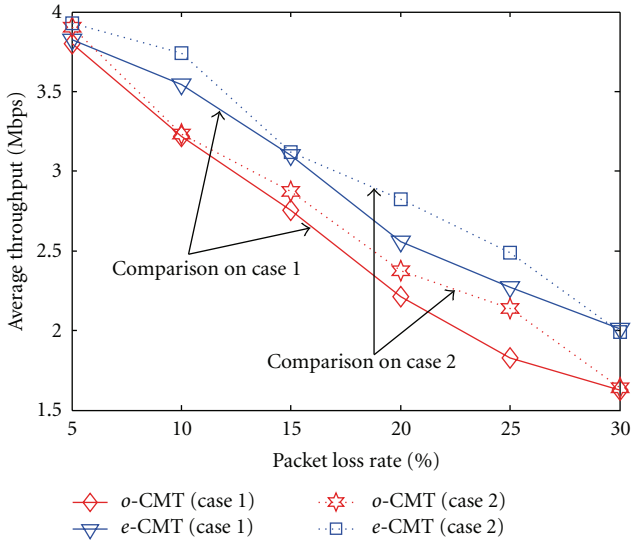


Figure 8: Comparison with an rbuf value of 128 KB.

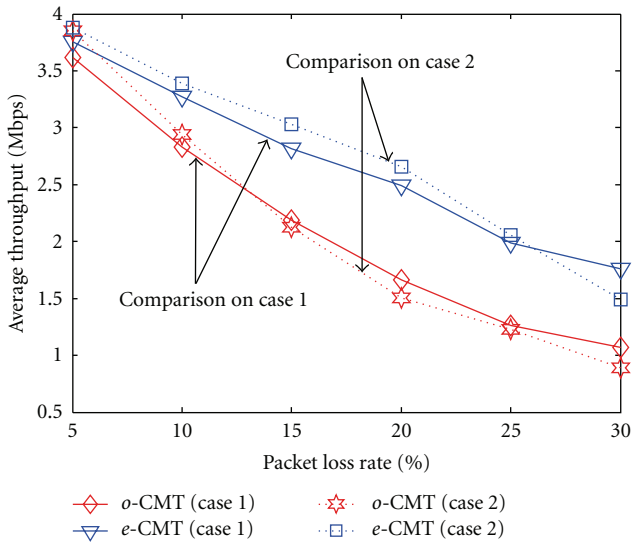


Figure 7: Comparison with an rbuf value of 64 KB.

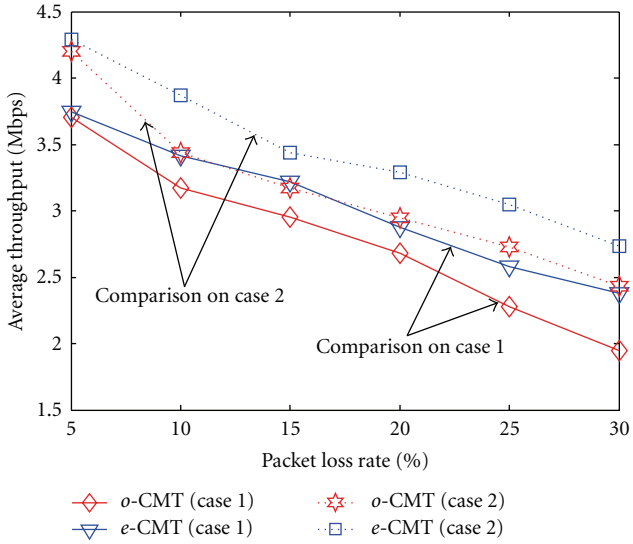


Figure 9: Comparison with an rbuf value of 256 KB.

results are calculated by averaging the results of 50 runs with different seeds.

5.3. Performance Evaluations. Firstly, we present a set of experiments to investigate the performance of *e*-CMT compliance with aforementioned Cases 1 and 2 condition. We compare the performance in terms of average throughput between the *e*-CMT and the original CMT with different rbuf value. Notice that we do not consider multimedia content in those experiments in order to present how *e*-CMT outperforms the original CMT with its cognitive feature for packet loss conveniently. Later experiments will focus on multimedia content to show how the *e*-CMT improves the quality of video data transmission. For convenience, we illustrate the results of original CMT as "*o*-CMT" in test result figures, and the results with our algorithm are illustrated as "*e*-CMT", respectively.

Figures 6, 7, 8, and 9 show the comparisons on average throughput for the *e*-CMT and the original CMT under an rbuf value of 32 KB, 64 KB, 128 KB, and 256 KB, respectively. As shown in those figures, owing to effect side occurred by background traffic, the available bandwidth that is insufficient and that leads to both the original CMT and the *e*-CMT could not achieve a better performance. Moreover, we can note that average throughput of either the original CMT or the *e*-CMT follows a decline in the relationship with the PLR of Path. However, since the *e*-CMT enables the cognitive feature for packet loss and starts fast retransmission behavior to resend the lost packets over candidate path selected by a well metric, thus, it outperforms the original CMT under different rbuf.

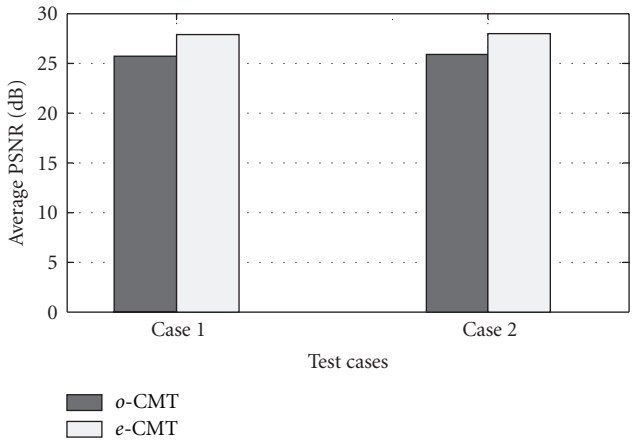


Figure 10: Comparison on PSNR.

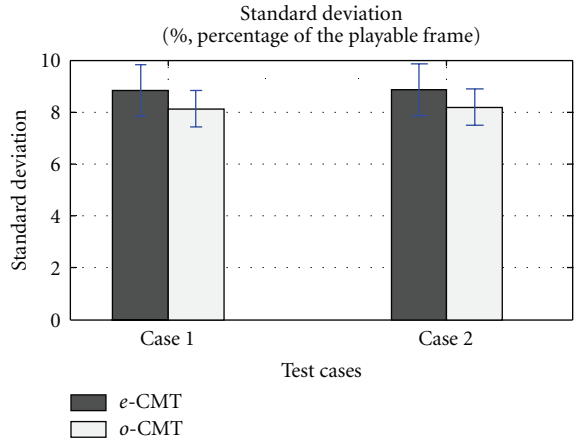


Figure 11: Standard deviations of other video sequences.

From Figure 6 to Figure 9, whatever the packet loss occurs on Path 1 or Path 2, we can figure out some significant phenomenons which are detailed as follows.

(1) As PLR comes higher, the gap between the *e*-CMT and the original CMT comes larger mostly. This phenomenon justifies that cognizing network state and the causes of change are very important for transport protocol. When the link unreliable condition comes deteriorated, these lost packets can be cognized and retransmitted by the *e*-CMT. Moreover, the *e*-CMT considers packet loss rate to construct a more reasonable metric. With the metric, the *e*-CMT can select a more satisfiable candidate path and redeliver lost packets efficiently. This feature makes *e*-CMT reduce the packet loss probability and improve the throughput when it reschedules a packet into a link with a smaller packet loss rate. This is why the *e*-CMT achieves more benefit over the original CMT as PLR on either Path 1 or Path 2 increases.

