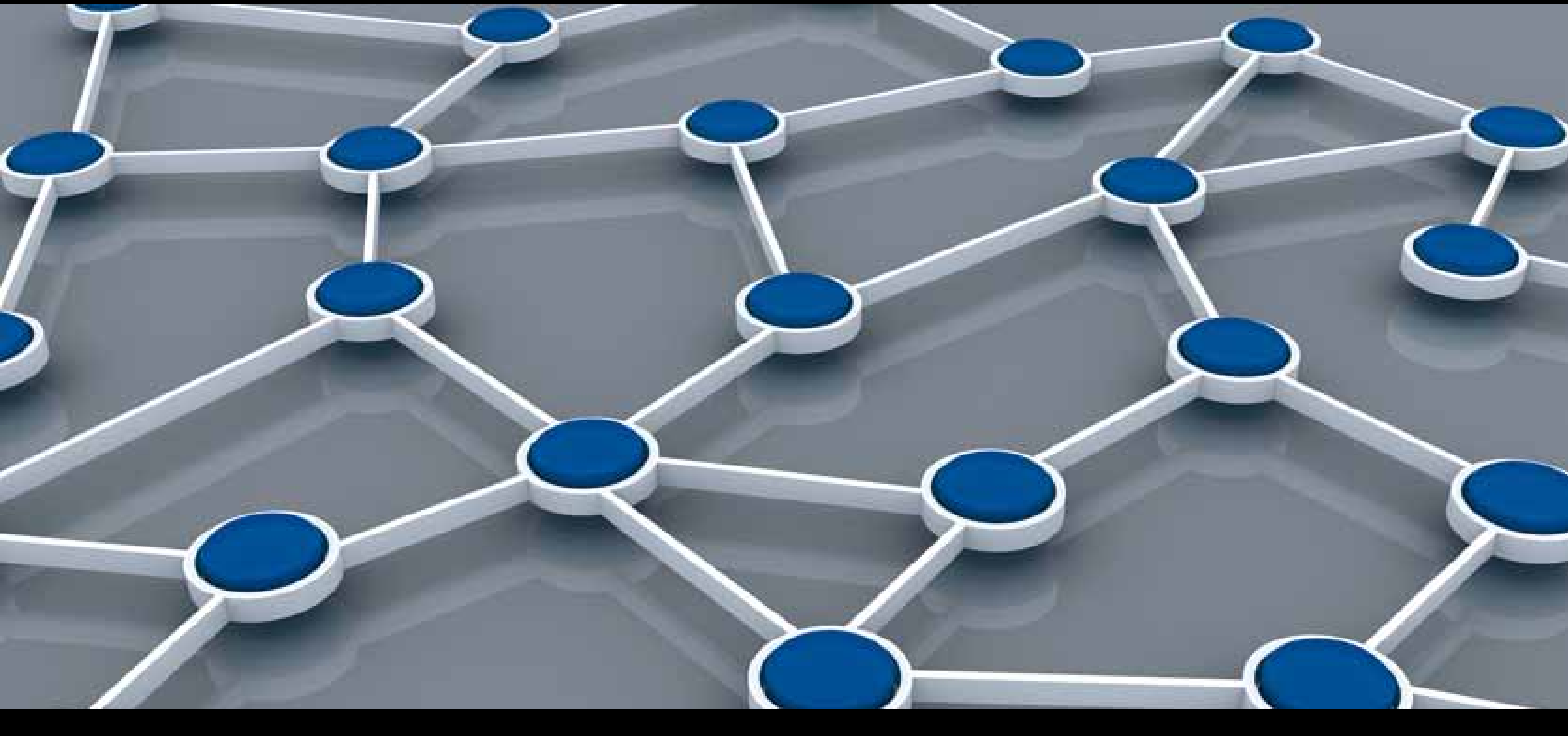
International Journal of Distributed Sensor Networks

Planning and Deployment of Wireless Sensor Networks

Guest Editors: Raouf Boutaba, Nadjib Achir, Marc St-Hilaire, and Eduardo Freire Nakamura



Planning and Deployment of Wireless Sensor Networks

International Journal of Distributed Sensor Networks

Planning and Deployment of Wireless Sensor Networks

Guest Editors: Raouf Boutaba, Nadjib Achir, Marc St-Hilaire, and Eduardo Freire Nakamura

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 198139, 2 pages
http://dx.doi.org/10.1155/2014/198139

Editorial
Planning and Deployment of Wireless Sensor Networks

Raouf Boutaba,[1] Nadjib Achir,[2] Marc St-Hilaire,[3] and Eduardo Freire Nakamura[4]

[1] *University of Waterloo, Waterloo, ON, Canada N2L 3G1*
[2] *University of Paris 13, 93430 Villetaneuse, France*
[3] *Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada K1S 5B6*
[4] *FUCAPI, Federal University of Amazonas, 69075-351 Manaus, AM, Brazil*

Correspondence should be addressed to Raouf Boutaba; rboutaba@uwaterloo.ca

Received 24 November 2013; Accepted 24 November 2013; Published 27 January 2014

The area of monitoring and control of physical environments has recently become a hot spot in the technology landscape. Currently, a number of companies are offering a plethora of sensing devices with different phenomena's monitoring capabilities. These devices, a.k.a. sensors, can be tiny with limited energy and processing capabilities; they can stand alone or be integrated in various structures (buildings, human bodies, vehicles, objects, etc.), and they can be deployed to form a wireless sensor network (WSN) in support to a variety of military, civil, and environmental applications.

In a WSN, the sensors are also capable of communicating with each other and in a multihop way propagating sensing information to data sinks and operations centers. In the last few years, wireless sensor networking has been a very active research area in both academia and the industry with a wide variety of applications such as area monitoring, environmental sensing, industry automation, structural monitoring, water and wastewater monitoring, surveillance, health monitoring, tracking of materials, and many others. While the set of challenges in sensor networks are diverse, researches have mainly focused on fundamental networking challenges, which include routing protocols, energy minimization, and data gathering. However, the performance of the proposed solutions strongly depends on the way sensors were positioned in the sensed area. To that end, the goal of this special issue is to report on recent advances in wireless sensor network planning, deployment, and management. Its scope includes both theoretical and practical contributions related to WSN architecture, planning, deployments, and applications.

This special issue starts with a paper focusing on the problem of cluster heads selections in cluster-based wireless sensor network architecture. The main objective is to select the optimal location and the optimal number of cluster heads that minimize both the intra- and intercluster energy consumption, when considering multihop routing protocol. To achieve this requirement, the authors start by modeling both the intra- and intercluster architecture and thus were able to determine the optimal size of each cluster and thus the optimal number of cluster heads.

"Optimal planning of distributed sensor layouts for collaborative surveillance" focuses on the WSN deployment issue in the case of surveillance coverage against moving targets. The authors develop a numerical optimization approach to place distributed sets of sensors to perform surveillance of moving targets over extended areas. To overcome the complexity of this problem the authors use a genetic algorithm based solution to find spatial sensor density functions that maximize effectiveness against moving targets. Finally, they numerically evaluate the performance of their solution using example use cases in general area surveillance and risk-based surveillance in protection of an asset.

"Optimal management of rechargeable biosensors in temperature-sensitive environments," "Dynamic sensor scheduling for thermal management in biological wireless sensor networks," and *"Wireless sensor network modeling and deployment challenges in oil and gas refinery plants"* focus on the usage of wireless sensor networks in specific application domains. In particular, "*Optimal management of rechargeable biosensors in temperature-sensitive environments*" and

"*Dynamic sensor scheduling for thermal management in biological wireless sensor networks*" focused on biological applications, where biosensors are attached or implanted into the body of a human or animal. Specifically, *"Optimal management of rechargeable biosensors in temperature-sensitive environments"* tackles the problem of finding an optimal policy for operating a rechargeable biosensor inside a temperature-sensitive environment characterized by a strict maximum temperature increase constraint. In this case, the authors model theoretically the problem using a Markov Decision Process (MDP). Moreover, to handle large-size MDP models they also show how operating policies can be obtained using a Q-learning approach. On the other hand *"Dynamic sensor scheduling for thermal management in biological wireless sensor networks"* focuses on the scheduling problem. Here also, the authors formulate the problem using an MDP approach and propose two specific types of states aggregation to produce a tractable solution. Finally, "*Wireless sensor network modeling and deployment challenges in oil and gas refinery plants*" considers the issue of wireless sensor network deployment in oil and gas refinery plants. More precisely, the authors propose different channel models based on the diffraction theory to assess the link quality in radio environments affected by highly dense metallic building blockage. The main idea is to split the wireless links into mutually exclusive attenuation classes based on the 3D structure of the building blockage. Each class is characterized by a different amount of obstruction loss; therefore a separate channel model is proposed to predict the QoS for each link type. Experimental results confirm the effectiveness of the proposed method as it provides a practical tool for virtual network planning.

"A multipath routing approach for secure and reliable data delivery in wireless sensor networks" focuses on the security issue in WSNs. More precisely, the authors present and evaluate a secure and reliable routing mechanism offering different levels of security in an energy efficient way for WSNs. The main idea of the authors is to use a node-disjoint routing approach in order to split the messages according to different paths with different coding methods. The authors show that using this approach security and reliability can be enhanced while respecting the resource constraints of WSNs.

"A cross-layer framework for network management in wireless sensor networks using weighted cognitive maps" tackled the problem of WSNs management and proposes a cross-layer framework for network management that considers different conflicting network objectives, such as network lifetime, connectivity, and coverage. To achieve this requirement, the authors use the Weighted Cognitive Maps (WCM) tool to provide a parameterized representation of conflicting system processes. The WCM continuously monitors the required QoS levels specified by the user and takes fast and efficient actions whenever those levels are violated.

"An Internet of Things approach for managing smart services provided by wearable devices" focuses on the generalization of the wireless sensor network concept to the emerging area of Internet of Things (IoT). Specifically, the authors presented an autonomous physical condition performance application, based on a WSN, bringing about the possibility of including several elements in an IoT scenario: a smart watch, a physiological monitoring device, and a smartphone.

Acknowledgments

Finally, we would like to express our thanks to all authors and reviewers for their contributions to this special issue.

Raouf Boutaba
Nadjib Achir
Marc St-Hilaire
Eduardo Freire Nakamura

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 480981, 10 pages
http://dx.doi.org/10.1155/2013/480981

Research Article

Multihop-Based Optimal Cluster Heads Numbers Considering Relay Node in Transmission Range of Sensor Nodes in Wireless Sensor Networks

Choon-Sung Nam,[1] Sang-Tae Bae,[2] Jin-Wook Chung,[1] and Dong-Ryeol Shin[1]

[1] *School of Information and Communication Engineering, Sungkyunkwan University, Suwon 440-774, Republic of Korea*
[2] *Knowledge & Information Division, Korea Institute of S&T Evaluation and Planning, Seoul 137-130, Republic of Korea*

Correspondence should be addressed to Dong-Ryeol Shin; drshin@skku.edu

Received 23 July 2012; Revised 11 April 2013; Accepted 21 April 2013

Academic Editor: Nadjib Achir

The data transmission of sensor nodes which detect events might be affected by their neighbor nodes located in their communication range. Thus, we analyze the energy consumption of sensor networks as a function of the number of cluster heads considering the above two options, multihop-based communication and the case where a neighbor node is in the transmission range for communicating data. This helps us to elect the number of energy-efficient cluster heads in a more practical manner. Also, we determine the effect of electing the number of cluster heads by restricting the local cluster size.

1. Introduction

Generally, wireless sensor networks (WSNs) are organized with sensor nodes and a wireless module in a specific area [1]. These sensor nodes have four basic components: a wireless network component, a sensor component to detect the environment, a power component for the electricity supply, and a process component for data processing. To adapt them for a specific purpose, a location finding system such as GPS, a mobility device, and a power generator can be added [2]. In WSNs, the sensor nodes have to establish a self-organizing network to collect data anywhere. To provide this, we have to consider the limitations of the sensor nodes. Because a sensor node is a lightweight, small, and low-power device, energy-efficient power consumption is an important issue [3]. Most of the energy consumption in WSNs occurs when the sensor node sends or receives event data.

The routing algorithms of sensor networks can be divided into flat- and hierarchical-based routing [4]. In flat-based routing, all of the nodes are on the same level in the network. Thus, global information is used for routing. All of the nodes have access to this global routing information. On the other hand, in hierarchical routing, local information is used for routing. Therefore, all of the nodes have to conserve the local routing information. This local information is bounded to a specific zone or event area. Hierarchical-based routing is superior to flat routing. The data transmission delay in hierarchical routing is lower than that in flat routing using contention for scheduling, because the former can reserve the transmission time in and out of the cluster area. For the same reason, in hierarchical routing, it is easier to synchronize information on the network. Through the process of reservation, as hierarchical routing can guarantee channel assignment, it can use collision avoidance. Thus, hierarchical routing makes it possible to achieve stable transmission. From the viewpoint of the energy consumption, hierarchical routing can maintain steady energy consumption regardless of the traffic pattern, because of the reservation of the transmission time. Hierarchical routing can prevent data duplication by aggregating similar or identical data by means of a cluster head. Due to these merits, hierarchical-based routing can achieve data load balancing between the nodes, distribute the energy consumption in the network, and increase the node and network lifetime [5, 6].

Generally, a local cluster in a clustering mechanism is formed by a cluster head, and a cluster head is elected by a sink node or among nodes which are scattered in the network. Previous proposals [7–10] based LEACH [7] that cluster head (CH) is able to communicate with a sink node via one single hop have tried to select the nodes whose energy consumption is the most minimum in the node groups as CHs compared with (or in proportion with) the number of CH and member node (MN) in the network. But in the real world this algorithm is possible to have a scalability problem when the network is extending, because its transmitting scope is always assumed as one single hop. The scope might be extended more depending on the cases. So, via previous papers, to migrate (or overcome) this limitation, multihop routing-based optimization of the number of cluster heads (MROCH) [11] has proposed the customized determining CH selection algorithm that is able to find the optimizing number of CHs in the sensor network which has multihop transmission way considering the size of nodes' communication scope. However, this proposal still has limitation, not considering the delay node which is for handling with additional energy consumptions and data transmitting costs between MN and CH on the phrase of the cluster group initiation in the network. Practical data transmission in cluster-based sensor networks (PDTC) [12] algorithm is designed for selecting the CH and defining the cluster space that is established as CH location like others. It means that local cluster, covered by CH, is automatically established with CH as the center: whenever a new CH is selected, a new local cluster is also built. This algorithm is also able to count CH number via measuring energy generation of clustering initiation based on the following principle. However, since not only, it requires additional energy consumption to build local clusters whenever a new CH is defined, but also its communication cost among local clusters is calculated as the distance of its communication scopes and detouring path simply, it needs to be revision and modified to get better performances. Like following limitations, to get optimistic number of CHs in sensor network, the process of establishing local clusters in the network is required firstly via doing CH selection process. Also, the process of counting CH number is essential considering additional communication costs among nodes. Therefore, this paper will mention extendable sensor network clusters based on multihops majorly. Also, we will propose new CH selection algorithm considering additional communication cost among relay nodes during CH selection.

The structure of this paper is as follows. In Section 2, we describe the existing clustering algorithm. In Section 3, we set up the equation for cluster modeling. In Section 3, we describe the performance evaluation and analysis of the proposed method. Finally, in Section 5, we conclude this paper.

2. Related Work

To collect the information of the sensor networks in an area without a network infrastructure, the sensor nodes need to create an ad hoc wireless network. WSNs, however, are not suitable for using existing routing mechanisms, because of the features of the sensor nodes. Thus, WSNs need an improved ad hoc routing algorithm considering self-organizing networks, data centric communication, the restricted capability of the nodes and so on. Generally, adjacent nodes in WSNs have similar environmental information. These nodes require a clustering mechanism to aggregate the event data from member nodes and to prevent redundant data communication in a local cluster. In WSNs, the representative clustering algorithm is LEACH [7]. The goal of LEACH is to distribute the energy consumption to all of the sensor nodes included in the cluster heads. To achieve this, LEACH circulates a cluster head and elects it randomly. Revised and expanded clustering mechanisms based on LEACH have been studied by many researchers. Increasing the distance between a cluster head and member node increases the energy consumption. The distance depends on the location of the cluster head in a local cluster. When the cluster head is located in the center of a local cluster, the distance is the shortest. The algorithm which moves a cluster head to the center of a local cluster is LEACH-C [7]. Handy et al. [8] proposed improved LEACH algorithm, called LEACH-DCHS. This proposal is a CH selection algorithm considering the status of nodes remaining energy in the network. But it has a critical problem (or limitation) that the quality of service ability is getting worse and worse as the time of CH selection process is going more and more because it considers (or calculates) the amount of the remaining energy resources of the nodes. To solve this problem, another improved algorithm, LEACH-DCHS CM (LEACH-DCHS cluster maintenance) [9] had been proposed, but there is still the same problem. Also, advanced low energy adaptive clustering hierarchy (ALEACH) [10] algorithm, calculating the nodes remaining energy amounts based on energy information, was proposed to overcome, but it was not easy to get optimistic number of CHs in the network like others. A cluster head consumes more energy than the member nodes, because it has to aggregate the data from the member nodes. So, HEED [13] selects a cluster head considering the remaining energy of the sensor nodes. These algorithms are based on single-hop communication between a cluster head and member nodes and between a cluster head and a sink node. linked cluster algorithm (LCA) [14] assumes the connection between a cluster head and member nodes, intracluster, to be a single hop. Adaptive clustering [15] makes the same assumption. Unlike LCA and adaptive clustering, CLUBS [16] uses multihop-based communication in the Intra-cluster. The communication range of the sensor nodes, however, is based on IEEE 802.15.4 (LR-WPAN), which is one of the transmission standards for WSNs. IEEE 802.15.4 typically extends up to 10 m in all directions [17]. Thus, WSNs have to use the multihop-based clustering mechanism, because it is impossible to communicate with the sensor nodes with a restricted transmission range by using single-hop communication [18]. In the multihop-based clustering mechanism, the node energy consumption is affected by the local cluster size. This means that as the size of the local cluster increases, the nodes need more relay nodes to send the event data to a sink node or a cluster head, and the cluster size depends on the number of cluster heads, as the local cluster is formed by a cluster head. So, it is important to determine the number of cluster heads in multihop-based clustering algorithms. The practical data transmission algorithm [12] is a method that determines the number of cluster heads based on Voronoi tessellation. This means that if there is one spot in a specific area, the local cluster area is determined by the location of this spot, and if there is another spot around it, the area is divided into two equal spaces containing these two spots. Whenever a new spot is added to it, the area can be divided into equal spaces, because of the iteration of this process. Multi-hop routing-based optimization [11] optimizes the energy consumption of the nodes as a function of the number of cluster heads. To achieve this, it models the distance between the nodes in the Intra-cluster and intercluster parts. Through this model, it can determine the effect on the energy consumption of the local cluster and whole network as a function of the number of cluster heads. Though these two mechanisms use multihop-based clustering inside and outside of a local cluster, they do not consider which nodes can relay the event node in the transmission range of the original node. In other words, they do not apply the detour caused by the relay node. This detour can be affected by the location of the nodes. The original node should not select all of the nodes in the transmission area as the relay node but should just select them as candidate nodes for detours. Among the candidate nodes, the node which is closest to the cluster head or sink node can be selected as the relay node. Then, the original node can set up the direction to the cluster head or sink.

3. Cluster Modeling of WSNs

In WSNs, the process of clustering starts by electing the cluster heads. The node elected as a cluster head sends its cluster head information to its neighbor nodes or member nodes in its transmission range, and the member nodes that receive this information also send it to their neighbor nodes. This work continues until the time the node meets a member node included in another cluster head. Through these processes, each local cluster is established. As the number of local clusters is the same as the number of cluster heads, determining the number of cluster heads is the same as determining the size of the local cluster. The greater the size of the local cluster, the greater the number of member nodes which a cluster head has. The energy consumption of a cluster head also increases. On the other hand, the smaller the size of a local cluster, the lower the number of member nodes. The energy consumption of a cluster head also decreases. Increasing the number of cluster heads means increasing the transmission cost for a sink node in Inter-cluster. Thus, to determine the optimal number of cluster heads, WSNs need to know the relations between the size of the local cluster and the number of cluster heads, and to minimize the data path, the nodes should send their event data to those neighbor nodes that are closer to a sink node or a cluster head. To achieve this, the nodes have to set up the possible transmission area based on the location of the nodes. Finally, WSNs have to know the size of the local cluster by limiting its maximum size.

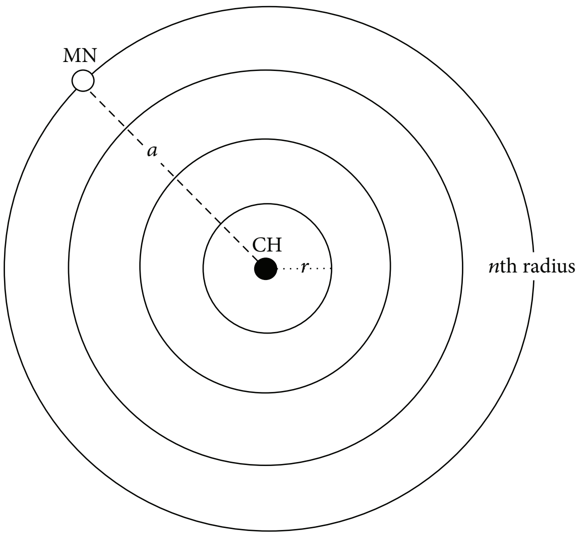


Figure 1: Local cluster modeling architecture.

3.1. Local Cluster Modeling. To model a local cluster, we assume the following: the network size is $A \times A$, the number of nodes is N, and the nodes are equally distributed in the network area. The transmission range of the nodes is restricted to 10 meters. The data transmission of the nodes is performed to collect and aggregate the event data or relay the data and to send the aggregated data to the neighbor nodes. The original node transmits the same amount of data in the same period of time. A cluster head is not any different from the other nodes, except that it can aggregate the event data from its member nodes and send it to a sink node by multihop-based communication.

Figure 1 shows the model of a local cluster. The location of the cluster head is at the center. The longest distance from the cluster head to a member node is "a" or "D_a" which is equal to the radius of the local cluster. The maximum transmission range is "r" which can describe 1 hop. So, a member node which is "n" hops away is located in the "nth" radius of the local cluster. As the network size is A^2, if there are "m" local clusters, it is equal to "$A^2 = m \times \pi a^2$." This means that the network size is the same as the total area of "m" local clusters. So, "a" can be described by the following (1)

$$A = D_a = \therefore D_a = \frac{A}{\sqrt{m\pi}}. \quad (1)$$

3.2. Intracluster Modeling. In Figure 1, the minimum number of hops between a cluster head and a member node can be described as "$\text{H}_{\min} = a/r$." If the neighbor nodes, or member nodes, are located at the maximum transmission range, the event data can be transmitted in the minimum transmission distance, as shown in Figure 2. Though the relay node is located at the nth radius, as shown in Figure 2, it can have different hop counts from the nth radius to the cluster head. In Figure 3, when node "d" transmits the event data to the cluster head, the data path is "d-c-b-a," which is the shortest path with the minimum hop count. On the other hand, node "1" can

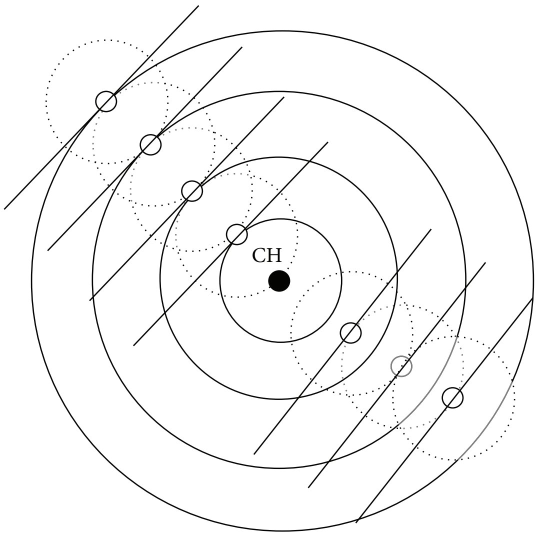


Figure 2: Transmission range of neighbor nodes.

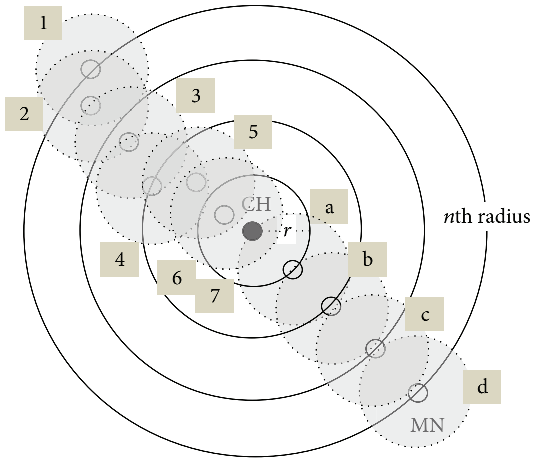


Figure 3: The shortest path and detour path.

transmit the event data through a detour path such as "1-2-3-4-5-6-7." Node "1" cannot help in selecting node "2," which is closer to the cluster head than itself, as the relay node because there are no nodes at the $n-1$th radius or with $n-1$th hop counts. After repeating the process, node "1" with a minimum of 4 hop counts transmits the event data to the cluster head over a data path with a hop count of 7. Thus, the data path or distance can be increased by the location of the nodes. To achieve this, we should select a relay node in the transmission range. The process of doing this is as follows.

To determine the number of nodes located in the nth area, which is the area between the nth and $n-1$th radius, we assume the ratio of the nth area to be as follows:

$$\mathrm{Ar}_n\mathrm{th} = \frac{\pi n^2 - \pi(n-1)^2}{\pi a^2}. \tag{2}$$

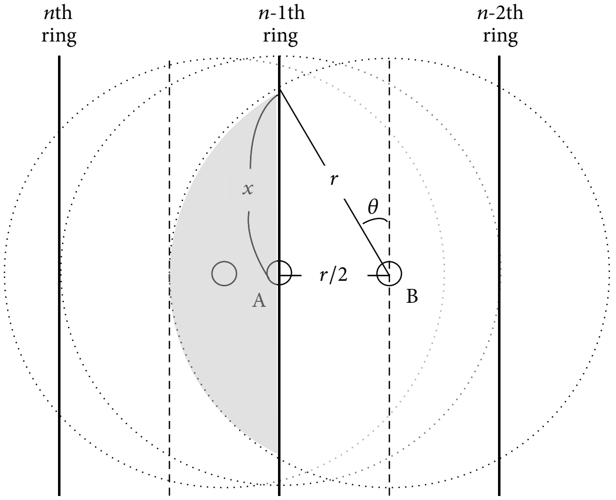


Figure 4: Transmission range of model architecture.

The number of nodes in the nth area is used to multiply the total nodes "N" by the rate of the nth area, Ar_nth. This is accomplished using the following (3):

$$\mathrm{C}_n\mathrm{th} = N \cdot \frac{\pi n^2 - \pi(n-1)^2}{\pi a^2}. \tag{3}$$

As described above, the data path can be increased by location of the nodes with nth area. Therefore, considering the best and worst node locations, the distance and hop counts of the data path should be calculated. In Figure 4, the transmission range of node "A" in the $n-1$th area can cover from the nth area to the $n-2$th area. In this case, the optimal relay node of node "A" is the node located at the nth radius, as shown in Figure 3. Though there are no nodes located at the nth radius, node "A" can select the nodes in the $n-1$th area as the relay node. They are better than the nodes in the $n-2$th area because their data path is shorter. Thus, the data transmission range for selecting the relay node is at least half of the transmission range. Equation (4) shows this:

$$\mathrm{Ar}_n = \frac{\pi r^2}{2}. \tag{4}$$

Node "B" is located at the center between the $n-1$th area and $n-2$th area. It is better for node "B" to pick the node in the $n-1$th area than the node in the $n-2$th area as the relay node. Therefore, the data relay area can be limited by the black spot, $\mathrm{Ar}_{\mathrm{relay}}$, in Figure 4.

This area can be calculated by formula Ar_nth in the $n-2$th area minus the area of the black spot not included in the $n-2$th area. This can be described as follows:

$$\mathrm{Ar}_{\mathrm{relay}} = \frac{\pi r^2}{2} - \left(\frac{\sin\theta}{360} \cdot \pi r^2 \cdot 2 + \frac{r}{2} \cdot x \cdot 2\right). \tag{5}$$

The number of relay nodes in the $n-1$th area is to multiply the number of nodes in the $n-1$th area, C_nth, by the rate of black spot area, $\mathrm{Ar}_{\mathrm{relay}}$, among Ar_nth. Therefore, the number

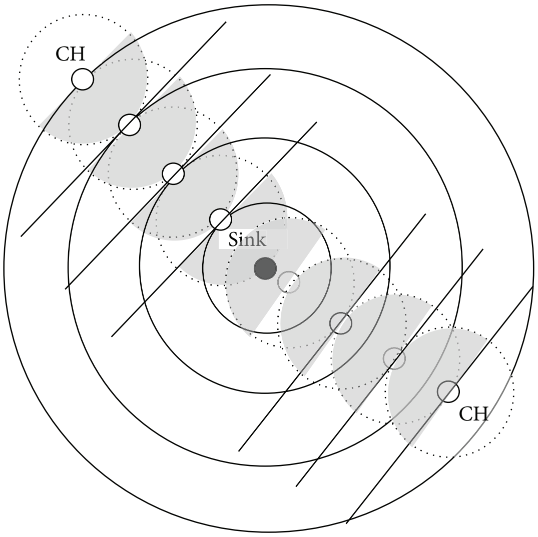


FIGURE 5: Data path to a sink node as a function of the transmission range of the sensor node.

of relay nodes in the $n-1$th area is obtained by the following equation in the Intra-cluster part:

$$\text{Intra\_C}_{\text{relay}} = \left(\frac{\text{Ar}_{\text{relay}}}{\text{Ar}_n\text{th}}\right) \cdot \text{C}_n\text{th}. \quad (6)$$

3.3. Intercluster Modeling. The aggregated data should be transmitted to a sink node by a cluster head in the intercluster. The number of relay nodes which send the aggregated data to a sink node is affected by the number of cluster heads. When a cluster head sends the aggregated data to a sink node, the transmission data range should also be considered as in the case of the Intra-cluster. As shown in Figure 5, a cluster head beyond the one-hop range has to send the aggregated data to a sink node using multi-hop communication. In this case, the cluster head also selects nodes in the black spot which are closer to the sink node and sends the data to the sink node. This is the same as the method used by member nodes to send the event data to a sink node in the Intra-cluster part. However, the transmission area of a cluster head is not affected by the nth area. In the Inter-cluster, the ratio of the nth area is always given by (7):

$$\text{Ar}_n\text{th} = \frac{\pi r^2/2}{\pi a^2}. \quad (7)$$

So, the number of relay nodes in the nth area is calculated by (8).

$$\text{Inter\_C}_{\text{relay}} = \frac{\pi r^2/2}{\pi a^2} \cdot N. \quad (8)$$

3.4. Cluster Depth Modeling. In the above intra- and inter-cluster modeling, we found that the number of cluster heads is related to the size of a local cluster. The cluster size is the radius of a local cluster. It can be presented as the hop count. If the depth of a local cluster is "d," other local clusters are located outside of the area defined by "d." This means that the number of local clusters can be increased by "d." As the cluster radius "d" can be presented in terms of the hop count, "d" can be given as "a/r." Therefore, "d" can be redefined by (9). Using (9), the number of clusters can be calculated:

$$d = \frac{a}{r} = \frac{A}{r\sqrt{m\pi}}. \quad (9)$$

4. Performance Evaluation and Analysis

4.1. Network Configuration. An ns2 [19], simulator is used in order to experiment with the supposed algorithm above. Ns2-allinone-2.27 is installed in cygwin of Windows XP SP3. Environments of sensor network are as follows. Channel is WirelessChannel, radio propagation model is TwoRayGroud, wireless physical and MAC protocols are 802.15.4, queue is Queue/DropTail/PriQueue, antenna model is OmniAntenna, and routing protocol is revised by LEACH. Data of sensor network references real sensor node, MiCaz based on data packet of 802.15.4 [20].

The network configuration is as follows. The networks size is 100 m × 100 m. The transmission range of a sensor node, "r," is 10 meters. The total number of sensor nodes is 100. They are scattered uniformly in the network. The sink node is located at a distance "r" from the network. The size of a data packet is 525 bytes. The receive energy and transmit energy are the same as in the existing research [11]. After the process of clustering, the operation of clustering is divided into two parts: the process of Intra-clustering and Inter-clustering. In the processes of Intra-clustering, the member nodes send the event data to a sink node once per a specific time. After the cluster heads collect and aggregate the event data from all of the member nodes, the process of Inter-clustering starts. The cluster head sends the aggregated data to a sink node. To know the node transmission range by the position of the nodes, the position of the nodes in the nth area is set up various position, line, center, left, and right (see Figure 6).

4.2. Performance Analysis. The network can be divided into local cluster heads according to the number of cluster heads. The size of each local cluster is affected by the number of cluster heads. In a local cluster, the number of relay nodes which can transmit the data to the neighbor nodes is shown in Figure 7.

As shown in Figure 7, the node located at the line, node_line, has more relay nodes than the nodes in the other locations. On the other hand, the node located on the right near the nth line, node_right, has less relay nodes. The number of member nodes of each local cluster decreases with increasing number of cluster heads, because the size of a local cluster becomes smaller. For this reason, the number of relay nodes is decreased by increasing the number of cluster heads. As shown in Figure 8, in the case where there is one cluster head in the WSN, the number of relay nodes of the node_line is about 5. In the case where there are 15 cluster heads in the WSN, the number of relay nodes is about 2. In this case, the node_line can transmit the event data to the neighbor nodes,

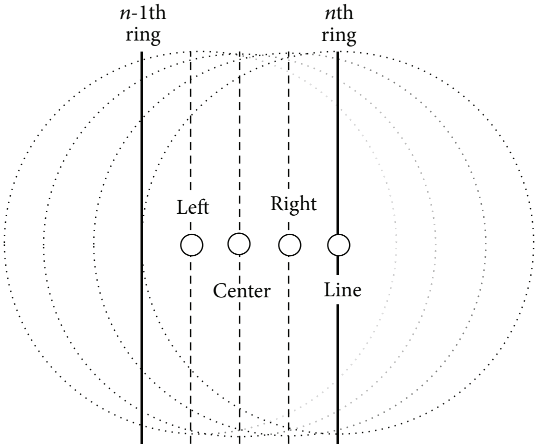


Figure 6: Position of a node in *n*th ring.

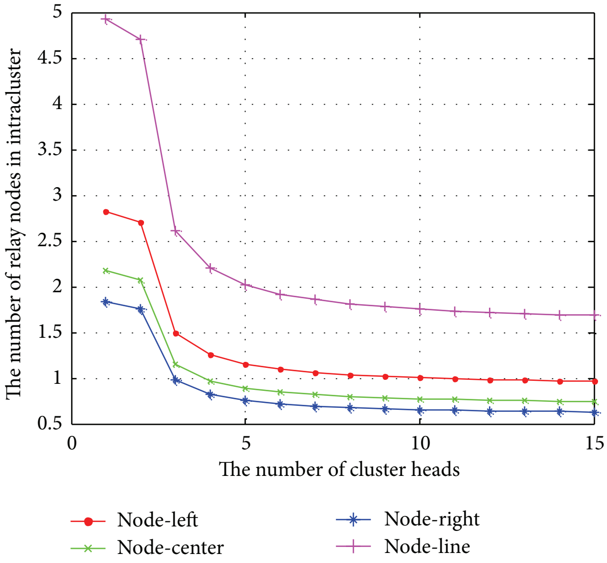


Figure 7: The number of relay nodes in intracluster.

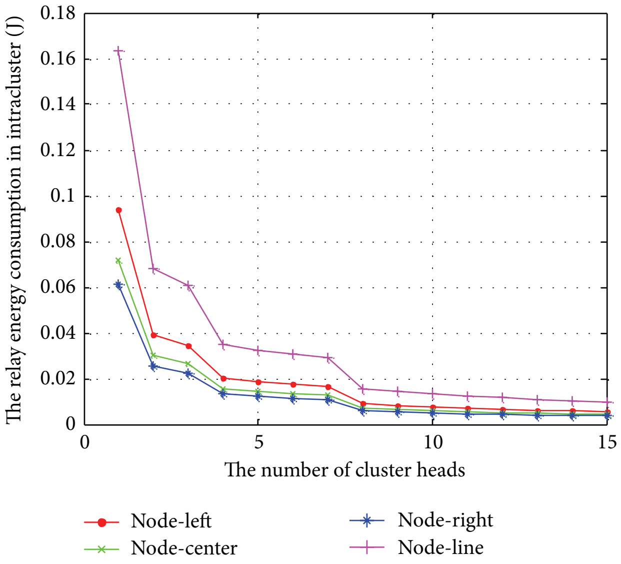


Figure 8: Transmission energy consumption in intracluster.

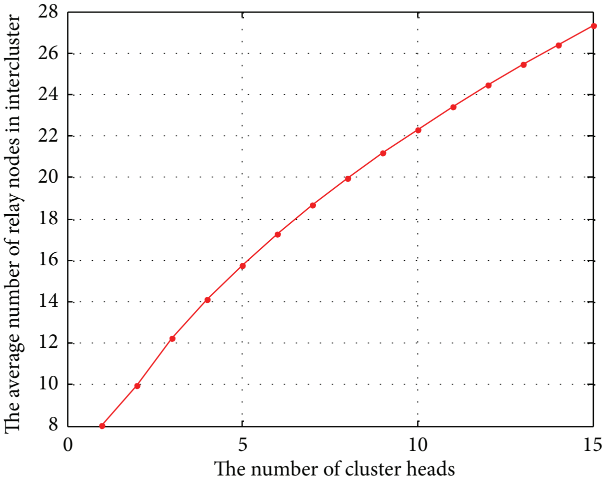


Figure 9: The average hops between a cluster and a sink node.

though the relay nodes are decreased. However, the node_line is the best case scenario for transmitting the event data. In the case of 15 cluster heads, the other nodes have lower relay nodes than 1 in its own transmission area. This means that they have to select detour.

Figure 8 shows the energy consumption of a local cluster. If the number of cluster heads is increased, the transmission energy of a local cluster is decreased, because the relay distance between the cluster head and its member nodes is shortened. As the number of cluster heads is increased, the amount of energy for a given location type is decreased. This is why the cluster size becomes smaller. The energy consumption of a local cluster would be the same in any location, as the cluster size is lower than 1 hop.

In the Inter-cluster operation, the number of relay nodes is not related to the local cluster size, unlike in the case of the Intra-cluster operation. It is related to the transmission range of the sensor nodes which play the role of relay nodes. As the transmission range of a cluster head is the same as that of the ordinary nodes, the energy consumption in the Inter-cluster operation is only related to the number of relay nodes within the transmission range, the distance between the cluster head and a sink node, and the number of cluster heads. Figure 9 shows the average number of hops in the case where the distance between a cluster head and a sink node is set to the shortest path. As the number of cluster heads increases, the distance between a cluster head and a sink node is increased.

Figure 10 shows the energy consumption as a function of the number of cluster heads in the Inter-cluster operation. The energy consumption increases rapidly with increasing number of cluster heads, because of increasing the number of relay nodes by distance.

In this paper, our selection algorithm proposes optimistic and suitable number of CH nodes before making cluster

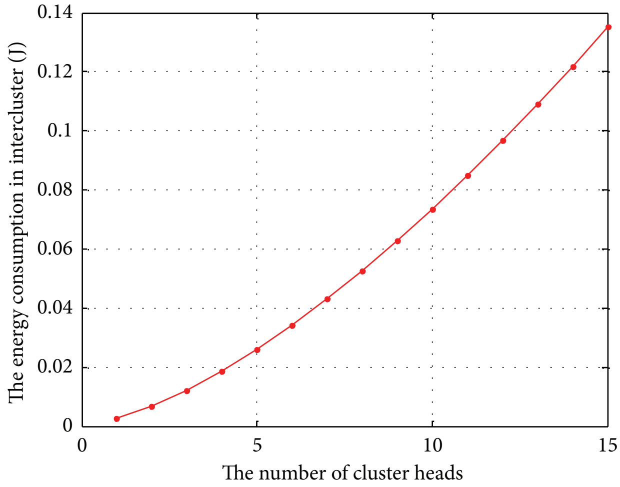


FIGURE 10: Transmission energy consumption between a cluster head and a sink node.