(2) As rbuf comes larger, the *e*-CMT gains less improvement in terms of average throughput over the original CMT. This phenomenon is reasonable since rbuf increases result in more packets that can be delivered over the wireless link and received by the receiver rapidly. Besides, our previous work [31] clarified that the impact of the background traffic on throughput is increased as the increasing of the rbuf. This reason leads to the *e*-CMT with less significant improvement on average throughput as PLR increases. However, since the *e*-CMT takes the packet loss rate into account during the retransmission to reduce the packet loss rate. Correspondingly, the *e*-CMT still can achieve a better performance over the original CMT.

Secondly, we compare the performance between the *e*-CMT and the original CMT in terms of multimedia data concurrent multipath transfer. To serve a best service for multimedia content delivery, the *e*-CMT will enable both the ART feature examined in Algorithm 1 and the CDD feature examined in Algorithm 2.

We use a YUV video sequence with a QCIF format (176 × 144) which consists of 2000 frames with average quality for experimental video trace. The frame rate of the encoded stream is 30 fps. After processing, a MPEG-4 video sequence which has 223 I frames, 445 P frames, and 1332 B frames is produced. Those frames are fragmented into 2250 packets which include 463 packets for I frames, 453 packets for P frames, and 1334 packets for B frames. The created MPEG-4 video trace file which includes these packets information is introduced to the NS2. Each frame has 10 s lifetime [9]. We have considered Cases 1 and 2 with 20% packet loss and the rbuf is set to the default 64 KB. 10 random seeds are used for our experiment, and simulation results are calculated by averaging the result of 10 runs.

Since Peak Signal to Noise Ratio (PSNR) is the most vital QoS parameter being used to measure the quality of video streaming services, we use PSNR to measure the performance of *e*-CMT and the original CMT. The results for the PSNR measurements for the *e*-CMT and the original CMT are illustrated in Figure 10. Under Case 1 condition, the *e*-CMT achieves a PSNR value of 27.84 dB and the original CMT gains about 25.64 dB. Under Case 2 condition, the *e*-CMT obtains a PSNR value of 27.98 dB and the original CMT gains about 25.82 dB. The reason the *e*-CMT can outperform the original CMT is that the *e*-CMT can cognize link loss and retransmit the lost packet timely in the high loss condition. Moreover, if deteriorated congestion condition is detected, the *e*-CMT gives up redundancy retransmission for unavailable frame; this feature makes the *e*-CMT can achieve more playable frames than the original CMT.

We also evaluate other several well-known video sequences in terms of playable frames. Figure 11 shows the statistical results of four different videos sequences. The standard deviations were evaluated based on the Percentage of the Playable Frames (PoPFs). These results show that the *e*-CMT outperforms the original CMT for these videos. The proposed *e*-CMT achieves PoPF about 77.3% and 77.6% under Cases 1 and 2 conditions, respectively. And the original CMT gains 71.2% under Case 1 and 71.7% under Case 2.

Based on variety of experiments, we observe that the *e*-CMT outperforms the original CMT because not only its efficient sense and retransmission feature but also with the capability of intelligent retransmission decision for congestion mitigation. Hence, the *e*-CMT can achieve high users' experience of quality of multimedia streaming services, it is good protocol selected for real time of video transmission over multihomed WMSNs.

6. Conclusions and Future Work

Video streaming transfer over wireless multimedia sensor networks is becoming an attractive research topic but with challenge of high data loss. On the other hand, multimedia content delivery over multipath has been recognized as a more promising resolution to overcome the challenges such as high-rate required and time-sensitive delivery in multihomed wireless networks. This paper presented a novel environment-aware CMT dubbed as *e*-CMT with goal of providing effective video content delivery over multihomed WMSNs. By distinguishing the causes of transmission condition change, the *e*-CMT can trigger retransmission behavior timely and redeliver lost packet over a candidate path selected by PQM. Before retransmission, an intelligent congestion-aware data distributor will be enabled to make decision if to give up retransmission or not accordance with transmission condition of the retransmission destination. Sufficient simulation results showed that the *e*-CMT can provide an efficient retransmission approach and achieve better users' experience of quality of video streaming services than the original CMT.

We note that the *e*-CMT just only focuses on how to improve CMT protocol itself that depended solely upon the information provided by transport layer. As cross-layer coordination design becomes a promising resolution for multimedia content delivery in wireless transmission [32], our future work will pay attention on the cross-layer coordination and communication among transport layer and other layers to design an efficient concurrent multipath multimedia data transfer protocol. Moreover, our future work will consider multimedia encoding characteristics to make more intelligent transmission, for example, when wireless link is recognized as deteriorated congestion condition (NC-1), the high important and available I-frame will be retransmitted while the low important but available P-frame and B-frame will reduce the number of retransmission or give up retransmission.

Acknowledgments

This work was partially supported by the National High-Tech Research and Development Program of China (863) under Grant no. 2011AA010701, in part by the National Basic Research Program of China (973 Program) under Grants 2013CB329100 and 2013CB329102, in part by the National Natural Science Foundation of China (NSFC) under Grant nos. 61001122, 61003283, and 61232017, Beijing Natural Science Foundation of China under Grant no. 4102064, in part by the Fundamental Research Funds for the Central Universities under Grant no. 2012RC0603, no. 2011RC0507, in part by Jiangsu Natural Science Foundation of China under Grant no. BK2011171 and in part by Jiangxi Natural Science Foundation of China under Grant no. 20122BAB201042.

References

[1] A. Prasanna and T. Melodia, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 1060–1072, 2012.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.

[3] H. Xu, L. Huang, C. Qiao, Y. Zhang, and Q. Sun, "Bandwidth-power aware cooperative multipath routing for wireless multimedia sensor networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 4, pp. 1532–1543, 2012.

[4] R. Stewart, "Stream control transmission protocol," IETF RFC 4960, Proposed Standard, 2007.

[5] B. Jinsuk, S. Paul Fisher, J. Minho, and H.-H. Chen, "A lightweight sctp for partially reliable overlay video multicast service for mobile terminals," *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 754–766, 2010.

[6] C. Xu, G.-M. Muntean, E. Fallon, and A. Hanley, "Distributed storage-assisted data-driven overlay network for P2P VoD services," *IEEE Transactions on Broadcasting*, vol. 55, no. 1, pp. 1–10, 2009.

[7] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 951–964, 2006.

[8] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: qualityaware adaptiveconcurrent multipath data transfer in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*. In press.

[9] 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.

[10] I. Politis, M. Tsagkaropoulos, T. Dagiuklas, and S. Kotsopoulos, "Power efficient video multipath transmission over wireless multimedia sensor networks," *Mobile Networks and Applications*, vol. 13, no. 3-4, pp. 274–284, 2008.

[11] N. Stephan, S. Varakliotis, and P. Kirstein, "Transport layer multipath on wireless sensor network backhaul links," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications*, pp. 469–472, Athens, Greece, June 2009.

[12] Y. Qiao, X. Yan, E. Fallon, and A. Hanley, "Handover strategies in multihomed body sensor networks," in *Proceedings of the Information Technology and Telecommunications Conference*, pp. 183–189, Dublin, Ireland, October 2007.

[13] C. Lu and Q. Wu, "Performance study on SNMP and SIP over SCTP in wireless sensor networks," in *Proceedings of the 14th International Conference on Advanced Communication Technology*, pp. 844–847, PyeongChang, South of Korea, February 2012.

[14] C. Xu, E. Fallon, Y. Qiao, L. Zhong, and G.-M. Muntean, "Performance evaluation of multimedia content distribution

over multi-homed wireless networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 204–215, 2011.

[15] C. Xu, E. Fallon, Y. Qiao, G. M. Muntean, X. Li, and A. Hanley, "Performance evaluation of distributing real-time video over concurrent multipath," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, Budapest, Hungary, April 2009.

[16] C.-M. Huang and M.-S. Lin, "Multimedia streaming using partially reliable concurrent multipath transfer for multi-homed networks," *IET Communications*, vol. 5, no. 5, pp. 587–597, 2011.