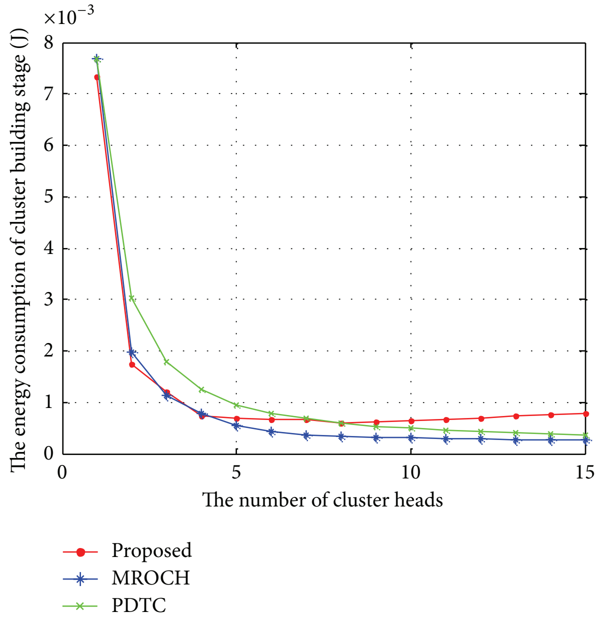


FIGURE 11: The energy consumption of cluster building stage per a local cluster.

group. So, the energy consumption of clustering formulation is depended (or relies) on the number of nodes which is selected as CH. In other words, the energy consumption of clustering formulation would be getting decreased if its size is getting smaller. Like Figure 11, as the number of CHs becomes more and more, the energy usage of local cluster formulation is getting lower and lower. While previous reviewed algorithms, MROCH [11] and PDTC [12], show that the energy consumption of cluster formulation is consistently getting lower in proportion in CH's number, our proposal shows that the energy consumption is getting higher and higher after CH number is eight percent because of adding the energy consumption of detouring. Via this chart, we can determine that this algorithm can be affected in local cluster formulation in the network than others since the detouring energy consumption in Inter-cluster is much more than Intra-cluster one.

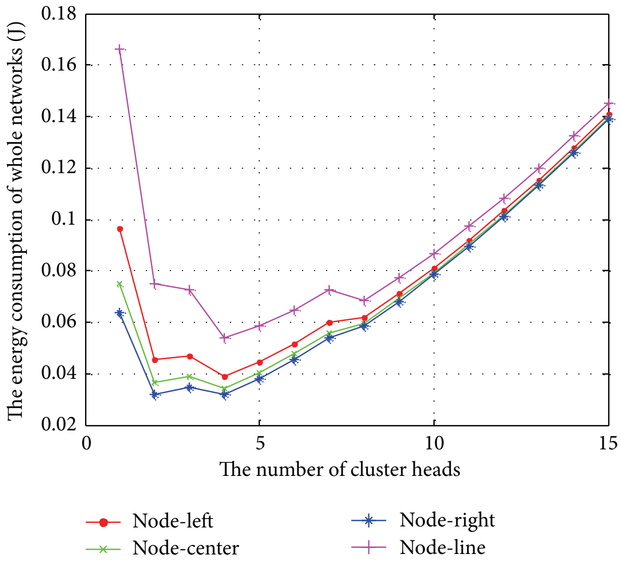


FIGURE 12: Total energy consumption of WSNs.

The total energy consumption is obtained by adding the Intra-cluster energy consumption to the Inter-cluster energy consumption. Figure 12 shows the total energy consumption as a function of the number of cluster heads. In all locations, the energy is decreased in the case where the number of cluster heads is 2, 3, or 4, because of the decrease in the number of relay nodes in a local cluster. In the case where there are more than 5 cluster heads, the total network energy increases rapidly as the Inter-cluster energy is higher than the Intra-cluster energy. Thus, we can see that total energy consumption is the lowest from 2% to 5%, under 0.04 Joule.

Figure 13 shows the number of relay node (RN) at the case when the node has the same hop-count and also has many data transferring request from many nodes compared with previous network steps during data transmission: the node selects a detouring node (DN) as its data transmission path. Compared with the previous literature MROCH [11], these algorithms show stable (of fixed) number of probable RNs between 5 and 14 when detouring because of not considering DN, but the number in our proposal is getting more and more since CH number is component rate is 3%. It means that increasing RN number considering detouring path is able to be affected with the energy consumption of the network.

Figure 14 shows and this following effect, component rate of RN considering detouring (DN) in the network visually. Also, we can find that the component rate of DN is over 15% when CH number is between 2 and 5% in the network via this figure. To sum up, like Figure 12, the entire amount of energy consumption is possible to be affected with the detouring path when CH number is around 2 and 5% in a whole network, the most minimum energy consumption case. It means that it is not easy to assume (or get) suitable

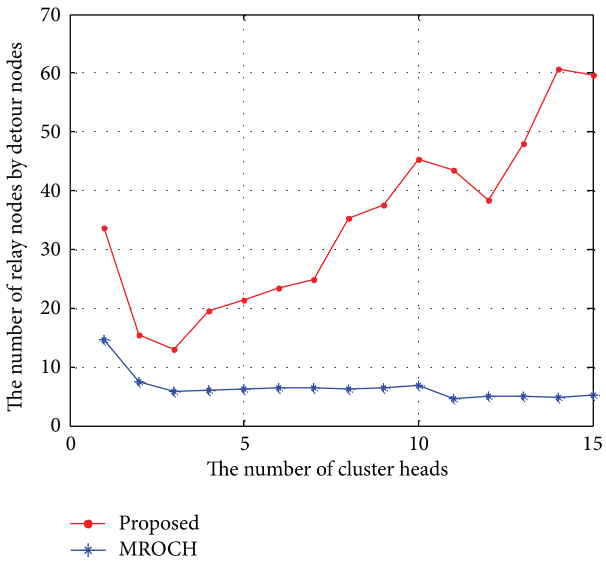


Figure 13: The number of relay nodes by detour nodes.

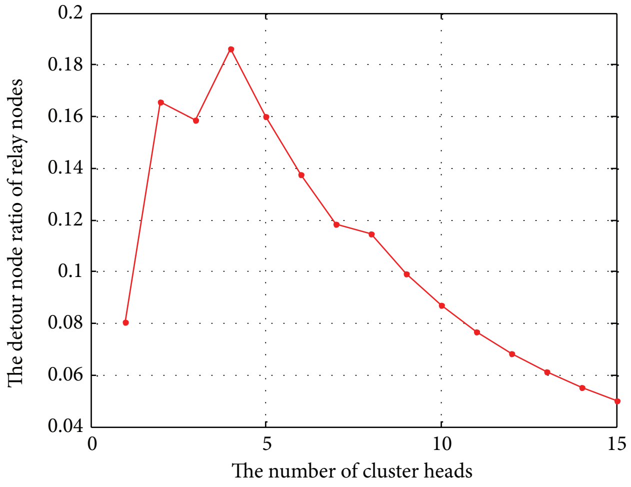


Figure 14: The detour node ratio of relay nodes.

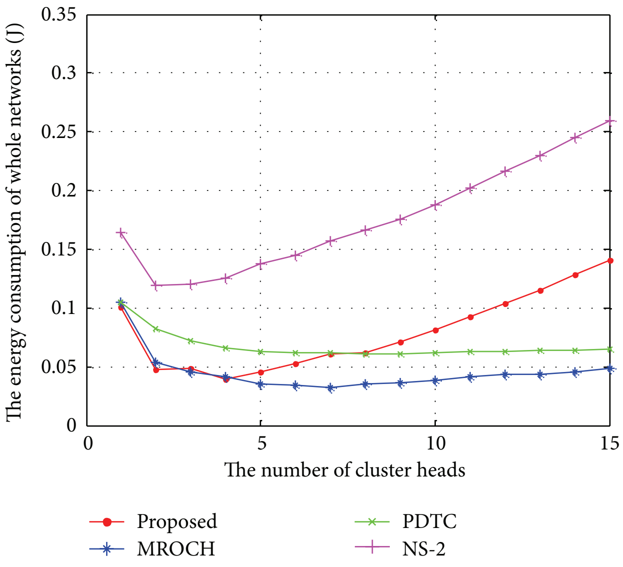


Figure 15: Compared with the proposed algorithm and other clustering-based algorithms.

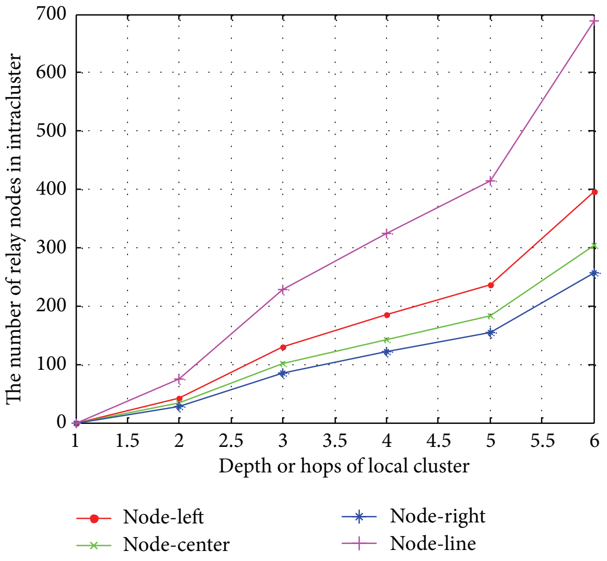


Figure 16: The number of relay nodes in intracluster.

CHs numbers exactly for the measuring of entire energy consumption in the network without detouring path (or DN).

To easily compare the variation of the entire energy consumption in the network with our proposal and other proposals, we need to focus on and analyze our clustering formulation algorithm and theirs firstly. But since LEACH assumed its communication scope as the only one single hop, we compared the performance of calculating optimistic CHs number with MROCH and PDTC via these mathematical equations. As a result, Figure 15 shows each variation of energy consumption and also represents that our proposal has suitable energy consumption states or performance if the component rate of CH number is around 2 and 5%: MROCH shows between 4 and 8%, and PDTC is around 5 to 9%. However, on "NS-2" network simulation, when the rate of CHs number in the network is around 2 to 4%, we can get optimistic energy consumption states. It means that in our proposal, considering additional energy consumption of detouring is able to calculate optimistic CHs number more correctly than other proposals.

If the local cluster size is less than or equal to 1 hop, there are no relay nodes. Therefore, when the size of a local cluster is "1," the number of relay nodes is "0" in Figure 16. The greater the size of a local cluster, the greater the number of relay nodes.

If the size of a local cluster is 1 hop or "r," the Intra-cluster energy consumption is almost equal to the total network energy consumption when the number of local clusters is 30. On the other hand, if the size of a local cluster is increased, the energy consumption of the Intra-cluster

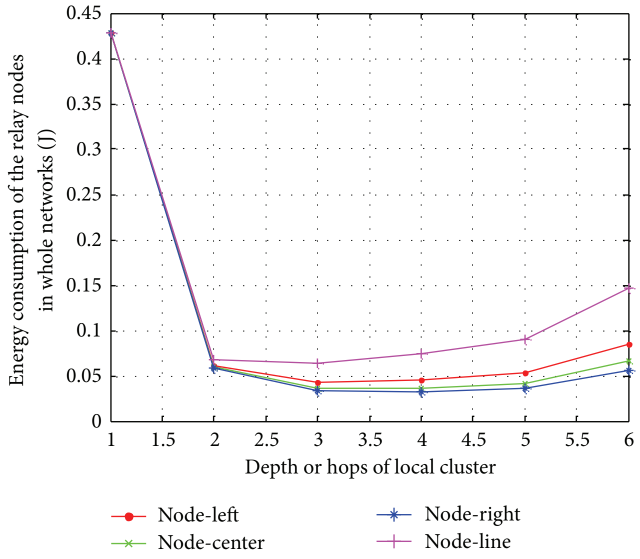


FIGURE 17: Energy consumption as a function of depth.

increases by increasing the number of relay nodes and the energy consumption of Inter-cluster decreases by decreasing the number of cluster heads. As the energy consumption of the Inter-cluster operation decreases rapidly with increasing the number of cluster heads in this case, the total energy consumption is decreased. Like Figure 17, until the size of a local cluster reaches 5 hops, the energy consumption continues to decrease or increase. However, when the size of a local cluster exceeds 5 hops, the number of relay nodes increases rapidly with the increasing distance between the cluster head and member nodes. Thus, the optimum depth of a local cluster from the viewpoint of the energy consumption is over 3 hops and under 5 hops. In the case of a depth of 3, that is 3 hops, the number of cluster heads is 3.5. In the case of a depth of 5, it is 1.3. Thus, the optimum number of cluster heads for energy efficiency is from 1.3 to 3.5.

5. Conclusion

Wireless sensor networks are networks used for monitoring and detecting environmental information using tiny sensor nodes with restricted capability in a specific area. WSNs have to use multi-hop-based communication by the sensor nodes with a limited transmission range and have to support energy efficiency mechanisms, as it is not easy to supply energy to a sensor node. Generally, sensor nodes tend to detect similar or the same event data. However, transmitting redundant data to other nodes is not energy efficient. To prevent this, a clustering mechanism is devised. The clustering mechanism of sensor networks can reduce duplicated data, as the cluster heads collect similar data from their neighbor nodes and thereby reduce the energy consumption. The clustering mechanism can collect the required data from a local cluster and make it possible to transmit the event data rapidly, as it forms local clusters based on the event features. The energy consumption associated with clustering is affected by the number of cluster heads and the size of a local cluster. The size of a local cluster is related to the number of cluster heads. The number of cluster heads affects the number of relay nodes, used for intra- and intercluster data transmission. Therefore, it affects the total energy consumption in WSNs. When the event node which detects the required data in the monitoring area selects a relay node, it is better to select a node which is closer to a cluster head or a sink node within its own transmission range in order to set up the shortest path. In this way, the path will be shortened. To achieve this, in this paper, we propose a method of selecting the node with less hops than its own hops in the transmission range. The locations of the nodes affecting the data path are also considered. Through equations, we determined the number of relay nodes and the energy consumption in the Intra- and Inter-cluster operations by means of the above considerations. We determined the variation of the energy consumption with the number of cluster heads and the size of a local cluster. By determining the energy consumption, we were able to determine the optimal size of a local cluster for a given number of cluster heads. Thus, we determined the optimal number of cluster heads in the clustering of WSNs.

Acknowledgment

This paper was supported by Faculty Research Fund, Sungkyunkwan University, 2011.

References

[1] A. Bharathidasan and V. A. S. Ponduru, Sensor Networks : An Overview, Dept. of Computer Science, University of California at Davis, 2002.

[2] I. F. Akylidiz, W. Su, Y. Sankarasubanmaniam, and E. Cayirci, "A survey on sensor networks," *Communication Magazine*, vol. 40, pp. 102–114, 2002.

[3] R. H. Katz, J. M. Kahn, and K. S. J. Pister, "Mobile Networking for Smart Dust," in *Proceeding of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, vol. 7, pp. 16–27, 1999.

[4] 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)*, January 2000.

[5] 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.

[6] M. Younis, M. Youssef, and K. Arisha, "Energy-aware management for cluster-based sensor networks," *Computer Networks*, vol. 43, no. 5, pp. 649–668, 2003.

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

[8] M. J. Handy, M. Haase, and D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *Proceedings of the 4th IEEE conference on mobile and wireless communication networks*, pp. 368–372, 2002.

[9] Y. Liu, J. Gao, Y. Jia, and L. Zhu, "A cluster maintenance algorithm based on LEACH-DCHS protoclol," in *Proceedings of the IEEE International Conference on Networking, Architecture, and Storage (IEEE NAS '08)*, pp. 165–166, Chongqing, China, June 2008.

[10] M. S. Ali, T. Dey, and R. Biswas, "ALEACH: advanced LEACH routing protocol for wireless microsensor networks," in *Proceedings of the 5th International Conference on Electrical and Computer Engineering (ICECE '08)*, pp. 909–914, December 2008.

[11] C. S. Nam, Y. S. Han, and D. R. Shin, "Multi-hop routing-based optimization of the number of cluster-heads in wireless sensor networks," *Sensors*, vol. 11, no. 3, pp. 2875–2884, 2011.

[12] D. Y. Kim, J. Cho, and B. S. Jeong, "Practical data transmission in cluster-based sensor networks," *KSII Transactions on Internet and Information Systems*, vol. 4, no. 3, pp. 224–242, 2010.

[13] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[14] D. J. Baker, A. Ephremides, and J. A. Flynn, "The design and simulation of a mobile radio network with distributed control," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 1, pp. 226–237, 1984.

[15] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[16] R. Nagpal and D. Coore, "An algorithm for group formation in an amorphous computer," in *Proceedings of the 10th International Conference on Parallel and Distributed Systems (PDCS '98)*, 1998.

[17] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks," *IEEE Network*, vol. 15, no. 5, pp. 12–19, 2001.

[18] K. Iwanicki and M. Van Steen, "Multi-hop cluster hierarchy maintenance in wireless sensor networks: a case for gossip-based protocols," *Lecture Notes in Computer Science*, vol. 5432, pp. 102–117, 2009.

[19] ns-allinon-2.27.tar.gz package, http://www.internetworkflow.com/downloads/ns2leach/ns-allinone-2.27.tar.gz.

[20] S. Sanka and G. Konchady, "Communication between Wireless Sensor Devices and GNU Radio," 2009, http://www.ece.uvic.ca/~elec499/2011-summer//group06/docs/papers/gaurav.pdf.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 145496, 13 pages
http://dx.doi.org/10.1155/2013/145496

Research Article

Optimal Planning of Distributed Sensor Layouts for Collaborative Surveillance

Thomas A. Wettergren and Russell Costa

Naval Undersea Warfare Center, 1176 Howell Street, Newport, RI 02841, USA

Correspondence should be addressed to Thomas A. Wettergren; t.a.wettergren@ieee.org

Received 27 September 2012; Accepted 13 March 2013

Academic Editor: Nadjib Achir

The use of a spatially distributed set of sensors has become a cost-effective approach to achieve surveillance coverage against moving targets. As more sensors are utilized in a collaborative manner, the optimal placement of sensors becomes critical to achieve the most efficient coverage. In this paper, we develop a numerical optimization approach to place distributed sets of sensors to perform surveillance against moving targets over extended areas. In particular, we develop a genetic algorithm solution to find spatial sensor density functions that maximize effectiveness against moving targets, where the surveillance performance of individual sensors is dependent on their absolute position in the region as well as their relative position to both the expected target(s) and any asset that is being protected. The density function representation of optimal sensor locations is shown to provide a computationally efficient method for determining sensor asset location planning. We illustrate the effective performance of this method on numerical examples based on problems of general area surveillance and risk-based surveillance in protection of an asset.

1. Introduction

Target surveillance in large areas is a difficult problem with many challenges; however, due to its importance for military operations it is one that has been studied extensively [1]. In the future, the importance of this problem will only grow as technical advances worldwide create more numerous and capable adversaries. This challenge has created more areas of the world where surveillance assets (sensors) must operate to achieve mission goals of varying scales. The descriptor “large area” is relative to the sensing capability of available (individual) sensors deployed in a specific region against specified targets of interest. A surveillance problem is deemed large area if the sensing capability of an individual sensor is small relative to the area to be searched (covered), in a fixed time scale. For example, problems can be defined in hours, days, weeks, and so forth, depending on tactical mission, and ultimately this will determine scale such as number of required sensors. Military surveillance problems may take the form of covering a bounded region against any intruders (the coverage problem), or may be more specific to covering a region around an asset of interest in order to protect the asset. In both of these situations, the selection of the best from a limited predefined set of surveillance configuration options is the standard practice [2].

Advances in sensor technology have made distributed sensor networks [3–5] a viable candidate technology for performing the military surveillance mission. In order for distributed sensor networks to achieve reasonable surveillance goals, some forms of collaboration must exist amongst the sensors. Historically, this collaboration has been managed in one of two ways. One method has been to partition the search region in such a way so that individual sensors are responsible for their own portion of the region of interest. This partitioning is done *a priori* using algorithms or human judgment to attempt to optimally split up the search effort among available sensors. The collaboration in this approach is limited to occasional reports (amongst sensors or to a central authority) which lead to suboptimal surveillance performance. The other common approach is to again partition the space, but with the emphasis on post processing of detection events. This approach focuses on the reactionary part of the problem (i.e., conditional on the presence and initial detection of a target) and thus, once again is suboptimal in its use of collaboration among sensors. In this work a methodology is developed to plan deployment of distributed sensors which includes a functional dependence on collaboration, as well as an explicit dependence on spatial variation in sensor performance.

With recent technological improvements in automation and communications networking capabilities, there has been an increase in the utilization of collaboration among sensors to perform target surveillance over large areas [6]. The focus of these studies for distributed sensor surveillance has been to spread out a number of sensors and use the spatial distribution of the individual sensors to cover a larger area (much larger than the coverage of any individual sensor) to monitor against intruders [7, 8]. These studies have been primarily for use in networks of sensors that are simple and autonomous in nature but have led to a fresh look at distributed surveillance particularly in the form of postdetection data fusion [9]. Other advances have used sensor repositioning after deployment to improve coverage, such as the use of virtual force algorithms [10] to move randomly deployed sensors to improve the coverage of the sensor network. While related, the problem of sensor network detection and classification algorithm design [11, 12] follows from the positioning of the sensors. We hold that the optimal placement of sensors will benefit from any further improvements gained from the detection and classification process. Similarly, the ability of the surveillance system to track any target of interest is critical to mission performance. Previous efforts in sensor network configuration have examined the positioning of sensors for target tracking applications [13], and it is well recognized that the target tracking performance of adaptively managed sensor networks is heavily dependent on the spatial deployment pattern [14]. In contrast to those efforts, the current paper is focused on the prior problem of maximizing the ability of gaining the initial detection for the surveillance application alone.

In this paper, we optimize distributed sensor configurations to achieve optimal surveillance performance. We utilize objectives based on a prescribed level of collaboration among sensors such that optimization of these objectives results in sensor placement that is optimal with respect to collaboration as well as individual sensor performance. This approach scales well with the number of sensors and, thus, is applicable to the large scale sensor network topologies down to the tactical scales more commonly found in current surveillance (search) problems. It is this latter scale that is the focus of this paper. In this distributed sensing objective, the sensors independently perform target detection and target detection decisions are made by comparing multiple noncollocated detections to check for kinematic consistency, as a form of target classification. If the individual detections are consistent (in spatiotemporal relation) with the anticipated target behavior, then the multiple detections corroborate and the collaborative sensors declare a target present.

The motivation of this work is to utilize a collaboration framework in a formal manner so that with modern computing resources, tactical decision aids can be developed to facilitate the command decisions with respect to collaborative sensors. With the formulation of a numerical objective, more target hypotheses can be considered than notional examination on which current approaches rely. To improve the performance of such a distributed sensor surveillance system, we consider the problem of determining the optimal layout of a group of sensors. Rather than optimizing all of the parameters of a system design, we focus instead on the key component of optimizing the opportunities for multiple sensor detections over time. Such opportunities are a critical first step in the many approaches to collaborative sensing. We consider this optimization of geometrical opportunities for collaboration as a fundamental goal of sensor layout planning, and secondary goals that are particular to a specific form of collaboration are beyond our scope. Our goal is the development of a computationally efficient numerical method that accounts for the geometric and environmental complexity of the problem, while maintaining enough generality to be useful in a variety of scenarios.

In the following, we refer generically to the platform that performs the sensing function as a "sensor," with the implied interpretation that all sensors have an underlying platform that holds them, be it a large manned asset or simply the device's housing. Thus a sensor may be as large as a manned radar or sonar platform, or as small as a simple proximity measurement device. The characterization of these devices, for our application, is given by their expected detection performance against the target of interest, which is presumed to be known from a prior model.

Given a fixed number of sensors, an expected distribution of target behavior, and a model of the sensors' detection performance in the region of interest, we develop an optimization framework that provides a sensor layout (set of sensor positions) for optimal surveillance protection of a region of interest under varying levels of collaboration. In particular, we consider three surveillance problems: (1) the detection of targets that are transiting throughout the region (typical area surveillance), (2) the detection of targets in the region that are far enough away from a high-valued unit (HVU) to provide reaction, and (3) the detection of objects that are weighted by their relative risk to the HVU. In the next section, we provide a mathematical model that accounts for all three of these distributed sensor surveillance problems. This common model has forms for both the cases of independent and of collaborative sensing and, thus provides a framework to study the implications of collaboration in the optimal positioning of sensors. The following develops a genetic algorithm-based optimization framework for optimizing sensor placement under this model. Finally, we conclude with some examples of the optimization to provide a comparison of the sensor layout patterns for various scenarios.

2. Mathematical Model of Distributed Surveillance

A crucial element in utilizing mathematical modeling to find practical solutions to problems such as optimal placement of distributed sensors (distributed assets) is in the formulation and numerical representation of the underlying objective. The formulation of the objective should be an accurate model of the problem that captures all parametric dependencies. In particular, any dependence on tactical parameters such as target behaviors and environmental characteristics requires a method that allows these parameters to be accounted for with varying levels of uncertainty. The numerical calculation of this objective should be a suitable approximation while being as efficient as possible to allow practical use in optimization approaches. The approach we follow is to model all of these dependencies in an integral formulation of expected performance over the search space. This integral form is then integrated with respect to any particular spatial distribution of sensors to arrive at probabilities representing expected surveillance field performance.

The first component in this model of distributed surveillance is the model for target motion (behavior). This model should allow varying levels of constraints on target motion to be of general use in a wide variety of problems. The model developed in this paper assumes the target motion to be Markovian in nature, such that its behavior can be decomposed into a sequence of short time behaviors. This assumption implies that the target motion path $\mathbf{y}(t)$ is effectively modeled by the sequence of intervals $\{[t_0, t_1), [t_1, t_2), \ldots\}$ and the path on the ith interval is $\mathbf{y}_i(t)$. The union of the collection of paths gives the total target path

$$\mathbf{y}(t) = \{\mathbf{y}_i(t) : t_i \leq t < t_{i+1}\}, \tag{1}$$

where each path $\mathbf{y}_i(t) = \mathbf{y}_{\tau,i} + v_i \cdot (t - t_i) \cdot [\cos(\theta_i), \sin(\theta_i)]^T$ represents a path of constant velocity target motion v_i in direction θ_i. Furthermore, each interval has motion parameters $\mathbf{p}_i = (\mathbf{y}_{\tau,i}, v_i, \theta_i)$ which are sampled from known distributions, and the specific values are only dependent on the previous time step, $\mathbf{p}_{i+1} = f(\mathbf{p}_i)$, as opposed to depending on the entire history (this is the Markov assumption). This Markov motion model is regularly assumed in modeling nonreactive targets [15] and is the basis of many Monte Carlo simulation approaches to target modeling [16]. We utilize the model to limit our analysis to optimizing the performance over a fixed, finite time step, where the motion of the target follows constant velocity during the interval of interest and the probability distributions of the motion parameters are all known *a priori*.

For a given interval $[t_i, t_{i+1})$, the probability of collaborative detection is a function of the random variables that describe target motion, as well as the location of the sensing assets and their detection performance. Consider a single given target motion track over this chosen interval. Assume that all of the sensors and the entire target track are contained within the surveillance region $\mathcal{S} \subseteq \mathbb{R}^2$ (we assume the region is large enough that edge effects are negligible). The probability of detecting this track by N_D individual sensor detections (the probability of successfully surveilling the track) is written as [17]

$$P_{\mathrm{ST}}(N_D \geq k) = 1 - \exp(-NP_D\phi) \sum_{m=0}^{k-1} \frac{(NP_D\phi)^m}{m!}, \tag{2}$$

where k is the minimum number of assets (sensors) independently detecting the target required for a collaborative detection of the target, while N is the total number of assets (sensors) in the surveillance region $\mathcal{S}$. The parameter P_D is the detection probability of an individual sensor, defined as constant within a given detection radius R_D. Note that the parameter k is used to define the level of collaboration in this framework. The variable ϕ represents the likelihood of a sensor being within distance R_D of a particular target track path to have an opportunity to detect the target (i.e., being within range of the target at some point during the track history). Explicitly, it is given by

$$\phi(\mathbf{y}_T, v, \theta) = \int_{\Omega_T(\mathbf{y}_T, v, \theta)} f(\mathbf{x})\, d\mathbf{x}, \tag{3}$$

where $f(\mathbf{x})$ is the distribution of sensors in the space, and the region Ω_T is the two dimensional region (defined by a specific target track) comprised of the subset of $\mathcal{S}$ that is within detection radius R_D of the track.

The relationship between target path and the probability of successful search criteria (for $k = 2$) is illustrated in Figure 1. Figure 1(a) shows a notional path through a rectangular search region in the presence of deployed passive sensors with detection circles as shown. A subset of this path is highlighted and magnified in Figure 1(b) to show a constant velocity segment of this path (from the given Markov parameters) and the subset of the sensors which detect this segment (and subsequently the given path). Note that one can view the detection process from a sensor frame of reference, that is, a detection occurs when two or more sensor circles contain the target segment, or through a target frame of reference, where the center of the sensor circles must be within the pill-shaped region (shaded area in Figure 1(b)) to detect the target. It is through this target frame of reference that we efficiently calculate P_{ST} given a distribution on sensor position. In particular, the expressions in (2) and (3) represent the random search [18] of a moving target "seeking" the fixed sensors when the problem is viewed from a target frame of reference. The resultant probability of successful search, P_{SS}, for this time interval is given by marginalizing the probability $P_{\mathrm{ST}}(N_D \geq k)$ over the uncertainty description of the target track as

$$\begin{aligned} P_{\mathrm{SS}}(N_D \geq k) &= \int_0^{2\pi} \int_{v_{\min}}^{v_{\max}} \int_{\mathcal{S}} P_{\mathrm{ST}}(N_D \geq k)\, f_T(\mathbf{y}_T) \\ &\quad \times f_v(v)\, f_\theta(\theta)\, d\mathbf{y}_T\, dv\, d\theta, \end{aligned} \tag{4}$$

where the functions $f_T(\mathbf{y}_T)$, $f_v(v)$, and $f_\theta(\theta)$ are probability density functions (PDF's) for target motion parameters of position, speed, and course, respectively. In addition, by increasing (or decreasing) k, we subsequently increase (decrease) the required level of collaboration among the distributed sensors.

In practice, a target track is successfully found by a collaborative sensor system based on sensors sharing detection information. Thus, it is not only dependent on the requisite

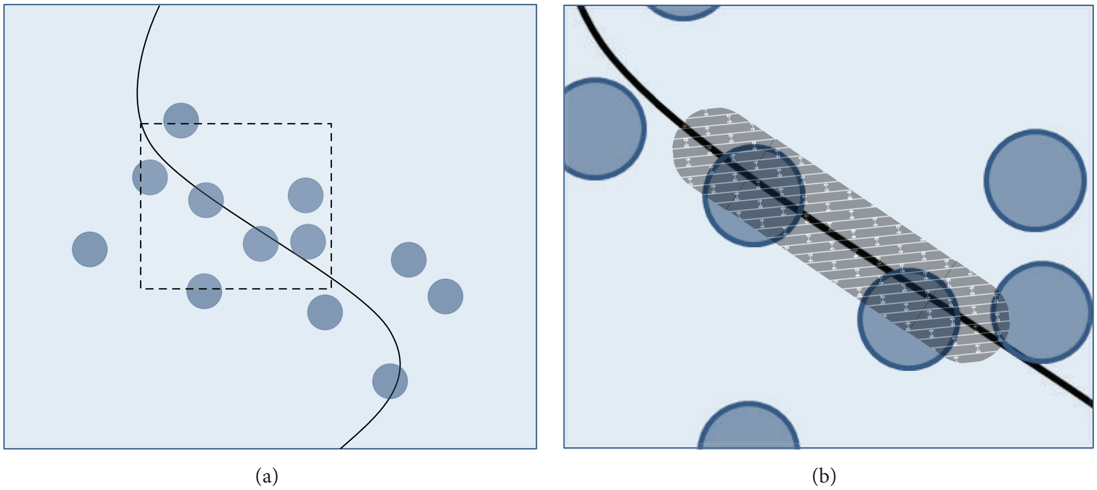


Figure 1: Notional example showing the connection between sensor placement and track coverage. (a) shows the target track path and the location of sensors with their coverage region. (b) is a blowup of the square box drawn in (a), showing a nominal "target pill" region representing a finite-time segment of the target track path.

number of sensors performing successful detections, but it also requires those sensors to communicate their results to other neighboring sensors. In the absence of full communication connectivity amongst the sensors, a graph model of the network node locations is used to assess the overall connectivity of the network in a probabilistic sense. Let $P_{\mathrm{CON}}(f(\mathbf{x}), N, r_c)$ represent the probability of connectivity of a sensor network of N nodes of communication range r_c that are spatially distributed according to the distribution $f(\mathbf{x})$. The probability of connectivity for the network is the probability that there exists some multihop path between any two nodes of the network, thus it is the probability that any network node can communicate with the rest of the network. The computation of $P_{\mathrm{CON}}(f(\mathbf{x}), N, r_c)$ for a fixed number N of nodes can be performed by known methods [19, 20]. Now the operational success of a search operation is the conditional probability of successful search conditioned on network connectivity. Mathematically, this is given by the joint probability expression $P_{\mathrm{SS}} \cdot P_{\mathrm{CON}}$ where P_{SS} is as given in (4). We note that both terms in this joint probability have a dependence on the sensor placement distribution $f(\mathbf{x})$. However, we assume that the sensors in our applications are densely spaced with respect to communications, such that every sensor can communicate reliably with all other nodes within the network. Such an assumption is common in passive detection systems, where the passive detection radius is often much smaller than the reliable communication distance. Thus, for the remainder of this paper, we consider only the case of completely connected networks (i.e., $P_{\mathrm{CON}} = 1$) and, therefore, the objective corresponding to an operational success of search is given by P_{SS} alone. The extension of our optimization technique to problems with limited connectivity is a subject of future work.

In order to numerically calculate the objective (4) over a variety of distributions from flat (uniform within search region $\mathcal{S}$) to highly nonuniform, we represent the sensor density function as a mixture of N_G circular Gaussian functions, as

$$f(\mathbf{x};\mathbf{w}) = \frac{1}{2\pi\sigma^2}\sum_{j=1}^{N_G} w_j \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{x}-\mathbf{x}_j\right)^T\left(\mathbf{x}-\mathbf{x}_j\right)\right), \quad (5)$$

with modal weights w_j, constant modal variance σ^2, and fixed (spaced equidistant in $\mathcal{S}$) positions $\mathbf{x}_j$, which has been shown to be a useful model for density approximation in many applications [21]. The number (and subsequent spacing) of the Gaussian modes required, as well as the value of the variance parameter σ, are chosen using a heuristic rule. The heuristic [22] is based on the flexibility of the overall density representation; that is, the relationship between the width of a Gaussian mode and the modal spacing; should be such that a wide variety of function forms of $f(\mathbf{x})$ can be represented. After experimentation on many test problems, we determined an appropriate relationship to be

$$N_G = \left(\left\lceil \frac{L}{3R_D} \right\rceil\right)^2, \quad (6)$$

where L is the length of the search region $\mathcal{S}$ along one dimension (assuming that $\mathcal{S}$ is square). The variance parameter is given as $\sigma = 2R_D$ for $R_D \ll L$. As R_D increases relative to L, this parameter should be made smaller relative to R_D; however, for the scale in this paper the given heuristic is applicable.

We note that P_{SS} provides a measure on the ability of multiple collaborative sensors to detect the target in a manner consistent with the spatiotemporal relationship of target motion and sensor position [17]. This is commonly referred to as track-before-detect and is an effective technique for reducing false alarms in distributed detection applications through collaboration defined by the aforementioned spatiotemporal relationship. It is also a method of multisensor filtering of contacts that is commonly used within many data fusion methods. Thus P_{SS} is, in general, a measure of track coverage in a surveillance region. However, as the length of the time interval tends to zero, P_{SS} becomes the more familiar metric of area coverage, that is, coverage independent of target motion. In that context, the expression in (2) is simply the composite area coverage provided by a set of independent sensors provided that their locations are randomly sampled from a common spatial distribution function $f(\mathbf{x})$.

3. Numerical Model

To optimize the placement of assets, the integrals in (3) and (4) must be evaluated with respect to changes in the sensor density function $f(\mathbf{x})$. In both cases these integrals do not have closed form solutions, and thus, must be evaluated numerically. Note first that the evaluation of the integral in (3) is significantly simplified by the representation of sensor density function defined in (5). Namely, through the fixed position and circular variance, the integral can be separated by mode (as a sum of the integrals of each mode) and in dimension (the two-dimensional integral can be separated by independence into the product of two one-dimensional integrals) independent of the modal weights. The latter property allows much of the computation to be done once, prior to entering the optimization, simplifying subsequent objective evaluations. This improvement in efficiency makes the optimization practical on standard desktop computers with no special coding requirements. Further simplification can be made by noting that for constant R_D (sensor performance independent of position in $\mathcal{S}$) the region of integration Ω_T given a target trajectory is a pill-shaped region with area $2R_D v\tau + \pi R_D^2$. This region can be well approximated by a rectangle of equal area for $v\tau \gg R_D$ which allows the integral in (3) to be evaluated using standard error functions commonly used for evaluation of integrals involving Gaussian functions [22]. The implementation utilized in this paper allows for spatial variability of R_D by including an additional step in which the equivalent rectangle is replaced by a series of rectangles (a partitioning) which approximate the track-dependent region (within which sensors have an opportunity to detect the target) by interpolation of an underlying R_D function. The number of segments that each track is partitioned (i.e., the number of rectangles) is determined *a priori* and depends on the extent of the spatial variability of R_D in the search region $\mathcal{S}$.

The mathematical detail required to evaluate (3) consists of the following. Consider an arbitrary target track of fixed length as defined above. Define an arbitrary point along this track $\mathbf{y}_{t_i}$ and a set $A_{t_i} = \{\mathbf{x} : \|\mathbf{y}_{t_i} - \mathbf{x}\| \le R_D(\mathbf{x})\}$, which is the set of all points in $\mathcal{S}$ from which an arbitrary sensor can detect a target at the specified point along the target track (with probability P_D). Then the general form of the "pill" shaped region of integration can be written as $\Omega_T = \bigcup_{i=1}^{\infty} A_{t_i}$. Next, define an approximation to this region as $\Omega^m = \bigcup_{j=1}^{m} A_{t_j}$ where t_j, $j = 1, \ldots, m$ refer to m equally spaced points spanning the length of the arbitrary track and construct disjoint sets B_{t_j} from sets A_{t_j} by the recursion $B_{t_j} = A_{t_j} - U^{j-1}$ given $U^k = \bigcup_{j=1}^{k} A_{t_j}$ and $U^0 = \emptyset$ (the empty set). Then the integral in (3) can be approximated as

$$\begin{aligned} \int_{\Omega_T} f(\mathbf{x})\,d\mathbf{x} &= \int_{\mathcal{S}} I_{\Omega_T}(\mathbf{x})\, f(\mathbf{x})\,d\mathbf{x} \\ &\approx \int_{\mathcal{S}} I_{\Omega^m}(\mathbf{x})\, f(\mathbf{x})\,d\mathbf{x} \\ &= \sum_{j=1}^{m} \int_{\mathcal{S}} I_{B_{t_j}}(\mathbf{x})\, f(\mathbf{x})\,d\mathbf{x}, \end{aligned} \tag{7}$$

where $I_A(\mathbf{x})$ is the set indicator function of the set A.

Assuming that the integral in (3) is well approximated by the above and that the function P_{ST} (as in (2)) changes slowly over the target track parameters in $\mathcal{S}$, the numerical evaluation of the integral in (4) can be done simply, provided that the PDFs of the target track parameters are continuously differentiable and slowly changing over their support. Since this is true for the examples in this paper, then in this work the integral is evaluated by gridding the track parameters (and associated weights) evenly over the track parameter space. This allows the triple integral in (4) to be well approximated by a triple sum weighted by the product of the corresponding values of the PDFs over the target parameter grid. The numerical evaluation of the sum as shown above provides a robust computation of the required integral of $f(\mathbf{x})$ over the region Ω_T.

4. Formulation of Optimization

The problem of optimum deployment of assets for collaborative multisensor surveillance, restricted to the mathematical model in (4), is one of maximizing search effectiveness in a fixed region. From an optimal planning perspective, the problem is one of maximizing the likelihood of achieving the surveillance mission, where the mission can take on various forms. We represent this as a minimization problem of the form

$$\min_{f} P_{\mathrm{MF}}, \tag{8}$$

where P_{MF} is the probability of mission failure and $f = f(\mathbf{x})$ is a function representing the sensor distribution over the region $\mathcal{S}$. In practice, this mission failure may take a variety of forms, but we are primarily concerned with the joint probability of not detecting a target within our surveillance region, combined with the risk associated with that target's presence. In particular, for the small time interval of interest which determines the path interval of interest, we have

$$\begin{aligned} P_{\mathrm{MF}} &= \int_0^{2\pi} \int_{v_{\min}}^{v_{\max}} \int_{\mathcal{S}} \left(1 - P_{\mathrm{ST}}\right) \psi\left(\mathbf{y}_T\right) f_T\left(\mathbf{y}_T\right) \\ &\qquad \times f_v(v)\, f_\theta(\theta)\; d\mathbf{y}_T\, dv\, d\theta \\ &= \int_{\mathcal{S}} f_T\left(\mathbf{y}_T\right) \psi\left(\mathbf{y}_T\right) d\mathbf{y}_T \end{aligned}$$

$$-\int_0^{2\pi}\int_{v_{\min}}^{v_{\max}}\int_{\mathcal{S}} P_{\mathrm{ST}}\left(\mathbf{y}_T\right)\cdot\left[f_T\left(\mathbf{y}_T\right)\psi\left(\mathbf{y}_T\right)\right] \times f_v\left(v\right) f_\theta\left(\theta\right) d\mathbf{y}_T\, dv\, d\theta, \tag{9}$$

where $\psi(\mathbf{y}_T)$ is a consequence (risk) function. The consequence function is dependent on the location of target track $\mathbf{y}_T$ and is defined to measure the relative risk posed by various tracks over that of others (such as those in proximity to an HVU, if that is the intention of the surveillance region). The first integral in (9) does not depend on the choice of the sensor location density $f(\mathbf{x})$, so it does not impact the optimization leading to an effective minimization objective of

$$J = -\int_0^{2\pi}\int_{v_{\min}}^{v_{\max}}\int_{\mathcal{S}} P_{\mathrm{ST}}\left(\mathbf{y}_T\right) \cdot\left[f_T\left(\mathbf{y}_T\right)\psi\left(\mathbf{y}_T\right)\right] f_v\left(v\right) f_\theta\left(\theta\right) d\mathbf{y}_T\, dv\, d\theta. \tag{10}$$

The optimization problem of (8) is now given in the form

$$\min_f J\left(f\right) \tag{11}$$

for the objective functional J given in (10).

If all target locations and tracks are equally important, then the consequence function $\psi(\mathbf{y}_T)$ is necessarily equal to unity, leading to $J = -P_{\mathrm{SS}}$ (see (4)). In such cases, the optimization problem of (11) is equivalent to

$$\max_f P_{\mathrm{SS}}. \tag{12}$$

We seek the $f(\mathbf{x})$ which maximizes the probability of successful search, leading to a density function from which sensors will then be placed [22]. When our goal is more specific, that is for protection of an HVU, the consequence function $\psi(\mathbf{y}_T)$ is used to represent the relative risk of various target tracks, and the solution of the same optimization problem ((10) and (11)) yields the solution of minimizing the expected risk to the HVU. Thus, the optimization problem of (10) and (11) is generically utilized as the asset layout optimization problem, with the understanding that a variety of specific problems are addressed by varying the form of the consequence function.

To compute the objective functional J as shown in (10), the target motion distribution parameters must be known, as well as the effective sensor performance P_D and R_D, which are generally functions of location in the region. In this work, we adopt a notional model for the spatial dependence of sensor performance in a rectangular region of interest, depicted in Figure 2. As in the previously defined model, sensor performance is measured by a spatially dependent radius of detection $R_D(\mathbf{x})$ which corresponds to a fixed probability of detection P_D (note that in Figure 2 we show $R_D(\mathbf{x})$ normalized to the size of the region L, where the R_D corresponds to a value of $P_D = 0.9$). In practice, sensor performance predictions such as these can be formulated using historical information on the environment and are common in passive acoustic sensor applications, both in air [23] and undersea [24] domains. The sensor performance and number of sensors provide the necessary inputs for computation of P_{ST} (as in (2)) for any specified target track. When combined with the target motion distribution parameters and consequence function, they provide a complete description of the objective J for any distribution $f(\mathbf{x})$ of sensor locations. Figure 3 shows a functional description of the overall approach, where it becomes clear that the "inputs" to the optimization, that is *a priori* knowledge of target, environment, and asset availability, are utilized to find the optimal distribution of assets. The specific placement of the individual sensor assets from the distribution is done using a sampling procedure from the resulting distribution, leading to a placement map for the surveillance region.

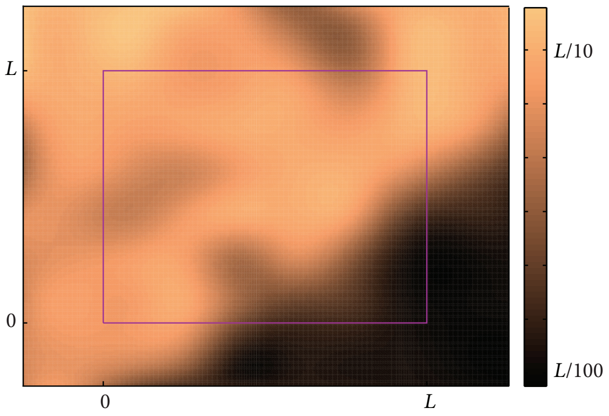


FIGURE 2: Sensor detection range map for the example problems. The region drawn in the center represents the surveillance region for sensor placement.

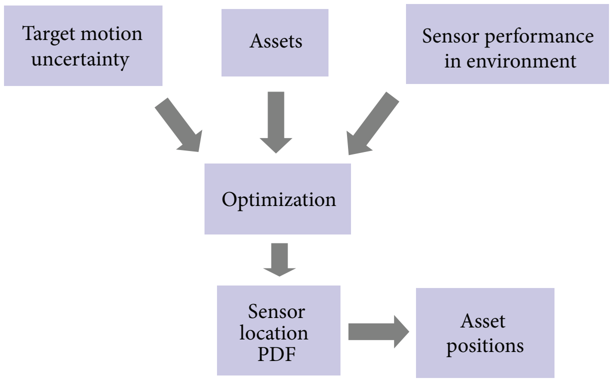


FIGURE 3: Sensor optimization algorithm flow diagram.

5. Numerical Procedure for Optimization

Recall from (5) that the modal positions representing the sensor distribution are fixed and thus the optimal distribution with respect to the surveillance objective (9) is parameterized only through the modal weights. Thus the numerical objective for optimization becomes

$$\min_{\mathbf{w}} J(\mathbf{w})$$
$$\text{subject to} \sum_{j}^{N_G} w_j = 1, \quad 0 \le w_j \le 1, \forall j. \tag{13}$$

We implement a genetic algorithm to perform the optimization defined in (13). The genetic algorithm cannot be run to any guarantee of convergence [25] but is rather run to a prescribed number of generations (iterations). If a theoretically optimal result is desired, the result of the genetic algorithm may be used as the starting point for a nonlinear program (NLP). These stages are complementary in that the genetic algorithm is insensitive to its start and will make significant progress toward a global solution but is devoid of satisfactory stopping criteria (i.e., no guaranteed final convergence). The NLP on the other hand can be quite sensitive to its starting solution but theoretically proven to converge to a local maximum [26]. Thus one goal in the design of this approach is for the potential use of the genetic algorithm to initialize an NLP in the neighborhood of a globally optimal solution, and thus we can attain convergence to a global maximum, if desired.

Genetic algorithms operate on a discrete set of parameters in the form of a binary string. The parameters in this problem are the weights $\{w_j\}_{j=1}^{N_G}$ representing the sensor distribution $f(\mathbf{x})$ in $\mathcal{S}$. In the numerical implementation, each weight parameter w_j is represented by a four-bit binary string, with N_G individual Gaussian modes for the representation in (5). Thus, the string length is $4N_G$.

A genetic algorithm starts with some random values of the parameters of interest represented in the form of a binary string as described above. A set of these strings is produced which is referred to as a *population*. This type of algorithm is an iterative search where iterations are referred to as *generations*. At each generation (iteration), the binary strings which make up the population undergo a series of operations. Thus, starting with a randomly generated population, each string is evaluated by the objective function J returning a value corresponding to each string. Typically, the value of the objective is mapped into a more convenient form (to improve scaling) referred to as fitness [27]. However, in this implementation fitness is set to the evaluated objective J, as this quantity is well scaled.

A standard form of genetic algorithm [27] was implemented with each generation consisting of three genetic operations defined in the evolutionary vernacular as *selection*, *mating*, and *mutation*. These operations utilize the fitness associated with each binary string in the population to pseudorandomly select the best (with respect to the objective J) parameter combinations, randomly combine the selected strings, and apply some random perturbations to the resulting strings, respectively. Specifically, the selection approach utilized, referred to as "roulette," selects binary strings by first scaling the fitness of the population members to sum to unity. Next, the cumulative sum of the fitness is calculated, creating an interval $(0, 1)$, with subintervals proportional to the fitness of each binary string. A random uniform number is then generated and the subinterval in which the number falls determines the string that is selected. Thus, a string (population member) with high fitness, relative to other strings, will be selected with high probability while one with low fitness will be selected with low probability. In this implementation, the string with the highest fitness (at each generation) is kept as a *survivor*; that is, the best string gets passed on to the next generation unchanged. Therefore, $N_{\text{pop}} - 1$ strings are selected to pass to the next generation (where N_{pop} is the fixed number of strings in a population), and these strings make up what is referred to as the *mating pool*. In the next phase, strings in the mating pool are randomly (without regard to fitness) paired up and then randomly combined to create new parameter strings. This operation is called *crossover*. Crossover consists of randomly breaking two strings (at the same point) and then combining the leading part of one with the trailing part of the other. Finally, each of these newly formed strings is passed to the *mutation* operation which flips bits (i.e., change 0 to 1, or vice versa) within each string randomly at some specified (*a priori*) probability. This is essentially a random perturbation of the parameters meant to avoid premature convergence to local minima. Once these operations are complete, the new strings are grouped with the survivor string and these strings become the new population passed on to the next generation (iteration). This process is repeated for some predefined set of generations. From numerical experimentation on a variety of problems, a population size of 100 run over 200 generations was suitable for producing meaningful results for the numerical examples in this paper.

On completion of the genetic algorithm, the optimal sensor density is obtained and the sensors are then placed using a numerical sampling procedure. The procedure consists of a sequential (conditional) sampling where an asset location is selected (among a grid of possible locations) which maximizes the relative entropy between the prior form of the PDF $f(\mathbf{x})$ (discretized and normalized to sum to unity, in order to convert to a probability mass function) and the posterior probability mass function (PMF) calculated by selecting the asset. The relative entropy between two PMFs is written as [28]

$$D(p_1 || p_0) = \sum_{\mathbf{s} \in \mathcal{S}} p_1(\mathbf{s}) \log\left(\frac{p_1(\mathbf{s})}{p_0(\mathbf{s})}\right) \tag{14}$$

and represents a measure of divergence of one PMF relative to the other. The conditional sampling procedure used to place sensors from $f(\mathbf{x})$ treats individual sensor placement as a Bayes recursion where a unique posterior is generated by a positional-dependent likelihood update, defined as corresponding to a possible sensor location. The procedure starts with the definition of two grids (uniformly spaced points in $\mathcal{S}$), written as $\mathbf{z}_i$, $i = 1, \ldots, m$ and $\mathbf{v}_j$, $j = 1, \ldots, n$ where $\mathbf{z}_i$ represents discretely sampled points of $f(\mathbf{x})$, and $\mathbf{v}_j$ represents all possible sensor locations for placement. Next, the prior is calculated from the final form (after optimization) of $f(\mathbf{x})$ as

$$p_0(\mathbf{z}_i) = \frac{I_{\mathbf{z}_i}(\mathbf{x}) f(\mathbf{x})}{\sum_{i=1}^{m} I_{\mathbf{z}_i}(\mathbf{x}) f(\mathbf{x})}, \tag{15}$$

where

$$I_{\mathbf{z}_i}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} = \mathbf{z}_i \\ 0, & \mathbf{x} \neq \mathbf{z}_i \end{cases} \tag{16}$$

is the indicator function. The posterior probability resulting from selecting (placing) a sensor at position $\mathbf{v}_j$ is defined as

$$\pi^i(\mathbf{z}) = \begin{cases} \dfrac{p_0(\mathbf{z})}{\psi_j}, & \mathbf{z} \in \overline{B}_j \\ \dfrac{\alpha \cdot p_0(\mathbf{z})}{\psi_j}, & \mathbf{z} \in B_j, \end{cases} \tag{17}$$

where $\mathcal{S} = B_j \cup \overline{B}_j$, $B_j \cap \overline{B}_j = \emptyset$ (i.e., a disjoint partitioning of $\mathcal{S}$ with respect to sensor position $\mathbf{v}_j$) and $B_j = \{\mathbf{z} : \|\mathbf{z} - \mathbf{v}_j\|_2 \leq R_D(\mathbf{v}_j)\}$ is a ball of spatially dependent radius R_D (with respect to constant P_D) centered at point $\mathbf{v}_j$. The sensor coefficient $\alpha = 1 - P_D$ plays the role of decreasing mass within the radius of the placed sensor, while the sensor-dependent normalizing constant is written as

$$\psi_j = \sum_{\mathbf{z} \in \overline{B}_j} p_0(\mathbf{z}) + \sum_{\mathbf{z} \in B_j} \alpha \cdot p_0(\mathbf{z}). \tag{18}$$

This normalizing constant is required so that each posterior probability is a proper PMF (i.e., sums to unity over its support). The posterior with respect to all possible sensor grid points is calculated as in (15), and a sensor is placed at a specified position, by choosing the posterior which maximizes the relative entropy with respect to the prior. This is formalized as

$$\pi^* = \arg\max_j D\left(\pi^j \| p_0\right), \tag{19}$$

where this process is repeated in a sequential fashion to place all N sensors. Upon the placement of each sensor, the posterior with respect to the chosen location acts as the prior for placing the next sensor.

6. Numerical Examples

The problem of sensor placement as defined above depends on many factors. These factors can be primarily sensor dependencies from environmental variability [29] or can be dominated by other factors such as target behavior [30]. To illustrate these dependencies, we present several numerical examples. Throughout the examples, we consider the number of available sensor assets N to be fixed. The planning problem is to place these sensors optimally in a square planar region $\mathcal{S}$ of size $L \times L$. For these examples, the optimality criteria are to maximize the probability of surveillance mission success, corresponding to minimization of the probability of mission failure P_{MF}.

As an introductory example, and to demonstrate the utility of the optimization approach, we define a nominal environment (constant detection range given by $R_D = L/15$) with target parameters for which intuitive solutions exist. We seek to optimally place $N = 28$ sensors, such that we obtain the maximum P_{SS} (corresponding to minimizing P_{MF}) with a requirement of at least two sensor reports during a time interval τ. The target is assumed to be traveling in a known fixed heading (assumed north) at constant speed v over the fixed time interval τ (where $v\tau = L/2$) with a start position randomly distributed within the search region. For this problem, the expected optimal placement is a "barrier" formation perpendicular to the target course [22]. In particular, due to random starting positions, we should observe a two-line barrier perpendicular to the target course. Figure 4(a) illustrates the optimization results from this problem, where we see that the barrier structure results, as expected. A second nominal example considers a similar problem but with target heading defined as random. In this case, the optimization result, shown in Figure 4(b), produces a sensor layout in a "box-like" structure, which may not be intuitively obvious but has been shown to be optimal [22]. These nominal examples show the dependence that the target behavior has on the optimal sensor layouts.

In a typical approach to deployment of sensors under limited knowledge of the environment, it is reasonable to consider some nominal sensor detection performance. However, given current environmental modeling capabilities, we assume that sensor detection performance can be provided to some acceptable level of fidelity. Figure 2 shows a $1.5L \times 1.5L$ region containing the region of interest, defined by the inner box. The underlying color map depicts sensor coverage as a function of position within the region. The sensor performance is limited by environmental factors that are beyond our control, and the optimization seeks to maximize mission performance (minimizing P_{MF}) in surveillance of the given region with a limited number of sensors. In particular, for the region in Figure 2, the lower right part of the region exhibits a sharp dropoff in individual sensor coverage.

An additional input to the optimization is the characterization of target behavior. The numerical examples that follow were produced assuming that target position and heading are uniformly random within the search region $\mathcal{S}$. That is, all reference track positions (previously defined as $\mathbf{y}_T$) are equally likely. Furthermore, assume that the target of interest travels at a fixed speed v over time intervals of length τ. This defines a track length of $v\tau$, which is given as $v\tau = L/8$ for these numerical examples. The track length is scalable over varying combinations of speed and time (as it is simply the product of the two) and represents *a priori* knowledge of the target of interest, which will result in increased surveillance performance over that of situations where there is very little known (and thus can be assumed) of the anticipated target behavior.

In Figure 5, we illustrate three example consequence (risk) functions $\psi(\mathbf{y}_T)$ of interest. The first function shown in Figure 5(a) is the nominal unity function that is equivalent to the problem of optimizing cumulative probability of detection (see (12) and the surrounding discussion). The second

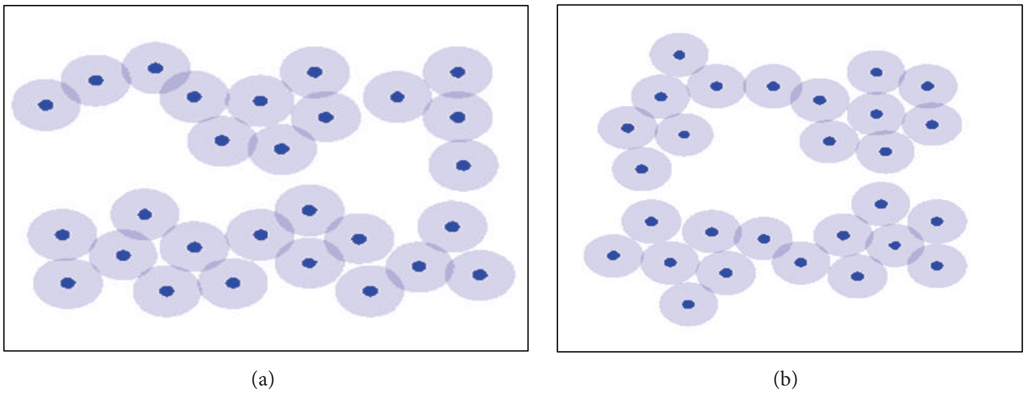


Figure 4: Optimal sensor placement for surveillance of fixed speed target in nominal environment. (a) is for a target with a known course; (b) is for a target with an unknown course.

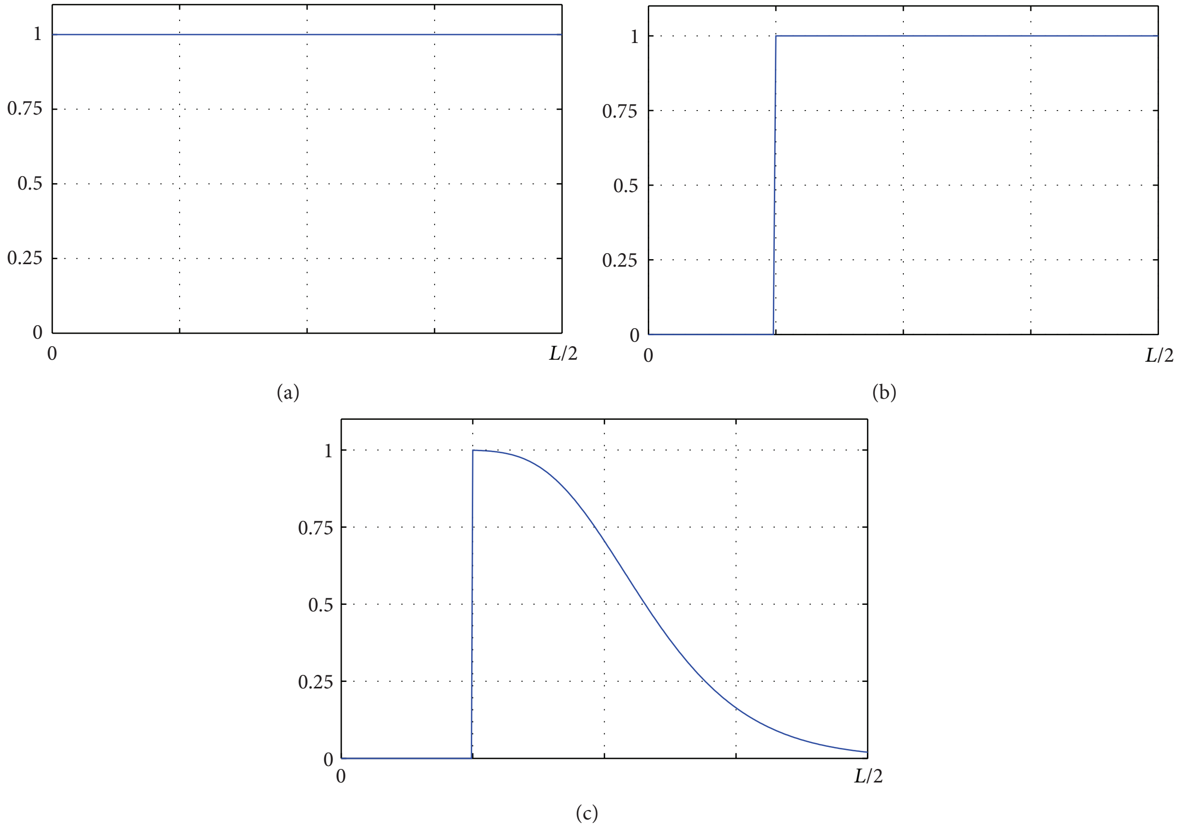


Figure 5: Consequence (risk) function $\psi(\mathbf{y}_T)$ versus range for each of the three example cases.

consequence function shown in Figure 5(b) represents a protection problem for an HVU, whereby any targets within a certain distance to the HVU are too close to provide a surveillance response, and, thus, provide zero surveillance risk, with all others providing nominal risk. This may seem counterintuitive, to have zero risk closest to the HVU, but the point here is to maximize surveillance performance, and this case illustrates the situation in which the surveillance mission is no longer operational when targets are too close to the HVU. Alternatively, if one were to weight the consequence very high near the HVU, an obviously optimal solution is to only try to detect those targets and ignore all targets that are not directly in proximity to the HVU, which is not desired if the risk is already passed. The third consequence function, shown in Figure 5(c), is perhaps the most operationally relevant, in that it incorporates the features of the second case along with degradation in risk for targets further from the HVU. In this case, the risk degradation follows a log-normal function, as described by [31]. Such a consequence is representative of scenarios in which there is a greater

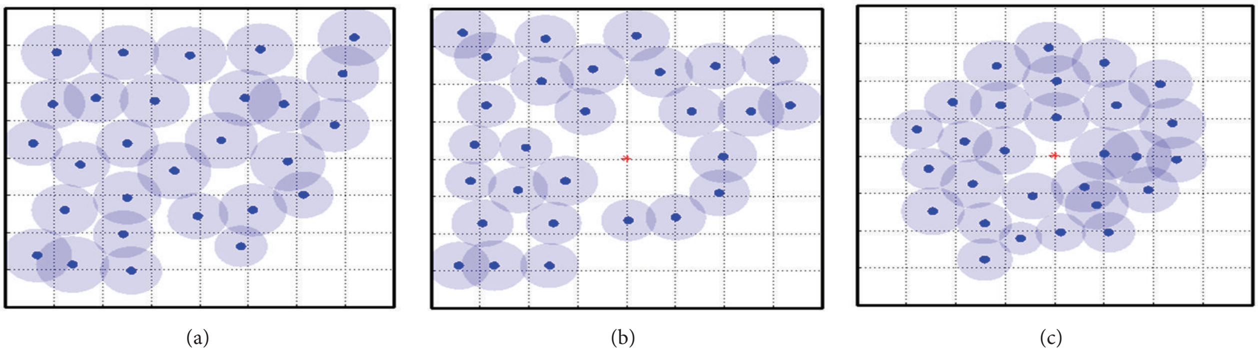

(a) (b) (c)

Figure 6: Optimal sensor distributions for each of the three consequence functions of Figure 5 for scenarios with no cooperation between sensors. Circle size represents the detection range of the sensor (which is a function of position).

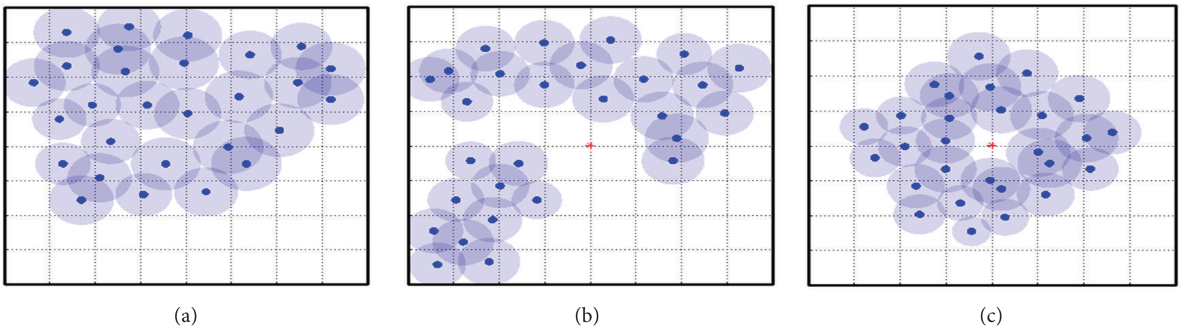

(a) (b) (c)

Figure 7: Optimal sensor distributions for each of the three consequence functions of Figure 5 for scenarios with $k = 2$ cooperation between sensors. Circle size represents the detection range of the sensor (which is a function of position).

importance to detect targets closer to the HVU, up to a point at which they are so close that response becomes impractical. Specifically, the log-normal consequence function takes the form

$$\psi(\mathbf{y}_T) = \begin{cases} 0, & \|\mathbf{y}_T - \mathbf{x}_0\| < r_0 \\ \frac{1}{2}\left[1 - \operatorname{erf}\left[\frac{\ln(\|\mathbf{y}_T - \mathbf{x}_0\| / \alpha)}{\sqrt{2}\beta}\right]\right], & \|\mathbf{y}_T - \mathbf{x}_0\| \geq r_0 \end{cases} \quad (20)$$

for an HVU at location $\mathbf{x} = \mathbf{x}_0$ with a minimal response distance r_0. The parameters α, β in (20) are shape parameters that control the slope and taper of the log-normal consequence function. These three example consequence functions illustrate various applications of the consequence function $\psi(\mathbf{y}_T)$ to show how seemingly different scenarios are solved using the same field optimization approach.

Figure 6 illustrates the results of the optimization process applied to the three consequence functions of Figure 5 for a scenario with $N = 28$ sensors performing noncollaborative surveillance. By noncollaborative surveillance, we consider the sensors to behave completely autonomously ($k = 1$ in (2)), and extended coverage is obtained only through the effective spacing of the individual sensors with respect to the target prior information, that is, since there is no collaboration between sensors, the surveillance relies only upon individual sensors to detect a target if present. In Figure 6, along with the sensor positions, we include opaque circles corresponding to the coverage capability radius R_D of each sensor. The circle size varies according to the local sensing capabilities attributable to the local environmental conditions, as shown in Figure 2. The effect for the first consequence function (for unity risk), as shown in Figure 6(a), is to place the sensors somewhat evenly to best cover the requirement of single sensor coverage in the field. Fewer sensors are located where there is lower detection capability (lower right corner) since the additive coverage is small. For the second consequence function, note that there are no sensors placed near the HVU (see Figure 6(b)), as expected. When compared to the first consequence function, note that the sensors are still spread evenly but now pushed slightly closer to the edges, in order to still cover as much of the area as possible. The third consequence function provides a different type of optimal configuration, as shown in Figure 6(c). In this case, the sensors tend to encircle the HVU in an annulus, as the annular region is the region of highest consequence if detections are missed. This effect appears more significant than the effect of avoiding the low coverage in the lower right corner, and more sensors are added to the lower right section of the annular region to make up for the lower individual sensor coverage. Note that each of these results were created from the same optimization procedure, with the only distinction between the three cases being the specific form of the consequence function $\psi(\mathbf{y}_T)$.

Table 1: Comparison of optimal and nominal values of the objective J for the distributions shown in Figures 6 and 7.

	Objective value for consequence function a		Objective value for consequence function b		Objective value for consequence function c	
	Uniform placement	Optimal placement	Uniform placement	Optimal placement	Uniform placement	Optimal placement
Noncollaborative Sensors	0.46	0.60	0.45	0.55	0.52	0.80
Collaborative Sensors	0.14	0.28	0.14	0.28	0.18	0.60

In practical situations with many sensors, there is performance enhancement opportunity through the use of collaboration [32, 33]. Historically, such problems are solved using optimal processing strategies given a fixed location of sensors [34, 35]. However, the optimization framework developed herein permits the optimization of sensor placements for a given level of collaboration. For instance, the parameter k in (2) may be adjusted to represent the number of sensors that must concurrently detect a target over the time interval of interest τ. Any detections that are spatially or temporally isolated will not count towards the probability P_{ST} used as the performance objective, as they are likely false positives. Recall that the requirement of multiple detections need not occur simultaneously, only over the time interval of interest. Thus, the performance objective cannot be translated into a simple geometrical overlap requirement, that is, a goal in which maximal overlap is sought. In fact, since this objective depends on target track parameters which have spatiotemporal features, there are many scenarios for which non-intuitive patterns of sensors will be optimal. In particular, as complexity (from such factors as environmental sensitivity or higher levels of collaboration) is added, results formed through intuition become less likely to approach optimal, reinforcing the need for an optimization framework which can factor in these complexities.

To illustrate the impact of multiple sensor collaboration on the optimal patterns, the examples of Figure 6 are repeated with a requirement of $k = 2$ detections to occur over the time interval of interest. In this case the goal is to optimally deploy the same sensors in the same variable environment, but we now require two separate sensor detections ($k = 2$) over the previously defined time interval τ. The resulting optimal patterns for the three consequence functions are shown in Figure 7. Comparison with Figure 6 shows that the increased detection requirement coupled with the relatively short target track length results in a more clustered approach to the deployment. For the third consequence function the deployment pattern has only subtle differences compared to Figure 6. This is attributed to the effect of having more than a suitable number of sensors for covering the annular region of primary interest.

In Table 1 we show the numerical values of the performance objectives for each of the scenarios presented in Figures 6 and 7. These objective values are also compared to the equivalent objective values obtained with uniform placement patterns of the assets for each situation. Observe that in each case the optimization approach resulted in better performance in the objective J than for the uniform distribution, as expected. In these examples the $N = 28$ sensors represent a sparse coverage with respect to the search region $\mathcal{S}$, particularly for general surveillance (consequence function a) and for cases requiring multiple detections. This sparsity explains some of the general trends seen in the results. For instance, for both consequence functions a and b there is little or no difference between the results for collaborating and independent sensors. This is because the reduction in the search space due to the presence of the HVU is not significant with respect to the level of sensor coverage sparsity. However, the coverage numbers increase significantly for consequence function c, where the form of the consequence function $\psi(\mathbf{y}_T)$ increases the spatial dependence of the objective with respect to the position of the HVU. Overall the increased coverage due to optimization is much more significant for collaborative sensors than for independent sensors. This is due to the added sensitivity of sensor placement when using collaboration based on spatiotemporal target dependence.

An important byproduct of these numerical results is that for a number of diverse surveillance missions, a common optimization procedure can be utilized for positioning sensors to either meet specific performance criteria, or to get the best performance possible. This can be applied in two ways, the obvious one being as a predeployment tool for positioning sensors for a specific mission, the other being a guide to repositioning sensors to react to a change in mission. In either case these examples show that through proper modeling of the problem, optimal positioning of sensor assets can be achieved, without resorting to costly simulations. In fact, the results attained for these examples were produced with a per case computation time of approximately 20 minutes on a Pentium IV 3 GHz processor with code implemented in MATLAB.

7. Conclusion

We have developed an optimization approach to place distributed sets of sensors to collaboratively perform surveillance against moving targets over extended areas. In particular, a genetic algorithm solution was provided to find the spatial sensor density functions that maximize effectiveness against moving targets. These density function representations provide a computationally efficient method for determining sensor locations for planning and were applied to situations with environmentally induced sensor spatial variability and varying forms of target risk. By illustrating the effective performance of our method on problems of general area surveillance and risk-based surveillance in protection of an asset, we have shown how the general technique applies to seemingly dissimilar problems. The numerical solutions that were obtained were shown to compare favorably against nominal layouts of sensors in the scenarios that were examined. Future work includes the extension of this method to problems with limited network connectivity between the sensor nodes.

Acknowledgments

This work has been supported by the Office of Naval Research code 321MS and by the In-house Laboratory Independent Research program of the Naval Undersea Warfare Center.

References

[1] D. H. Wagner, W. C. Mylander, and T. J. Sanders, *Naval Operations Analysis*, Naval Institute Press, Annapolis, Md, USA, 1999.

[2] S. Olariu and J. V. Nickerson, “Protecting with sensor networks: perimeters and axes,” in *Proceedings of the Military Communications Conference (MILCOM ’05)*, vol. 3, pp. 1780–1786, October 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. 102–114, 2002.

[4] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, “Connecting the physical world with pervasive networks,” *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.

[5] H. Qi, S. S. Iyengar, and K. Chakrabarty, “Distributed sensor networks—a review of recent research,” *Journal of the Franklin Institute*, vol. 338, no. 6, pp. 655–668, 2001.

[6] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, “Sensor deployment strategy for detection of targets traversing a region,” *Mobile Networks and Applications*, vol. 8, no. 4, pp. 453–461, 2003.

[7] S.S. Dhillon and K. Chakrabarty, “Sensor placement for effective coverage and surveillance in distributed sensor networks,” in *Proceedings of the Wireless Communications and Networking Conference*, vol. 3, pp. 1609–1614, March 2003.

[8] P. K. Biswas and S. Phoha, “Self-organizing sensor networks for integrated target surveillance,” *IEEE Transactions on Computers*, vol. 55, no. 8, pp. 1033–1047, 2006.

[9] M. F. Duarte and Y. H. Hu, “Vehicle classification in distributed sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 826–838, 2004.

[10] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization in distributed sensor networks,” *ACM Transactions on Embedded Computer Systems*, vol. 3, no. 1, pp. 61–91, 2004.

[11] E. H. Aboelela and A. H. Khan, “Wireless sensors and neural networks for intruders detection and classification,” in *Proceedings of the International Conference on Information Networking (ICOIN ’12)*, pp. 138–143, 2012.

[12] A. Arora, P. Dutta, S. Bapat et al., “A line in the sand: a wireless sensor network for target detection, classification, and tracking,” *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[13] S. Martínez and F. Bullo, “Optimal sensor placement and motion coordination for target tracking,” *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[14] K. Hadi and C. M. Krishna, “Management of target-tracking sensor networks,” *International Journal of Sensor Networks*, vol. 8, no. 2, pp. 109–121, 2010.

[15] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*, Wiley-Interscience, 2001.

[16] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, 2004.

[17] T. A. Wettergren, “Performance of search via track-before-detect for distributed sensor networks,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 314–325, 2008.

[18] B. O. Koopman, *Search and Screening: General Principles with Historical Applications*, Pergamon Press, 1980.

[19] M. Desai and D. Manjunath, “On the connectivity in finite ad hoc networks,” *IEEE Communications Letters*, vol. 6, no. 10, pp. 437–439, 2002.

[20] A. Ghasemi and S. Nader-Esfahani, “Exact probability of connectivity in one-dimensional ad hoc wireless networks,” *IEEE Communications Letters*, vol. 10, no. 4, pp. 251–253, 2006.

[21] G. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley & Sons, 2000.

[22] T. A. Wettergren and R. Costa, “Optimal placement of distributed sensors against moving targets,” *ACM Transactions on Sensor Networks*, vol. 5, no. 3, article 26, pp. 1–25, 2009.

[23] R. J. Kozick, B. M. Sadler, and D. K. Wilson, “Signal processing and propagation for aeroacoustic sensor networks,” in *Frontiers in Distributed Sensor Networks*, S. S. Iyengar and R.R. Brooks, Eds., CRC Press LLC, 2004.

[24] C. Ferla and M. B. Porter, “Receiver depth selection for passive sonar systems,” *IEEE Journal of Oceanic Engineering*, vol. 16, no. 3, pp. 267–278, 1991.

[25] G. Rudolph, “Convergence analysis of canonical genetic algorithms,” *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.

[26] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, 2nd edition, 1987.

[27] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Longman, 1989.

[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 1991.

[29] R. Stolkin, L. Vickers, and J. V. Nickerson, “Using environmental models to optimize sensor placement,” *IEEE Sensors Journal*, vol. 7, no. 3, pp. 319–320, 2007.

[30] S. A. Musman, P. E. Lehner, and C. Elsaesser, “Sensor planning for elusive targets,” *Mathematical and Computer Modelling*, vol. 25, no. 3, pp. 103–115, 1997.

[31] J. S. Przemieniecki, *Mathematical Methods in Defense Analyses*, American Institute of Aeronautics and Astronautics, Reston, Va, USA, 3rd edition, 2000.

[32] J. F. Chamberland and V. V. Veeravalli, “Decentralized detection in sensor networks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 407–416, 2003.

[33] S. Ferrari, “Track coverage in sensor networks,” in *Proceedings of the American Control Conference*, pp. 2053–2059, Minneapolis, Minn, USA, June 2006.

[34] R. Niu, P. K. Varshney, and Q. Cheng, “Distributed detection in a large wireless sensor network,” *Information Fusion*, vol. 7, no. 4, pp. 380–394, 2006.

[35] D.E. Penny, “Multi-sensor management for passive target tracking in an anti-submarine warfare scenario,” in *IEE Colloquium on Target Tracking: Algorithms and Applications*, vol. 3, pp. 1–5, 1999.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 635637, 10 pages
http://dx.doi.org/10.1155/2013/635637

Research Article

Optimal Management of Rechargeable Biosensors in Temperature-Sensitive Environments

Yahya Osais,[1] F. Richard Yu,[2] and Marc St-Hilaire[2]

[1] *Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*
[2] *Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada K1S 5B6*

Correspondence should be addressed to Yahya Osais; yosais@kfupm.edu.sa

Received 26 September 2012; Accepted 13 March 2013

Academic Editor: Nadjib Achir

Biological sensors (biosensors, for short) are tiny wireless devices attached or implanted into the body of a human or animal to monitor and detect abnormalities and then relay data to physician or provide therapy on the spot. They are distinguished from conventional sensors by their biologically derived sensing elements and by being temperature constrained. Biosensors generate heat when they transmit their measurements and when they are recharged by electromagnetic energy. The generated heat translates to a temperature increase in the tissues surrounding the biosensors. If the temperature increase exceeds a certain threshold, the tissues might be damaged. In this paper, we discuss the problem of finding an optimal policy for operating a rechargeable biosensor inside a temperature-sensitive environment characterized by a strict maximum temperature increase constraint. This problem can be formulated as a Markov Decision Process (MDP) and solved to obtain the optimal policy which maximizes the average number of samples that can be generated by the biosensor while observing the constraint on the maximum safe temperature level. In order to handle large-size MDP models, it is shown how operating policies can be obtained using Q-learning and heuristics. Numerical and simulation results demonstrating the performance of the different policies are presented.

1. Introduction

Biosensors are tiny wireless devices attached or implanted into the body of a human or animal to monitor and detect abnormalities and then relay data to physician or provide therapy on the spot. Unlike conventional wireless sensors, biosensors are energy as well as temperature constrained. Also, their sensing elements are biological materials such as enzymes and antibodies which are integrated into transducers for producing electrical signals in response to biological reactions and changes. Biosensors are powered by either rechargeable built-in batteries or by continuously sending electric energy to them in the form of electromagnetic waves.

The use of batteries necessitates periodic recharging which can be performed using energy resulting from vibration, motion, light, and heat. However, a more mature approach is to wirelessly collect energy from a Radio Frequency (RF) source and then convert it into usable power. This approach is widely used in the industry to transfer data and power to biosensors. It is also more practical since many biosensors can be recharged simultaneously. In essence, a charging station generates a magnetic field that can convey energy through the skin. From the penetrating magnetic field, an electric voltage is produced by induction in the receiver circuit. The induced voltage is then rectified, filtered, and stabilized to run the biosensors or recharge their batteries.