[17] C.-M. Huang and M. S. Lin, "Fast retransmission for Concurrent Multipath Transfer (CMT) over vehicular networks," *IEEE Communications Letters*, vol. 15, no. 4, pp. 386–388, 2011.

[18] L. Cui, S.-J. Koh, and W. J. Lee, "Fast selective ACK scheme for throughput enhancement of multi-homed SCTP hosts," *IEEE Communications Letters*, vol. 14, no. 6, pp. 587–589, 2010.

[19] S. Shailendra, R. Bhattacharjee, and K. Sanjay Bose, "MPSCTP: a simple and efficient multipath algorithm for SCTP," *IEEE Communications Letters*, vol. 15, no. 10, pp. 1139–1141, 2011.

[20] Y. Wang, J. Ostermanm, and Y. Q. Zhang, *Video Processing and Communications*, IEEE Transactions on Multimedia, Prentice Hall, 2002.

[21] Q. Zhang, W. Zhu, and Y. Q. Zhang, "Resource allocation for multimedia streaming over the Internet," *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 339–355, 2001.

[22] F. Zhang, C. K. Wu, P. X. Cheng, and S. Xiao, "Research on an improved scalable video coding and the network transmission," *Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology*, vol. 27, no. 1, pp. 108–111, 2005.

[23] S. Floyd and E. Kohler, "Internet research needs better models," *ACM Computer Communications Review*, vol. 33, no. 1, pp. 29–34, 2003.

[24] M. Fomenkov, K. Keys, D. Moore, and K. Claffy, "Longitudinal study of Internet traffic in 1998–2003," in *Proceedings of theWinter International Symposium on Information and Communication Technologies*, pp. 1–6, Cancun, Mexico, January 2004.

[25] The Network Simulator,ns-2, http://www.isi.edu/nsnam/ns/.

[26] A. Chan, K. Zeng, P. Mohapatra, S. Lee, and S. Banerjee, "Metrics for evaluating video streaming quality in lossy IEEE 802.11 wireless. Networks," in *Proceedings of the IEEE Conference on Computer Communications*, pp. 1–6, Cancun, Mexico, March 2010.

[27] P. Natarajan, F. Baker, P. D. Amer, and J. T. Leighton, "SCTP: what, why, and how," *IEEE Internet Computing*, vol. 13, no. 5, pp. 81–85, 2009.

[28] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1–4, pp. 18–39, 2008.

[29] E. Callaway, P. Gorday, L. Hester et al., "Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 70–77, 2002.

[30] A. Caro, P. Amer, J. Iyengar, and R. Janardhan Iyengar, "Retransmission policies with transport layer multihoming," in *Proceedings of the 11th IEEE International Conference on Networks*, pp. 255–260, Sydney, Australia, October 2003.

[31] Y. Cao, C. Xu, J. Guan, and H. Zhang, "Background traffic-based retransmission algorithm for multimedia streaming transfer over concurrent multipaths," *International Journal of Digital Multimedia Broadcasting*, vol. 2012, Article ID 789579, 10 pages, 2012.

[32] Y. Cao, C. Xu, J. Guan, and H. Zhang, "Cross-layer retransmission approach for efficientVoD transfer over multi-homed wireless networks," *International Journal of Digital Content Technology and its Applications*. In press.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 135054, 9 pages
doi:10.1155/2012/135054

Research Article

Holes Detection in Anisotropic Sensornets: Topological Methods

Wei Wei,[1] Xiao-Lin Yang,[2] Pei-Yi Shen,[3] and Bin Zhou[4]

[1] *School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*
[2] *College of Management Science, Chengdu University of Technology, Chengdu 610059, China*
[3] *National School of Software, Xidian University, Xi'an 710071, China*
[4] *College of Science, Xi'an University of Science and Technology, Xi'an 710054, China*

Correspondence should be addressed to Xiao-Lin Yang, yangxlcdut@gmail.com

Received 11 May 2012; Revised 2 September 2012; Accepted 12 September 2012

Academic Editor: Chuan Foh

Wireless sensor networks (WSNs) are tightly linked with the practical environment in which the sensors are deployed. Sensor positioning is a pivotal part of main location-dependent applications that utilize sensornets. The global topology of the network is important to both sensor network applications and the implementation of networking functionalities. This paper studies the topology discovery with an emphasis on boundary recognition in a sensor network. A large mass of sensor nodes are supposed to scatter in a geometric region, with nearby nodes communicating with each other directly. This paper is thus designed to detect the holes in the topological architecture of sensornets only by connectivity information. Existent edges determination methods hold the high costs as assumptions. Without the help of a large amount of uniformly deployed seed nodes, those schemes fail in anisotropic WSNs with possible holes. To address this issue, we propose a solution, named PPA based on Poincare-Perelman Theorem, to judge whether there are holes in WSNs-monitored areas. Our solution can properly detect holes on the topological surfaces and connect them into meaningful boundary cycles. The judging method has also been rigorously proved to be appropriate for continuous geometric domains as well as discrete domains. Extensive simulations have been shown that the algorithm even enables networks with low density to produce good results.

1. Introduction and Motivation

Sensornets are appearing as promising techniques for pervasive data exchange and information sharing. Sensornets are tightly linked with the geometric environment in which they are deployed. Detecting topological holes is a very important task in wireless sensor networks [1]. In many crucial safe-related scenarios, such as earthwork construction and mine exploitation cases, we need to determine whether topological holes in space topological structure exist, thus, we can send the urgent warning for users so as to prevent the disasters that happen suddenly and have enough time to deal with the accidents in time. Many existing countermeasures usually do strong assumptions. As we know, all of mathematical theorems have their own used field. That is to say, before we use these mathematical methods to solve the practical problems, we need to proof at least explain that these mathmatic ways can be used in the specialized domain. Simultaneously, those current methods either enquire customerized hardware devices or have strong assumptions on the network environment, leading to low efficiency and applicability. In this work, we fundamentally analyze the detecting mind of space holes issue by topology methodology and by observing the inevitable topology deviations introduced by holes. We generalize the definition of space holes in practical scenarios and propose a topological approach. Mathmatical proof and simulation results show that our approach can detect and locate various holes and relies solely on topological information of the network. To the best of our knowledge, we try our best to make the first attempt towards a purely topological approach to detect holes distributedly without any rigorous requirements and assumptions. At the same time, we also solve the applied domain problem of mathmaitical theorem by removing the theoretical barriers to finish it. Our approach achieves superior performance and applicability with the least limitations.

On one hand, sensor network applications for example environment monitoring and data collection demand wealthy coverage over the region of interest. On the other hand, the global topology of a WSN has a great influence on the design of basic networking functionalities, for example, point-to-point routing and data collecting mechanisms. In this paper we study the problem of discovering the global geometry of the sensornets field, especially, inspecting sensor nodes on the boundaries (both inner and outer boundaries). The standpoint we take is to regard the sensornet as a discrete sampling of the underlying geometric environment. This is inspired by the fact that sensornets are to offer dense monitoring of the potential space. Therefore, the shape of the sensor field, that is, the boundaries, indicates significant characters of the underlying environment. These boundaries usually have physical correspondences, such as a building floor plan, a map of a transit network, topography changes, and barriers (skyscrapers, subsidence areas, etc). Holes can also map to events that are being monitored by the sensornet. If we consider the sensors with readings above a threshold to be "invalid", then the hole borders are essentially isocontours of the landscape of the property of interest.

Cases include the identification of regions with overheated sensors or abnormal chemical contamination. Holes are also important indicators of the universal health of a sensornet, for example insufficient coverage and connectivity. The detection of holes divulges groups of destroyed sensors because of physical destruction or power consumption, where additional sensor deployment is demanded. Besides the real scenario mentioned above, understanding the global geometry and topology of the sensor field is of great importance in the design of basic networking operations. For example, in the sensor deployment problem, if we are desirous to spread some mobile sensors in an unknown region formed by static sensor nodes, knowing the border of the region permits us to guarantee that newly added sensors are deployed only in the expected region.