In this paper, we study a stochastic control problem which arises when a rechargeable biosensor operates in a temperature-sensitive environment like the human body. In this problem, the state of the biosensor is characterized by its current temperature and energy levels, and uncertainty exists due to the random behavior of the wireless channel between the biosensor and base station. The objective is to operate the biosensor in such a way that the average number of samples generated by the biosensor is maximized while the maximum safe temperature level is not exceeded. This control problem can be formulated as an MDP and solved to obtain an optimal operating policy.

Since the size of the MDP model increases with the number of biosensors and their states, Q-learning which is a form of reinforcement learning is used to obtain the optimal policy. The optimal policy is learned by interacting with a simulation model of the system. Another way to handle large MDP models is through the use of heuristic policies. This paper proposes a simple heuristic policy whose performance is sufficiently close to that of the optimal policy. A greedy policy is also proposed and used as a baseline for comparing the performance of the different policies.

The remainder of the paper is organized as follows. In the next section, the necessary background information is given. Then, the limited available literature is reviewed. After that, the model of the system under study is described followed by the presentation of its MDP formulation. Next, it is shown how large-size MDP models can be handled using Q-learning. Besides, greedy and heuristic policies are described. Then, numerical and simulation results are presented and an example is given. Finally, conclusions are summarized and directions for further research are suggested.

2. Calculating the Temperature Increase

Radiation due to wireless communication and recharging are the major sources of heat in biosensor networks. The level of radiation absorbed by the human body when exposed to RF radiation is measured by the Specific Absorption Rate (SAR) which is expressed in units of W/Kg. SAR records the rate at which radiation energy is absorbed per unit mass of tissue [1]. The mathematical relationship between SAR and radiation is given by

$$\mathrm{SAR} = \frac{\sigma |E|^2}{\rho}, \tag{1}$$

where E is the induced electric field due to radiation, and ρ and σ are the density and electrical conductivity of the tissue, respectively. As an example to appreciate the importance of this measure, it was reported in [2] that an exposure to a SAR of 8 W/Kg in any gram of tissue in the head for 15 minutes may result in tissue damage.

SAR is a point quantity. That is, its value varies from one location to another. In this paper, we consider only the SAR in the near field (i.e., the space around the antenna of the biosensor). The extent of the near field is given by $d_0 = \lambda/2\pi$, where λ is the wavelength of the carrier signal used in wireless communication. SAR in the near field is given by the following equation [3]:

$$\begin{aligned}\mathrm{SAR}_{\mathrm{NF}} = {} & \frac{\sigma\mu\omega}{\rho\sqrt{\sigma^2+\epsilon^2\omega^2}} \\ & \cdot \left(\frac{Idl \sin\theta e^{-\alpha R}}{4\pi}\left(\frac{1}{R^2}+\frac{|\gamma|}{R}\right)\right)^2,\end{aligned} \tag{2}$$

where μ and ϵ are the permeability and permittivity of the tissue, respectively. dl is the length of the wire representing the antenna, I is the current provided to the antenna, α is the attenuation constant, R is the distance from the biosensor to the observation point, θ is the angle between the observation point and the x-y plane, γ is the propagation constant, and ω is the angular frequency. We assume that the radiation patterns are omnidirectional on the 2D plane and thus $\sin\theta = 1$.

The Pennes bioheat equation [4] is the standard for calculating the temperature increase in the body due to heating. The general form of the equation is

$$\rho C_p \frac{dT}{dt} = K\nabla^2 T - b\left(T - T_b\right) + \rho \mathrm{SAR} + P_c + Q_m, \tag{3}$$

where ρ is the mass density, C_p is the specific heat of the tissue, dT/dt is the rate of temperature increase, K is the thermal conductivity of the tissue, b is the blood perfusion constant which indicates how fast the heat can be taken away by the blood flow inside the tissue, and T_b is the temperature of the blood and the tissue. Terms on the right side indicate the heat accumulated inside the tissue. The terms $K\nabla^2 T$ and $b(T - T_b)$ are the heat transfer due to the thermal conduction and the blood perfusion, respectively. The terms ρSAR, P_c, and Q_m are the heat generated due to radiation, the power dissipation of circuitry, and the metabolic heating, respectively.

The Finite-Difference Time-Domain (FDTD) method [5] is a technique that transforms the previous bioheat equation to a discrete form with discrete time and space steps. The area under consideration is divided into cells of side δ, and the temperature is evaluated in a grid of points defined at the centers of the cells. Temperatures are computed at equally spaced time instants with a time step equal to δ_t. Therefore, from [6], the new bioheat equation is

$$\begin{aligned}T_{m+1}(i,j) = {} & \left[1 - \frac{\delta_t b}{\rho C_p} - \frac{4\delta_t K}{\rho C_p \delta^2}\right] T_m(i,j) \\ & + \frac{\delta_t}{C_p}\mathrm{SAR}_{\mathrm{NF}} + \frac{\delta_t b}{\rho C_p} T_b + \frac{\delta}{\rho C_p} P_c + \frac{\delta_t K}{\rho C_p \delta^2} \\ & \cdot \begin{bmatrix} T_m(i+1,j) + T_m(i,j+1) \\ +T_m(i-1,j) + T_m(i,j-1) \end{bmatrix},\end{aligned} \tag{4}$$

where $T_{m+1}(i,j)$ is the temperature of cell (i,j) at time $m+1$, δ_t is the time step, and δ is the space step.

Using (2) and (4), the temperature increase at the location of the biosensor (i,j) can be found. It is assumed that the temperature of the surrounding cell points is the normal body temperature (i.e., 37°C).

3. Related Work

The research on the possible biological effects caused by biosensors and how to mitigate those effects is very recent. Most of the existing research deals with other technical issues such as energy efficiency and quality of service. In this section, the limited available literature is briefly reviewed.

Tang et al. [6] were the first to propose rotating the cluster leadership in a cluster-based biosensor network to minimize the heating effects on human tissues. They proposed a genetic algorithm for computing a minimal temperature increase rotation sequence. Since using (4) in computing the temperature increase due to a sequence is computationally expensive, they proposed a scheme for estimating the possible temperature increase due to a sequence.

In another work, Tang et al. [7] addressed the issue of routing in implanted biosensor networks. They proposed a thermal-aware routing protocol that routes the data away from high-temperature areas referred to as hot spots. The location of a biosensor becomes a hot spot if the temperature of the biosensor exceeds a predefined threshold. The proposed protocol achieves a better balance of temperature increase and shows the capability of load balance.

The above two works have motivated us to explore further the bioeffects of implanted biosensor networks. As a result, we noticed a lack of information on how to optimally operate an implanted biosensor network when bounds such as the maximum temperature increase exist. Most of the existing works assume that energy is the only limiting factor in the operation of Wireless Sensor Networks (WSNs). However, this is not the case in biosensor networks where the increase in temperature is a serious limiting factor.

We have approached the problem of how to optimally operate a biosensor network from the perspective of sensor scheduling and activation in conventional wireless sensor networks. Sensor scheduling is concerned with the problem of how to dynamically choose a sensor for communication with the base station. On the other hand, sensor activation is concerned with the problem of when a sensor should be activated. Many interesting works have been done in this regard. Next, these works are briefly reviewed.

In [8], the sensor scheduling problem is formulated as an MDP. The objective is to find an operating policy that maximizes the network lifetime. The state of a sensor is characterized by its current energy level only. Three kinds of channel state information are considered: global, channel statistics, and local. Considering only the energy level at each sensor gives rise to an acyclic (i.e., loop-free) transition graph which enables the MDP model to converge in one iteration. On the other hand, if the temperature of each sensor is included in the model, the transition graph of the underlying MDP becomes cyclic. This is because when the sensor cools down (i.e., its temperature decreases), it transitions back to a less hot state. An MDP model whose transition graph is cyclic needs more time to converge.

Dynamic sensor activation in networks of rechargeable sensors is considered in [9]. The objective is to find an activation policy that maximizes the event detection probability under the constraint of slow rate of recharge of the sensor. The state of the system is characterized by the energy level of the sensor and whether or not an event would occur in the next time slot. The recharge event is random and recharges the sensor with a constant charge. The model does not include the state of the wireless channel which is very crucial when temperature is considered.

Body sensor networks [10] with energy harvesting capabilities are another kind of WSNs in which each sensor has an energy harvesting device that collects energy from ambient sources such as vibration, light, and heat. In this way, the more costly recharging method which uses radiation is avoided. The interaction between the battery recharge process and transmission with different energy levels is studied in [11]. The proposed policies utilize the sensor's knowledge of its current energy level and the state of the processes governing the generation of data and battery recharge to select the appropriate transmission mode for a given state of the network.

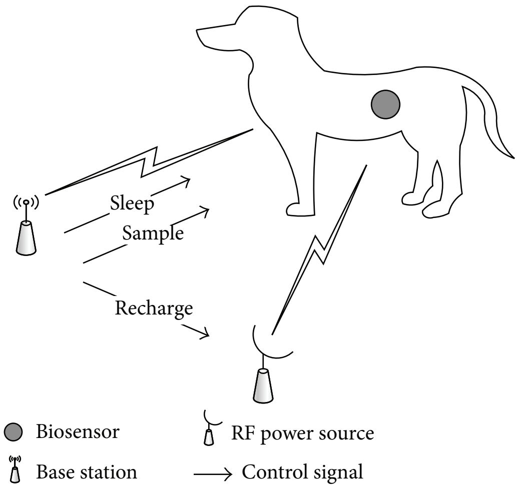


Figure 1: Setup of the system under study.

4. System Model

Figure 1 shows the system under study where a mobile subject, in this case an animal, has a biosensor implanted into its body. The biosensor has a built-in battery which is recharged by an RF power source. The role of the biosensor is to monitor and report interesting physiological events such as heart rate and blood pressure. The biosensor becomes incapable of detecting and reporting events if it does not have enough energy for transmission under any channel condition or the increase in its temperature exceeds a prespecified threshold. The latter condition causes a halt in system operation to allow the system to cool down.

Both the biosensor and RF power source are under the control of the base station which initiates the measurement process. The base station generates three control signals: *Sleep* and *Sample* targeted at the biosensor and *Recharge* targeted at the RF power source. The system state information is assumed to be available to the base station before it generates a control signal.

Mathematically, the system can be modeled as a discrete-state system which evolves in discrete time. Therefore, the time axis is divided into slots of equal durations Δt. At the beginning of each time slot, the state of the system is observed and a control signal is generated by the base station accordingly. Each time slot is long enough to transmit a complete packet carrying a measurement. Next, the elements of the system are described.

4.1. Biosensor. A biosensor typically contains four essential components: biorecognition, transducer, radio and battery.

The biorecognition system is made of elements such as enzymes and antibodies whose role is to produce a physiochemical change which is detected and measured by the signal transducer. The transducer carries out signal processing tasks. The radio circuitry is responsible for wireless communication. The battery provides power for all active modules in the biosensor and is recharged using RF energy. During a recharging period, the biosensor uses its radio module to collect energy and recharge the battery. Therefore, while its battery is being recharged, the biosensor cannot perform sensing and communication.

The location of a biosensor represents a critical point since it experiences the maximum temperature increase. This is because the tissues surrounding the biosensor might be heated continuously due to the local radiation generated by the biosensor itself and the radiation generated by the base station while recharging the biosensor.

In each time slot t, the state of the biosensor is characterized by two variables which are the current temperature T_t and energy level E_t. There are $\tau + 1$ safe temperature levels; that is, $T_t \in \{0, 1, \ldots, \tau\}$, where the zero temperature level represents the normal body temperature and τ is an upper limit which must not be exceeded. Initially, the biosensor has a total energy of $\mathscr{E}_0$ which is also the capacity of its battery. The energy required for the biosensor to successfully transmit its measurement to the base station is determined by the state of the wireless channel at the time of transmission. This transmission energy is denoted by $\mathscr{E}_{w_i}$, where w_i is the ith state of the wireless channel. The temperature increase due to a transmission energy of $\mathscr{E}_{w_i}$ units is denoted by $\mathscr{T}_{w_i}$.

At the beginning of each time slot, the base station may decide to recharge the biosensor; let it transmit its measurement or put it into sleep. The time required for a full recharge is random since it depends on the current temperature and energy levels. During this time, interesting events may occur but they will not be reported by the biosensor since it is being recharged. Also, the biosensor may be put into sleep for a random amount of time during which no measurements can be produced.

At the beginning of the next time slot (i.e., $t + 1$), the energy level at the biosensor is given by the following equation:

$$E_{t+1} = \begin{cases} E_t & \text{if } a_t = \mathit{Sleep} \\ E_t - \mathscr{E}_{w_i} & \text{if } a_t = \mathit{Sample} \\ E_t + \mathscr{E}_r & \text{if } a_t = \mathit{Recharge}, \end{cases} \tag{5}$$

where a_t is the action taken by the base station at time t and $\mathscr{E}_r$ is the amount of energy gained by the biosensor. Similarly, the temperature of the biosensor at $t+1$ is given by the following equation:

$$T_{t+1} = \begin{cases} \max\{T_t - \mathscr{T}_s, 0\} & \text{if } a_t = \mathit{Sleep} \\ T_t + \mathscr{T}_{w_i} & \text{if } a_t = \mathit{Sample} \\ T_t + \mathscr{T}_r & \text{if } a_t = \mathit{Recharge}, \end{cases} \tag{6}$$

where $\mathscr{T}_s$ is the amount by which the temperature of the biosensor decreases when it is put to sleep. In the same way, $\mathscr{T}_r$ and $\mathscr{T}_w$ are the amounts by which the temperature of the biosensor increases when it is recharged and when it is allowed to transmit its measurement, respectively. $\mathscr{T}_r$ and $\mathscr{T}_w$ can be calculated using (4). $\mathscr{T}_r$ is constant since the SAR due to the base station is assumed to be constant. On the other hand, $\mathscr{T}_w$ is not constant since the SAR due to the biosensor changes with the change in transmission energy. Therefore, before $\mathscr{T}_w$ can be calculated, the SAR due to the radiation from the antenna of the biosensor is calculated using (2). Since (2) is a function of I, it is assumed that the current (I) corresponding to each transmission power level is known.

4.2. Wireless Channel. The communication between the biosensor and base station occurs over a Rayleigh fading channel with additive Gaussian noise. Hence, the instantaneous received Signal-to-Noise Ratio (SNR) denoted by γ is exponentially distributed with the following probability density function [12]:

$$P(\gamma) = \frac{1}{\gamma_0} \exp\left(-\frac{\gamma}{\gamma_0}\right), \tag{7}$$

where γ_0 is the average received SNR.

Such a wireless channel can be modeled as a Finite-State Markov Chain (FSMC) [13, 14]. The model can be built as follows. For a wireless channel with K states, the state boundaries (i.e, SNR thresholds) are denoted by $\Gamma_1, \Gamma_2, \ldots, \Gamma_K, \Gamma_{K+1}$, where $\Gamma_1 = 0$ and $\Gamma_{K+1} = \infty$. The channel is said to be in state s_i if the SNR is between Γ_i and Γ_{i+1}, where $i = 1, 2, \ldots, K$. It is assumed that the SNR remains the same during packet transmission, and only transitions to the current or adjacent states are allowed.

The steady-state probability of the ith state of the FSMC is given by

$$P(s_i) = \exp\left(-\frac{\Gamma_i}{\gamma_0}\right) - \exp\left(-\frac{\Gamma_{i+1}}{\gamma_0}\right) \tag{8}$$

and thus the state transition probabilities are

$$\begin{aligned} P(s_{i+1} \mid s_i) &\approx \frac{N(\Gamma_{i+1})\,\Delta t}{P(s_i)}, \\ P(s_{i-1} \mid s_i) &\approx \frac{N(\Gamma_i)\,\Delta t}{P(s_i)}, \end{aligned} \tag{9}$$

where $N(\Gamma_i)$ is the average number of times per unit interval that the SNR crosses level Γ_i and Δt is the packet duration. $N(\Gamma_i)$ can be computed using the following equation [15]:

$$N(\Gamma_i) = \sqrt{2\pi\Gamma_i}\, f_d e^{-\Gamma_i}, \tag{10}$$

where f_d is the maximum Doppler frequency defined as $f_d = v/\lambda$ with v being the speed of the subject and λ being the wavelength.

The above channel model has been verified to be precise when the fading process is slow [13], such as in biosensor applications.

5. MDP Formulation

An MDP is a model of a dynamic system whose behavior varies with time. The elements of an MDP model are the following [16]:

(1) system states,

(2) possible actions at each system state,

(3) a reward or cost associated with each possible state-action pair, and

(4) next state transition probabilities for each possible state-action pair.

The solution of an MDP model (referred to as a policy) gives a rule for choosing an action at each possible system state. If the policy chooses an action at time t depending only on the state of the system at time t, it is referred to as a stationary policy. An optimal stationary policy exists over the class of all policies if every stationary policy gives rise to an irreducible Markov chain. This means that one can limit the attention to the class of stationary policies.

In order to obtain a policy from an MDP model, it is necessary to form and solve the so-called optimality equation (or Bellman's equation). The following is the standard form of this equation with the maximization operator [17]:

$$V_n(s) = \max_{a \in A(s)} \left[f(s,a) + \sum_{s' \in S} \mathbb{P}(s,s',a) V_{n-1}(s') \right], \tag{11}$$

where n is the iteration index, S is the set of system states ($s \in S$), $A(s)$ is the set of actions possible when the system is at state s, $f(s,a)$ is the reward/cost per step, $\mathbb{P}$ is the system state transition probability matrix, and $V(s)$ is the optimal value of the objective function when the system is started at state s and the optimal policy is followed. Equation (11) can be solved using the classical policy iteration, value iteration, and relative value iteration algorithms [17]. Next, the details of the MDP model are given.

5.1. State Set. The state of the system at time t is described by the following 3-dimensional vector:

$$s_t = (T_t, E_t, W_t), \tag{12}$$

where T_t, E_t, and W_t are the current temperature of the biosensor, its energy level, and transmission power required for successful transmission at time t, respectively. The total number of possible states is $|S| = |T| \times |E| \times |W|$, where $|T|$, $|E|$, and $|W|$ are the numbers of possible temperatures, residual energies, and transmission energy levels, respectively.

5.2. Action Set. In each time slot, the base station chooses an action based on the current state of the system. In each state s, there are three possible actions:

$$A(s) = \{Sample, Recharge, Sleep\}, \tag{13}$$

where the *Sample* action lets the biosensor generate a measurement and transmit it to the base station, *Recharge* action recharges the biosensor, and *Sleep* action puts the biosensor into sleep.

The *Sleep* action can be performed at every system state. The other two actions, however, can only be performed at system states, where the next temperature of the biosensor is within the safe temperature range. In addition, the *Sample* action can only be performed at system states, where the remaining energy is sufficient to make a successful transmission.

5.3. Reward Function. Since the objective is to maximize the expected number of samples that can be generated by the biosensor, the reward function is defined as

$$R(s, Sample) = 1. \tag{14}$$

This means that one unit of reward is earned every time the *Sample* action is performed. The long-run expected sum of rewards represents the average number of samples that can be generated by the biosensor with an initial energy of $\mathscr{E}_0$ units and maximum temperature increase of τ units.

5.4. Transition Probability Function. After the action taken by the base station is performed, the system transits to a new state according to the transition probabilities of the present state of the wireless channel. Thus, the behavior of the system is described by $|A|$ transition probability matrices, and each such matrix is of size $|S| \times |S|$. Each matrix is denoted by $P_{s_t,s_{t+1}}(a)$ which is the probability that choosing an action a when in state s_t will lead to state s_{t+1}. More formally, $P_{s_t,s_{t+1}}(a)$ can be written as follows:

$$P[s_{t+1} \mid s_t, a_t] = P[W_{t+1} \mid W_t]. \tag{15}$$

5.5. Value Function. The problem of finding an optimal policy for maximizing the average number of samples is formulated as an infinite-horizon MDP using the average reward criterion [16]. So, let $V_\pi(s_0)$ be the expected number of samples given that the policy π is used with an initial state s_0. Then, the maximum expected number of samples $V^*(s_0)$ starting from state s_0 is given by

$$V^*(s_0) = \max_\pi V_\pi(s_0). \tag{16}$$

The optimal policy π^* is the one that achieves the maximum expected number of samples at all system states.

The famous value iteration algorithm [17] is used to numerically solve the following recursive equation for $n > 0$:

$$V_n(s) = \max_{a \in A(s)} \left[R(s,a) + \sum_{s_{t+1} \in S} \mathbb{P}(s_t, s_{t+1}, a) V_{n-1}(s_{t+1}) \right]. \tag{17}$$

In (17), the subscript n denotes the iteration index. As $n \to \infty$, $V_n \to V^*$.

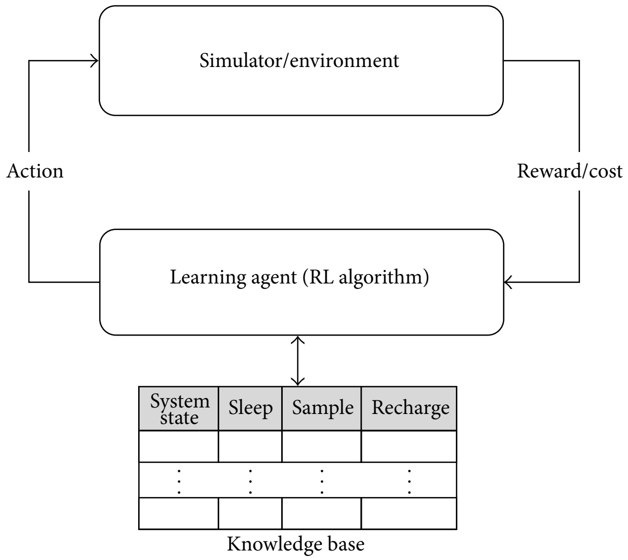


FIGURE 2: Using an RL algorithm (like Q-learning), the decision-making agent gradually learns the optimal policy. An action is applied to the system or simulated and then the resulting reward/cost is fed back into the knowledge base of the decision maker. The new knowledge obtained over time helps the decision maker to make better actions.

6. Handling Large-Size MDP Models

The size of the proposed MDP model depends on the number of biosensor states which is a function of the number of possible temperature and energy levels. As the number of biosensor states increases, the process of computing the transition probability matrices for the system becomes very time consuming. Also, the value iteration algorithm used for solving the MDP model becomes impractical. This section presents two methods (namely, Q-learning and heuristics) for handling MDP models with a large number of states.

6.1. Q-Learning. Reinforcement Learning (RL) offers an alternative for obtaining the optimal policy at a significantly lower computational cost. Using a simulation model of the system under study, the decision maker in an MDP is viewed as a learning agent whose task is to learn the optimal action in each possible state of the system. As Figure 2 shows, the optimal policy is learned while the system is being driven (i.e., simulated) by the actions selected by the learning agent which stores the results of its actions in a knowledge base. The actions of the decision maker become better over time as new knowledge is obtained. Eventually, the RL algorithm converges to an optimal policy which can be used in the physical system.

Q-Learning is an RL algorithm which was introduced in [18]. It is used for learning from experience. It requires that each entry in the decision-maker's knowledge base corresponds to a state-action pair. The value stored in each entry is referred to as the Q-value and is a measure of the goodness of executing an action in a particular system state. The Q-value for a state-action pair (s, a) is updated as follows:

$$Q(s,a) = Q(s,a) + \alpha * \left(r + \gamma * \max_{j \in A(s')} Q(s', j) - Q(s,a) \right), \tag{18}$$

where r is the immediate reward obtained after executing action a in state s, s' is the next state, and $A(s')$ is the set of possible actions in state s'. α and γ denote the learning rate and discount factor ($0 < \gamma < 1$), respectively.

```
for $i = 1$ to $NumEpisodes$ do
    $ps \leftarrow InitialState$; {E.g., $[0\ \mathscr{E}_0\ 1]$}
    for $j = 1$ to $NumIter$ do
        if $rand \le 1 - \epsilon$ then
            $a \leftarrow \max_{j \in A(ps)} Q(ps, j)$; {rand $\in$ (0,1)}
        else
            $a \leftarrow Random(A(ps))$; {Random Action}
        end if
        if $a == Sample$ then
            $r \leftarrow 1$; {Reward}
        else
            $r \leftarrow 0$;
        end if
        $ns \leftarrow SimulateAction(ps, a)$;
        $Update(Q(ps, a))$; {Using (18)}
        $ps = ns$
    end for
end for
```

ALGORITHM 1: The Q-Learning algorithm.

The Q-learning algorithm is shown as Algorithm 1. The interaction between the learning agent and the simulator (or environment) is divided into episodes. In each episode, the system transits through a sequence of states. The length of this sequence is controlled by the parameter *NumIter* which is the number of simulated time slots. In each simulated time slot, based on the current state of the system, the learner chooses an action either based on the ϵ-policy or randomly. If the former is selected, the action with the highest Q-value is selected. After that, if the action is *Sample*, a reward of one unit is earned; otherwise, the reward is zero. The action is then simulated and the next system state is observed. Next, the Q-value is updated using (18). Also, the new system state becomes the current one and the cycle repeats.

Although Q-learning is theoretically guaranteed to obtain an optimal policy, it requires that each state-action pair be tried infinitely often in order to learn the optimal policy. The quality of the learned policy depends on how much time is spent in learning and if every state-action pair can be tried. On the other hand, depending on the application, a certain percental difference between the learned and optimal policies might be tolerated. This is because the system states differ in the likelihood of being visited. Thus, a default action (like *Sleep*) can be assigned to system states with a low likelihood of being visited.

```
               S: Set of possible systemstates
Require:       A: Set of possible actions at
               each system state
  for $i$ = 1 to $|S|$ do
    if Action ($i$, Sample) is True then
        Policy(i) = Sample
    else if Action ($i$, Recharge) is True then
        Policy(i) = Recharge
    else
        Policy(i) = Sleep
    end if
  end for
```

Algorithm 2: Greedy policy.

```
               S: Set of possible system states
Require:       A: Set of possible actions at
               each system state
  $\alpha = T/\tau$
  $\beta = E/E_0$
  for $i$ = 1 to $|S|$ do
    if Action ($i$, Sample) is True and $\alpha \leq \beta$ then
        Policy ($i$) = Sample
    else if Action ($i$, Recharge) is True and $\alpha \geq \beta$ then
        Policy ($i$) = Recharge
    else
        Policy ($i$) = Sleep
    end if
  end for
```

Algorithm 3: Heuristic policy.

6.2. Heuristic. Since it is difficult to describe the structure of the optimal policy, a heuristic policy is proposed in this section. The goal is to design a policy which mimics the behavior of the optimal policy as close as possible. However, before presenting such a policy, a greedy one is given to provide insight into the design of any heuristic policy.

The greedy policy is computed using Algorithm 2. The inputs to this algorithm are the set of possible system states and the set of feasible actions for each system state. The computed policy is greedy in the sense that for each system state, the feasibility of actions is checked in the following order: *Sample*, *Recharge,* and then *Sleep*. The first feasible action is associated with the corresponding system state.

As will be shown by simulations in the next section, the greedy policy is poor since it is based on a fixed order of actions. Therefore, Algorithm 2 needs to be extended to allow for a dynamic selection of actions. This objective is accomplished by introducing two control parameters: α and β. With these two control parameters, the *Sample* and *Recharge* actions are not selected in a specific order or whenever they are feasible. Algorithm 3 shows how the control parameters and new heuristic policy are computed.

The essence of Algorithm 3 is as follows. If the current temperature (denoted by T) of the biosensor is low and its current energy level (denoted by E) is high, then the condition $\alpha \leq \beta$ would more likely be true and thus the *Sample* action could be executed. However, this would not be the case when the available energy is very close to zero. In this case, the opposite condition (i.e., $\alpha \geq \beta$) would more likely be true and thus a *Recharge* could be performed. If neither of the two conditions is true, the biosensor is put to sleep and thus its temperature decreases.

Table 1: Values assigned to the parameters of the example and Q-learning algorithm.

	Parameter	Value
Example	Δt	0.25 ms
	f_d	60 Hz
	$\mathscr{E}_0$	5
	T	5
	$\mathscr{E}_{w_1}, \mathscr{E}_{w_2}$	1, 2
	$\mathscr{T}_{w_1}, \mathscr{T}_{w_2}$	1, 2
	$\mathscr{T}_s$	1
	$\mathscr{T}_r$	1
	$\mathscr{E}_r$	1
Q-Learning	*NumEpisodes*	100
	NumIter	10000
	ϵ	0.5
	α	0.2
	γ	0.1

7. Numerical and Simulation Results

In this section, an example is first presented to illustrate the viability of the proposed MDP model. Then, the performance of the optimal policy is compared to that of the approximate policies using simulation. The impact of various system parameters on the performance of the system is also evaluated. The simulation was performed using a simulator written in Matlab [19]. Each simulation was run for a duration of 100000 time slots, and each data point is the average of 10 simulation runs. The number of channel states (W) is four, and the channel state boundaries are randomly generated.

7.1. Illustrative Example. In this example, a wireless channel with two states is considered. The channel state transition probabilities are calculated using (9). Table 1 shows the values of the parameters involved. Figures 3(a) and 3(b) show the expected number of samples when there is no recharge and when recharge is allowed, respectively. The expected number of samples is expressed as a function of the maximum safe temperature level (τ) and initial energy ($\mathscr{E}_0$). The first observation is that if recharge is allowed, more samples are expected to be generated by the biosensor. In the case when recharge is not allowed, the expected number of samples is limited only by the amount of initial energy. This is confirmed by Figure 3(a) where for the same initial energy, the same expected number of samples is obtained when τ is varied.

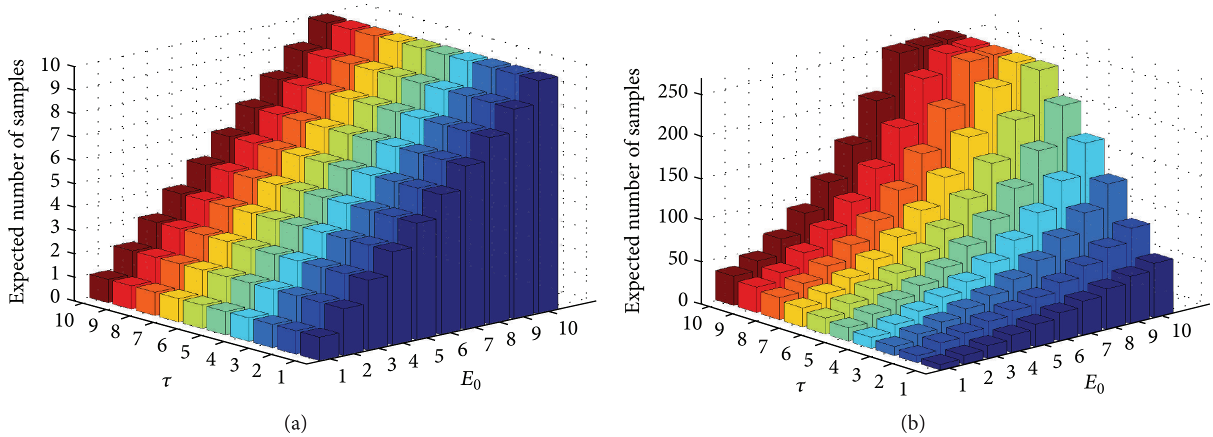


FIGURE 3: Expected number of samples. (a) No recharge. (b) With recharge.

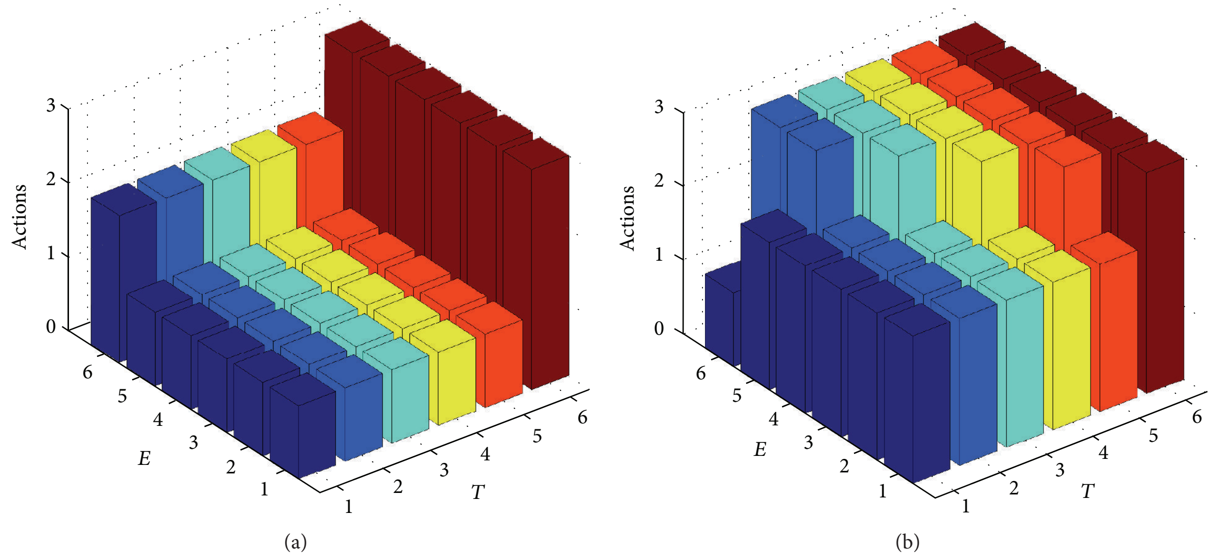


FIGURE 4: Optimal policy for $\tau = 5$ and $\mathscr{E}_0 = 5$. Actions 1, 2, and 3 denote *Sample*, *Recharge*, and *Sleep*, respectively. (a) Policy for channel state 1. (b) Policy for channel state 2.

When recharge is allowed, the maximum safe temperature level (τ) plays a critical role. This is due to the temperature increase caused by the recharge action. In Figure 3(b), for the same initial energy, the expected number of samples increases as τ is varied. Increasing τ enables the *Recharge* action to be performed more often. On the other hand, as one would expect, if τ is fixed and ($\mathscr{E}_0$) is varied, the expected number of samples slightly increases when τ is small. However, when τ is large (≥6), the maximum possible expected number of samples can be achieved when $\mathscr{E}_0$ is at its maximum value. Therefore, for this particular example, if $\mathscr{E}_0 = 10$, the optimal value for τ is 6.

Figures 4(a) and 4(b) show the optimal action for each possible system state. In Figure 4(a), for channel state 1, the *Sample* action is performed in 70% of the system states. The *Sleep* action is performed whenever the temperature reaches the maximum safe level (τ), and the *Recharge* action is performed when the remaining energy is zero and the temperature is below τ.

By contrast, in Figure 4(b), for channel state 2, the *Sample* action is performed only once at the initial system state. For this channel state, due to the higher cost of transmission, the biosensor is put in the sleep mode most of the time. However, since the cost of the *Recharge* action is independent of the channel state, the system recharges itself more often to enable more samples to be generated when the wireless channel switches to a state with a lesser transmission energy requirement (i.e., channel state 1).

7.2. Comparative Analysis. In order to be able to appreciate the merit of any approximate policy, a more meaningful performance criterion is needed. In this work, *the average number of time slots needed to generate a sample* is used as a criterion to distinguish between the different policies

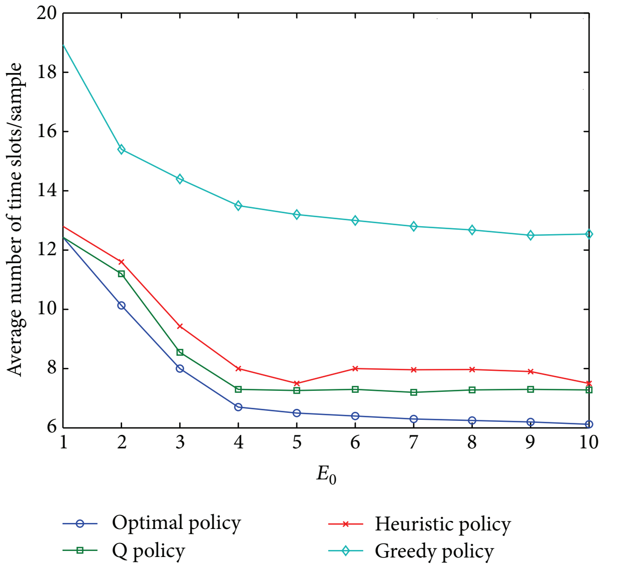


FIGURE 5: Average number of time slots needed to generate a sample when τ is fixed at five and $\mathscr{E}_0$ is varied.

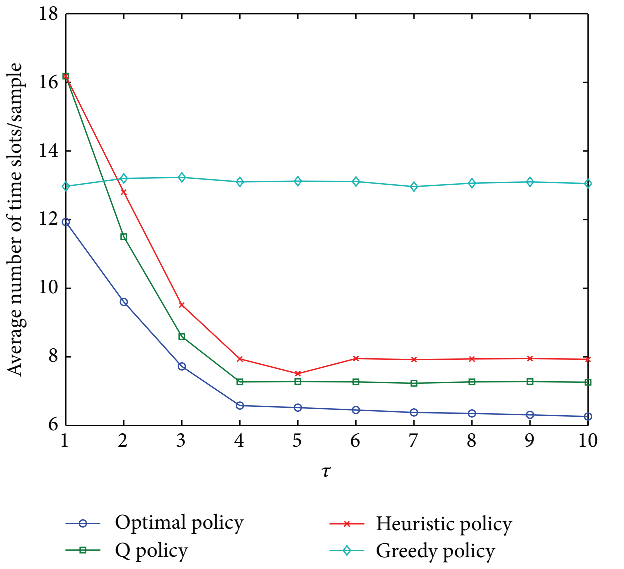


FIGURE 6: Average number of time slots needed to generate a sample when $\mathscr{E}_0$ is fixed at five and τ is varied.

available to run a system. It is calculated as the total simulation time divided by the average number of samples generated by the system while being operated by a certain policy. This measure takes into account the effect of the *Recharge* and *Sleep* actions.

For example, consider Figure 5. In this figure, τ is fixed at five while $\mathscr{E}_0$ is varied. The greedy policy is very costly since it requires the largest amount of time before a sample can be generated. The difference in the amount of time required by the heuristic policy and that required by the optimal policy stays around two time slots. This is a 75% reduction in time when compared to the greedy policy. The Q policy is the best approximate policy. On average, the difference with the optimal policy stays around 1.1 time slots.

Figure 6 shows the amount of time required to generate a sample when $\mathscr{E}_0$ is fixed at five and τ is varied. In this figure, when $\tau = 1$, the greedy policy outperforms both the Q policy and heuristic policy. A difference of three time slots is observed. This can be explained as follows. In the Q and heuristic policies, the *Recharge* action can be performed in one state only (i.e., when $T = E = 0$). On the other hand, with the greedy policy, the *Recharge* action can be performed in more than one state (i.e., whenever $T = 0$). This, of course, leads to a reduction in the average amount of time needed to generate a sample. Other than that, for $\tau \geq 2$, the Q and heuristic policies are always better than the greedy policy, and their performance is close to that of the optimal policy.

8. Conclusions

The increase in temperature due to the heat generated by biosensors is a limiting factor in the operation of biosensor networks. This problem can be modeled as a stochastic control problem using the framework of Markov decision processes. The solution is an optimal policy which ensures that the maximum safe temperature level is not exceeded. In order to handle large-size MDP models, it is shown how Q-learning can be used for obtaining the optimal policy. In addition, a heuristic policy is proposed. Its performance is comparable to that of the policies obtained by the MDP model and Q-learning.

This work can be extended in the following directions. First, the scenario of more than one rechargeable biosensor should be studied. In this case, the number of possible system states is exponentially huge. Thus, techniques for eliminating equivalent states would be necessary. Second, the performance of other reinforcement learning techniques should be investigated, especially for models with a huge state space. Third, algorithms for computing better heuristic policies should be developed to mitigate the problem of finding better approximate policies.

Acknowledgment

The first author would like to acknowledge the financial support of King Fahd University of Petroleum and Minerals (KFUPM) while conducting this research.

References

[1] National Council on Radiation Protection and Measurements (NCRP), “A practical guide to the determination of human exposure to radiofrequency fields,” Tech. Rep. 119, NCRP, 1993.

[2] International Electrotechnical Commission (IEC), “Medical electrical equipment, part 2–33: particular requirements for the safety of magnetic resonance equipment for medical diagnosis,” IEC 60601-2-33, 1995.

[3] S. K. S. Gupta, S. Lalwani, Y. Prakash, E. Elsharawy, and L. Schwiebert, “Towards a propagation model for wireless biomedical applications,” in *Proceedings of the International*

Conference on Communications (ICC 2003), pp. 1993–1997, IEEE, May 2003.

[4] H. H. Pennes, "Analysis of tissue and arterial blood temperatures in the resting human forearm," *Journal of Applied Physiology*, vol. 85, no. 1, pp. 93–122, 1948.

[5] D. M. Sullivan, *Electromagnetic Simulation Using the FDTD Method*, IEEE Press, New York, NY, USA, 2000.

[6] Q. Tang, N. Tummala, S. K. S. Gupta, and L. Schwiebert, "Communication scheduling to minimize thermal effects of implanted biosensor networks in homogeneous tissue," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 7, pp. 1285–1294, 2005.

[7] Q. Tang, N. Tummala, S. K. S. Gupta, and L. Schwiebert, "TARA: Thermal-Aware Routing Algorithm for implanted sensor networks," in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, vol. 3560, pp. 206–217, Springer, July 2005.

[8] Y. Chen, Q. Zhao, V. Krishnamurthy, and D. Djonin, "Transmission scheduling for optimizing sensor network lifetime: a stochastic shortest path approach," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2294–2309, 2007.

[9] N. Jaggi, K. Kar, and A. Krishnamurthy, "Rechargeable sensor activation under temporally correlated events," *Wireless Networks*, vol. 15, no. 5, pp. 619–635, 2009.

[10] G. Z. Yang, *Body Sensor Networks*, Springer, Berlin, Germany, 2006.

[11] A. Seyedi and B. Sikdar, "Energy efficient transmission strategies for Body Sensor Networks with energy harvesting," in *Proceedings of the 42nd Annual Conference on Information Sciences and Systems (CISS '08)*, pp. 704–709, IEEE, March 2008.

[12] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA, 2000.

[13] H. S. Wang and N. Moayeri, "Finite-state Markov channel—a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, 1995.

[14] Q. Zhang and S. A. Kassam, "Finite-state markov model for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1688–1692, 1999.

[15] W. C. Jakes, *Microwave Mobile Communications*, John Wiley & Sons, New York, NY, USA, 1974.

[16] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, New York, NY, USA, 2005.

[17] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Volume I*, Athena Scientific, Belmont, Mass, USA, 2000.

[18] C. Watkins, "Learning from delayed rewards," 1989, Springer, http://www.cs.rhul.ac.uk/~chrisw/thesis.html.

[19] The MathWorks Inc., http://www.mathworks.com/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 794920, 10 pages
http://dx.doi.org/10.1155/2013/794920

Research Article

Dynamic Sensor Scheduling for Thermal Management in Biological Wireless Sensor Networks

Yahya Osais,[1] F. Richard Yu,[2] and Marc St-Hilaire[2]

[1] *Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*
[2] *Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada K1S 5B6*

Correspondence should be addressed to Yahya Osais; yosais@kfupm.edu.sa

Received 26 September 2012; Accepted 13 March 2013

Academic Editor: Nadjib Achir

Biological sensors are a very promising technology that will take healthcare to the next level. However, there are obstacles that must be overcome before the full potential of this technology can be realized. One such obstacle is that the heat generated by biological sensors implanted into a human body might damage the tissues around them. Dynamic sensor scheduling is one way to manage and evenly distribute the generated heat. In this paper, the dynamic sensor scheduling problem is formulated as a Markov decision process (MDP). Unlike previous works, the temperature increase in the tissues caused by the generated heat is incorporated into the model. The solution of the model gives an optimal policy that when executed will result in the maximum possible network lifetime under a constraint on the maximum temperature level tolerable by the patient's body. In order to obtain the optimal policy in a lesser amount of time, two specific types of states are aggregated to produce a considerably smaller MDP model equivalent to the original one. Numerical and simulation results are presented to show the validity of the model and superiority of the optimal policy produced by it when compared with two policies one of which is specifically designed for biological wireless sensor networks.

1. Introduction

Biological wireless sensor networks (BWSNs) are networks made up of biological sensors (*biosensors*, for short) which are tiny wireless devices attached or implanted into the body of a human or animal to monitor and control biological processes. They have originated because of the need to improve and modernize healthcare. The sensing elements in biosensors are biological materials such as enzymes and antibodies. They are integrated into transducers for producing electrical signals in response to biological reactions and changes.

A famous application of BWSNs is the geodesic sensor network developed by EGI corporation [1]. In this application, a cap-based system of electrodes is worn by a patient for continuous brain monitoring. Figure 1 shows a girl wearing a geodesic sensor network. The sensor network collects electroencephalographical (EEG) measurements of the brain and delivers them to a controller which processes them and displays the results. Another example is glucose biosensors which monitor the blood glucose level in a diabetic patient. They can be used to optimally control the infusion of insulin into the patient or to initiate a prompt medical intervention. An example of glucose biosensors can be found in [2].

In addition to being energy-constrained, biosensors are also temperature-constrained. This is due to the heat generated as a result of their operation in temperature-sensitive environments like the human body. Radiation which is mainly due to wireless communication is the major source of heat. (Another major source of heat is the radiation due to RF recharging in rechargeable BWSNs. This source of heat is not considered here since we are assuming nonrechargeable biosensors.) The tissues surrounding biosensors absorb the RF energy which gets transformed into heat. This effect is balanced by the human thermoregulatory system. However, if the generated heat is larger than what can be drained, the temperature of the tissues rises. If the blood flow is not sufficient, the affected tissues might be damaged.

Thermal effects caused by biosensors are a major obstacle in the road to realizing the full vision for BWSNs. These effects can be mitigated through the use of effective thermal

Figure 1: A girl wearing a geodesic sensor network [1].

management techniques. One such technique is the dynamic scheduling of the transmission of biosensor measurements. As will be shown, this technique is very effective in reducing the temperature rise in the tissues due to heating. In this paper, the thermal management problem in BWSNs is studied. It is shown how it can be modeled as a stochastic control problem. Randomness is present due to the random behavior of the wireless channel between biosensors and the base station where measurements are collected and processed.

Toward that end, the framework of MDPs is used to build a mathematical model of the BWSNs under study. The model is then solved to obtain a policy which dictates how the BWSN should be operated in order to avoid a hazardous temperature increase. The obtained policy can achieve the best balance between transmission energy consumption and temperature increase. It also results in the minimum temperature increase when compared to existing policies.

In order to reduce the size of the MDP model, state aggregation is used. Two classes of system states are identified. A considerable reduction in the size of the MDP model is achieved when the states in these two classes are aggregated. The equivalence of the reduced MDP model to the original one is established and the reduction in model size is shown. A reduction of as high as 79% can be achieved.

The remainder of the paper is organized as follows. First, the necessary background information is given. Second, the available literature is briefly reviewed. Third, the system under study is described. Then, its MDP model is presented. After that, the minimization of the size of the MDP model using state aggregation is discussed. Then, numerical and simulation results are presented within the context of an example to illustrate the viability of the MDP model. Finally, conclusions and directions for further research are given.

2. Background

This section presents the necessary information to understand how temperature increase is calculated. It also briefly explains MDPs and points out some approaches for handling their state explosion problem.

2.1. Calculating Temperature Increase. RF signals used for wireless communication and recharge of implanted biosensors produce electrical and magnetic fields. When a human gets exposed to electromagnetic fields, the absorbed radiation gets converted into heat which manifests itself as a temperature increase inside the tissues. This phenomenon is balanced by the human thermoregulatory system. If the generated heat is larger than what can be drained, the temperature of the tissues will rise. The tissues might be damaged if the generated heat cannot be regulated by the blood circulation system.

The level of radiation absorbed by the human body when exposed to RF radiation is measured by the specific absorption rate (SAR) which is expressed in units of watts per kilogram (W/Kg). SAR records the rate at which radiation energy is absorbed per unit mass of tissue [3]. SAR is a point quantity. That is, its value varies from one location to another. SAR in the near field (i.e., the space around the antenna of the biosensor) causes the heating of the tissue surrounding the biosensor. It is a function of the current provided to the antenna of the biosensor. As an example to appreciate the importance of this measure, it was reported in [4] that an exposure to an SAR of 8 W/Kg in any gram of tissue in the head for 15 minutes may result in tissue damage.

The Pennes's bioheat equation [5] is the standard for calculating the temperature increase in the body due to heating. This equation can be transformed into a discrete form by using the finite-difference time-domain (FDTD) method [6]. Basically, the area under consideration is divided into cells and the temperature is evaluated in a grid of points defined at the centers of the cells. It is assumed that the temperature of the surrounding cell points is the normal body temperature (i.e., 37°C).

2.2. Markov Decision Processes. An MDP is a model of a dynamic system whose behavior varies with time. The elements of an MDP model are the following [7]:

(1) system states,

(2) possible actions at each system state,

(3) a reward or cost associated with each possible state-action pair,

(4) next state transition probabilities for each possible state-action pair.

The solution of an MDP model (referred to as a policy) gives a rule for choosing an action at each possible system state. If the policy chooses an action at time t depending only on the state of the system at time t, it is referred to as a stationary policy. An optimal stationary policy exists over the class of all policies if every stationary policy gives rise to an irreducible Markov chain. This means that one can limit the attention to the class of stationary policies.

An interesting class of MDPs is the class of MDPs with a terminating state. This state is reached with probability one in a finite number of steps. The number of steps represents the lifetime of the Markovian process induced by the MDP model (hence, the lifetime of the modeled system). The solution of the model is a policy which drives the system into the terminating state while optimizing an objective function which may include the lifetime of the system as a parameter.

In order to obtain a policy from an MDP model, it is necessary to form and solve the so-called optimality equation (or Bellman equation). The following is the standard form of this equation with the maximization operator [8]:

$$V_n(s) = \max_{a \in A(s)} \left[f(s,a) + \sum_{s' \in S} \mathbb{P}\left(s, s', a\right) V_{n-1}\left(s'\right) \right], \quad (1)$$

where n is the iteration index, S is the set of system states ($s \in S$), $A(s)$ is the set of actions possible when the system is at state s, $f(s,a)$ is the reward/cost per step, $\mathbb{P}$ is the system state transition probability matrix, and $V(s)$ is the optimal value of the objective function when the system is started at state s and the optimal policy is followed.

Equation (1) can be solved using the classical policy iteration, value iteration, and relative value iteration algorithms [8]. However, these algorithms become impractical when the number of system states is large. In such situations, one typically resorts to approximate techniques such as in [9–12]. Another solution for the problem of state explosion is *state aggregation* [13–17]. In this technique, using some notion of equivalence, equivalent states are combined into one class which is represented by a single state in the reduced MDP model. The new MDP model is equivalent to the original one but with significantly fewer states. In this paper, this technique is used to aggregate two kinds of system states.

3. Related Work

The research on the possible biological effects caused by biosensors and how to mitigate those effects is very recent. Most of the existing research deals with other technical issues such as energy efficiency and quality of service. In this section, we briefly review the limited available literature.

The effect of leadership rotation on a cluster-based biological WSN was studied in [18]. It was observed that rotating the role of which node collects measurements from other sensors and delivers them to the base station can significantly reduce the temperature increase in tissues due to wireless communication. The computation of an optimal rotation sequence involves using the Pennes's bioheat equation [5] and the finite-difference time-domain (FDTD) method [6] to calculate the temperature increase due to a sequence. Because of its time requirement, the authors proposed another scheme to calculate the temperature increase. It is referred to as the temperature increase potential (TIP). It efficiently estimates the temperature increase of a sequence. Using this scheme and a genetic algorithm, they were able to find the minimum temperature increase rotation sequence. They, however, did not consider the effect of the wireless channel and limited energy.

The issue of routing in biological WSNs was studied in [19]. The authors proposed a thermal-aware routing protocol that routes the data away from high temperature areas referred to as hot spots. The location of a biosensor becomes a hot spot if the temperature of the biosensor exceeds a predefined threshold. The proposed protocol achieves a better balance of temperature increase and shows the capability of load balance.

In [20], the sensor scheduling problem is formulated as an MDP. The objective is to find an operating policy that maximizes the network lifetime. The state of a sensor is characterized by its current energy level only. Three kinds of channel state information are considered: global, channel statistics, and local. Considering only the energy level at each sensor gives rise to an acyclic (i.e., loop-free) transition graph which enables the MDP model to converge in one iteration. On the other hand, if the temperature of each sensor is included in the model, the transition graph of the underlying MDP becomes cyclic. This is because when the sensor cools down (i.e., its temperature decreases), it transitions back to a less hot state. An MDP model whose transition graph is cyclic needs more time to converge.

Dynamic sensor activation in networks of rechargeable sensors is considered in [21]. The objective is to find an activation policy that maximizes the event detection probability under the constraint of slow rate of recharge of the sensor. The state of the system is characterized by the energy level of the sensor and whether or not an event would occur in the next time slot. The recharge event is random and recharges the sensor with a constant charge. The model does not include the state of the wireless channel which is very crucial when temperature is considered.

Body sensor networks [22] with energy harvesting capabilities are another kind of WSNs in which each sensor has an energy harvesting device that collects energy from ambient sources such as vibration, light, and heat. In this way, the more costly recharging method which uses radiation is avoided. The interaction between the battery recharge process and transmission with different energy levels is studied in [23]. The proposed policies utilize the sensor's knowledge of its current energy level and the state of the processes governing the generation of data and battery recharge to select the appropriate transmission mode for a given state of the network.

4. System Model

Figure 2 shows a BWSN consisting of three biosensors implanted into the body of a patient. The biosensors communicate with an access point (or base station) over a wireless channel. The wireless access point initiates the data collection process by determining which biosensor is going to transmit the next measurement. A biosensor is selected for transmission based on the current network state and some policies. The wireless access point is assumed to know the global channel state information (CSI) of the wireless channel and the state of each biosensor at each point in time. It is assumed that the instantaneous received signal-to-noise ratio (SNR) fully characterizes the state of the wireless channel.

The setup in Figure 2 can mathematically be modeled as a discrete-state system which evolves in discrete time. Thus, the time axis is divided into slots of equal duration ΔT and time $t \in \mathbb{Z}^{+}$ is the time interval $[t\Delta T, (t+1)\Delta T)$. The state

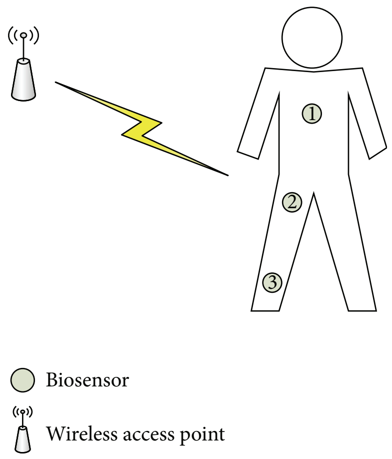


Figure 2: A patient with three biosensors implanted into his body.

of the system represents its condition at the beginning of a time slot. Control (i.e., which biosensor to choose next) can only be exercised at the beginning of a time slot and not at any other time during the slot. For example, the current temperature and remaining energy of each biosensor and the CSI of the wireless channel are used to represent the state of the system in Figure 2. Also, the number of biosensors is used to represent the number of possible control actions that can be used to control the evolution of the system.

The system in Figure 2 works as follows. At the beginning of each time slot, a biosensor is selected by the base station to transmit its measurement. As a result, the energy and temperature of the selected biosensor change according to its transmission energy requirement which is determined by the state of the wireless channel. Also, the temperature of the neighbors of the selected biosensor increases based on the amount of energy used in the transmission. On the other hand, the temperature of the nonneighboring biosensors decreases. The change in the temperature of the biosensors can be calculated using the Pennes's bioheat equation and the FDTD method (for more details, see Section 2.1.). However, due to the large simulation time required before the temperature change reaches a steady state, this approach is not followed here. Instead, the temperature decrease is assumed to be a constant reduction which occurs whenever the biosensor is not transmitting and not a neighbor of a transmitting biosensor. The temperature increase is also assumed to be directly proportional to the energy consumed by the transmitting biosensor.

Clearly, from the previous description, the location of a biosensor represents a critical point since it experiences the maximum temperature increase. This is because the tissues surrounding a biosensor might be heated continuously due to the local radiation generated by the biosensor itself and the radiation generated by its neighbors.

Let χ be the set of biosensors which have been surgically implanted in the body of a patient and at known locations. Also, let Υ_i be the set of biosensors which are neighbors to biosensor i. Different criteria can be used to compute this set. In this work, the Euclidean distance between biosensors is used. Each biosensor $i \in \chi$ has a battery with an initial energy of $\mathscr{E}_0$ and a maximum safe temperature level τ which must not be exceeded. In each time slot t, the state of a biosensor i is characterized by two variables which are the current temperature $T_t(i)$ and remaining energy $E_t(i)$. The energy required for a biosensor i to successfully transmit its measurement to the base station is determined by the state of the wireless channel in time slot t in which it is scheduled. This transmission energy is a random variable that is denoted by $W_t(i)$ and is IID over all sensors and time slots. Due to hardware and power limitations, $W_t(i)$ is discretely distributed over a finite set $\{\epsilon, \epsilon_2, \ldots, \epsilon_L\}$, where $0 < \epsilon_1 < \epsilon_2 < \cdots < \epsilon_L < \infty$ and ϵ_j is the energy consumed by a biosensor in transmitting its measurement at the jth power level.

At the end of each time slot, the energy level at each biosensor i is given by the following equation:

$$E_{t+1}(i) = \begin{cases} E_t(i) & \text{if } i \neq a \\ E_t(i) - W_t(a) & \text{if } i = a, \end{cases} \tag{2}$$

where a is the index of the sensor selected for transmission. Similarly, the temperature of each biosensor i is given by the following equation:

$$T_{t+1}(i) = \begin{cases} \mathscr{F}(T_t(i), W_t(a)) & \text{if } i = a \mid i \in \Upsilon_a \\ T_t(i) - \kappa & \text{if } i \neq a \ \& \ i \notin \Upsilon_a, \end{cases} \tag{3}$$

where $\mathscr{F}$ is a function of the transmission power and current temperature of the sensor scheduled for transmission and κ is the amount by which the temperature of a nonneighboring sensor decreases. The symbol | denotes the logical OR operator. It should be noted that the change in temperature experienced by the scheduled biosensor and its neighbors is assumed to be the same. This is a realistic assumption since biosensors in the same neighborhood experience the same amount of radiation.

Finally, the communication between the biosensor and base station occurs over a Rayleigh fading channel with additive Gaussian noise. Hence, the instantaneous received SNR denoted by γ is exponentially distributed with the following probability density function [24]:

$$P(\gamma) = \frac{1}{\gamma_0} \exp\left(-\frac{\gamma}{\gamma_0}\right), \tag{4}$$

where γ_0 is the average received SNR.

Such a wireless channel can be modeled as a finite-state Markov chain (FSMC) [25, 26]. The model can be built as follows. For a wireless channel with K states, the state boundaries (i.e, SNR thresholds) are denoted by $\Gamma_1, \Gamma_2, \ldots, \Gamma_K, \Gamma_{K+1}$ where $\Gamma_1 = 0$ and $\Gamma_{K+1} = \infty$. The channel is said to be in state s_i if the SNR is between Γ_i and Γ_{i+1} where $i = 1, 2, \ldots, K$. It is assumed that the SNR remains the same during packet transmission and only transitions to the current or adjacent states are allowed.

The steady-state probability of the ith state of the FSMC is given by

$$P(s_i) = \exp\left(-\frac{\Gamma_i}{\gamma_0}\right) - \exp\left(-\frac{\Gamma_{i+1}}{\gamma_0}\right) \tag{5}$$

and thus the state transition probabilities are

$$P(s_{i+1} \mid s_i) \approx \frac{N(\Gamma_{i+1})\,\Delta t}{P(s_i)},$$
$$P(s_{i-1} \mid s_i) \approx \frac{N(\Gamma_i)\,\Delta t}{P(s_i)}, \tag{6}$$

where $N(\Gamma_i)$ is the average number of times per unit interval that the SNR crosses level Γ_i and Δt is the packet duration. $N(\Gamma_i)$ can be computed using the following equation [27]:

$$N(\Gamma_i) = \sqrt{2\pi\Gamma_i} f_d e^{-\Gamma_i}, \tag{7}$$

where f_d is the maximum Doppler frequency defined as $f_d = v/\lambda$ with v being the speed of the subject and λ being the wavelength.

Therefore, the transmission energy requirement for a biosensor i follows a Markov chain with L states and transition probabilities $P[W_{t+1}(i) = w' \mid W_t(i) = w]$, where $w, w' \in \{\epsilon_j\}_{j=1}^{L}$. This channel model has been verified to be precise when the fading process is slow [25] such as in biosensor applications.

5. MDP Model

5.1. Formulation. The purpose of the MDP formulation of the system described in the previous section is to find a policy π that prescribes the best action to take in each state of the system so as to maximize the long-term expected lifetime of the system. The policy π is a stationary policy which means that it is independent of time and depends only on the state of the system. Next, we give the details of the MDP model.

5.1.1. State Set. The state of the system with $|\Pi|$ biosensors at time t is described by a $(3 \times |\Pi|)$-dimensional vector. That is,

$$s_t = \{(T_t(1), E_t(1), W_t(1)), (T_t(2), E_t(2), W_t(2)), \ldots, (T_t(|\Pi|), E_t(|\Pi|), W_t(|\Pi|))\}. \tag{8}$$

Let S be the set of possible system states. Then, the number of possible system states is $|S| = |T|^{|\Pi|} \times |E|^{|\Pi|} \times |W|^{|\Pi|}$, where $|T|$, $|E|$, and $|W|$ are the numbers of possible temperatures, residual energies, and transmission energy levels, respectively.

The system enters a terminating state when any one of the following two conditions is true:

(1) temperature of any biosensor is harmful (i.e., $T_t(i) \geq \tau$, where τ is a maximum threshold on the allowed temperature increase);

(2) a biosensor cannot transmit its measurement due to lack of enough energy (i.e., $E_t(i) < W_t(i)$) (this condition also accounts for the case when $E_t(i) = 0$).

Once the system is in a terminating state, the system must be halted to protect the patient. The system can then be restored to an initial state by recharging the biosensors and letting them cool down.

5.1.2. Action Set. In each time slot, based on the current state of the system, the base station chooses an action (i.e., a biosensor to transmit its measurement). The set of possible actions consists of the indexes of all biosensors. In other words, the set of actions available in each state $s \in S$ is $A(s) = \{1, 2, \ldots, |\Pi|\}$.

5.1.3. Reward Function. Let $R(s, a)$ be the instantaneous reward earned by the network due to action $a \in A(s)$ when the system is in state $s \in S$. Since the goal is to maximize the expected network lifetime, the reward function can be defined as

$$R(s, a) = 1 \tag{9}$$

which assigns a unit reward to each time slot as long as the network is in a nonterminating state. Therefore, the expected sum of rewards obtained before the network reaches a terminating state represents the network lifetime. It should be pointed out that the expectation is taken over all possible state sequences generated by a given policy.

5.1.4. Transition Probability Function. The behavior of the system is described by $|A|$ $|S| \times |S|$ transition probability matrices. Each matrix is denoted by $\mathbb{P}_{s_t, s_{t+1}}(a)$ which is the probability that choosing an action a when in state s_t will lead to state s_{t+1}. More formally, $\mathbb{P}_{s_t, s_{t+1}}(a)$ can be rewritten as follows:

$$\mathbb{P}[s_{t+1} \mid s_t, a = k] = \prod_{i \in \Pi} \{P[T_{t+1}(i) \mid T_t(i), W_t(i), a = k] \times P[E_{t+1}(i) \mid E_t(i), W_t(i), a = k] \times P[W_{t+1}(i) \mid W_t(i)]\}. \tag{10}$$

5.1.5. Value Function. The thermal management problem is formulated as an infinite-horizon MDP using the average reward criterion [7]. So, let $V_\pi(s_0)$ be the expected network lifetime given that the policy π is used with an initial state s_0. Then, the maximum expected network lifetime $V^*(s_0)$ starting from state s_0 is given by

$$V^*(s_0) = \max_{\pi} V_\pi(s_0). \tag{11}$$

The optimal policy π^* is the one that achieves the maximum expected network lifetime at all nonterminating states. Hence, it gives the optimal sensor transmission schedule.

The relative value iteration (RVI) algorithm [8] is used to numerically solve the following recursive equation for $n > 0$:

$$V_n(s) = \max_{a \in A(s)} \left[R(s, a) + \sum_{s_{t+1} \in S} \mathbb{P}(s_t, s_{t+1}, a) V_{n-1}(s_{t+1}) \right]. \tag{12}$$

In (12), the subscript n denotes the iteration index. As $n \rightarrow \infty, V_n \rightarrow V^*$.

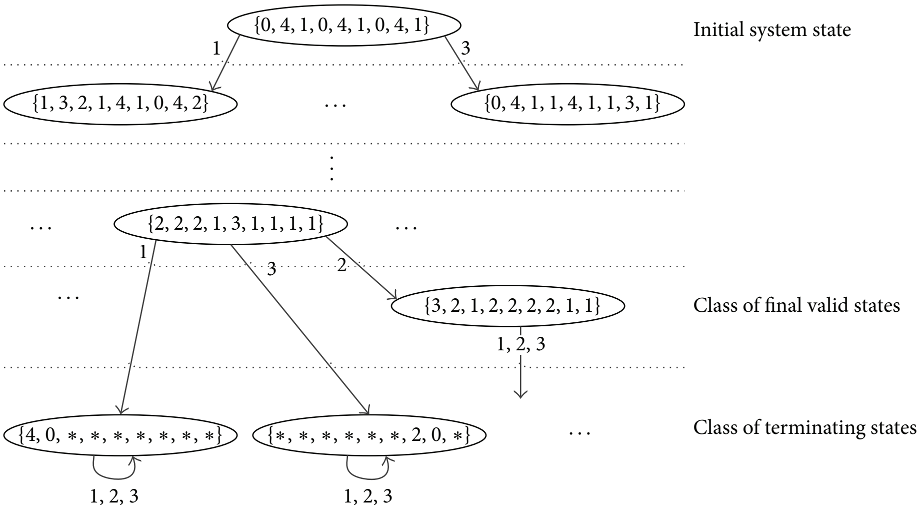


Figure 3: Excerpt of the system state space showing three classes of states.

5.2. Minimizing the Size of the MDP Model through State Aggregation. The large state space of the MDP model makes the computation of the optimal policy a highly intensive process and thus only feasible for small-scale networks. This is due to the storage and runtime requirements which are both functions of the number of possible system states. State aggregation can be used to mitigate this problem. With this technique, the state space is partitioned and the states belonging to the same partition are aggregated into one new state. Partitioning is performed by using some notion of equivalence between system states. The final result is a reduced MDP model with the same properties as the original one but significantly fewer states.

In this work, the definition of state equivalence in MDPs introduced in [14] is utilized. This definition can be stated as follows.

Definition 1 (state equivalence [14]). Two states are equivalent if and only if for every action:

(1) they achieve the same immediate reward,

(2) they transit to the same next states with the same transition probabilities.

For example, consider Figure 3 which shows an excerpt of the state space of an instance of the MDP model of the system in Figure 2. In this case, τ and $\mathscr{E}_0$ are both 4. The state space has a tree-like structure in which the root is the initial state and the leaves are the terminating states. Two important classes of states are the class of *terminating* states and the class of *final valid* states (the name is just a convention to indicate that the final working state of the system before entering a terminating state always belongs to this class of states). In the former, the states are equivalent since for each action, no reward is generated and the next state is the same as the present one with a probability of one. This class of states can be identified in $O(|S|)$ time. Similarly, in the latter, the states are equivalent since for each action, a reward of one unit is generated and the next state is a terminating state with probability one. This class of states can be identified in $O(|S||\Pi|)$ time. Additional classes of states can be identified in $O(|S|^2|\Pi|)$ time. However, this is very costly in practice due to the huge number of states. Therefore, we consider only the classes of final valid states and terminating states since they are not costly to compute and provide a considerable reduction in the size of the MDP model.

The following theorem asserts that system states identified as final valid (terminating) are equivalent and thus can be represented by one final valid (terminating) state in the reduced MDP model.

Theorem 2. *The system states in the class of final valid states (terminating states) are equivalent.*

Proof. We provide the proof for any two system states belonging to the class of final valid states. The proof for any two states belonging to the class of terminating states is the same.

By definition, a valid system state is one at which each biosensor can make a transmission (i.e., all actions are possible). Also, by definition, a final valid system state is one at which the execution of an action generates a reward of one unit and causes the system to enter a terminating state. Since all terminating states are equivalent, the system transits to a terminating state with a probability of one. □

The equivalence of the optimal policy produced by solving the reduced MDP model is established by the following theorem.

Theorem 3. *The reduced MDP model produced by combining the final valid states and terminating states induces an optimal policy for the original MDP model.*

Proof. Let S^* be the new reduced set of system states. Also, let i and j be two equivalent system states such that $i \in S$ and $j \in S^*$. Using mathematical induction, it can be shown that i and j have the same optimal value. First, we start with

Table 1: Reduction in the number of system states when terminating states and final valid states are aggregated. The number of biosensors is 3. τ and L are 7 and 2, respectively.

$\mathscr{E}_0$	Total no. of states	Reduced no. of states	Percentage of reduction
5	884736	184319	79.17
6	1404928	341802	75.67
7	2097152	569849	72.83
8	2985984	881510	70.48
9	4096000	1289835	68.51
10	5451776	1807874	66.84

the base case where $n = 0$ and $V_0(k) = 0$ for all $k \in S^*$. In this case, the optimal value for any state is just the reward for that state; that is, $V_1(k) = \max_{a \in A(k)} R(k, a)$. Since states i and j are equivalent, it is implied that $R(i, a) = R(j, a)$ for all $a \in A$ and thus $V_1(i) = V_1(j)$. This proves the base case.

For the inductive case (i.e., $n \geq 2$), using the induction hypothesis, the following can be shown for states i and j:

$$\begin{aligned} V_n(j) &= \max_{a \in A(j)} \left[R(j, a) + \sum_{k \in S^*} \mathbb{P}(j, k, a) V_{n-1}(k) \right] \\ &= \max_{a \in A(i)} \left[R(i, a) + \sum_{k \in S^*} \mathbb{P}(i, k, a) V_{n-1}(k) \right] \\ &= \max_{a \in A(i)} \left[R(i, a) + \sum_{l \in S} \mathbb{P}(i, l, a) V_{n-1}(l) \right] = V_n(i). \end{aligned} \tag{13}$$

This proves the inductive case. Therefore, it can now be established that any optimal action for state $j \in S^*$ is also an optimal action for state $i \in S$. □

Table 1 shows the percentage reduction obtained for a network with three biosensors. $\mathscr{E}_0$ is varied while fixing τ and L at 7 and 2, respectively. This considerable reduction is achieved just by aggregating the final valid and terminating states. Clearly, most of the system states fall into these two classes of system states. This can be attributed to the fact that the state space of the MDP model has a tree-like structure in which the number of leaf nodes representing terminating states is substantially large. The next substantially large number is the number of final valid states.

6. Numerical and Simulation Results

The numerical and simulation results are obtained by using the following example. Consider again the biosensor network shown in Figure 2. The biosensors are indexed from one to three. The neighbors of each biosensor are as follows:

(i) $\Omega_1 = \{2\}$,

(ii) $\Omega_2 = \{1, 3\}$,

(iii) $\Omega_3 = \{2\}$.

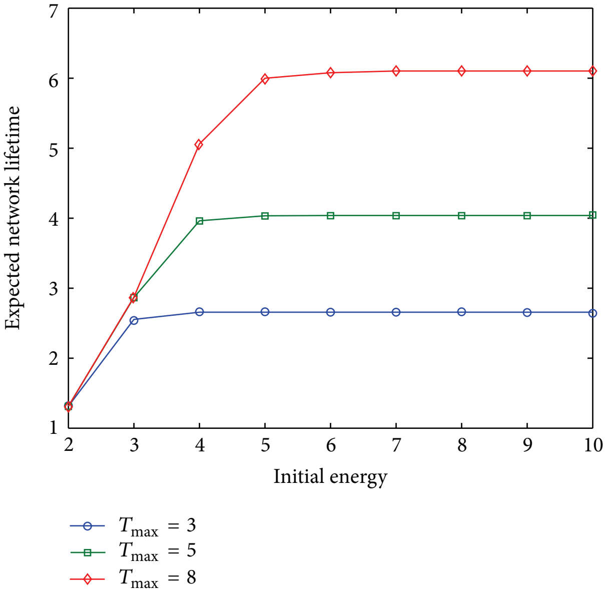


Figure 4: Expected network lifetime versus initial energy for different values of τ.

Also, the $\mathscr{F}$ function in (3) is defined for each biosensor i as

$$\mathscr{F}(T_t(i), W_t(a)) = T_t(i) + W_t(a). \tag{14}$$

The channel for each biosensor is modeled as a two-state Markov chain whose state boundary is randomly generated. A biosensor requires ϵ_k units of energy to successfully transmit its measurement when its channel is in state $k \in \{1, 2\}$. It is assumed that $\epsilon_k = k$. The transition probability matrix is

$$\begin{bmatrix} 0.2 & 0.8 \\ 0.6 & 0.4 \end{bmatrix}. \tag{15}$$

The MDP model of the biosensor network is solved using the RVI algorithm. The initial state of the network is assumed to be $\{(0, \mathscr{E}_0, 1), (0, \mathscr{E}_0, 1), (0, \mathscr{E}_0, 1)\}$. The expected network lifetime is the value calculated by the RVI algorithm for the initial state.

Figure 4 shows the expected network lifetime for different levels of initial energy ($\mathscr{E}_0$) and maximum allowed temperature increase (τ). For example, for $\tau = 3$ (i.e., a maximum temperature of three units is allowed), the maximum expected network lifetime is 2.875. This can be achieved with an initial energy of 4 units. As the curve for $\tau = 3$ shows, increasing the initial energy will not increase the expected lifetime due to the limit on the maximum allowed temperature increase.

The initial energy of a biosensor might also become a limiting factor. For example, for $\tau = 8$, $\mathscr{E}_0$ limits the maximum expected lifetime over the range of initial energies from 2 to 6. After that, τ becomes the limiting factor. In this example, the maximum expected network lifetime which can be achieved with $\tau = 8$ is 7.265 with an initial energy of 7 units.

Another interesting issue is the amount of energy which remains in biosensors after the system is halted due to a high temperature increase. For example, from Figure 4, it can be seen that for $\mathscr{E}_0 = 4$, increasing τ leads to a noticeable increase in the expected lifetime of the network. This indicates that the amount of initial energy must be determined carefully. This is because an excessive amount of remaining energy means that the patient has been exposed to an unnecessary temperature increase when the biosensors implanted in his body were charged. Thus, the measurement process has been started on already heated organs.

Figure 5 shows the actions the optimal policy makes when the remaining energy at each biosensor is fixed at three and the transmission energies of biosensors 1 and 2 are both two and that of biosensor 3 is one. $\mathscr{E}_0$ and τ are both 5. After analyzing the data, it is found that biosensor 3 is selected for transmission in 64% of the system states since it results in the minimum temperature increase. This is obvious since only one unit of energy is required for a successful transmission and the size of its neighborhood is one. Biosensor 2 is never selected. Biosensor 1, however, is selected when the temperature at biosensor 3 or its neighbor (biosensor 2) is 4. This is because if any one of them is selected, the system will enter a terminating state. So, biosensor 1 is selected to let biosensor 3 cool down and thus lengthen the network lifetime or to distribute heat evenly if the network is going to enter a terminating state.

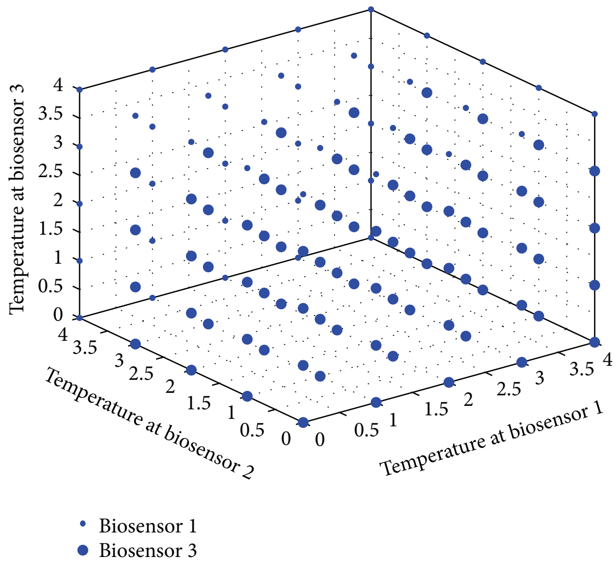


Figure 5: Optimal actions when $E(1) = E(2) = E(3) = 3$, $W(1) = W(2) = 2$ and $W(3) = 1$, $T = 5$ and $\mathscr{E}_0 = 5$.

Next, the biosensor network is simulated to compare the performance of the optimal policy with that of the TIP-based and most residual energy policies. Also, the impact of varying the initial energy and maximum safe temperature level is evaluated. The simulator is written in Matlab [28] and each data point is the average of 1000 simulation runs. The TIP-based policy (or the optimal rotation sequence) is computed as described in [18]. The optimal sequence is $(3, 1, 2)$. The peak potential is 0.148 and is experienced by biosensor 2. On the other hand, the most residual energy policy selects the biosensor whose transmission will result in the smallest reduction in energy.

First, the impact of varying the initial energy on the network lifetime is studied using simulation. Figure 6 shows the simulated lifetime of the biosensor network when the initial energy is varied from 2 to 10. Essentially, the network lifetime increases as the initial energy increases. However, after a threshold (around 4), the lifetime curve starts to level off for all policies. This is because the limit on the maximum allowed temperature increase is reached. Therefore, unless τ is increased, the average network lifetime will not increase with the increase of the initial energy.

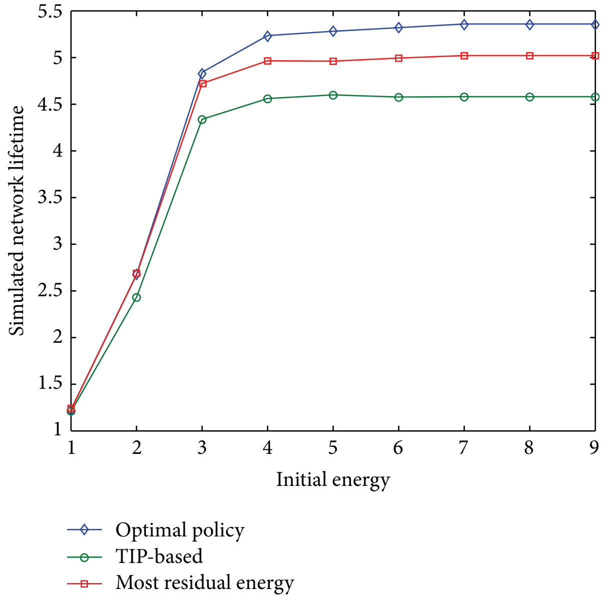


Figure 6: Simulated network lifetime versus initial energy for the different policies.

Figure 6 also shows that the optimal policy outperforms the other two policies. The TIP-based policy performs the worst. The main reason for its poor performance is that the TIP-based policy does not account for the effects of the wireless channel. On the other hand, the policy based on the most residual energy performs better than the TIP-based policy. This is because it always chooses the sensor which consumes the least amount of energy for transmission. Hence, the gap between its curve and that of the optimal policy is smaller. Nevertheless, its performance cannot reach the performance of the optimal policy since temperature is not considered explicitly.

Figure 7 shows the impact on the network lifetime when fixing the initial energy and varying the upper limit on the safe temperature level. As expected, the network lifetime increases as τ increases. However, this increase eventually levels off due to the lack of energy. Clearly, the optimal policy gives the best network lifetime. The policy based on the most residual energy gives the next best network lifetime. The worst network lifetime is achieved by the TIP-based policy.

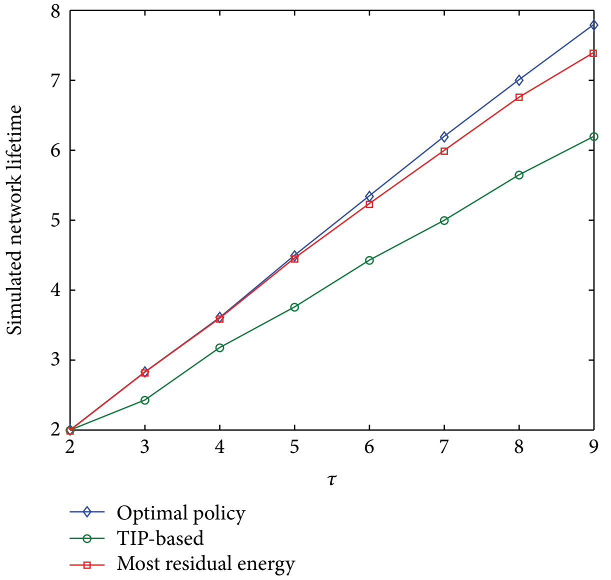


Figure 7: Simulated network lifetime versus maximum safe temperature level for the different policies.