A number of networking protocols also exploit geometric intuitions for simplicity and scalability, for instance geographical greedy forwarding [2, 3]. Such algorithms based on local greedy advances may fail at local minima if the sensor networks have nontrivial topology. Backup methods, for instance face, routing on a explanate subgraph, can assist packets avoid local minima, but build high traffic on hole boundaries, and eventually destroy the network lifetime [2, 3]. This artificial product is not amazing because any algorithm with a strong geometrical application, for example geographical forwarding, ought to stick to the genuine shape of the sensor field. Currently, there are lots of routing schemes that address explicitly the importance of topological properties and propose routing with virtual coordinates that are adaptive to the inner geometric features [4, 5]. The construction of these virtual coordinate systems needs the identification of topological features. We focus on developing a judgment method that detects hole boundaries based on the Poincare Conjecture theory.

The rest of this paper is organized as follows. We first give a brief overview of this scheme in Section 2. And then, we present the PPA design principle in a continuous domain and offer the solid and complete theoretical proof to describe how the traditional and continuous topological theory (Poincare-Perelman Theorem) can be suitably (appropriately) applied to discrete and practical scenarios. As a result, we utilize the Poincare-Perelman Theorem to judge (determine) whether there are existing holes in real topological spaces. Namely, the constructing topological structure of continuous deployment of sensors over the Euclidean plane can also be used to justify whether holes in practical applications exist. We can efficiently detect holes danger and therefore send alert notice in real and safe field applications. In Section 3, we perform the problem formulation and holes detection in discrete environments. Section 4 extends the discussion into the practical discrete context. Section 5 evaluates the proposed scheme through comprehensive simulations and compares it with state of the art-area-based approaches localization schemes. We conclude the work in Section 5.

2. Prior Works

A lot of methods have been presented to judge sensor locations in WSNs. A universal overview of the state-of-the-art localization schemes is available in [6].

Existing researches on edges recognition can be separated into three classifications: geometric, statistical, and topological methods. Geometric methods that were proposed by Fang et al. [1] for boundary detection use geographical location information. This method assumes that the sensor nodes can sense their geographical locations and that the communication graph follows the UDG (Unit Disk Graph) assumption, when two nodes are connected by an edge if and only if their interval is at most 1. The description of holes in [1] is closely interrelated with geographical forwarding so that a packet can only get stuck at a node of hole edges. Fang et al. also presented a simple algorithm that greedily sweeps along hole boundaries and eventually discovers boundary cycles. Statistical methods for boundary detection usually make assumptions about the probability distribution of the sensor deployment. Fekete et al. [7] proposed a border detection algorithm for sensors (uniformly) randomly deployed inside a geometric region. The primary idea is that boundaries nodes have much lower average degrees than nodes in the "interior" of the network. Statistical arguments cause an appropriate degree threshold to differentiate border nodes. An statistical way is to calculate the "restricted stress centrality" of a vertex v, which measures the quantity of shortest paths going through v with a bounded length [7]. Nodes in the interior tend to have a higher centrality than nodes on the boundary. With a sufficient nodes density, the centrality of the nodes holds dual features so that it can be used to detect boundaries. The dominating weak points of these two algorithms are the idealized request on sensor deployment and density: the mean density needs to be 100 at least. In real scenario, the sensors are not as dense and they are unnecessarily arranged uniformly and randomly. There are also topological methods to prime deficient sensor coverage and holes. Ghrist and Muhammad [8] presented an algorithm that detects holes via homology with no knowledge of sensor locations; on the contrary,

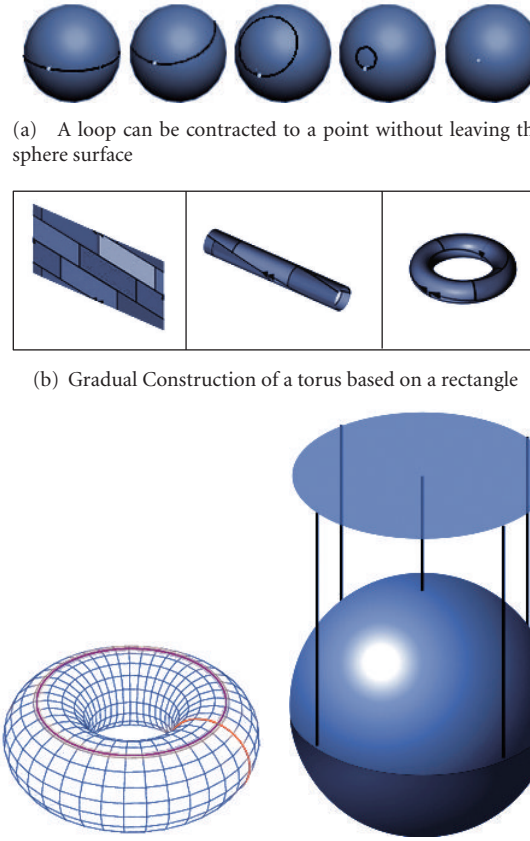
the algorithm is centralized, with assumptions that both the sensing range and communication range are disks with radii carefully tuned. Kröller et al. [9] presented an algorithm by probing for combinatorial structures called flowers and augmented cycles. They make less restrictive assumptions on the problem setup, modeling the communication graph by a quasi UDG, with nodes p and q demonstrably linked by an edge if $d(p,q) \leq \sqrt{2}/2$ and not connected if $d(p,q) > 1$. The success of this algorithm critically depends on the identification of at least one flower structure, which might not often be the case specially in a sparse network. For a real scenario, Funke [10] developed a simple heuristic with only connectivity information. The essential idea is to build iso contours with hop count from a root node and identify where the contours are broken. Under the unit-disk graph assumption and adequate sensor density, the algorithm outputs nodes marked as border with certain guarantees. Definitely, for each node of the geometry boundary, the algorithm enables to mark a corresponding sensor node within distance 4.8, and each node marked as boundary is within distance 2.8 from the actual geometry boundary [11]. The simplicity of the algorithm is appealing; however, the algorithm only identifies nodes that are near the boundaries but does not show how they are connected in a meaningful way. The density requirement of the algorithm is also rather high; so as to obtain good results, the average degree generally needs to be at least 16.

From mathematics aspect, the Poincare Conjecture [12] is a theorem about the specification of the three-dimensional sphere among three-dimensional manifolds. Original conjectured is proposed by Henri Poincare, the claim considers a space that locally resembles ordinary three-dimensional space but is connected, finite in size, and lacks any boundary (a closed three-dimensional manifold). The Poincare Conjecture states that if each loop in such a space can be continuously tightened to a point, then it must be a three-dimensional sphere. An similar result has been proved in higher dimensions. (Some related content is partially referred to the Wikipedia information).

2.1. Definitions of Manifold. A manifold is a space made by conglutinating together pieces of Euclidean space, which is called charts. For example you could take two-dimensional disks and bend them around two hemispheres and then stick them together to form a two-dimensional sphere. (See also in Figure 1(d)).

A torus (the surface of a donut) can be established utilizing a rectangular diagram as shown in this image. The colored parallelograms explain how a pattern on the associated surface would arise in case the edges were once again disconnected. (See also in Figures 1(b) and 1(c)). A pair of solid balls can made a three-dimensional sphere. It should be required to discern every point of the first ball boundary with the corresponding point of the second one. Other kinds of manifolds can be established by the mimetic ways.

2.2. Explanations of Homeomorphic. Generally, two shapes are homeomorphic if one of these shapes can be transformed into the other without pause or discontinuity. *A* homeomorphism is a function of continuous domain that maps points of one object into another.