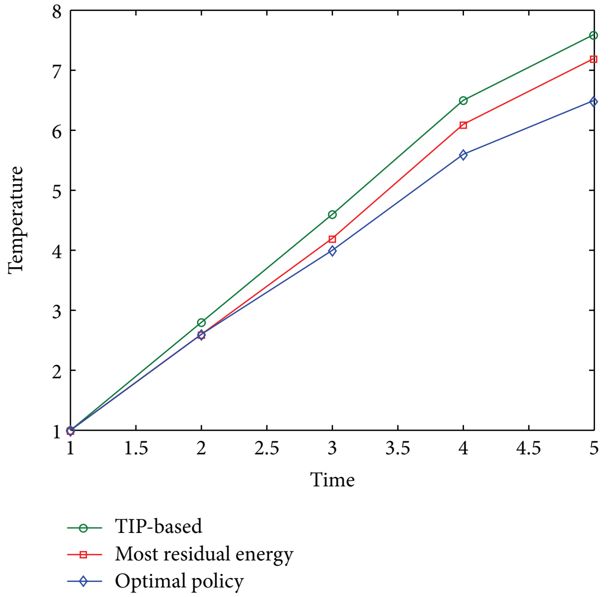


Figure 8: Temperature at biosensor 2 for the different policies.

The performance of the three policies in terms of temperature increase is compared. The initial energy is fixed at $\mathcal{E}_0 = 7$. The temperature at biosensor 2 is chosen as a metric. This is because biosensor 2 belongs to the neighborhoods of both biosensors 1 and 2. Thus, it might be heated continuously.

Figure 8 shows the temperature at biosensor 2 over four time slots. As expected, the TIP-based policy gives the maximum temperature increase. A closer examination of the simulation data reveals that biosensor 2 has indeed been continuously heated. This in turns leads to a larger temperature increase and thus shorter lifetime since the maximum allowed temperature is approached very fast.

Both the most residual energy and optimal policies give a significant improvement over the TIP-based policy. The performance of the two policies is slightly the same over the first two time slots. Then, the optimal policy shows a lower temperature increase over the remaining time slots.

The previous observation is very interesting since the goal of the TIP-based policy is to give a minimal temperature increase rotation sequence. However, since the wireless channel and its dynamics are not taken into account, the precomputed rotation sequence will most probably lead to a larger temperature increase when implemented in practice.

7. Conclusions and Directions for Further Research

The future of BWSNs is bright. However, much remains to be done to define the full potential of this technology. In this paper, we have taken one step further in understanding the thermal management problem in BWSNs. The problem is modeled as an MDP to obtain an optimal operating policy for the network. Further, the aggregation of final valid and terminating system states is proposed as a way for minimizing the number of states in the proposed MDP model. The equivalence of the reduced MDP model is established. Also, numerical results show a substantial reduction in model size which is obtained by aggregating just two types of system states. The optimal policy produced by the MDP model outperforms the policies based on the most residual energy and temperature increase potential. This is because the optimal policy gives the best balance between transmission energy consumption and the resulting temperature increase.

The following directions for further research are suggested. First, the notion of state equivalence used in this work is too strict and too sensitive. It is too strict because it requires that its conditions be met exactly. And, it is too sensitive because any perturbation of the transition probabilities can make two equivalent states no longer equivalent. More flexible metrics for state equivalence are needed. The works in [16, 17] can be used as a starting point. Second, in some applications like ours, the state transition probability matrix is built programmatically. This means a runtime which largely grows with the number of system states and thus state aggregation might not always be helpful. Hence, approximate techniques based on reinforcement learning are recommended (see [8–12]). Third, the possibility of obtaining effective policies based on simple heuristic techniques should be investigated. Heuristic techniques are typically characterized by their low runtime and storage requirements.

Acknowledgment

The first author would like to acknowledge the financial support of King Fahd University of Petroleum and Minerals (KFUPM) while conducting this research.

References

[1] EGI Corporation, "Geodesic sensor networks," http://www.egi.com/.

[2] "Pinnacle Technology," http://www.pinnaclet.com/glucose.html.

[3] National Council on Radiation Protection and Measurements (NCRP), "A practical guide to the determination of human exposure to radiofrequency fields," NCRP Report 119, 1993.

[4] International Electrotechnical Commission (IEC), *Medical Electrical Equipment, Part 2-33: Particular Requirements for the Safety of Magnetic Resonance Equipment for Medical Diagnosis*, IEC 60601-2-33, 2nd edition, 1995.

[5] H. H. Pennes, "Analysis of tissue and arterial blood temperatures in the resting human forearm," *Journal of Applied Physiology*, vol. 1, no. 2, pp. 93–122, 1948.

[6] D. M. Sullivan, *Electromagnetic Simulation Using the FDTD Method*, IEEE Press, 2000.

[7] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, 2005.

[8] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, Wiley, 1995.

[9] W. B. Powell, *Approximate Dynamic Programming—Solving the Curse of Dimensionality*, Wiley, 2007.

[10] H. S. Chang, M. C. Fu, J. Hu, and S. I. Marcus, *Simulation-Based Algorithms for Markov Decision Processes*, Springer, 2007.

[11] X.-R. Cao, *Stochastic Learning and Optimization: A Sensitivity-Based Approach*, Springer, 2007.

[12] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*, Wiley-IEEE Press, 2004.

[13] Z. Ren and B. Krogh, "State aggregation in markov decision processes," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 3819–3824, December 2002.

[14] R. Givan, T. Dean, and M. Greig, "Equivalence notions and model minimization in Markov decision processes," *Artificial Intelligence*, vol. 147, no. 1-2, pp. 163–223, 2003.

[15] P. Castro, P. Panangaden, and D. Precup, "Equivalence relations in fully and partially observable markov decision processes," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 1653–1658, Morgan Kaufmann, July 2009.

[16] N. Ferns, P. Panangaden, and D. Precup, "Metrics for finite markov decision processes," in *Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence*, pp. 162–169, AUAI Press, July 2004.

[17] N. Ferns, P. Castro, D. Precup, and P. Panangaden, "Methods for computing state similarity in markov decision processes," in *Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence*, pp. 174–181, AUAI Press, July 2006.

[18] Q. Tang, N. Tummala, S. K. S. Gupta, and L. Schwiebert, "Communication scheduling to minimize thermal effects of implanted biosensor networks in homogeneous tissue," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 7, pp. 1285–1294, 2005.

[19] Q. Tang, N. Tummala, S. K. S. Gupta, and L. Schwiebert, "Tara: thermal-aware routing algorithm for implanted sensor networks," in *Distributed Computing in Sensor Systems*, vol. 3560, pp. 206–217, Springer, 2005.

[20] Y. Chen, Q. Zhao, V. Krishnamurthy, and D. Djonin, "Transmission scheduling for optimizing sensor network lifetime: a stochastic shortest path approach," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2294–2309, 2007.

[21] N. Jaggi, K. Kar, and A. Krishnamurthy, "Rechargeable sensor activation under temporally correlated events," *Wireless Networks*, vol. 15, no. 5, pp. 619–635, 2009.

[22] G. Z. Yang, *Body sensor networks [Ph.D. thesis]*, Cambridge University, Cambridge, UK, 2006.

[23] A. Seyedi and B. Sikdar, "Energy efficient transmission strategies for Body Sensor Networks with energy harvesting," in *Proceedings of the 42nd Annual Conference on Information Sciences and Systems (CISS '08)*, pp. 704–709, March 2008.

[24] J. G. Proakis, *Digital Communications*, McGraw-Hill, 2000.

[25] H. S. Wang and N. Moayeri, "Finite-state Markov channel—a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, 1995.

[26] Q. Zhang and S. A. Kassam, "Finite-state markov model for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1688–1692, 1999.

[27] W. C. Jakes, *Microwave Mobile Communications*, Wiley, 1974.

[28] The MathWorks, http://www.mathworks.com/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 383168, 17 pages
http://dx.doi.org/10.1155/2013/383168

Research Article

Wireless Sensor Network Modeling and Deployment Challenges in Oil and Gas Refinery Plants

Stefano Savazzi,[1] Sergio Guardiano,[2] and Umberto Spagnolini[3]

[1] *National Research Council (CNR), IEIIT Institute, 20133 Milano, Italy*
[2] *Saipem S.p.A. (ENI Group), San Donato, Italy*
[3] *DEIB, Politecnico di Milano, 20133 Milano, Italy*

Correspondence should be addressed to Stefano Savazzi; stefano.savazzi@cnr.it

Received 1 November 2012; Revised 23 January 2013; Accepted 5 February 2013

Academic Editor: Marc St-Hilaire

Wireless sensor networks for critical industrial applications are becoming a remarkable technological paradigm. Large-scale adoption of the wireless connectivity in the field of industrial monitoring and process control is mandatorily paired with the development of tools for the prediction of the wireless link quality to mimic network planning procedures similar to conventional wired systems. In industrial sites, the radio signals are prone to blockage due to dense metallic structures. The layout of scattering objects from the existing infrastructure influences the received signal strength observed over the link and thus the quality of service (QoS). This paper surveys the most promising wireless technologies for industrial monitoring and control and proposes a novel channel model specifically tailored to predict the quality of the radio signals in environments affected by highly dense metallic building blockage. The propagation model is based on the diffraction theory, and it makes use of the 3D model of the plant to classify the links based on the number and density of the obstructions surrounding each individual radio device. Accurate link classification opens the way to the optimization of the network deployment to guarantee full end-to-end connectivity with minimal on-site redesign. The link-quality prediction method based on the classification of propagation conditions is validated by experimental measurements in two oil refinery sites using industry standard ISA SP100.11a compliant devices operating at 2.4 GHz.

1. Introduction

The increasing demand of oil and gas supplies frequently requires the design of very large production and processing plants over remote locations with harsh environmental conditions and challenging logistics. The adoption of cabling to fully interconnect machines for process monitoring/control lacks flexibility when in large plants, and it is becoming unfeasible due to the increasing fluctuations of wiring costs to high values. The opportunity to replace cabling by deploying a network of wireless sensors is now becoming of strategic interest for several industrial applications ranging from oil and gas refining, smart factories, transport processes [1], and more recently oil and gas exploration [2].

The status of current technology allows the deployment of low-power, cost-effective network nodes in a battery-powered configuration that substitute the traditional wired devices in a very cost-effective way [3]. The installation of wireless devices may give significant cost savings for a variety of typical plants [4]. Current wireless networks for industrial control and monitoring are based on the IEEE 802.15.4 standard [5] and are mostly considered for monitoring tasks and supervised/regulatory control. The typical locations of wireless devices used for remote control and monitoring of industrial oil and gas refinery sites are characterized by harsh environments where radio signals are prone to blockage and multipath fading due to metallic structures (structural pipe racks, metallic towers and buildings, etc.) that obstruct the direct path [6].

With the widespread use of the wireless technology in industrial environments, the development of virtual (computer aided) network planning software tools is now becoming crucial for accurate system deployment. Inaccuracies during the radio planning design phase will turn into issues during the commissioning phase. As an example, when adding new wired nodes such as gateways and/or access

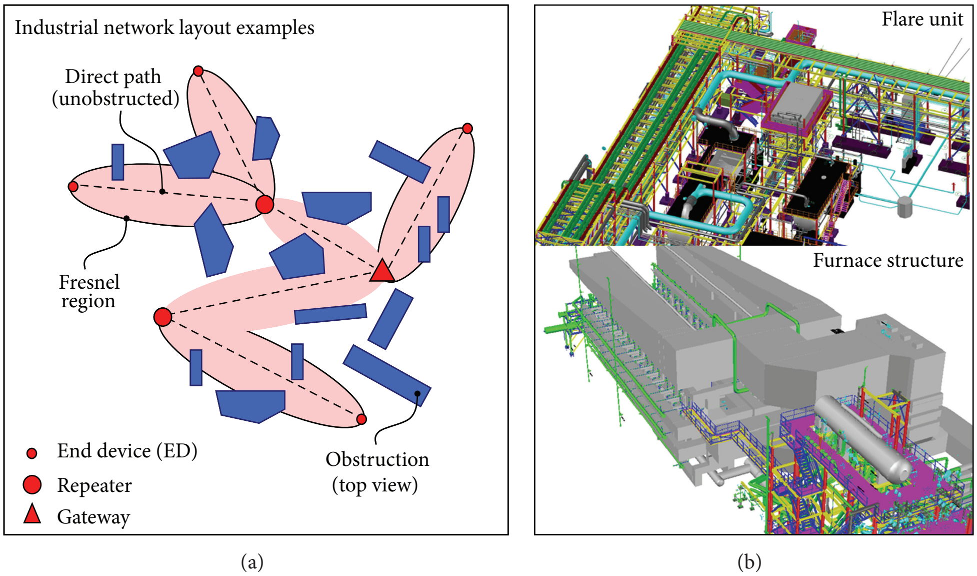


Figure 1: (a) Two-hop network architecture (ISA SP100 compliant) for deployment testing; (b) 3D-CAD model of the industrial sites for testing: flare unit (on top) and furnace structure (at bottom).

points to improve the coverage, it might be required to reopen excavations along the cable route which is totally unacceptable during the commissioning (or even before the commissioning) phase of the plant. Accurate network planning limits the need to oversize the design of the overall system, which is obviously an extra cost for the contractor. Therefore, it is crucial to develop consistent design guidelines and tools that can guarantee a reasonable accuracy in the prediction of the wireless coverage. Making use of the 3D model of the deploying area (if available) during the design phase is also of utmost importance to achieve this result. An example of a 3D view of two oil refinery sites is illustrated in Figure 1: the wireless end devices (EDs), also referred to as sensors, can be connected by star or mesh mode towards a Gateway device, with the help of intermediate Repeater nodes serving as decode and forward relays. The Gateway device is collecting data and rerouting to a wired network. Network planning is based on the prediction of the pairwise wireless link qualities among all the devices in the distributed network: the link quality is expressed in terms of the strength of the received signal. The prediction can be supported by independent radio measurement campaigns over typical refinery environments and/or by models based on propagation theory and statistical or ray-tracing tools.

Conventional empirical channel models [7] cannot fully capture the unique propagation characteristics of the industrial environments; in addition, the ray-tracing-based models [8] turn out to be not practical to process the high number of structures observed in large industrial sites [9]. This motivates the development of accurate site-specific channel models based on a small fraction of measurements taken in the refinery area.

This paper addresses a novel channel model based on the diffraction theory to assess the link quality in radio environments affected by highly dense metallic building blockage. The wireless links are partitioned into mutually exclusive classes: for each class, a separate channel model is proposed to predict the quality of the radio link. The link classification is based on the analysis of the characteristics of the obstructions that impair the wireless propagation. The 3D-CAD model of the refinery site (see Figure 1(b)) is used to identify the structure of the building blockage. Based on link classification, an optimization tool is developed for the prediction of the radio coverage and for wireless connectivity optimization. Although the channel modeling and the classification methodologies proposed in this paper are fairly general and applicable in different scenarios, the model is validated by experimental measurements using industry standard ISA SP100.11a compliant [10] devices operating at 2.4 GHz based on the IEEE 802.15.4-2011 physical layer. The measurement campaigns have been carried out in two sites located in a large-size oil refinery plant. Different practical deployment cases for coverage testing are discussed in environments characterized by blockage due to a high-density of metallic structures.

1.1. Wireless Industrial Networks: Applications and Technologies.

A typical industrial environment shows relevant similarities with dense urban microcellular sites characterized by a harsh environment for short-range (10–50 m) radio-frequency propagation with metallic structures [6], changing environmental conditions, nonline of sight (NLOS), and possible colocated wireless applications running over unlicensed spectrum [11]. Industrial networks typically require low-jitter sampling period for monitoring, high-integrity data delivery of critical messages, automatic reconfiguration, and usage of redundancy in case of communication failures. The most representative application cases for wireless technology [12] are commissioning, open-loop maintenance monitoring, closed-loop supervisory, and regulatory remote control. Notice that regulatory control is characterized by stricter reliability and delay requirements compared to supervisory control (some relevant application cases are primary flow and pressure control).

The commercial wireless systems predominantly use the so-called ISM bands at 2.4 GHz. Early experiments for cable replacing in regulatory control applications revealed that the traditional single-hop carrier sense multiple access (CSMA) schemes supported by WiFi (IEEE 802.11) perform poorly when adopted in a factory environment [13]. More recently, wireless extensions to PROFIBUS protocol for critical control have been analyzed by real-time simulations [14]. Today, commercial battery-operated systems are based on the IEEE 802.15.4 standard and enable data to be transmitted at a typical rate of 250 kbit/s, with up to a maximum of 10 dBm output RF power to meet the RF regulations for hazardous environments. The IEEE 802.15.4 physical layer also constitutes the basis for the WirelessHART [15] and ISA100.11a [10] industry standard protocols.

2. Wireless Standards for Industrial Monitoring and Control

Low-power wireless architectures and standards widely adopted in industrial automation are reviewed in this section. This introduction is instrumental to the definition of a design tool for coverage prediction and connectivity optimization. Industrial organizations such as HART and the International Society of Automation (ISA) are currently pushing towards the definition of common specifications for wireless industrial monitoring and process automation based on the IEEE 802.15.4 standard. Below we summarize the characteristics of the most relevant network solutions.

WirelessHART has been ratified by the HART Communication Foundation in 2007 as the first open wireless communication standard designed for process control applications and monitoring. Although WirelessHART adopts the IEEE 802.15.4 standard for the physical layer, the MAC layer is slightly modified as it is based on TDMA (while contention access is not allowed [15]) with guaranteed time slots assigned to the network devices. Frequency hopping spread spectrum access (FHSS) is used as proven technology to provide further improvements in terms of link gain compared to direct sequence spread spectrum (DSSS) option. The adoption of TDMA technology with precisely network-wide time synchronization is the key technology that makes WirelessHART different from other industry standards. Time synchronization is based on the Time Synchronized Mesh Protocol (TSMP). This method allows to synchronize transmitting and receiving node pairs by periodically correcting the relative time offsets misalignments. The offsets corrections are typically transmitted using standard ACK reply messages (with limited extra power consumption). The synchronous TDMA MAC sublayer is built upon the IEEE 802.15.4 physical layer for mesh network communication and defines superframes of 1 sec, fixed timeslot of 10 ms, channel hopping scheme supporting flexible blacklisting options and industry-standard AES-128 block ciphers with related keys.

ISA SP100.11a standard for wireless industrial automation is meant to provide the specifications for reliable and secure wireless operations for monitoring, alerting, open/closed-loop quality control, and predictive monitoring applications [10]. The standard supports the interoperability of multiple radio technologies. The envisioned applications include wireless process control systems (with maximum latencies in the order of 1 sec). The protocol suite, system management, and security specifications are defined for low data-rate wireless connectivity based on IEEE 802.15.4 standard. Network and transport layers are based on UDP with support of IPv6-based solutions (6LoWPAN). Coexistence with other wireless services based on IEEE 802.11x, and IEEE 802.16x standards is also addressed. Although the logical link layer of ISA SP100.11a standard has a similar structure compared to WirelessHART, the standard specifies configurable timeslots with variable durations from 10 ms to 12 ms on a superframe base. Configurable timeslots ease the development of advanced architectures based on duo-casting mechanisms, optimized coexistence, and flexibility. In ISA SP100, a transaction may consist of multiple timeslots; longer transactions can be used to extend the waiting time for multiple consecutive ACKs as required in multicast transmission. The ISA standard supports both slow and fast channel hopping schemes, thus allowing devices with imprecise timing settings to perform resynchronization and neighbor discovery.

The wireless architecture supported by the standard ISA is adopted here as reference for deployment testing. As depicted in Figure 1(a), the network infrastructure consists of the following components.

(i) The end devices (ED) are the input/output field instruments with the minimum set of functions that are necessary to join the network. The EDs take the role of reduced-function devices and typically do not provide any mechanism for relaying messages of other devices.

(ii) The Repeaters are field EDs specifically configured to serve as relay nodes for other EDs by forming a two-hop (or multihop) mesh network. In typical industrial settings where the real-time responsiveness of the monitoring network is a crucial issue, the number of hops is limited to 2, therefore, the Repeater devices act as ED range extenders.

(iii) The Gateways act as access points (or sinks) and collect the measurements acquired by the field devices. In practical settings, the Gateways are connected by cables (or by broadband wireless technology) to a common network manager node and thus also act as a translator between the ISA standard and other wired protocols (Foundation Fieldbus, HART, etc.).

3. Channel Modeling

In this section we introduce the channel model as instrumental to the proposed link classification approach. The wireless links without a clear line-of-sight (LOS) path undergo more severe received signal power attenuations than those where the line-of-sight (LOS) path is fully unobstructed. This additional attenuation is almost uncorrelated from the distance between the transmitter and the receiver [16]. The main scatterers/objects that are responsible for the received signal power attenuation are mostly confined within the first and second Fresnel zones as these can be considered to contribute to the main propagating energy in the wavefield [17]. For a wireless link where the direct path between the transmitter and the receiver has length d, the nth Fresnel zone is the region inside an ellipsoid with circular cross-section. The radius of the nth Fresnel zone at distance $q \leq d$ is

$$r_n(q) = \sqrt{n\lambda q(d-q)d^{-1}}, \tag{1}$$

with λ the signal wavelength.

We assume that any pair of wireless devices connected with an arbitrary link ℓ are deployed at fixed locations and distance d. The nodes are equipped with radio devices characterized by single omnidirectional antenna transceivers. As for typical scenarios, the Gateway antenna is mounted on an elevated point while flat terrain is assumed.

The propagation model describes the correlation between the size of the (mostly metallic) obstructions located within the Fresnel volumes and the total received signal strength (RSS) experienced along the propagation path. The RSS γ_ℓ is thus the metric (in decibel scale) used to assess the quality of the radio link

$$\gamma_\ell|_{\text{dB}} = \underbrace{g_0(d,\alpha) - \sigma}_{g_\ell} + s. \tag{2}$$

It combines (i) a static component g_ℓ characterized by a distance-dependent LOS term $g_0(d,\alpha)$ and an excess attenuation σ that accounts for the building blockage; (ii) a zero-mean random term s accounting for the fluctuations of the received power with typical standard deviation around $\sqrt{\mathbb{E}[s^2]} = 3 \div 5$ dB in static environments [6].

In what follows, it is derived a model for the static RSS component g_ℓ. The model is instrumental to the prediction of the average radio link quality for connectivity optimization (see Section 5). The distance-dependent static component $g_0(d,\alpha)$ describes the channel gain observed over the flat terrain and without obstructions. The term σ denotes the additional signal attenuation as a function of the size and the density of the metallic objects located within the Fresnel volume (i.e., blocking the LOS path). The model used to characterize the additional attenuation component σ is derived in Section 3.1. As observed in [18], the reflection of the radio signals from the flat terrain does not influence the attenuation parameter σ but only the term $g_0(d,\alpha)$ and the path-loss exponent α. The model is validated based on measurements over the refinery sites (see Section 6).

The distance-dependent loss factor $g_0(d,\alpha)$ can be modeled as a function of the path-loss exponent α (see [16]):

$$g_0(d,\alpha) = g_0 - 20\log_{10}\left(\frac{1+d}{d_0}\right) - 10(\alpha-2)\log_{10}\left(\frac{1+d}{d_F}\right), \tag{3}$$

where $g_0 = g_0(P_T)$ is the channel gain function of the transmit power P_T and measured at a reference distance d_0 (d_0 = 2 m typical), while

$$d_F = \frac{2h_m h_p}{\lambda}, \tag{4}$$

is the Fresnel distance being a function of the antenna heights from the ground h_m and h_p for the pair of devices (m,p), respectively. Path loss exponent is typically set to $\alpha = 2$ in short-range environments [16] where ground reflections can be neglected, for $d < d_F$. Larger path loss exponents $\alpha > 2$ are caused by reflections from the ground and can be experimented in long-range cases for $d > d_F$.

The probability P_E of successful communication depends on the random fluctuations of the RSS as in (2). Successful communication is modeled by outage probability such that $P_E = \Pr[\gamma_\ell \geq \beta]$. The threshold β is typically set to $\beta = -85$ dBm such that $P_E \leq 10^{-6}$ [5]. Any link experiencing $\gamma_\ell < \beta$ is assumed as unreliable, and thus, it should not be accounted for during network planning.

3.1. Diffraction Model for Prediction of Building Blockage. It is assumed that the additional attenuation σ in (2) is due to propagating wavefronts diffracting around the building blockage consisting of metallic obstacles with different dimensions. Obstacles are acting as perfectly absorbing interfaces.

The diffraction model for the building blockage term σ is based on the Fresnel-Kirchhoff method [19]. The attenuation σ in (2) is obtained as a function of the received electric field E:

$$\sigma = -20\log_{10}\left|\frac{E}{E_{\text{free}}}\right|. \tag{5}$$

The ratio E/E_{free} describes the obstruction loss in excess of the free space field E_{free}. Large-size metallic objects obstructing the wireless link absorb a large amount of the signal intensity and limit the received field to a small fraction (being $E/E_{\text{free}} \ll 1$) of the one that would be observed under free-space propagation (without obstructions). A simplified description of the propagation environment (with obstacles blocking the LOS path) is considered in Figure 2. To simplify the reasoning, we assume that the obstacles surrounding the transmitter and the receiver antennas lie in the far-field region. In addition, the shape of the obstacles obstructing the Fresnel zones is squared or rectangular. Shapes that are more typical in refinery sites (tubes, structural pipe racks, etc.) have been approximated by matching a number of rectangles in the (x,y) plane to get the same shape of the obstructed areas; this is also illustrated in [20]. For the ith object, the clearance zone $\mathcal{F}_i$ in the (x,y) plane denotes the region corresponding to the Fresnel volume cross-section that is free from any obstacle. The shaded region $\mathcal{R}_i$ in the same plane indicates instead the complementary portion of the surface occupied by the obstacle.

The Fresnel-Kirchhoff approach is used to model the field loss $E(q_i)/E_{\text{free}}$ caused by a single ith obstacle located at distance $q = q_i \leq d$. The Huygens principle is used to predict

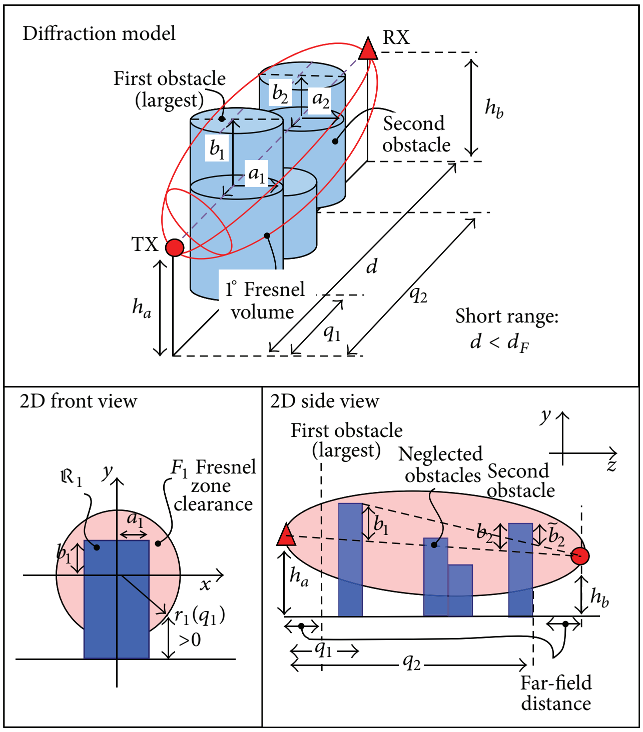


Figure 2: Fresnel-Kirchhoff method for modeling the attenuation caused by objects acting as perfectly absorbing 2D interfaces. Any hidden obstacle located in the shadow area caused by larger structural blockage can be neglected as irrelevant for additional loss.

the actual field strength diffracted by one obstacle modeled as a knife edge. The 2D model takes into account both the lateral and the vertical profiles of the obstruction by integrating the exponential phase term of the spherical wavefields over the two dimensions [19]. The electric field $E(q_i)$ measured at the receiver may be interpreted as generated by a virtual array of Huygens sources located in the plane of the single obstacle i at distance d from the receiver. Considering an object located at distance q_i from the transmitter and occupying an area $(x, y) \in \mathcal{R}_i$, the field loss $E(q_i)/E_{\text{free}}$ can be approximated for $(x, y) \ll q_i$, $d - q_i$ as [19]

$$\left|\frac{E(q_i)}{E_{\text{free}}}\right| \simeq \left|1-j\int_{(x,y)\in\mathcal{R}_i}\frac{1}{r_1^2(q_i)}\exp\left[\frac{-j\pi\left(x^2+y^2\right)}{r_1^2(q_i)}\right]dx\,dy\right|, \tag{6}$$

where $r_1(q_i)$ defined in (1) refers to the radius of the 1st Fresnel volume circular section corresponding to the location of the obstruction. The approximation reasonably fits with the considered environment (see Section 6) as far as the obstacle is confined within the Fresnel volume.

To gain further insight into the interplay between the obstruction size and the corresponding field loss, in what follows we focus on the example of a single obstacle obstructing the LOS path with rectangular cross-section described by lateral and vertical half-dimensions (a_i, b_i). The loss term in (6) simplifies for the case of large obstacle $|a_i|$, $|b_i| \gg r_1(q_i)$ as (see the appendix)

$$\left|\frac{E(q_i)}{E_{\text{free}}}\right| \approx \left|1-2j\times\Gamma\left(\frac{\sqrt{2}b_i}{r_1(q_i)}\right)\Gamma\left(\frac{\sqrt{2}a_i}{r_1(q_i)}\right)\right| \tag{7}$$

with

$$\Gamma(x)=\left[\frac{1}{2}+\frac{1}{\pi x}\sin\left(\frac{1}{2}\pi x^2\right)\right]-j\left[\frac{1}{2}-\frac{1}{\pi x}\cos\left(\frac{1}{2}\pi x^2\right)\right]. \tag{8}$$

Figure 3 compares the diffraction loss for a single object obstructing the LOS path with varying square cross-sections ($a_1 = b_1$) measured with respect to the Fresnel radius $r_1(q_1)$. Model (6) and approximation (7) are in solid and dashed lines, respectively. The field loss caused by an object fully obstructing the 1st and the 2nd Fresnel circular section such that $a_1/r_1 > \sqrt{2}$ lies below $E/E_{\text{free}} < 40\%$.

The general model (6) for a single obstacle can be extended to multiple obstacles by following the Deygout approach [21]. For multiple obstacles, the lateral a_i and vertical b_i dimensions of the shaded region $\mathcal{R}_i$ for the ith obstacle are calculated with respect to the size of the largest obstacle (obstacle $i = 1$ in the example of Figure 2). For each dimension, the Deygout method requires to find the ith object (edge) with the largest value of parameters (a, b) compared to the Fresnel size, such that $a = \arg\max_{a_i}[a_i/r_1(q_i)]$ and $b = \arg\max_{b_i}[b_i/r_1(q_i)]$ and ignoring all the other edges. Based on the selected set of obstacles, a new reference plane is created

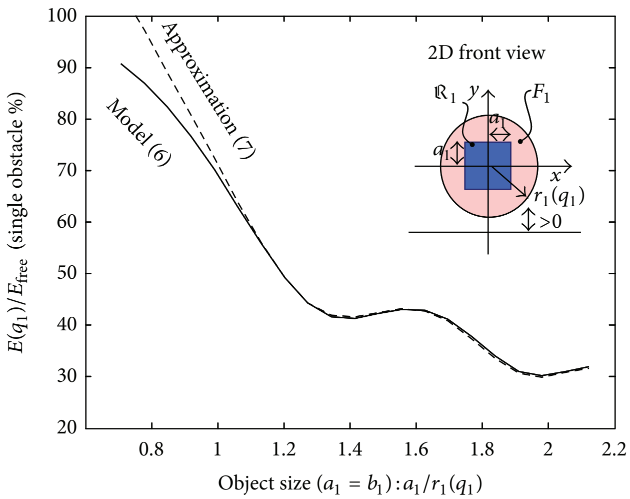


Figure 3: Diffraction loss caused by an object with square cross-section ($a_1 = b_1$) and obstructing the LOS path. Energy loss E/E_{free} is analyzed with respect to the ratio $a_1/r_1(q_1)$.

for each dimension and used to compute the contributions of all the intermediate edges with modified size, $a_i = \tilde{a}_i$, $b_i = \tilde{b}_i$ (see Figure 2). The overall obstruction loss E/E_{free} for $B > 1$ obstacles with meaningful obstructing size at distance q_i is obtained by multiplying each contribution along the LOS path so that [21]

$$\left|\frac{E}{E_{\text{free}}}\right| = \prod_{i=1}^{B} \left|\frac{E(q_i)}{E_{\text{free}}}\right|, \tag{9}$$

where each term $E(q_i)/E_{\text{free}}$ is in (6) or approximated as in (7). In spite of the simplicity of this method, in Section 6, it is proved to be accurate enough for wireless link quality prediction.

4. Wireless Link Classification

The proposed approach for the evaluation of the pairwise link channel qualities is validated by a database of radio measurements taken in different refinery sites to cover the most representative scenarios. Based on the experimental measurements, 5 mutually exclusive link categories have been defined to account for the different sizes and the positions of the most typical obstructions inside the (1st and 2nd) Fresnel volumes surrounding the considered links. The analysis of the building blockage property is based on the inspection of the full 2D/3D model of the plant. Each link type is characterized by a specific configuration of the Fresnel zone clearance that corresponds to a reference value for the obstruction loss E/E_{free} according to the model outlined in Section 3. For each link type ℓ, the loss $\sigma = \sigma(\ell)$ is computed as in (5), and it is used to predict the average link quality g_ℓ in (2).

Based on the experimental activity, five different link-types are considered (see Figure 4).

Type I. LOS ($\ell = 1$) link type is characterized by the absence of obstacles (with dimensions larger than the signal wavelength λ) within the first and second Fresnel volume, while obstacles might instead occupy the remaining Fresnel volumes. The nominal (such that from (2) $E[\gamma_{a,b}] \geq \beta$) maximum range to guarantee a reliable connection is found as $R \simeq 150$ m (for RSS above $\beta = -85$ dBm). In the worst-case scenario where obstacles completely obstruct the nth Fresnel volumes with $n \geq 3$, the observed received electric field intensity from (6) is the $E/E_{\text{free}} = 90\%$ fraction of the one that would be measured in the free-space case (thus corresponding to an attenuation of $\sigma(\ell) \simeq 1$ dB [22]).

Type II. Near-LOS ($\ell = 2$) link type is observed in environments where the obstacles are located in the first Fresnel outer region at distance $0.6 \times r_1(q)$ from the direct path. The shaded subregion in Figure 4 can be considered as a "forbidden" region: if this region is kept clear, then the total path attenuation will be practically the same as for the unobstructed case (Type I). This clearance zone is thus used here as a criterion to decide whether an object is to be treated as a relevant obstruction. The radio propagation for this link category is characterized by an additional signal energy loss compared to Type I. Based on the radio measurement campaigns and the diffraction model in (6), the Type II links typically retain the $E/E_{\text{free}} = 70\%$ of the electric field observed in the free-space case ($\sigma(\ell) \simeq 3$ dB). The theoretical maximum range reduces to $R \simeq 108$ m.

Type III. Obstructed-LOS ($\ell = 3$) link type is observed in environments where the obstacles are located inside the forbidden region, although the direct path connecting the transmitter and the receiver is still unobstructed. The links belonging to this category retain approximately $E/E_{\text{free}} = 40\%$ of the electric field that would be measured in the free-space case. The theoretical maximum range is $R \simeq 60$ m.

Type IV. NLOS ($\ell = 4$) link type is characterized by large objects obstructing the direct path between transmitter and receiver; therefore, $E/E_{\text{free}} < 40\%$ ($\sigma(\ell) \simeq 8$ dB): the size of those objects is such that a clearance zone is still visible, $\bigcap_i \mathcal{F}_i \neq \emptyset$, suggesting that there might be the possibility of reliable communication. Being the forbidden region and the LOS path both obstructed, the reference value for the field loss is chosen as $E/E_{\text{free}} = 20\%$ ($\sigma(\ell) \simeq 14$ dB). The theoretical maximum range further reduces to $R \simeq 32$ m.

Type IV-S. Severe-NLOS ($\ell = 5$) link type refers to a severe NLOS environment where the first and the second Fresnel regions are *completely* obstructed by one or more obstacles with significant size (and dimensions scaling as $\sim 4 \div 5r_1(q)$), so that the observed received electric field falls below $E/E_{\text{free}} = 10\%$ compared to the one that would be measured in the free-space case ($\sigma(\ell) \simeq 21$ dB). The theoretical maximum range is $R \simeq 15$ m. This model type resembles a propagation environment where the line-of-sight path is blocked by large-size concrete buildings [18].

5. Radio Planning Optimization

The wireless network deployment problem refers to the determination of the positions of the wireless nodes such that

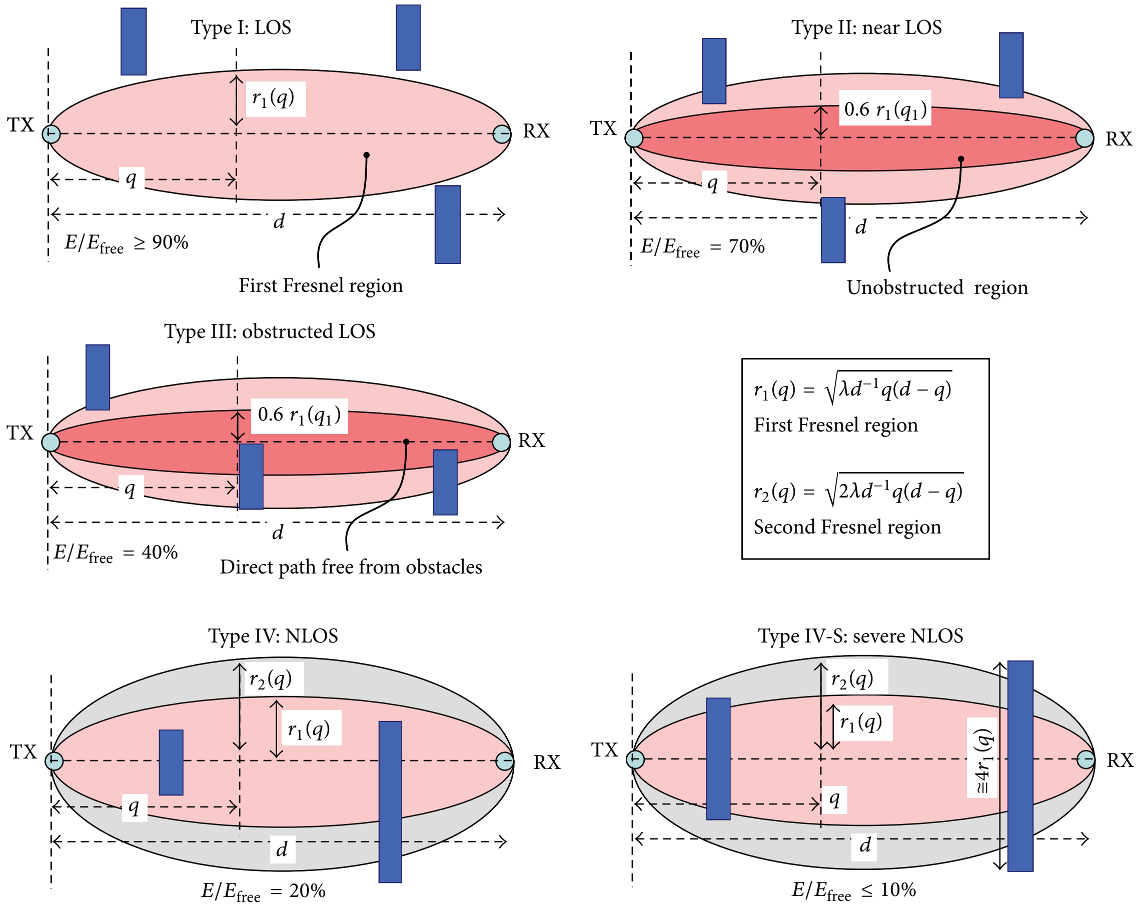

Type I: LOS
TX
RX
$r_1(q)$
q
d
$E/E_{\text{free}} \geq 90\%$
First Fresnel region
Type II: near LOS
$0.6\ r_1(q_1)$
$E/E_{\text{free}} = 70\%$
Unobstructed region
Type III: obstructed LOS
$E/E_{\text{free}} = 40\%$
Direct path free from obstacles
$r_1(q) = \sqrt{\lambda d^{-1} q(d-q)}$
First Fresnel region
$r_2(q) = \sqrt{2\lambda d^{-1} q(d-q)}$
Second Fresnel region
Type IV: NLOS
$r_2(q)$
$E/E_{\text{free}} = 20\%$
Type IV-S: severe NLOS
$\geq 4r_1(q)$
$E/E_{\text{free}} \leq 10\%$

FIGURE 4: Proposed link classification and Fresnel clearance zones.

some limiting values of coverage, connectivity, and energy efficiency can be achieved [23]. Wireless device deployment strategies for coverage and connectivity enhancement play a crucial role in providing better quality of service (QoS) to the network. The coverage and the connectivity problems are two fundamental issues that have been widely studied in the literature [24]. In coverage problems, the objective is to deploy wireless sensor devices in strategic ways such that an optimal area coverage is achieved given the requirements of the underlying application [23]. The coverage problem therefore deals with placing a minimum number of nodes so that every measurement point in the sensing field is optimally covered according to application-specific constraints. In industrial monitoring and control applications, the position of the measurement points (sensors or actuators) is constrained by the application; therefore, the coverage optimization is typically carried out based on the structure of the process unit. The focus of this section is thus on connectivity optimization, as this is the most crucial problem for cable replacing in the industrial networking context.