(a) A loop can be contracted to a point without leaving the sphere surface

(b) Gradual Construction of a torus based on a rectangle

(c) Any chromatic loops cannot be contracted to a point without leaving the torus surface

(d) Hemispheres mapped to a Whole Sphere

Figure 1: Diagram of manifold.

Two spaces are regarded as homeomorphic if a homeomorphism between them exists. Such as, a two-dimensional sphere is homeomorphic to the surface of a cube; homoplastically, a three-dimensional sphere is homeomorphic to the three-dimensional boundary of a four-dimensional hypercube. In the above-mentioned primary concepts, they can aid to comprehend the specification of the Poincare-Perelman Theorem: the Poincare-Perelman Theorem says that a three-dimensional manifold which is compact, has no boundary and is simply connected must be homeomorphic to a three dimensional-sphere. The original phrasing was as follows: consider a compact three dimensional manifold V without boundary. Is it possible that the fundamental group of V could be trivial, even though V is not homeomorphic to the three-dimensional sphere? This centurial challenging problem is proofed by Grigori Perelman [12] in 2006.

Here is the standard form of the conjecture: every simply connected, closed tridimensional manifold is homeomorphic to the triaxial sphere. Therefore, a centralized method

of collecting all of the information to a central server is not feasible for large sensor networks.

2.3. Preliminaries. Materials for topology theory. The following definitions in topology theory can be found in [13].

Topology. Let X be a set. $\mathcal{T} \subseteq 2^X$ is called a topology on X if (1) $\emptyset, X \in \mathcal{T}$; (2) if $A, B \in \mathcal{T}$, then $A \cap B \in \mathcal{T}$; (3) if $\{A_i \mid i \in I\} \subseteq \mathcal{T}$, then $\bigcup_i A_i \in \mathcal{T}$. The pair $(X, \mathcal{T})$ is called a topological space. The members of $\mathcal{T}$ are called open sets. If Y is a subset of X, then $\mathcal{T} \mid Y = \{U \cap Y \mid U \in \mathcal{T}\}$ is a topology on Y and called the induced topology of $(X, \mathcal{T})$. A bijection $f : (X_1, \mathcal{T}_1) \to (X_2, \mathcal{T}_2)$ between two topological spaces is called a homeomorphism if $B \in \mathcal{T}_2$ iff $f^{-1}(B) \in \mathcal{T}_1$ for any $B \subseteq X_2$. In this case, $(X_1, \mathcal{T}_1)$ and $(X_2, \mathcal{T}_2)$ are said to be homeomorphic to each other.

Dense Set. Let $(X, \mathcal{T})$ be a topological space and $C \subseteq X$. A point $x \in X$ is called a cluster of C if $U \cap C \neq \emptyset$ any $U \in \mathcal{T}$ with $x \in U$. Denote C^- as the set of all cluster of C, called the closure of C. The set C is called a closed set if $C = C^-$. A set is called a clopen set if it is simultaneously open and closed. A set C is called dense of $(X, \mathcal{T})$ if $C^- = X$. Dense set is an important and useful concept in topology. For example, every continuous map from a dense set of a topological space to another topological space can be extended onto the whole topological space. Thus dense sets in a topological space may share some same topological properties as the whole topological space, for example the connectedness as Theorem 1 shows.

Partition. The specification of partition £ for a set X, a family of subsets $\{X_i \mid i \in I\}$ is called a partition of X, if $\bigcup_i X_i = X$ and $X_i \cap X_j = \emptyset$ for all $i, j \in I$ with $i \neq j$.

2.4. Our Contributions. We develop a practical and efficient determination solution for boundary detection in sensor networks, using only the communication graph and not making unrealistic assumptions. We do not assume any location information, angular information, or distance information. More importantly, we do not request that the communication graph obeys the unit disk graph model or the quasi-unit disk graph model. Actual communication ranges are not circular disks and are often quite irregularly shaped [14]. Algorithms that depend on the unit disk graph model fail in practice (e.g., the extraction of a planar subgraph by the relative neighborhood graph or Gabriel graph [15].

Our PPA method also readily provides other topological and geometric information, such as the number of holes (genus), the nearest hole to any given sensor, and the sensor field's medial axis (the collection of nodes with at least two closest boundary nodes), which is useful for virtual coordinate systems for load-balanced routing [4]. Simulation results show that our algorithm correctly determines useful borders for sensor networks with rational node density (average degree 10 and above) and distribution (e.g., uniform). The algorithm also works well for nonuniform distributions. The algorithm is efficient. The entire procedure involves only three network flooding procedures and greedy shrinkage of paths or cycles. Further, as a theoretical ensure, we prove that for a continuous geometric space bounded by polygonal obstacles, the case in which node density approaches infinity, the algorithm correctly discovers all of the boundaries. More definitely, we investigate the fact that a legitimate multihop sensor network deployed on the surface of a geometric terrain, (even possibly including irregular boundaries, inner obstacles, or even on a non-2D plain) PPA solution is able to accurately estimate the node-to-node distances and calculate node locations with only 3 seeds, thus increasing system scalability and usage as well as lowering hardware costs. In addition, PPA does not presume the superior communication capability of seeds, that is, with much larger radio range than those of the ordinary nodes [16].

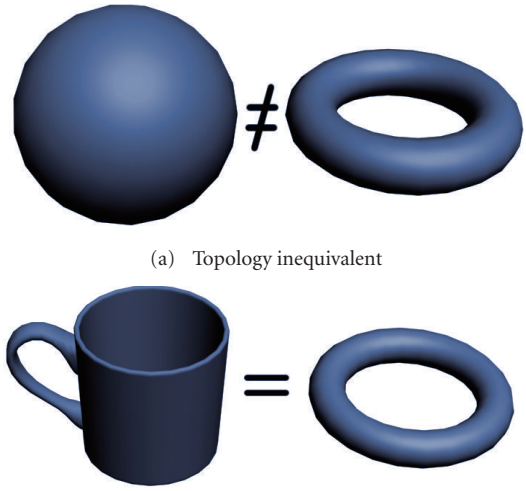
Due to all mentioned above assumptions based on UDG graph model and its basis on the symptom of packing number, it is thus inaccurate under non-UDG graphs. Indeed, there are still no perfect symptoms found to establish an all-round method in the resource-limited sensornets. Our design is originated from the perspective of topological observation and is based on the theory of Poincare Conjecture, our solution is orthogonal to existing approaches and takes a step towards relaxing these assumptions and expanding the applicability of methods.

3. Problem Formulation and Holes Detection

The definition has been given under the constraints of the UDG communication graph model, which has been proven far from practical in many analytical and experimental works. Second, the distance-based definition in Euclidean space naturally binds the hole features with external geometric environments and thus neglects the inherent topological impacts introduced by holes. We hereby present a more general and fundamental definition of the hole based only on network topologies and aim to present the inherent characteristics of holes. According to the Poincare-Perelman Theorem, in the three dimensions space, the donut topology is homeomorphous to the coffee cup topology (see also Figure 2(b)). As shown in Figure 2(a), since these two topologies are not equivalent (namely, not homeomorphism), we can determine that the holes in the monitored areas based on the Poincare-Perelman Theorem exist (see also Figures 2(a) and 2(b)). Since these two topologies are not equivalent (namely, not homeomorphism), we can determine that the holes in the monitored areas based on the Poincare-Perelman Theorem exist.

In real scenario, we will treat the multihole condition. But in this proposed solution, we currently do not differentiate the numbers of holes. In future work, we will discuss and deal with this condition.