In what follows, we first report on the current state of the research on optimized node placement in wireless sensor networks (Section 5.1). Next, we discuss relevant practical issues and rules that are specifically tailored for network and connectivity optimization in industrial networks (Section 5.2). Finally, we propose an optimization framework tailored for commercially available ISA SP100 two-hop networks that allows the optimal selection of the devices that need to be configured as Repeaters (Section 5.3). The goal is to optimize the number and the position of the infrastructure devices (e.g., the Repeater nodes and/or the Gateways) to guarantee a reliable connection between the measurement points and the control unit with some degree of redundancy [25]. Optimally deployed wireless infrastructure devices guarantee adequate QoS (i.e., outage probability), long network lifetime, and thus reduced costs for network maintenance. The proposed deployment problem is based on the prediction of the RSS for all the pairwise wireless links according to channel modeling and classification outlined in Sections 3 and 4.

5.1. Node Deployment Strategies in Wireless Sensor Networks: A Survey. Extensive work has been reported in the literature relating to wireless sensor and relay node deployment. Deployment of nodes has been considered for targeting connectivity, coverage, node lifetime, and/or QoS. The deployment strategies can be classified into static and dynamic [26] depending on whether the optimization is performed during network setup or during network operation (for node repositioning, see [26]). In static environments where data is periodically collected over preset routes, the problem of optimal node placement for connectivity maximization has been proven to be NP-hard for most of the formulations [27]. Several heuristics and rules have been therefore proposed to find suboptimal solutions based on graph theory. Several approaches to the problem of placing nodes are addressed in [24] to achieve K-connectivity at the network setup time so that K independent paths are identified for every pair of devices. The majority of published work on sensor network deployment limits its focus on simplified and analytically tractable 1D and 2D environments where connectivity can be considered as a primary/secondary objective or as a constraint in the deployment problem [26]. For example, in [28] an outdoor random deployment which targets the connectivity as a primary objective in 2D space is investigated. In [29], a constrained multivariable nonlinear programming problem is analyzed to determine the locations of the sensor nodes to maximize the network lifetime, given a fixed number of sensor nodes with certain coverage and connectivity requirements. A deployment strategy for sensor networks is introduced in [30] to balance the network lifetime and connectivity goals for single- and two-hop networks.

Focusing on large-scale sensor network applications, controlled placement of nodes is often focused on a subset of network devices (e.g., Repeaters or relays) with the goal of designing the network topology to achieve the desired application requirements [31]. The problem of relay placement in two-hop networks is analyzed in [32]: the objective is to place the fewest number of relay nodes so that each sensor node can communicate with at least one relay node, and the network of relay nodes is connected. The goal is to guarantee a reliable communication between each pairs of sensor nodes while the same reasoning can be extended for sensors communicating with a common Gateway node. Recent literature considers the problem of connectivity in massively dense sensor networks [33]. The problem of deploying relay nodes in heterogeneous sensor network scenarios is considered in [34] where sensor and relay nodes possess different transmission ranges (e.g., through the use of different hardware, antennas, or high-power radio modules). The work [35] considers a scenario where sensor devices are equipped with directional antennas: the goal is to find an optimal subset of locations to minimize the total network cost while satisfying the requirements of coverage and connectivity.

The network connectivity problem is mostly considered for 2D planning with the assumption of simple binary communication disk model without looking at site-specific environmental constraints (see also [34–37]). Those approaches are very prone to failure in practical large-scale industrial applications. Some attempts in the literature have been made towards the analysis of deployment and connectivity problems in 3D environments, although the topic is still considered an open issue [38]. The problem of modeling and connectivity optimization in random 3D networks has been recently addressed in [38, 39] where the deployment problem considers the maximization of network connectivity satisfying lifetime constraints.

5.2. Connectivity Optimization in Wireless Industrial Networks. The connectivity optimization for industrial networks can be in general applied to two-hop large-scale networks consisting of Gateways, relays, and sensors, operating in time (and safety) critical applications [36, 37]. Three general practical rules [40] should be followed during system design and configuration. These are summarized below.

Gateway Deployment Planning. The wireless network is first divided from a single process unit into subsections (subnetworks). Within each subsection, the position of the measurement points, and thus the degree of coverage, is designed to satisfy application-dependent requirements. The devices (or end devices, EDs) are deployed to collect data from the nearby measurement points (depending on the monitored process, EDs might consists of a single or multiple measurement points). Each process unit subsection is served by one Gateway (acting as access point for the corresponding devices). The Gateway should be able to allocate resources for two-way communication in real time with the EDs. For small-size projects (as those analyzed in Section 6), a single Gateway is sufficient if the total number of measurement points is less than the capacity C of the Gateway point. Instead, if the project is large with several hundreds of wireless devices and process units, a single network manager should manage multiple Gateways. The required number of Gateways can be defined as a function of the number of measurement points. The following simple calculation can be used in practice to approximate the number of Gateways G needed:

$$G = N \times \left[C \times (1 - \rho_{sc})\right]^{-1}, \tag{10}$$

where ρ_{sc} is the spare (or residual) fraction of the available capacity C to be reserved for emergency signalling with capacity measured in terms of number of measurement points served. N is the number of measurement devices assuming that each ED is serving as a single measurement point. A typical design rule prescribes that $\rho_{sc} = 40\%$ [40]. The Gateway capacity C depends on the wired/wireless protocol used for data transfer towards the network manager.

Connectivity Optimization. The use of site-specific radio propagation models (empirical or ray-tracing based) enables the optimization of the connectivity for virtual network planning. A propagation model can be therefore exploited as instrumental to the prediction of the RSS, from which the quality of the radio link (and of the end-to-end connectivity) can be inferred with some degree of accuracy. Prediction errors are typically caused by modeling mismatches (e.g., link classification errors) or unpredictable RSS fluctuations (see Section 3) due to interference over the 2.4 GHz band or fading induced by objects or people moving in the area. The solution to the connectivity and the Gateway deployment problems is generally well understood in the literature (see, e.g., [41]). In the industrial context, three practical rules are defined to ensure a sufficiently high link reliability. The rules are summarized as follows (see also [40]).

(i) Rule 1. Every network with more than 5 devices should have a minimum of 25% of devices within the effective range of the Gateway to ensure mesh connection (typically over a maximum number of 3 ÷ 4 hops). In any case, every network should have a minimum of 5 infrastructure devices within the effective range of the Gateway. Example: a network consisting of 100 EDs requires 25 EDs at minimum within the effective range of the Gateway (directly connected).

(ii) Rule 2. Gateway RF antenna should be mounted at least 2 m from the ground level and should not be surrounded by obstacles. Obstacles should lie at distance 2λ from the antenna.

(iii) Rule 3. Every device should have a minimum of 3 neighbors in the effective range. This ensures that when implemented, there will be at least one reliable routing path to the Gateway alternative to direct connection (to guarantee $K = 2$ connectivity).

On-Site Stress Testing. Stress testing of the deployment design is recommended during an on-site survey to verify potential weaknesses highlighted during the virtual network configuration. Stress testing is performed by altering the position of the EDs from the nominal position and thus by measuring the fluctuations of the RSS field.

Although the context may vary slightly depending on the structure of the environment, almost all these basic steps could be applied regardless of the specific commercial system and standard (i.e., WirelessHART or ISA SP100, see Section 2). The first and the second steps are known to be the most critical for high density applications [12].

5.3. Optimal Repeater Configuration for Two-Hop ISA Industrial Networks. The wireless network for industrial environment under consideration conforms with the standard ISA SP100.11a and is characterized by one Gateway collecting data from wireless end devices (EDs). A subset of EDs might serve as Repeater nodes acting as range extenders. The Gateway node is an electrically powered device, serving as access point for the EDs. It manages both wireless and wired interfaces. The Repeaters are configured as EDs with superior functionalities: these allow to connect to the Gateway and simultaneously serve as decode and forward relays for extending the range of the neighboring EDs. Repeater nodes are more expensive than standard EDs since they must be preconfigured to multiplex different sensor data and could be more powerful in terms of processing and transmission capabilities.

The connectivity optimization problem is therefore focused on the Repeater configuration. The candidate sites for the deployment of the EDs and of the Gateway node are assumed to be assigned: each candidate site might host either a Repeater node multiplexing sensor data or a standard ED without relaying functionalities. Optimal placement of the Gateway is not addressed in this paper, although we only assume that Gateway locations satisfy connectivity Rule 2 (see Section 5.2). Optimal deployment for the Gateway might be carried out as illustrated in [41], even if, in practical industrial scenarios, the exact position is subject to stringent environmental constraints.

The optimization approach is based on the selection of the smallest subset of devices that need to be configured as Repeaters to guarantee network connectivity. The optimization jointly minimizes the number of EDs connected to the corresponding Gateway over two hops [32] and guarantees a minimum quality of service for all links, so that the static RSS component g_ℓ is kept for all the configured links above the system threshold β (see Section 3) herein adopted as the minimum tolerable link quality. The static RSS component is predicted based on the 3D model of the plant, as done in Section 3.

Let the wireless network be represented by a set $\mathcal{S}$ of N nodes located at known positions within a specific area of the plant. A sequence of messages is continuously transmitted by the EDs towards a common Gateway node labeled as "0" possibly with the help of one intermediate node serving as a Repeater. To comply with the real-time responsiveness constraints typically required by industrial closed-loop control applications, the maximum number of hops to reach the Gateway node is herein limited to 2: the same constraint is also adopted in recent ISA compliant network implementations. Any wireless node $a \in \mathcal{S}$ is said to be connected with reasonable quality to the Gateway node "0" if and only if $i_{a,0} = 1$ where the indicator $i_{a,0}$ for an arbitrary link $\ell := (a, 0)$ is defined as

$$i_{a,0} = 1 \quad \text{iff } g_\ell > \beta, \quad i_{a,0} = 0 \text{ otherwise}, \tag{11}$$

while sensitivity threshold β accounts for the random fluctuations of the static RSS component g_ℓ. Deployment optimization consists of three phases.

Selection of Candidate Repeaters. First, it is defined the subset $\mathcal{S}_0 \subset \mathcal{S}$ of N_0 nodes without direct connection to the Gateway

$$\mathcal{S}_0 := \{a \in \mathcal{S} \mid i_{a,0} = 0 \ \forall a\}. \tag{12}$$

The same nodes $a \in \mathcal{S}_0$ are preconfigured as EDs, and they should not provide relaying functionalities. The remaining subset $\mathcal{S}_1 := \mathcal{S} \setminus \mathcal{S}_0$

$$\mathcal{S}_1 := \{a \in \mathcal{S} \mid i_{a,0} = 1 \ \forall a\}, \tag{13}$$

of $N_1 = N - N_0$ devices observing a reliable connection with the Gateway can be assigned either as Repeaters or EDs. The optimal configuration of devices in subset $\mathcal{S}_1$ is carried out in the following steps.

Feasibility Region for the Connectivity Problem. Assuming all nodes $b \in \mathcal{S}_1$ be initially configured as Repeaters, a solution to the connectivity problem satisfying Rule 3 for the EDs $a \in \mathcal{S}_0$ without a reliable direct connection with the Gateway exists if

$$\sum_{b \in \mathcal{S}_1} i_{a,b} > 0, \quad \forall a \in \mathcal{S}_0, \tag{14}$$

such that all the EDs $a \in \mathcal{S}_0$ can exploit an alternative two-hop link through one Repeater $b \in \mathcal{S}_1$ serving as a range extender. In case the condition is not satisfied, the number of candidate points is not sufficient for a feasible solution to the coverage problem; additional candidate sites must be therefore identified. New candidate sites must be assigned during the precommissioning of the plant; the deployment should thus account for application and site-specific environmental constraints.

Repeater Configuration. Among the $K = \sum_{n=1}^{N_1} \binom{N_1}{n}$ potential subsets $\mathcal{R}_k \subseteq \mathcal{S}_1$ of devices configured as Repeaters, with $k = 1, \ldots, K$, the optimal subset is defined as the one satisfying the feasibility region (14) and with the smallest cardinality. By letting $|\mathcal{R}_k|$ be the cardinality of the kth subset $\mathcal{R}_k$, the algorithm identifies the optimal $\overline{k}$th subset of the Repeater devices $\mathcal{R}_{\overline{k}} \subseteq \mathcal{S}_1$ such that

$$\begin{aligned} \mathcal{R}_{\overline{k}} &:= \arg\min_{k} \left|\mathcal{R}_k\right| \\ \text{s.t.} &\sum_{b \in \mathcal{R}_k \subseteq \mathcal{S}_1} i_{a,b} > 0, \quad \forall a \in \mathcal{S}_0. \end{aligned} \tag{15}$$

The devices $b \in \mathcal{R}_{\overline{k}}$ are thus configured as Repeaters while the other devices $a \in \mathcal{S} \setminus \mathcal{R}_{\overline{k}}$ take the role of EDs. Notice that $\mathcal{S}_0 \subseteq \mathcal{S} \setminus \mathcal{R}_{\overline{k}}$.

The iterative algorithm described as follows is used to find a solution to problem (15). Let the ordering of the Repeater subsets be such that $\forall k \ |\mathcal{R}_k| \geq |\mathcal{R}_{k+1}|$; the algorithm starts by picking the largest feasible set of Repeaters, so that $\mathcal{R}_1 \equiv \mathcal{S}_1$ and iteratively identifies new feasible subsets $\mathcal{R}_k \subset \mathcal{S}_1$ with smaller cardinality ($k > 1$) by randomly removing nodes from $\mathcal{S}_1$. The optimal subset $\mathcal{R}_{\overline{k}}$ solution to (15) is such that any smaller subset of Repeaters $\mathcal{R}_h$ with $h > \overline{k}$ is not feasible as

$$\prod_{a \in \mathcal{S}_0} \sum_{b \in \mathcal{R}_h} i_{a,b} = 0, \quad \forall h > \overline{k}, \tag{16}$$

or, equivalently, for any Repeater subset $\mathcal{R}_h$ with smaller cardinality $\mathcal{R}_h \subset \mathcal{R}_{\overline{k}}$ the sum $\sum_{b \in \mathcal{R}_h} i_{a,b} = 0$ for some ED $a \in \mathcal{S}_0$ without reliable direct connection.

6. Experimental Activity

The experimental validation of network connectivity is based on the link classification and channel modeling described in Sections 3 and 4. The optimization tool used for the optimal selection of the Repeater nodes is described in Section 5. The connectivity design consists of three steps. At first, the candidate positions for the wireless devices are chosen to highlight practical cases of meaningful interest for the deployment of an industrial network. The Gateway node is mounted above ground (according to Rule 2 in Section 5.2) and collects the data received from all the EDs. Second, the pairwise link RSSs are predicted based on channel modeling and classification as outlined in Sections 3 and 4. Finally, the optimal sub-set of the wireless devices that should act as Repeaters is computed based on the connectivity optimization tool illustrated in Section 5.3.

In the proposed experimental set-up, we deployed absolute and gauge pressure transmitters communicating with a Gateway node by star or two-hop mesh topology. Compared to mesh topology, deploying a star topology network should be preferred in practice as it provides better performance in terms of per-link real-time responsiveness that is required for monitoring and control of critical plant parameters. The radio transceivers conform with the ISA SP100.11a protocol [10] with radio transmit power set to $P_T = 11.6\,\text{dBm}$. The experiments have been carried out in two sites within the same oil refinery: the first site is a 100 m × 200 m area around a flare unit; the second one is a 60 m × 30 m area surrounding a furnace structure. All the environments under consideration are characterized by metallic objects and concrete buildings with high-reflectivity surfaces. Before the test, we used a signal analyzer to characterize the interferers in the area. Since no significant activity was detected, the IEEE 802.15.4 channels selected for the experiments have center frequencies 2.405 GHz and 2.480 GHz, corresponding to the ISA SP100.11a channel numbers 1 and 15, respectively.

The static RSS component g_ℓ in (2) characterizing the radio propagation over each link is predicted by following three steps.

(i) *Step 1*. The number and size of the objects blocking the direct path between the transmitter and receiver pair (or the corresponding Fresnel volumes) are identified by analyzing the 3D model of the plant.

(ii) *Step 2*. The link is classified by exploring the 3D maps of the corresponding sites. Based on the link types identified in Section 4, the size of the obstructions is compared with the Fresnel volumes to identify the corresponding clearance zones $\mathcal{F}_i$ for the obstacles with relevant size compared to the wavelength λ. The link category is then selected by comparing the resulting clearance zones with the ones characterizing each link type.

(iii) *Step 3*. The static RSS component g_ℓ is predicted according to the chosen link type. The signal attenuation $\sigma = \sigma(\ell)$ in (2) for the chosen link category ℓ is computed based on the predicted field loss E/E_{free} as in (5). The distance-dependent loss factor $g_0(d, \alpha)$ is defined according to the position of the transmitter and receiver devices. In all of the considered short range cases for $d < d_F$, the path loss exponent is $\alpha = 2$. The propagation over long ranges such that $d > d_F$ suffers from a larger path loss due to ground reflections: for a typical case of $d_F \simeq 50\,\text{m}$ (Gateway at 6 m from the ground), the available measurements indicate a path loss exponent of $\alpha = 2.5$.

The measurements analyzed in the following sections highlight the accuracy of the proposed channel characterization and modeling approach. Tightness of the proposed model is verified by comparing the predicted RSSs with the corresponding measurements obtained during the on-line testing. The model accuracy is found as reasonably high in all the considered settings (with errors below 4 dB for all the considered cases).

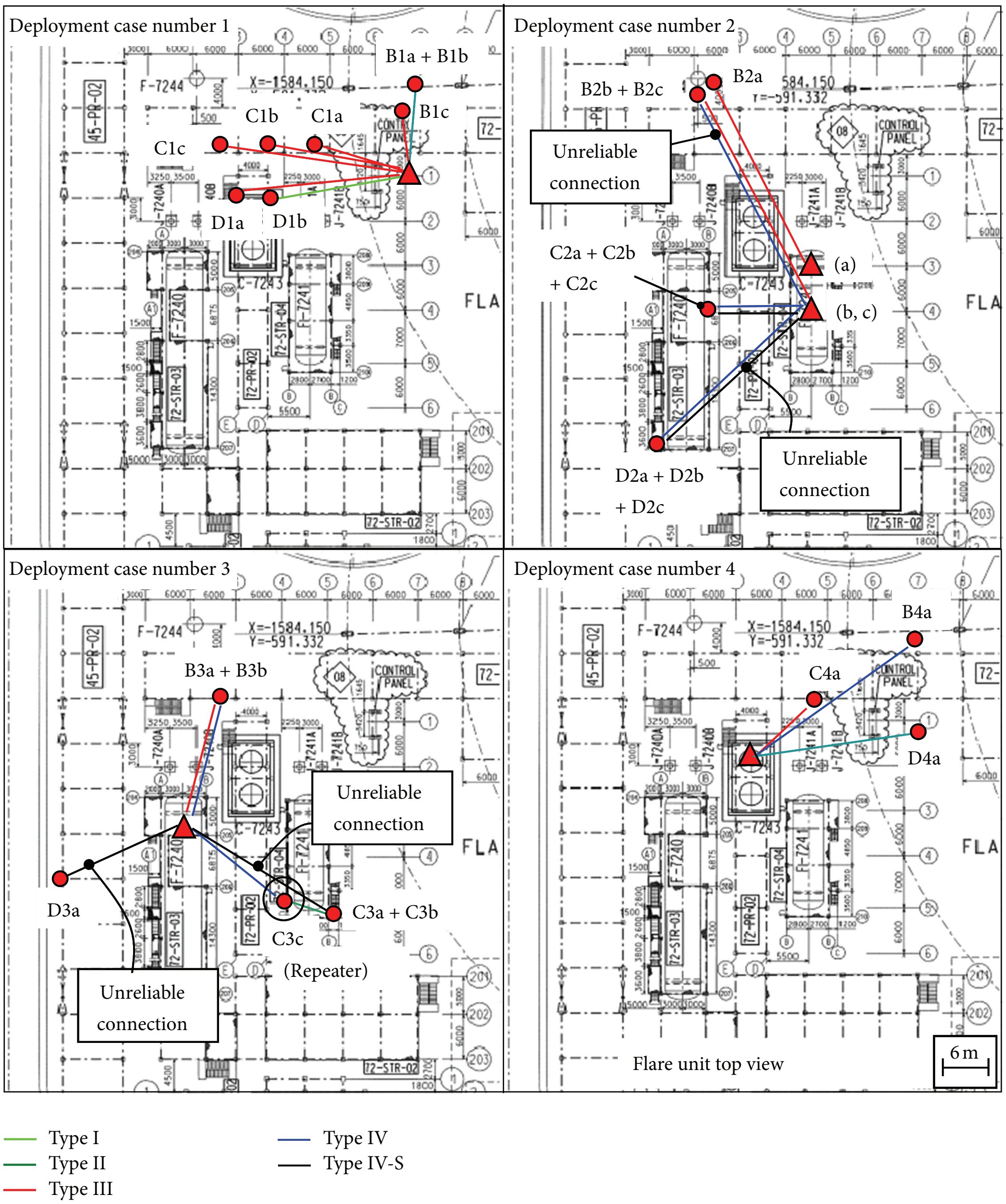


Figure 5: Flare unit test sites and link classification according to the categories defined in Section 4. Links are colored based on the selected link type; unreliable links are also highlighted.

6.1. Site Test No. 1: Flare Unit. In this test, the Gateway is mounted in 4 different locations corresponding to different deployment cases as illustrated in the floor plan maps of Figure 5. For deployment case no. 1 the height from the ground of the Gateway is 1.5 m ($d_F \simeq 25$ m); for the remaining cases; the height is above 6 m ($d_F \simeq 50$ m). For all cases, the 3 EDs labelled as B, C, and D are acting as input/output field devices and are moved in different positions labeled by lowercase letters (a, b, and c). The corresponding RSS measurements are reported by circle markers for all the deployment cases and analyzed in Figure 6 for devices at ground level and in Figure 7 for devices at 1 m above ground. The markers have different colors to identify the link category while the link classification is based on the inspection of 2D and 3D-CAD maps. The predicted static RSS component g_ℓ is represented by solid lines as a function of the distance d and for each link category (see Section 4). The same color code used for the measurements is adopted to highlight the prediction and link classification accuracy.

The effectiveness of the proposed channel characterization and modeling approach can be appreciated in several settings as highlighted in Figure 8. To focus on a relevant example, in the deployment case 3, the ED transmitters located at positions C3a (ground level) and C3b (1 m height

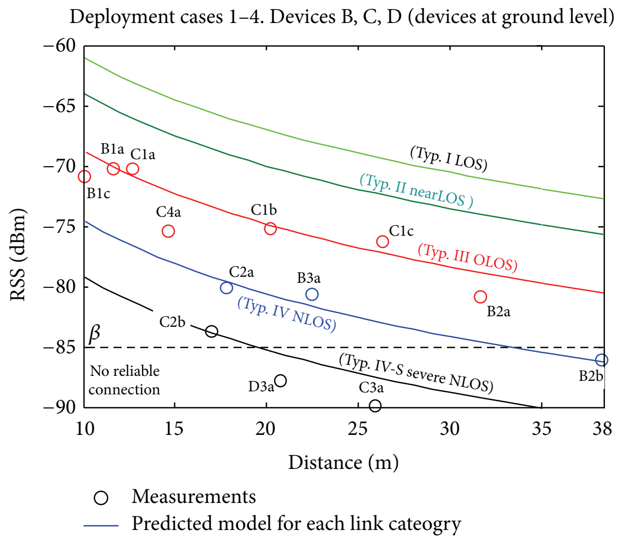


Figure 6: RSS measurements (circle markers) for devices B, C, and D over the flare unit sites (1–4) at ground level. Positions of devices are indicated by lowercase letters and correspond to the maps in Figure 5. Colors identify the link types; the predicted model for each link category is superimposed by solid lines, using the same color code.

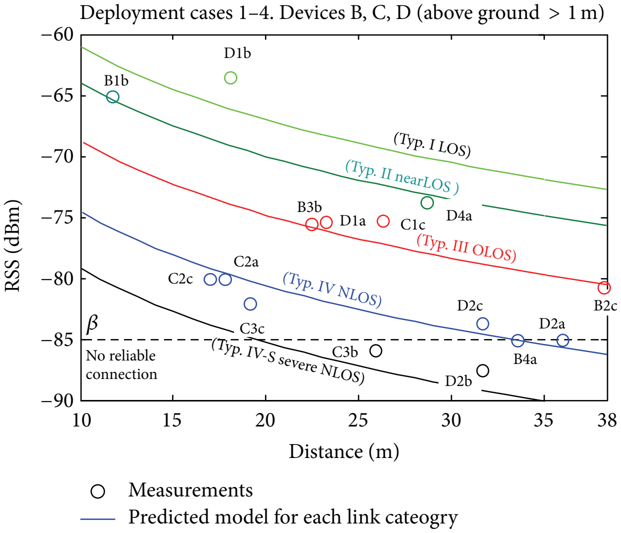


Figure 7: RSS measurements (circle markers) for devices B, C, and D over the flare unit sites (1–4) located at 1 m above ground. Positions of devices are indicated using lowercase letters and correspond to the maps in Figure 5. Colors identify the link types; predicted model is illustrated using the same color code.

from the ground) are hidden behind a big cylindrical vessel that completely obstruct the 1st Fresnel region. The wireless links connected to the Gateway retain the $E/E_{\text{free}} = 13\%$ and the $E/E_{\text{free}} = 9\%$ of the received field that would be measured in free space, respectively. Therefore, they can be reasonably classified as Type IV-S. As confirmed by measurements, the predicted RSS is below the critical $\beta = -85$ dBm reliability threshold (distance $d = 26$ m) suggesting the need for a Repeater device acting as relay. The same transmitter is now moved at position C3c to circumvent the large obstruction and create more favorable propagation conditions. In this case, by analyzing the corresponding 3D map, the 1st Fresnel region is slightly unobstructed: the link retains a larger fraction ($E/E_{\text{free}} = 17\%$) of the received electric field and thus can be reasonably classified as Type IV. As confirmed by the chosen model, the connection with the Gateway is now reliable as RSS −82 dBm: this suggests to deploy a Repeater device at position C3c multiplexing the source data received from the devices obstructed by the cylindrical vessel at position C3a and C3b. As confirmed by analysis of the 3D model, the links connecting the Repeater with devices located at the other side of the vessel can be classified as Type II, being the forbidden region free from obstacles.

Figure 8 highlights other relevant deployment cases: the links connecting the Gateway with the EDs at position B4a and C4a are classified as Type IV ($E/E_{\text{free}} = 20\%$) and Type III ($E/E_{\text{free}} = 32\%$), respectively. For both links the forbidden region is found as partially obstructed: in addition, at position B4a, the LOS path is blocked by concrete and metallic structures located around the corresponding ED location. For position D4a instead, the forbidden region is found as unobstructed; the corresponding link can be thus classified as Type II ($E/E_{\text{free}} = 63\%$).

6.2. Site Test No. 2: Furnace Structure. In this test, the Gateway is mounted on the stairway in the south-east of the furnace at 10 m above the ground level. In this scenario, devices C and D are moved over four different floors of the furnace structure according to Figure 9. Device B instead is located at ground level, moved in 5 positions in front of the furnace structure. The distance between each device and the Gateway ranges between 14 m and 57 m and is lower than the Fresnel distance $d_F \simeq 80$ m in all cases. Measurements and predicted model for each link category are reported in Figure 10 using the same color code adopted for the flare unit scenario. By exploring the 2D and the 3D maps of the site, the links corresponding to positions B5 (d and e), D5e, and C5a can be reasonably classified as Type III ($E/E_{\text{free}} = 40\%$ or $\sigma(\ell = 3) \simeq 8$ dB), being the forbidden region (see Figure 4) partially obstructed. Measured attenuations are $\sigma = 5 \div 10$ dB and confirm this choice. As highlighted in Figure 11, the NLOS links (Type IV) correspond to positions B5c, D5a, and C5e with observed attenuation ranging from $\sigma = 11 \div 17$ dB. For positions D5d (4rd floor) and C5d (3rd floor), the metallic structure produces a waveguide effect on propagation such that reliable communication occurs even across the whole furnace structure. The wireless signals propagate all around the furnace environment without obstacles and take advantage of the constructive interference. The positions D5b (2rd floor) and C5b (1rd floor) are instead surrounded by the furnace building that fully obstructs the 1st Fresnel volume and absorbs approximately the 84% and the 88% of the free-space field intensity, with $E/E_{\text{free}} = 16\%$ and $E/E_{\text{free}} = 12\%$, respectively (Type IV-S). As confirmed by measurements, the predicted RSS for Type IV-S link (with distance $d = 57$ m) is below the critical $\beta = -85$ dBm reliability threshold. The installation of one Repeater device located in the example at position C5a is therefore the optimal choice to relay the data

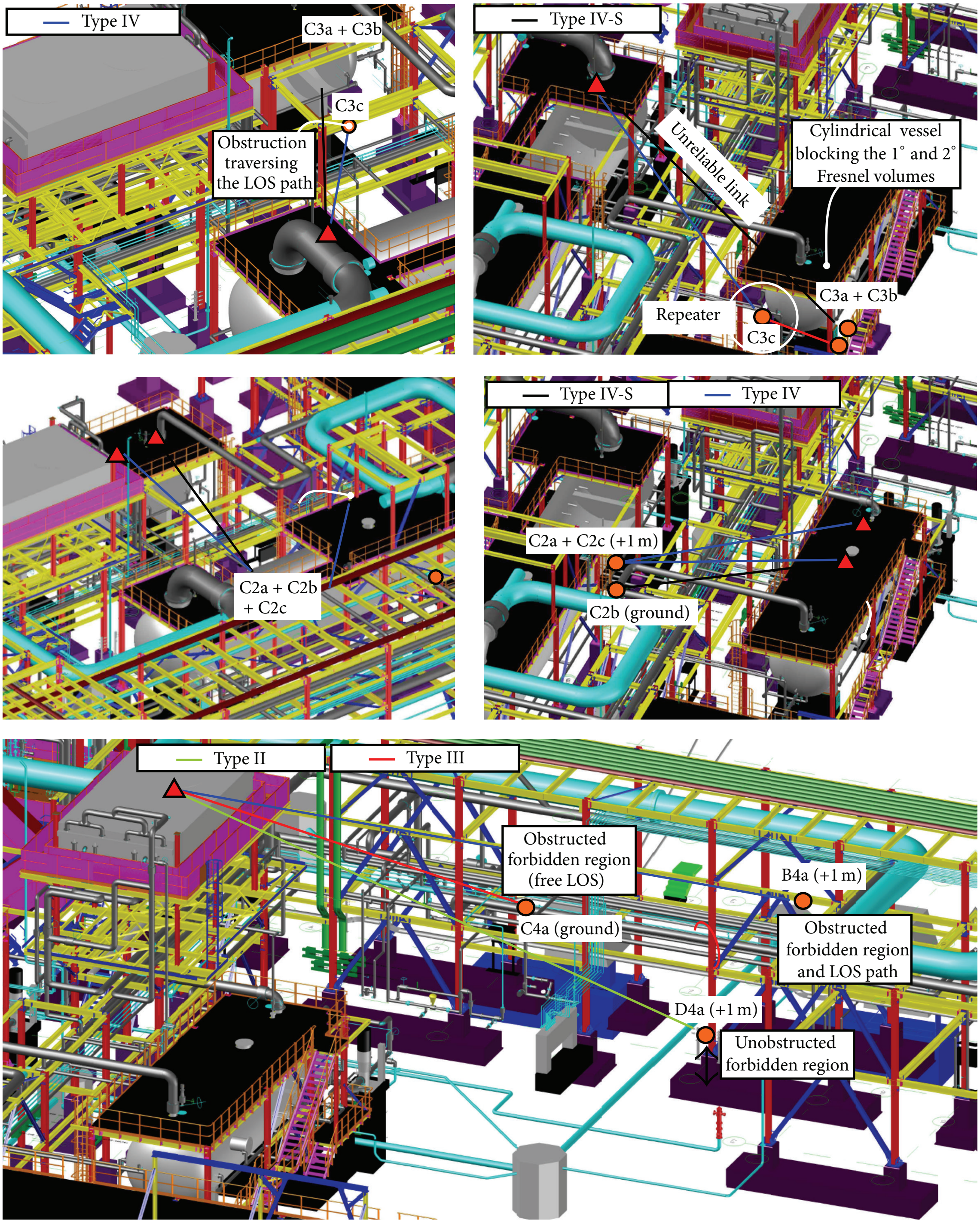
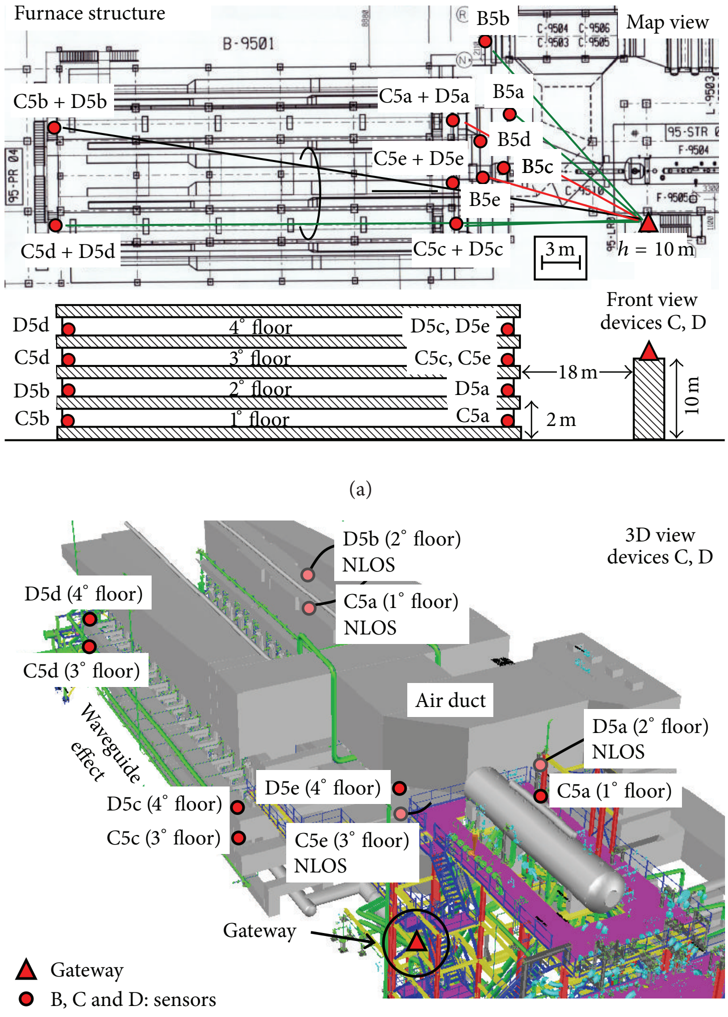

Type IV
C3a + C3b
C3c
Obstruction traversing the LOS path
Type IV-S
Unreliable link
Cylindrical vessel blocking the 1° and 2° Fresnel volumes
C3a + C3b
Repeater
C3c
C2a + C2b + C2c
Type IV-S
Type IV
C2a + C2c (+1 m)
C2b (ground)
Type II
Type III
Obstructed forbidden region (free LOS)
C4a (ground)
B4a (+1 m)
Obstructed forbidden region and LOS path
D4a (+1 m)
Unobstructed forbidden region

FIGURE 8: Flare unit scenario: relevant deployment example cases.

acquired by the measurement points located at positions D5b and C5b.

6.3. Long-Range Testing. Although the focus of this paper is mostly on short-range networking modeling and optimization of network deployment in industrial environments, a long-range test have been also carried out as depicted in Figure 12 (deployment case 5) with the Gateway located in the same position of case 4 while the device C at ground level has been moved in two sites. The first one is an open area classified as near LOS environment (Type II) on the right side of the flare unit at distance 109 m from the Gateway, the second site was located at distance 132 m from the Gateway in the southern part of the flare unit where the LOS path is obstructed by a building. The path loss g_0 caused by the ground reflections (flat terrain) can be reasonably modeled



(a)

(b)

Figure 9: Furnace structure test site: top and front view (a) and 3D view (b) of the environment.

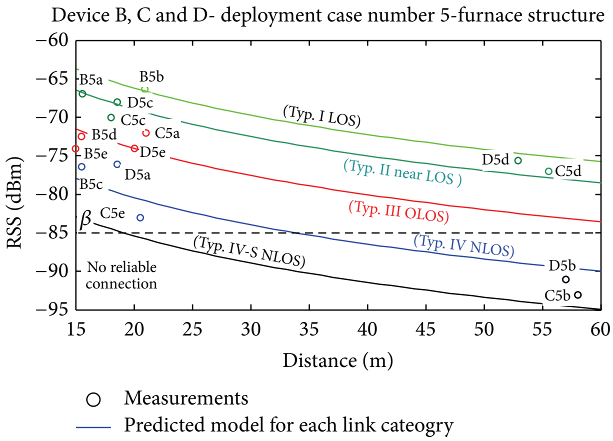


Figure 10: RSS measurements (circle markers) and predicted model (solid lines) for the furnace site. Same color code as for Figure 9.