Owing to constructing the network topological structure of monitored areas, in given the surface S, we first select an arbitrary point in S as the root and run a continuous Dijkstra shortest path algorithm [17, 18] to construct the topology structure (manifold) of monitored areas. As shown in Figure 3(a) and Figure 3. Consequently, we can determine whether any closed and simply connected manifold is



(a) Topology inequivalent

(b) Topology inequivalent

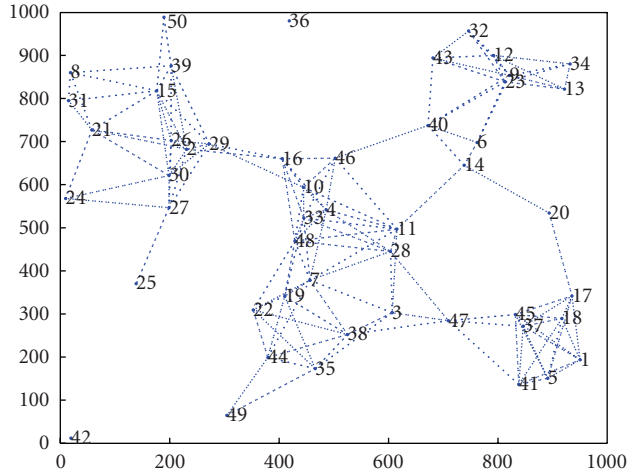
Figure 2: Diagram of manifold.

homeomorphic to the three dimensions sphere. If it is not homeomorphic to the three dimensions sphere. It refers to there is/are hole(s) in the monitored areas. (See also in Figure 3(b)).

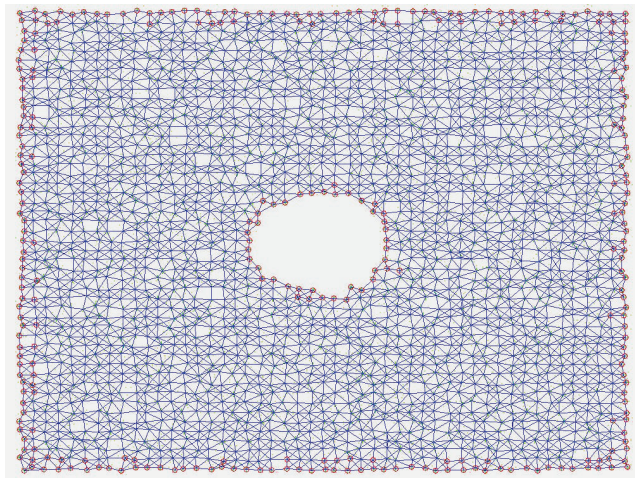
3.1. Holes Detection in Discrete Environments. We have characterized the impact of holes and described the principles of holes detection under continuous settings in the previous section. In a real multihop network, however, nodes are deployed discretely on the field. In this section, we focus on solving the mapping question from the discrete domain to continuous geometric domain. Since the theory of Poincare-Perelman Theorem belongs to the judgement of continuous geometric domain, we need to proof the correctness and applicability of this topological judgement. By means of the following solid proof, we can transfer the discrete topological space to the continuous geometric domains. Namely, we utilize the partial discrete topology structure to substitute the whole continuous geometric topology. As a result, we can apply the theory of Poincare-Perelman Theorem to judge the existence of holes in the monitored areas by WSNs.

Let $\mathcal{T}$ be a topology on a set X and $Y \subseteq X$. Then $\mathcal{T} \mid Y = \{U \cap Y \mid U \in \mathcal{T}\}$ is a topology on Y, called the induced topology on Y and in this case $(Y, \mathcal{T} \mid Y)$ is always called a subspace of $(X, \mathcal{T})$ and $\mathcal{T}$ an extension of $\mathcal{T} \mid Y$ from Y to X. We confirm that these two topological spaces $(X, \mathcal{T})$ and $(Y, \mathcal{T} \mid Y)$ have the same topological properties. In a topological space X, a subset U is called dense if $U^- = X$, where U^- is the closure of U in $(X, \mathcal{T})$. A topological space $(X, \mathcal{T})$ is called connected if there exists no clopen (simultaneously closed and open) subset except empty and whole set X.

Theorem 1. *Suppose that $(Y, \mathcal{T}_1)$ is a dense subspace $(X, \mathcal{T}_2)$, then $(Y, \mathcal{T}_1)$ is connected if and only if $(X, \mathcal{T}_2)$ is connected.*



(a) Dijkstra process



(b) Hole detection process

Figure 3: Diagram of hole boundary construction.

Proof. Suppose that $(Y, \mathcal{T}_1)$ is connected. If $(X, \mathcal{T}_2)$ is not connected, then there exists a clopen set U in $(X, \mathcal{T}_2)$ and $U \notin \{X, \varnothing\}$. Put $V = U \cap Y$, we have that V is a clopen set in $(Y, \mathcal{T}_1)$, which implies that $V = Y$ and $Y \subseteq U$ (obviously, $V \neq \varnothing$). Since Y is dense in $(X, \mathcal{T}_1)$, we have $X = Y^- \subseteq U^- = U$, which contradicts to $U \neq X$. Conversely, suppose that $(X, \mathcal{T}_2)$ is connected. If $(Y, \mathcal{T}_1)$ is not connected, then it contains a clopen set V which is neither Y nor $\varnothing$. For this V, there exist a clopen set U in $(X, \mathcal{T}_2)$ such that $V = U \cap Y$. In order to induce a contradiction, we only need to show that $U \neq X$. If $U = X$, then $V = U \cap Y = Y$, which is another contradiction to $V \neq Y$. The proof is complete. □

Remark 2. If a topological space $(X, \mathcal{T})$ is n-connected topological space $(n \geq 2)$, then $(X, \mathcal{T})$ can exactly be separated into $n - 1$ connected subspace $\{(X_i, \mathcal{T} \mid X_i) \mid i = 1, 2, \ldots, n-1\}$ such that $\{X_i \mid i = 1, 2, \ldots, n-1\}$ is a partition of X. By Theorem 1, if $(Y, \mathcal{T}_1)$ is a dense subspace $(X, \mathcal{T}_2)$, then $(Y, \mathcal{T}_1)$ is n-connected if and only if $(X, \mathcal{T}_2)$ is n-connected. As mentioned in the above Remark, the definition

of concept partition for a set X, a family of subsets $\{X_i \mid i \in I\}$ is called a partition of X, if $\bigcup_i X_i = X$ and $X_i \cap X_j = \varnothing$ for all $i, j \in I$ with $i \neq j$.

The following verdict is usually held for any set that satisfies the requirement of the theory. Whatever the set is finite or infinite. We assume that a certain area deployed the WSNs. This area can be considered as a smooth curve equipped with the traditional Euclidean topology, the set of all sensors equipped with their own topology can be considered as a subspace of the former one. Furthermore, we assume that the set of all sensors is dense in this area.

We assume that the whole sensor nodes set, which completely cover the monitored area, constructs a dense set. A dense set is the monitored area which is abundantly and completely covered by the large quantities of sensor nodes. Therefore, the network topology can be continuously expanded to the monitored area. Specially, some part of sensornets can be destroyed by some accidents so that it will lead to form a hole in the architecture of topology. As a result, there exists a hole in the corresponding practical area. The sensornets corresponding geometric structure is a universal Euclid topology, particularly, if a hole in this monitoring area exists. If and only if the topology of sensornets is sub dense space of area topology space. Furthermore, If and only if the geometric topology of monitored area is connected completely, consequently, the constructing topology of sensornets is interconnected. Simultaneously, if and only if there exists holes in the geometric topology of monitored area, as a result, there exists holes in the constructing topology of sensornets.

Thus detecting whether there are holes existing in sensornets topology is equivalent to detecting whether there are holes in the monitored area.

Steps. Symbolic Interpretation. Area S, $\mathcal{T}$ is the Euclidean topology of S. The set C denotes the sensornets while C_1 denotes the efficient sensornets. Precondition: set C is dense in $(S, \mathcal{T})$.

(1) Let S_1 be the closure of C_1 on $(S, \mathcal{T})$. If it exists holes, then $S1 \subseteq S$. (2) Obtaining $\mathcal{T}_1$ while the topology $\mathcal{T}$ of S is constrained in the S_1. Therefore, $(S_1, \mathcal{T}_1)$ is a subspace of $(S, \mathcal{T})$. (3) In the above mentioned, S_1 is continuous set. Consequently, we can depend on the Poincare Conjecture theory to determine whether there are holes that existed in monitoring area. If there are holes in the topology structure of $(S_1, \mathcal{T}_1)$, then there are holes in the topology structure of C. The above-mentioned theory can guarantee this determination.

3.2. Topological Boundary Recognition. Suppose a large number of sensor nodes are scattered in a geometric region with nearby nodes communicating with each other directly. Our goal is to discover the nodes on the boundary of the sensor field, using only local connectivity information. We propose a solution that identifies boundary cycles for the sensor field. For compact 2-dimensional surfaces without boundary, if every loop can be continuously tightened to a point, then the surface is topologically homeomorphic to a 2 spheres, usually just called a sphere. The Poincare Conjecture asserts that the same is true for 3-dimensional surfaces. (See also in Figure 1(a)). Practically, for obtaining the topology of monitored areas, we firstly use the Dijkstra Shortest Path algorithm [17] to construct the topology (manifold) of monitored areas. Consequently, we can determine whether any closed and simply connected manifold is homeomorphic to the three dimensions sphere. If it is not homeomorphic to the three dimensions sphere, it refers to there is/are hole(s) in the monitored areas (see also Figure 3(b)).

In the following, we first outline the Dijkstra Shortest Path algorithm and then explain each step in detail.

Algorithm allows the node at which we are starting to be called the initial node. Let the distance of node Y be the distance from the initial node to Y. Dijkstra's algorithm that allocates some initial distance values and will try to increase them step-by-step. Assign to every node a distance value. Set it to zero for our initial sensor node and to infinity for all other nodes. Mark all nodes as unvisited. Set initial sensor node as current. For current node, consider all its unvisited neighbors and calculate their distance (from the initial node). For instance, if current node (A) has distance of 6, and an edge connecting it with another node (B) is 2, the distance to B through A will be $6 + 2 = 8$. If this distance is less than the previously recorded distance (infinity in the beginning, zero for the initial node), overwrite the distance. When we are done considering all neighbors of the current node, mark it as visited. A visited node will not be checked ever again; its distance recorded now is final and minimal. If all nodes have been visited, finish. Otherwise, set the unvisited node with the smallest distance (from the initial node) as the next "current node". Suppose you want to find the shortest path between two intersections on a map, a starting point and a destination. To accomplish this, you could highlight the streets (tracing the streets with a marker) in a certain order, until you have a route highlighted from the starting point to the destination. The order is conceptually simple: at each iteration, create a set of intersections consisting of every unmarked intersection that is directly connected to a marked intersection, and this will be your set of considered intersections. From that set of considered intersections, find the closest intersection to the destination (this is the "greedy" part, as described above) and highlight it and mark that street to that intersection, draw an arrow with the direction, then repeat. In each stage mark just one new intersection. As getting to the destination, follow the arrows backwards. There will be only one path back against the arrows, the shortest one. The basic idea is to detect the existence of holes by judging whether if the existing topology is equivalent to sphere in the three-dimension space. Based on the mentioned above, we can construct a topology of monitored areas. Intuitively, it is very hard to determine the existence of holes by the two-topology structure. We assume our method can obtain the whole monitored topology, and then we can compare this obtained topology with sphere topology. Finally, we can determine whether holes in the monitored area exist.

4. Simulations

We performed extensive simulations in various scenarios, with the goal to evaluate the performance of the algorithm with respect to the network topology, node density and distribution, so on. We particularly note that our method works well even in cases of very low average degree, such as less than 10, or even as low as 10 in some models. Its ability is also similar to average degree 20 condition. Degree 6 has been shown to be optimal for mobile networks [19]. For each figure in this part, we assume a root node in the upper left corner and middle to illustrate the communication range of the sensor field.

4.1. Random Distribution of Sensors. In this experiment, we first assume that the network connectivity and link quality are good enough. In terms of a uniform distribution, we randomly deploy 1600 nodes in a square region with one hole. The average degree of the graph is discriminated by regulating the communication radius. As expected, Figures 4(a) and 4(b) show the results of our method. We can efficiently judge the hole existing in the monitored area. Connectivity is necessary for computing the shortest path tree. Practically, this low-degree graph with insufficient connectivity is the major troubling issue for prior boundary detection methods. Since our method only requires the communication graph, we can use several simple policies to raise artificially the average degree. For a disconnected network, we use the largest connected component of the graph to build our shortest path tree. Then we artificially enlarge the communication radius by taking two/three hops neighbors as fake one-hop neighbors. According to this means, the connectivity of the graph will be made better, and the results will be improved correspondingly by this simple strategy. The result using three hops neighbors has fewer incorrectly marked extremal nodes, and the final judgement is in good result except that the boundary cycle is not very tense. This is understandable since we make the communication range artificially larger, so that more nodes could be equivalently to distribute on the boundary now. Therefore, based on our solution, we can efficiently find holes in the supervised area.

4.2. Grid with Random Perturbation. In this simulation, we put about 1600 nodes on a grid and then perturbed each point by a random shift. Especially, for each original grid node we create two random numbers modulo the length and the width of each block of the grid and use these two small numbers to perturb the positions of the nodes. This distribution may be a good approximation of manual deployments of sensors; it also gives an alternative means of modeling "uniform" distributions, while avoiding clusters and holes that can arise from the usual continuous uniform distribution or Poisson process. As the theoretic verification considering, our method generates very good results, while average degree of graphs is ten or more.

4.3. Low Density, Sparse Graphs. In the experiments, we spread sensor nodes in a square region with one hole. In order to guarantee good connectivity, the nodes are distributed on a randomly perturbed grid. Our experiments show that if we amend the communication radius and decrease the density of nodes, our solution is performed very well, even for low density or sparse condition, as long as the average degree is at about ten or more. See also Figure 4(c).

5. Conclusion and Discussion

We devote our most efforts to explore the application of Poincare Conjecture to resolve the holes detection of safety-monitored areas in WSNs. Based on the theoretic specification, we can judge whether there are holes in the detected area. Because the detected network topology is not homeomorphous to the three dimensions sphere, it can be confirmed to have holes in the detected topology architecture. Therefore, we can accomplish the detecting holes purposes. The proposed new detection solution enables us to find holes in the continuous case, in discrete sensors networks several implementation issues arise. First, even for a given homotopy type, there needs not be a unique shortest path between two nodes. Thus, the boundary topology discovered by our solution, as shown in the simulations, may not tightly surround the real boundaries. Currently, we have two approaches to improve it. One is to make use of the fact that the nodes with lower degree are more likely to be on the boundary; thus, we implemented a preferential scheme for low-degree nodes when computing shortest paths. Another approach is to use an iterative method to find more extremal nodes and then refine the topology; this can also help to address the issue that several extremal points may have the same positions because we use hop counts to approximate true distances. Second, deciding the correct orderings of the extremal nodes requires some care. In the continuous case, extremal nodes project to their nearest node. In the discrete case, since we employ hop counts to approximate the true distance, it is possible that different extremal points are mapped to the same position on the inner boundary, obscuring their ordering. Again, by using an iterative procedure, we delete all the extremal nodes with duplicate positions except one and then iteratively find more extremal points and refine the boundary gradually. In real scenarios, the sensor nodes often know some partial location information or relative angular information. Such positional information can help to improve the performance of our holes detection solution, for example, when we utilize the shortest path algorithm to construct the topology of monitored areas. If the nodes have knowledge of a general arctic direction, it is easier to distinguish the extremal nodes in the interior and exterior of rough boundary. Also, if we have estimated distance or other rough localization information, other than pure hop count, the procedure to find shortest paths will become more reliable. Finally, our method discussed until now assumes a sensor field with holes. We remark that the case with no holes can be solved as well.

Finally, our method discussed until now assumes a sensor field with holes. We remark that the case with no holes can be solved as well. If a network topology is equivalent to

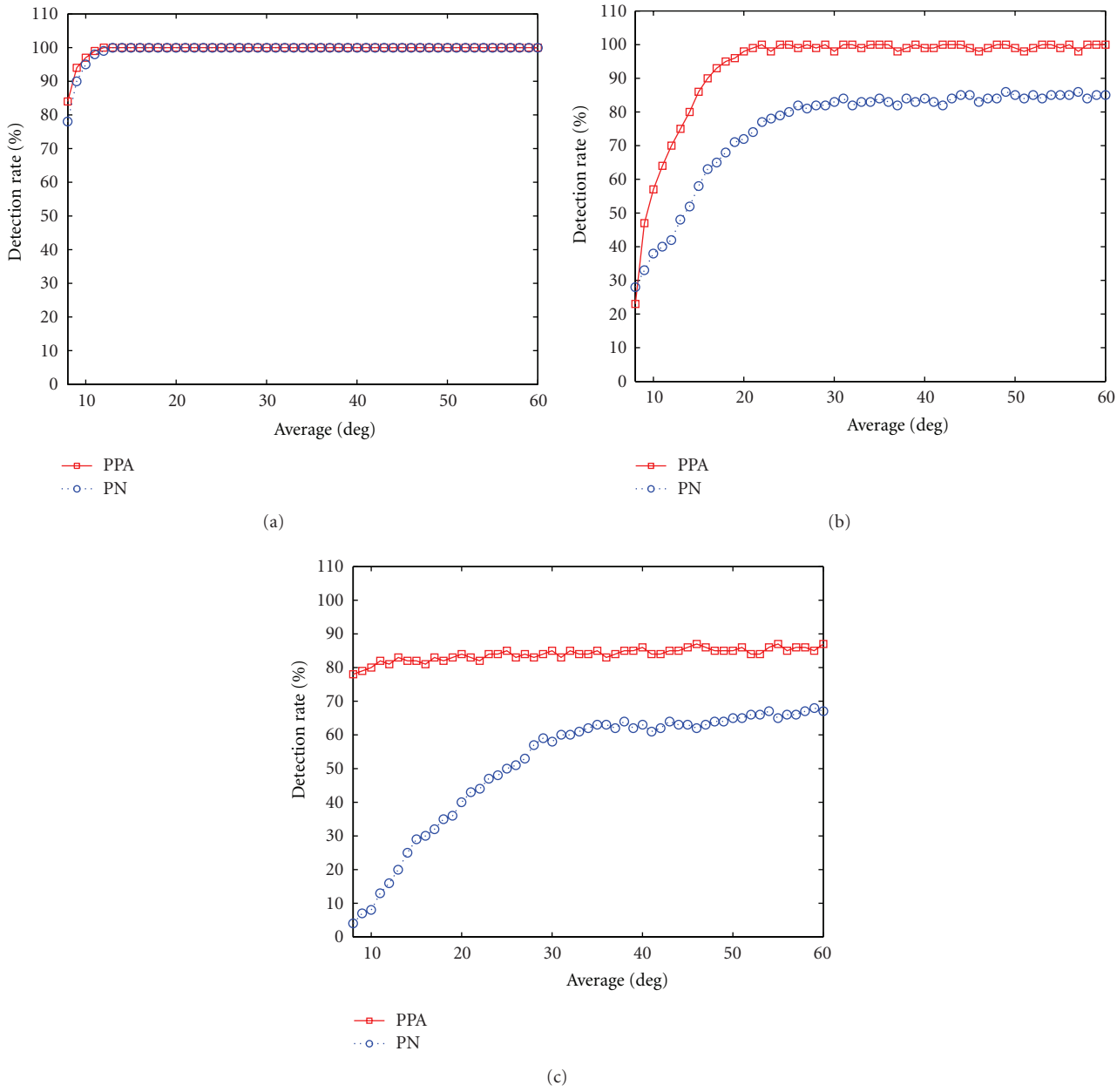


Figure 4: Percent Detected against various node density.

(homeomorphous) the three dimensions sphere, then it have no holes on the monitored areas based on the proofed of Poincare Conjecture, vice versa.

6. Conclusion

In this paper, a novel CT reconstruction model is proposed based on the approximate inverse where the kernel of the FDK method is derived and is used to complete the reconstruction. In order to eliminate the imposed ring artifacts, the kernel is truncated with proper radius. Reconstruction results show that the compact support FDK kernel reconstruction model can suppress the ring artifacts. The proposed reconstruction model preserves the simplicity of the FDK reconstruction method and also provides an alternative to realize the approximate inverse method for circular trajectory. And when the kernel of an algorithm is modified, the corresponding reconstruction formula is also modified accordingly. And this give us another way to improve the existing reconstruction methods.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive feedback and valuable input. Thanks are due for the supports to our program from the TI, the

XILINX, and the Software School of Xidian University. This program is partially supported by NSFC (Grant no. 61072105, 61007011) and also supported by the Open Projects Program of National Laboratory of Pattern Recognition. The project is also partially supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program no. 2010JM8005) and Scientific Research Program Funded by Shaanxi Provincial Education Department (Program no. 11JK0504).

References

[1] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing holes in sensor networks," *Mobile Networks and Applications*, vol. 11, no. 2, pp. 187–200, 2006.

[2] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.

[3] B. Karp and H. T. Kung, "GPSR: greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

[4] J. Bruck, J. Gao, and A. Jiang, "MAP: medial axis based geometric routing in sensor networks," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, pp. 88–102, September 2005.

[5] Q. Fang, J. Gao, L. J. Guibas, V. De Silva, and L. Zhang, "GLIDER: Gradient Landmark-Based Distributed Routing for sensor networks," in *Proceedings of the 24th Conference of the IEEE Communication Society (INFOCOM '05)*, pp. 339–350, March 2005.

[6] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[7] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann, "Neighborhood-based topology recognition in sensor networks," *Lecture Notes in Computer Science*, vol. 3121, pp. 123–136, 2004.

[8] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 254–260, April 2005.

[9] A. Kröller, S. P. Fekete, D. Pfisterer, and S. Fischer, "Deterministic boundary recognition and topology extraction for large sensor networks," in *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1000–1009, January 2006.

[10] S. Funke, "Topological hole detection in wireless sensor networks and its applications," in *Proceedings of the Joint Workshop on Foundations of Mobile Computing*, pp. 44–53, 2005.

[11] S. Funke and C. Klein, "Hole detection or: 'how much geometry hides in connectivity?'," in *Proceedings of the 22nd Annual Symposium on Computational Geometry (SCG '06)*, pp. 377–385, June 2006.

[12] Poincare and Jules Henri, *The American Heritage Dictionary of the English Language*, Houghton Mifflin Company, Boston, Mass, USA, 4th edition, 2000.

[13] J. L. Kelly, *General Topology*, D. Van Nostrand, Princeton, NJ, USA, 1955.

[14] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: an experimental study of low-power wireless sensor networks," Tech. Rep. UCLA/CSD-TR 02-0013, UCLA, 2002.

[15] Y. J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proceedings of the 2nd USENIX/ACM Symposium Networked System Design and Implementation*, pp. 217–230, 2005.

[16] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, September 2003.

[17] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[18] F. Li, J. Luo, C. Zhang, S. Xin, and Y. He, "UNFOLD: uniform fast on-line boundary detection for dynamic 3D wireless sensor networks," in *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'11)*, pp. 141–152, 2011.

[19] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, "An analysis of the optimum node density for ad hoc mobile networks," in *International Conference on Communications (ICC '01)*, pp. 857–861, June 2000.