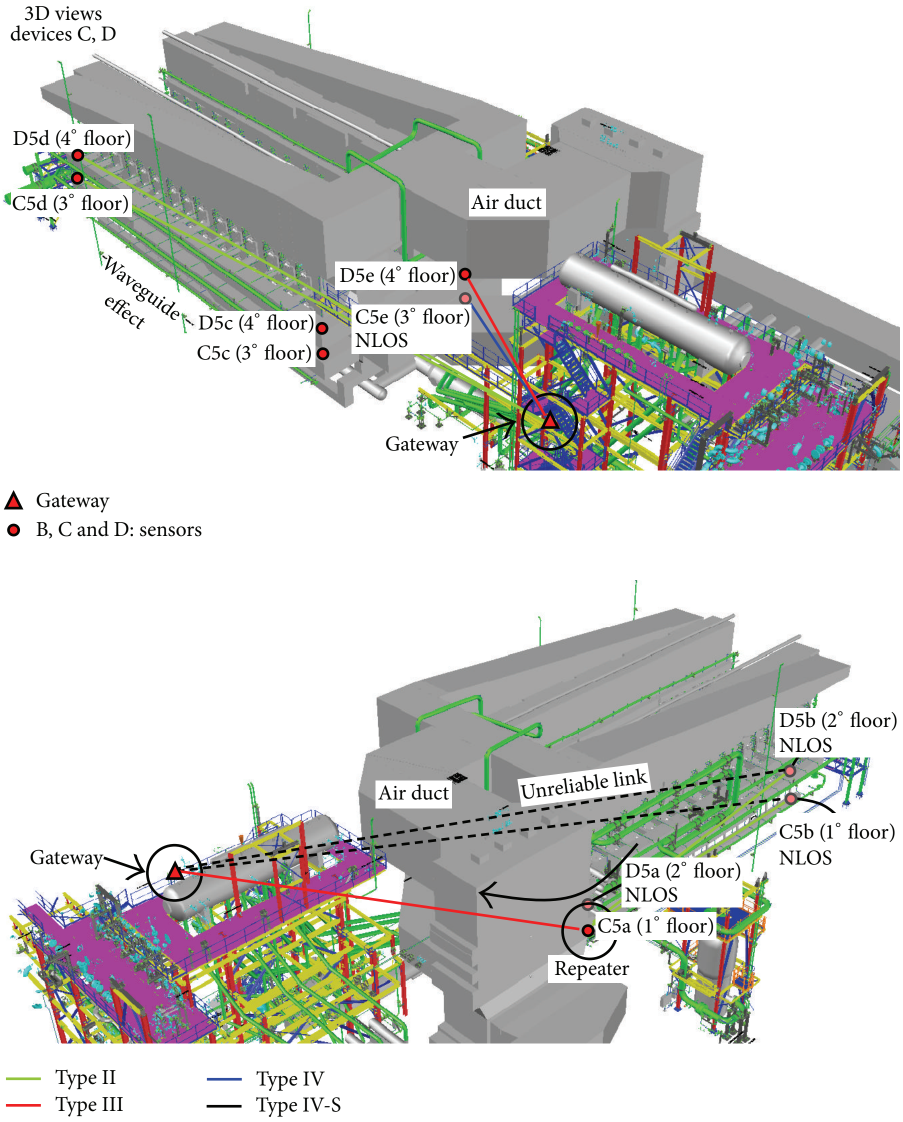
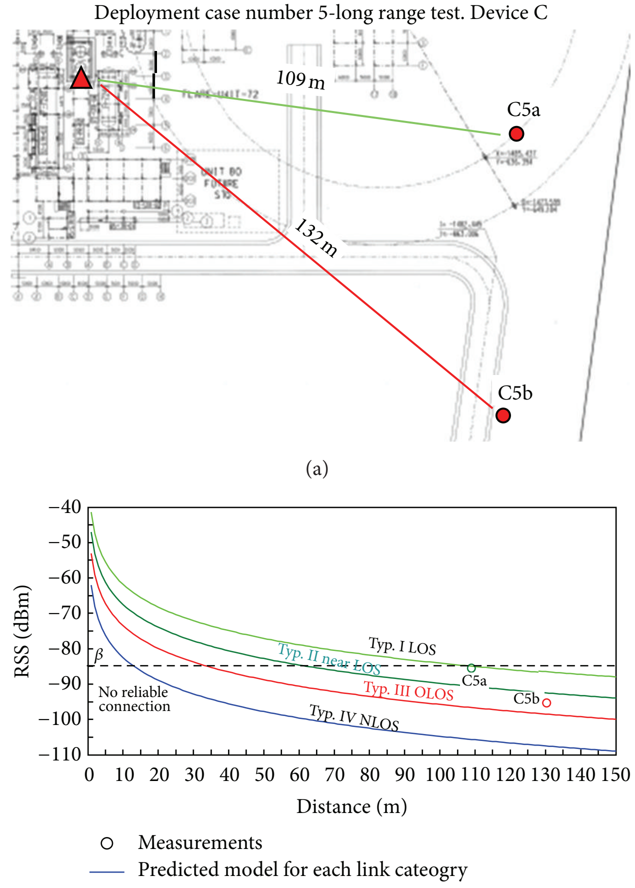


FIGURE 11: Furnace site scenario: deployment example cases.

as in (2) with exponent $\alpha = 2.5$. For the first test, the wireless link is characterized by $E/E_{\text{free}} = 83\%$. The measured RSS of −85 dBm confirms the predicted range for the corresponding Type II link category (see Section 3). In the second test, the link only retains the $E/E_{\text{free}} = 44\%$ of the free-space electric field (the attenuation caused by the building is 7 dB), and it is classified as unreliable with RSS of −91 dBm.

7. Concluding Remarks

Network deployment in industrial settings with dense metallic structures can be based on a simple but effective channel model that makes use of the diffraction theory for 3D environments. The model proves to characterize the wireless propagation in industrial environments with an accuracy that is reasonably high to predict the average quality of the wireless links in different sites. The wireless links are partitioned into mutually exclusive attenuation classes (link types) based on the 3D structure of the building blockage. Each class is characterized by a different amount of obstruction loss; therefore, a separate channel model is proposed to predict the QoS for each link type. The diffraction model is then adopted for virtual planning of two-hop ISA networks: the problem of optimal Repeater configuration of the Repeater devices is addressed to guarantee reliable connectivity between the end devices and the Gateway. The proposed classification approach has been validated by extensive experimental measurements in critical areas within an oil refinery plant characterized by highly dense metallic structure. Industry ISA SP100.11a standard devices operating at 2.4 GHz are adopted. Experimental results from the surveys confirm the effectiveness of the proposed method as it provides a practical tool for virtual network planning with reasonable accuracy that meets the expectations in several industrial settings.



(a)

(b)

Figure 12: Long-range testing for two sensors: layout (a), measurements (b).

Appendix

For the symmetric rectangular obstacle case, the loss term in (6) simplifies as

$$\left|\frac{E\left(q_i\right)}{E_{\text{free}}}\right| \simeq \left|1-2j\int_0^{\sqrt{2}b_i/r_1(q_i)}\exp\left[-j\pi\frac{y_1^2}{2}\right]dy_1 \times \int_0^{\sqrt{2}a_i/r_1(q_i)}\exp\left[-j\pi\frac{x_1^2}{2}\right]dx_1\right|, \tag{A.1}$$

where we used the substitutions $x_1 = \sqrt{2}x/r_1(q_i)$ and $y_1 = \sqrt{2}y/r_1(q_i)$. Using an asymptotic expansion [42] for the integrals in the form $\int_0^x \exp[-j\pi(x^2/2)]dx$ valid for large enough x

$$\int_0^x \exp\left[-j\pi\frac{x^2}{2}\right]dx \approx \Gamma\left(x\right), \tag{A.2}$$

with $\Gamma(x)$ defined in (8), the loss term can be written now as in (7).

Acknowledgment

This work has been performed in the framework of the European Research Project DIWINE (Dense Cooperative Wireless Cloud Network) under FP7 ICT Objective 1.1—The Network of the Future.

References

[1] D. Zuehlke, "Smart factory—towards a factory-of-things," *Annual Reviews in Control*, vol. 34, no. 1, pp. 129–138, 2010.

[2] S. Savazzi, U. Spagnolini, L. Goratti, D. Molteni, M. Latva-aho, and M. Nicoli, "Ultra-wide band sensor networks in oil and gas explorations," *IEEE Communications Magazine*. In press.

[3] A. Willig, "Recent and emerging topics in wireless industrial communications: a selection," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102–124, 2008.

[4] S. Savazzi, S. Guardiano, and U. Spagnolini, "Wireless critical process control in oil and gas refinery plants," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '12)*, pp. 1003–1008, Athens, Greece, March 2012.

[5] Standard IEEE 802.15.4-2006, "Part 15.4: Wireless Medium Access Control (MAC) and Physical layer (PHY) specifications for low-rate Wireless Personal Area Networks," 2006.

[6] L. Tang, K.-C. Wang, Y. Huang, and F. Gu, "Channel characterization and link quality assessment of IEEE 802.15.4-compliant radio for factory environments," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 99–110, 2007.

[7] V. Erceg, L. J. Greenstein, S. Y. Tjandra et al., "An empirically based path loss model for wireless channels in suburban environments," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 7, pp. 1205–1211, 1999.

[8] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela, and M. H. Wright, "WISE design of indoor wireless systems: practical computation and optimization," *IEEE Computational Science and Engineering*, vol. 2, no. 1, pp. 58–68, 1995.

[9] J. Lei, L. Greenstein, and R. Yates, "Link gain matrix estimation in distributed wireless networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 848–852, New Orleans, La, USA, December 2008.

[10] Standard ISA100.11a-2009, "Wireless systems for industrial automation: process control and related applications," ISA, July 2009.

[11] M. Baldi, R. Giacomelli, and G. Marchetto, "Time-driven access and forwarding for industrial wireless multihop networks," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 2, pp. 99–112, 2009.

[12] D. Christin, P. S. Mogre, and M. Hollick, "Survey on wireless sensor network technologies for industrial automation: the security and quality of service perspectives," *Future Internet*, no. 22, pp. 96–125, 1999.

[13] G. Cena, I. C. Bertolotti, A. Valenzano, and C. Zunino, "Evaluation of response times in industrial WLANs," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 3, pp. 191–201, 2007.

[14] P. B. Sousa and L. L. Ferreira, "Hybrid wired/wireless profibus architectures: performance study based on simulation models," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, Article ID 845792, 25 pages, 2010.

[15] J. Song, S. Han, A. K. Mok et al., "WirelessHART: applying wireless technology in real-time industrial process control,"

in *Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '08)*, pp. 377–386, April 2008.

[16] D. J. Y. Lee and W. C. Y. Lee, "Propagation prediction in and through buildings," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 5, pp. 1529–1533, 2000.

[17] S. R. Saunders and A. A. Zavala, *Antennas and Propagation for Wireless Communications Systems*, Wiley, New York, NY, USA, 2nd edition, 2007.

[18] W. C. Y. Lee and D. J. Y. Lee, "Microcell prediction in dense urban area," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 1, pp. 246–253, 1998.

[19] H. Mokhtari and P. Lazaridis, "Comparative study of lateral profile knife-edge diffraction and ray tracing technique using GTD in urban environment," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 1, pp. 255–261, 1999.

[20] A. Nyuli and B. Szekeres, "An improved method for calculating the diffraction loss of natural and man made obstacles," in *Proceedings of Personal, Indoor and Mobile Radio Communications*, pp. 426–430, Boston, Mass, USA, October 1992.

[21] C. L. Giovanelli, "An analysis of simplified solutions for multiple knife-edge diffraction," *IEEE Transactions on Antennas and Propagation*, vol. 32, no. 3, pp. 297–301, 1984.

[22] G. E. Athanasiadou, "Incorporating the Fresnel zone theory in ray tracing for propagation modelling of fixed wireless access channels," in *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*, pp. 3–7, September 2007.

[23] C. F. Huang and Y. C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.

[24] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: a survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.

[25] R. Chandra, L. Qiu, K. Jain, and M. Mahdian, "Optimizing the placement of integration points in multi-hop wireless networks," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP '04)*, pp. 271–282, Berlin, Germany, 2004.

[26] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[27] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

[28] M. Ishizuka and M. Aida, "Performance study of node placement in sensor networks," in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW '04)*, vol. 7, pp. 598–603, Tokyo, Japan, March 2004.

[29] P. Cheng, C. N. Chuah, and X. Liu, "Energy-aware node placement in wireless sensor networks," in *Proceedings of the 47th IEEE Global Telecommunications Conference (GLOBECOM '04)*, pp. 3210–3214, Dallas, Tex, USA, December 2004.

[30] K. Xu, H. Hassanein, and G. Takahara, "Relay node deployment strategies in heterogeneous wireless sensor networks: multiple-hop communication case," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and AdHoc Communications and Networks (SECON '05)*, pp. 575–585, Santa Clara, Calif, USA, September 2005.

[31] M. Haghpanahi, M. Kalantari, and M. Shayman, "Topology control in large-scale wireless sensor networks between information source and sink," *Ad Hoc Networks*, 2012.

[32] J. Tang, B. Hao, and A. Sen, "Relay node placement in large scale wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 490–501, 2006.

[33] A. Konstantinidis and K. Yang, "Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D," *Applied Soft Computing*, vol. 12, no. 7, pp. 1847–1864, 2012.

[34] X. Han, X. Cao, E. L. Lloyd, and C. C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 5, pp. 643–656, 2010.

[35] Y. E. Osais, M. St-Hilaire, and F. R. Yu, "Directional sensor placement with optimal sensing range, field of view and orientation," *Mobile Networks and Applications*, vol. 15, no. 2, pp. 216–225, 2010.

[36] J. Lee, T. Kwon, and J. Song, "Group connectivity model for industrial wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1835–1844, 2010.

[37] T. M. Deyab, U. Baroudi, and S. Z. Selim, "Optimal placement of heterogeneous wireless sensor and relay nodes," in *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC '11)*, pp. 65–70, July 2011.

[38] F. M. Al-Turjman, H. S. Hassanein, and M. A. Ibnkahla, "Efficient deployment of wireless sensor networks targeting environment monitoring applications," *Computer Communications*, vol. 36, no. 2, pp. 135–148, 2013.

[39] X. Bai, C. Zhang, D. Xuan, and W. Jia, "Full-coverage and k-connectivity (k =14, 6) three dimensional networks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 388–396, April 2009.

[40] WirelessHART, "IEC 62591, System Engineering Guide," Revision 2, October 2010.

[41] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, "Gateway placement optimization in wireless mesh networks with QoS constraints," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 2127–2136, 2006.

[42] M. Abramovitz and I. A. Stegun, *Handbook of Mathematical Functions*, Applied Mathematics Series 55, Dover, New York, NY, USA, 1972.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 232798, 10 pages
http://dx.doi.org/10.1155/2013/232798

Research Article

A Multipath Routing Approach for Secure and Reliable Data Delivery in Wireless Sensor Networks

Hind Alwan and Anjali Agarwal

Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada H3G 1M8

Correspondence should be addressed to Hind Alwan; h_alwan@encs.concordia.ca

Received 28 September 2012; Revised 22 December 2012; Accepted 30 January 2013

Academic Editor: Marc St-Hilaire

The severe resource constraints and challenging deployment environments of wireless sensor networks (WSNs) pose challenges for the security and reliability of data transmission for these networks. In this paper, we present and evaluate a secure and reliable routing mechanism offering different levels of security in an energy-efficient way for WSNs. Our approach uses node-disjoint routing and the selection mechanism of these paths depends on different application requirements in terms of security. The original data message is split into packets that are coded using Reed-Solomon (RS) codes and, to provide diverse levels of security, different number of fragments is encrypted related to the requested security level before being transmitted along independent node-disjoint paths. This technique makes encryption feasible for energy-constrained and delay-sensitive applications while still maintaining a robust security protection. We describe how to find the secure multipath, the number of these paths, and how to allocate fragments on each path seeking to enhance security and improve data reliability. Extensive analysis and performance evaluation show that data transmission security and reliability can be enhanced while respecting the resource constraints of WSNs.

1. Introduction

Advances in wireless sensor networks have enabled a wide range of application across many fields. Many of these applications have high quality of service (QoS) requirements in terms of security and reliability of data transmission.

Wireless sensor networks (WSNs) are characterized by severe resource constraints of sensor nodes, unreliable nature of the wireless links, dynamic changing in the size and density of the network, and the high risk of physical attacks to sensors. Many routing protocols have been proposed to overcome these constraints and improve the QoS in wireless networks. However, most of the existing protocols provide either secure [1] or QoS [2–5] routing. Few protocols have combined these two requirements [6–9].

Secure multipath routing protocols in WSNs can be divided into three categories based on the security-related operational objective [1]. The multipath routing protection only, the attack-specific, and the security operations support. The security-based multipath routing protection protocol is the interest of this paper in which the multipath routing is used to improve the security, increase reliability of data transmission, provide load balancing, and decrease the end-to-end delay.

A common approach to provide reliability in WSNs is to use forward error correction (FEC) technique as a replication mechanism in multipath routing to increase data transmission reliability, decrease energy consumption, and increase the network lifetime while avoiding the costly or impossible data retransmission due to the severe resource constraints of sensor nodes [10]. However, this approach required sending more data than necessary over the multipath in order to tolerate a certain number of path failures.

This paper was motivated mainly by the observations that most traditional encryption algorithms are complex and may introduce a severe delay in sensor nodes. For instance, the encryption time of each 128-bit block using the AES algorithm is about 1.8 ms on a MicaZ platform [11]. Our approach therefore proposes to encrypt only a certain fraction of the RS [12] codewords while the remaining portion is transmitted unprotected. Our scheme makes encryption feasible for energy-constrained and delay-sensitive applications while still maintaining a robust security protection.

Our major contributions in this paper are the following. First, we introduce a new mechanism for secure and reliable data transmission in WSNs multipath routing, derived from node-disjoint multipath and combined with source coding in order to enhance both security and reliability of data transmission in the network. Second, we define different levels of security requirements and depending on these requirements, a selective encryption scheme is introduced to encrypt selected number of coded fragments in order to enhance security and thereby reduce the time required for encryption. Finally, an allocation strategy that allocates fragments on paths is introduced to enhance both the security and probability of successful data delivery.

The remainder of this paper is organized as follows. In the next section, we review the related work on secure and reliable multipath routing protocols. The routing problem metrics are formulated in Section 3. Section 4 provides a detailed description of the proposed secure mechanism. In Section 5, we describe our methodology for evaluating the security and reliability. A detailed case study is presented with different required security levels and possible attack scenarios. The simulation model and the performance evaluation are presented. Finally, we conclude our work in Section 6.

2. Related Work

In the literature, encryption techniques have been developed for secure multipath routing protocols in WSNs. In [1], an extensive survey has been conducted on the current state of the art for secure multipath routing protocols. The security-related issues, threats, and attacks in WSNs and some of the solutions can be found in [13].

One of the possible solutions to support secure and reliable data transmission is to combine multipath routing protocols with secret sharing algorithm. In (T, N) threshold secret sharing algorithm [14], the original data message is divided into N shares and sent to the destination over different paths. The original message can be reconstructed from any T shares, while no information about the original message can be obtained with less than T shares. The main drawback of using the secret sharing method is the large amount of traffic and redundancy involved. H-SPREAD [6] protocol is proposed as an extended version of SPREAD protocol [7] which used multipath between a single source-destination pair to deliver multiple secret message shares in order to enhance the data confidentiality in mobile ad hoc networks. H-SPREAD proposed for WSNs a distributed many-to-one multipath discovery protocol by employing two phases of flooding in order to enhance the security and reliability of data transmission. To enhance reliability, H-SPREAD uses an active per-hop packet salvaging strategy; the sender forwards the packet over another path instead of dropping it when unsuccessful transmission occurs to increase the probability that the data packet is delivered to the sink. Although, H-SPREAD protocol provides security in terms of resilience against node capture, it does not provide any authentication mechanism. Thus, many network layer attacks such as Sinkhole or Wormhole on routing protocols that attract traffic by advertising high-quality route to the sink are related with the goal of affecting the construction of paths. Furthermore, the construction of the spanning tree used in this protocol introduces high overhead.

Other possible solutions to support secure and reliable data transmission is the combination of data encryption and FEC technique [8, 9]. The main concept of this combination is to encrypt the original data message, encode the encrypted message using FEC coding, and then route it to the destination. A secure, multiversion, multipath protocol, MVMP, is proposed in [9] to offer a secure and reliable data communication in WSNs. MVMP consists of four steps: divide the original data message into groups, encrypt each group using different cryptographic algorithms, code the encrypted packets using RS codes, and transmit the coded packets on multiple disjoint paths that are assumed to be established before the data transmission. The data packet can be compromised when certain amount of codewords over different paths are intercepted and all the encryption algorithms used for the transmission are known. Moreover, to reconstruct the original message, the attacker needs to make all possible packet combinations, which is a resource challenging task. Although MVMP protocol uses different cryptographic algorithms in order to enhance data transmission security; this strategy could be expensive in resource-constrained environments such as WSN.

In [15], a secure and reliable node-disjoint multipath routing protocol is proposed in order to minimize the worst case security risk and to maximize the packet delivery ratio under attacks. The multipath routing problem is modeled as an optimization problem and solved by a heuristic algorithm using game theory, and a routing solution is derived to achieve a tradeoff between route security and delivery ratio in worst scenarios. The protocol focuses on the worst case attack scenarios to achieve the design objective of providing the best security and/or delivery ratio. Although the protocol assumes using link reliability history in the computations, in WSN the sensors and the communication links change frequently and are time varying. This required a frequent update of the computation of paths to discover the most reliable and secure paths. Also, the protocol assumes that each node has a full knowledge of the whole network topology which is considered an expensive assumption in WSN.

An intrusion-fault tolerant routing scheme proposed in [16] offers a high level of reliability by a secure multipath routing construction topology and uses one-way hash chains to secure the construction of a multipath, many-to-one dissemination topology.

A secure and energy-efficient multipath routing protocol for wireless sensor networks is proposed in [17]. Disjoint and braided paths are constructed using a modification of the breadth first search algorithm. The sink executes the paths discovery, selection, and maintenance in a centralized way. The authors claim that network layer attacks such as Sinkhole and Wormhole are not related since routing paths are selected by the sink node and periodically changed to prolong the lifetime of the network. Also, the protocol addresses the replayed attack by having each packet identified by a unique sequence number to be transmitted only once. However, the protocol does not use any encryption and authentication mechanism to protect against a number of attacks; this means that an attacker can affect the paths construction process. Moreover, the sink needs to have information of the whole network topology which requires that each node sends its neighbors list to the sink, and this process consumes huge energy and introduces extra overhead.

Enhancing data security in ad hoc networks based on multipath routing is proposed in [18], which is designed on the multipath routing characteristics of ad hoc networks and uses a route selection based on the security costs without modifying the lower layer protocols. The authors claim that the proposed protocol can be combined with solutions which consider security aspects other than confidentiality to improve significantly the efficiency of security systems in ad hoc networks. The protocol in [18] is designed for an ad hoc network where the number of nodes in the network is considerably low and the capability of node is usually better than that of sensor networks. Thus, the protocol cannot directly fit the properties of sensor networks.

Our work differs from the above existing schemes by considering different levels of security requirements to encrypt limited number of packets contingent to these requirements in order to enhance data transmission security at lower cost than full packet encryption. The new mechanism proposed adapts to the resource constraints of WSNs by combining FEC technique and selective cryptographic algorithms to achieve secure and reliable data transmission in an energy-efficient way for WSNs. Unlike [9], the original message is split into packets that are first coded using RS codes. Then depending on the required security level, the selective encryption scheme is used to encrypt a selected number of coded fragments before being transmitted along different disjoint paths. Thus, the security can be achieved while respecting the resource constraints of WSNs.

3. QoS Routing Problem Formulation

3.1. Replication and Erasure Coding. Erasure coding has been used in distributed systems to achieve load balancing and fault tolerance, but recently [10] it has been used for WSNs as a replication mechanism in multipath routing to increase the data transmission reliability while decreasing energy consumption and increasing network lifetime. The advantage of using data replication is to avoid the costly or impossible data retransmission in WSNs due to the severe resource constraints of sensor nodes. RS code is the simplest and the widely used FEC codes for achieving reliable data transmission in networks.

In the network layer, we assume that there are totally n available disjoint paths between the source node and the sink. Only the source node and the sink are active participants in the coding/decoding process while no processing is needed at the intermediate nodes. Using RS codes, the source node codes each data packet of size Mb bits it receives into M fragments each of size b bits and generates another K parity fragments to have in total a set of $M + K$ fragments. If the sink receives any M fragments, it can recover the original data packet allowing at most K lost fragments. Denote the fragments allocation as $X = [x_1, x_2, \dots, x_n]$, where x_i is an integer and is the number of fragments allocated to path$_i$ and n is the number of node-disjoint paths from source node to sink, as shown in Figure 1 [10]. The allocation of fragments on each path is determined with a load balancing algorithm where $\sum_{i=1}^{n} x_i = M + K$. The value of K determines the loss recovery capability of the code. Given a fixed value of $M + K$, smaller M means less data information and more redundancy contained in each encoded block, thus the loss, recovery capability is better. If z_i is a random variable that indicates the number of fragments received on path$_i$, then we have $\sum_{i=1}^{n} z_i \geq M$. Typically, the code rate is $\lambda = M/(M+K)$, the redundancy ratio is $r = K/(M + K)$, the maximum codeword length for a RS code is $c = 2^b - 1$, and the coding overhead is $h = K/M$.

3.2. Security. A path is compromised when one or more node in the path is compromised. In this paper, node-disjoint paths are used; vthus the probability of compromising of a single path is not correlated with the probability of compromising of other paths. We assume that the source node and the sink are trustworthy. The source node selects np paths out of the n node-disjoint paths to route the data packet to the sink. The probability that the data packet is compromised, P_{pkt}, is defined as

$$P_{\text{pkt}} = \prod_{i=1}^{np} P_{\text{path}_i}, \tag{1}$$

where P_{path_i} is the probability that path$_i$ is compromised and is given as

$$P_{\text{path}_i} = 1 - \prod_{u=1}^{l} \left(1 - p_u\right), \tag{2}$$

where p_u is the probability that a sensor node is compromised, $u \in l$, l is the number of sensor nodes on path$_i$ and $0 \leq P_{\text{path}_i} \leq 1$.

Note that the probability p_u indicates the security level of node u and could be estimated from the feedback of some security-monitoring software or hardware such as firewalls and intrusion detection devices [18].

The proposed mechanism uses RS coding to send the $M + K$ fragments on np node-disjoint paths. To improve the security of the data transmission consider the following.

(1) Allocate fragments on as many paths as possible in order to minimize the probability P_{pkt}. The total number of fragments for each packet is equal to np, that is $M + K = np$. In this case, one fragment is transmitted on each path. With such allocation, the probability that the data packet is compromised, P_{pkt}, is equal to the probability that M out of np paths are compromised, $P_{\text{pkt}} = \prod_{i=1}^{M} P_{\text{path}_i}$. Thus, the more paths are used, the less P_{pkt} is, and the better the security is, Figure 2.

However, this strategy could be expensive in resources constraint networks like WSNs since it introduces a large storage and communication overhead. Moreover, fragments might be dropped on some paths due to the error-prone nature of sensor nodes and wireless links and to reconstruct the original data packet, a minimum of M paths are needed

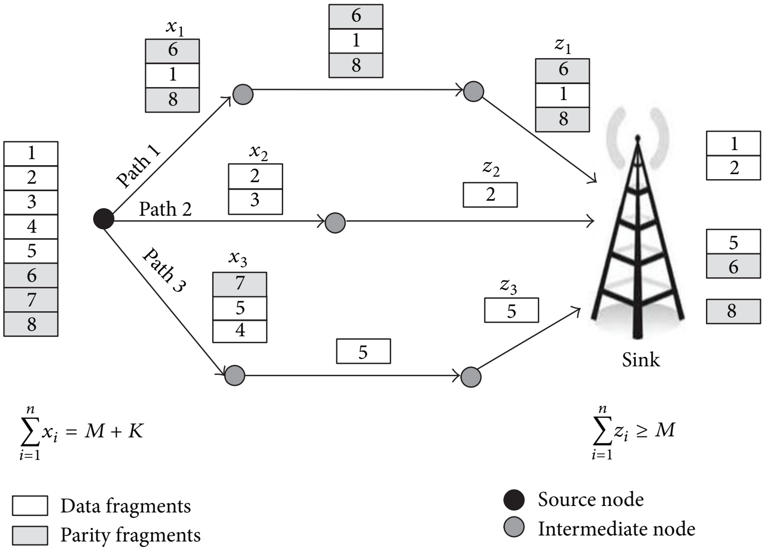


Figure 1: Example of data transmission using erasure coding [10]. Note that the data packet $M = 5$ fragments, the added redundancy $K = 3$ fragments and $n = 3$ paths.

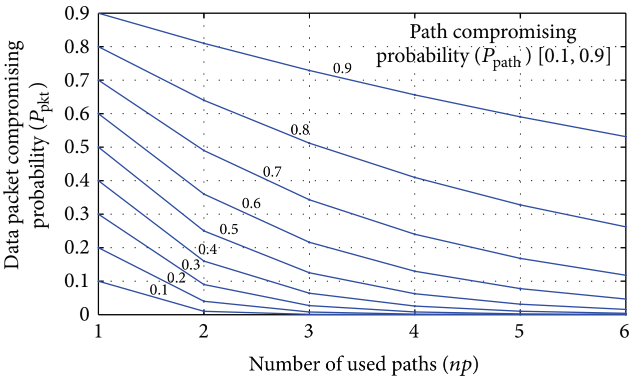


Figure 2: Relationship between data packet compromising probability, P_{pkt}, and the number of used paths, np, for different path compromising values, $P_{\text{path}_i}[0.1, 0.9]$.

to successfully deliver the required number of fragments to the sink.

(2) To achieve the highest security level, the allocated fragments on any path, x_i, should be less than M. With such allocation an attacker must intercept more than one path to get the M fragments required to reconstruct the data packet. The allocated fragments on each path should be as follows:

$$1 \leq x_i \leq M - 1. \tag{3}$$

This strategy is used in the proposed security mechanism.

(3) Minimize P_{path_i} such that P_{pkt} is minimized, (1). By using a path that contains as few nodes as possible, the shortest path and/or, path that contains the highest secure nodes among others minimizes P_{path_i}, (2).

3.3. Reliability. Multipath routing is one way of improving the reliability of data transmission by sending duplicated data via multiple paths. Thus, a packet is delivered to the destination even if some paths fail. The main drawbacks of the multipath routing are the higher energy consumption and the high probability of network congestion due to the increased number of messages which in turn impact the performance of the network. However, to improve the reliability of data transmission while respecting the network energy constraint, redundancy is applied using erasure coding on multipath routing. The idea is to send more fragments, $M + K$, than the minimum required fragments, M, to recover the original packet at the sink. In our proposed routing mechanism, the reliability of data transmission, the successful end-to-end data delivery, is achieved by sending the fragments of RS codeword on np selected node-disjoint multipath and to guarantee that the codeword packet is recoverable from any $\lceil np/2 \rceil$ paths, we need to ensure that fragments allocation on any $\lceil np/2 \rceil$ paths follows,

$$\sum_{i=1}^{\lceil np/2 \rceil} x_i \geq M. \tag{4}$$

3.4. Delay. The total path delay, D_{path}, includes the sum of time required for processing, queuing, transmission and propagation for all the nodes along the path. If coding and encryption are used, the path delay equals ($D_{\text{path}} + D_{\text{cod}} + D_{\text{enc}}$), where D_{cod} and D_{enc} are the coding time and the encryption time, respectively. D_{enc} is related to number of bits to be encrypted, n_{bit}, the unit-block encryption time, T_{blk}, and the encryption block size, L_{blk}, [19]. This is given as follows,

$$D_{\text{enc}} = \left(\frac{n_{\text{bit}}}{L_{\text{blk}}} \right) T_{\text{blk}}. \tag{5}$$

Request ID	Source ID	Sender ID	*hop*	P_{path}	S_{req}	Sink ID

(a)

Request ID	Source ID	Sender ID	No. of paths *np*	Sink ID

(b)

Figure 3: Control messages format (a) route request message, *RREQ*, (b) route reply message, *RREP*.

Encryption block size varies between different encryption algorithms and may also vary within the same encryption algorithm while the unit-block encryption time can be measured on specific platforms. Thus, choosing the appropriate block size as well as the total amount of bits to be encrypted can affect the delay performance of the network. Therefore, in our proposed selective encryption approach, a minimum amount of data is selected for encryption contingent to the security requirements. In this way, encryption time is reduced due to the need to encrypt fewer packets. Also, the energy required to encrypt the extra packets is conserved while still maintaining the required security level.

4. Proposed Protocol

An on demand routing protocol [20] is used to build multiple disjoint paths using route request/reply phases. Each sensor node is assumed to update the local states of its one-hop neighbors by broadcasting a *HELLO* message in which the links conditions are reported. Each node then maintains and updates its neighboring table information to record the link performance between itself and its direct neighbor nodes in terms of the probability that a sensor node is compromised, p_u. When the source node has data packet to transmit to the sink to which it has no available route, it starts the route discovery phase by transmitting a short route request message, *RREQ,* as shown in Figure 3(a). An *RREQ* message is broadcasted to all the neighbors of the source node within its transmission range, in which the required security level (in terms of message compromising probability), S_{req}, the path information (hop, P_{path}) are transferred to the sink. Each intermediate node updates the information of its one-hop local states, including the path compromising probability and hop count information. The route discovery phase is therefore introduced.

4.1. Next Node Selection. In order to achieve the shortest hop count from the current node to the sink, we assume that only the neighbors that are closer to the sink than the current node are added to the neighbor list as a candidate node. Since security is the essential metric in choosing different paths and to maximize the path security (Section 3), and to ensure constructing node-disjoint paths, each intermediate node selects one node as the next hop from its neighbor list to forward the *RREQ*, the neighbor with the highest security among all, smallest p_u. However, if the selected node is already reserved then the next neighbor with the smallest p_u will be selected and so on. The selected node then modifies the path information in the *RREQ* message (*hop* and P_{path_i} in Figure 3(a)), before forwarding the message to the next selected neighbor. The probability of path compromising, P_{path}, is updated according to (2) and the value of hop count, *hop*, is increased by one. Note that the initial values of *hop* and P_{path_i} at the source node are zero.

4.2. Number of Path Selection. The sink estimates the number of all available node-disjoint paths to the source from the number of the *RREQ* messages received to decide on choosing the first *np* most secure paths that satisfy the required security level. From these *RREQ* messages it obtains information about security and number of hops on each path. The sink sends back the route reply message, *RREP*, Figure 3(b), via the selected paths. Algorithm 1 is used to determine the number of node-disjoint multipath, *np*, which are used to transmit data message between the source and the sink. For each data transmission, given *n* available node-disjoint paths between the source and the sink, the sink sorts these available paths according to the security characteristics of each path (in terms of the probability that path *i* is compromised), such that the first path is the highest secure one and so on. The sink then calculates the probability that a packet is compromised, P_{pkt}, using (1). According to (1) more paths are chosen to lower P_{pkt} and enhance the security in order to deliver the data packet. Our proposed protocol only needs to select the first *np* paths ($np \geq 2$) satisfying $P_{\text{pkt}} \leq (1 - S_{\text{req}})$.

4.3. Security Mechanism. The following consecutive steps are involved in the routing mechanism to ensure the communication security level and are illustrated in Figure 4 [21].

(1) Divide the original data message of size *S* into *j* packets each of *M* fragments of size *b* bits. Assume the number of packets is equivalent to the number of paths used to transmit the data, *np*, such that $Mb = \lceil S/np \rceil$. If the last packet is less than *M* fragments, zero padding [9] is applied to meet the length requirements of RS codes.

(2) Encode each packet using RS codes to generate *M* data fragments and *K* parity fragments as a codeword of size $M+K$ fragments such that $K \leq M$. For each codeword packet, allocate one fragment on each path starting from the highest secure path and repeat this process till all the $M+K$ fragments are assigned on the selected multipath and ensure that the number of allocated fragments on each path, x_i, follows

$$x_i = \left\lceil \frac{(M+K)}{np} \right\rceil < M, \quad i = 1, 2, \ldots, np. \tag{6}$$

(3) Depending on the required security level, the number of fragments to be encrypted, N_{enc}, is calculated as follows:

$$N_{\text{enc}} = K + E, \tag{7}$$

where *E* is determined according to the required security level and $1 \leq E \leq M$.

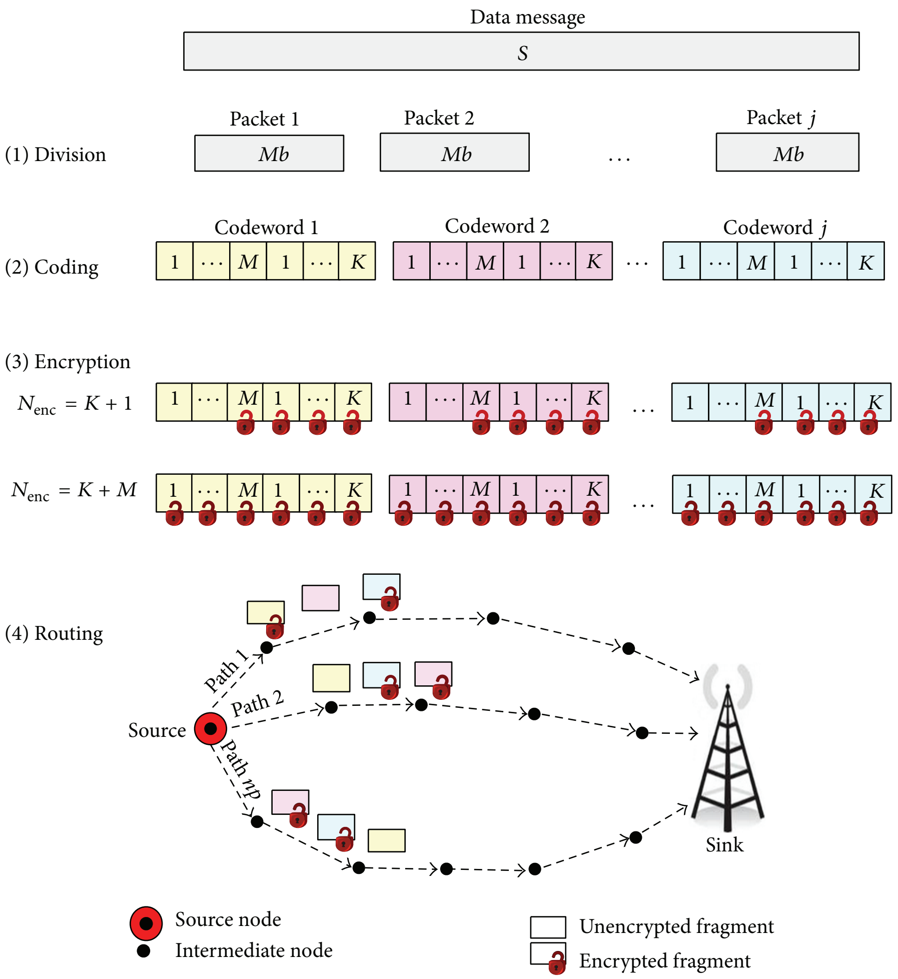


FIGURE 4: Proposed security mechanism.

As shown in Figure 4, for a low security requirement, $E = 1$, source node only encrypts any $N_{\text{enc}} = K + 1$ of $M + K$ fragments from the codeword. For each codeword, an attacker must receive at least M of the $M + K$ fragments and be able to decrypt the encrypted fragments to restore the codeword. On the other hand, when the required security level is high, then $E = M$, which requires to encrypt $N_{\text{enc}} = K + M$ fragments for each codeword. In order to compromise the data packet, the attacker must receive and be able to decrypt all M fragments to reconstruct the codeword.

(4) Route all the fragments on the np node-disjoint paths to the sink with each path carrying x_i fragments according to (4) and (6). To enhance security the encrypted fragments from the same codeword are transmitted on different paths.

(5) At the sink side, the encrypted fragments are decrypted first and then all the fragments are decoded to reconstruct the original data packet.

5. Evaluation Methodology

In this section, we precisely explain the security and reliability behaviors of the proposed mechanism. For security metric, we describe different scenarios to compromise the data packet, and for the reliability metric, we describe the failure models for which we evaluate the resiliency of our mechanisms.

5.1. Case Study. To help illustrate, we present an example on how the proposed mechanism functions with diverse security levels and attacker scenarios. Suppose we have a 9-byte data message to be transmitted to the sink. Let $np = 3$ and assume using packet-level RS (5, 3) code, where $M = 3$ and $M+K = 5$. Bit-level RS can also be used. The RS codeword packet has the following matrix format:

$$\text{RS codeword} = \begin{pmatrix} d_{j,1} \\ \vdots \\ d_{j,M} \\ p_{j,1} \\ \vdots \\ p_{j,K} \end{pmatrix}, \qquad (8)$$

where $d_{j,1} \cdots d_{j,M}$ and $p_{j,1} \cdots p_{j,K}$ are the data and parity fragments for codeword j, respectively.

```
$n$ = number of available node-disjoint paths (source to sink)
Sort for $P_{\text{path}}$ such that $P_{\text{path}_1} < P_{\text{path}_2} < \cdots < P_{\text{path}_n}$
$np = 1$;                    //Initialization
$P_{\text{pkt}_1} = P_{\text{path}_1}$              //Calculate the probability of compromising a packet on the first path
for ($i = 2$; $i \le n$; $i$++)
{
$np = np$++;
$P_{\text{pkt}_i} = P_{\text{pkt}_{i-1}} \times P_{\text{path}_i}$
If ($P_{\text{pkt}_i} \le (1 - S_{\text{req}})$)        //if the required security is reached
    {
        number of paths to be used = $np$;
        break;
    }
}
```

Algorithm 1: Calculating the number of paths related to the required security level.

Step 1 (*division*). For $np = 3$, divide the 9 byte data message to three packets of the size of 3-byte.

Step 2 (*coding*). The three packets are coded using RS code to generate three codewords each of the size of 5-byte as follows:

$$\text{Codeword } 1 = \begin{pmatrix} d_{1,1} \\ d_{1,2} \\ d_{1,3} \\ p_{1,1} \\ p_{1,2} \end{pmatrix},$$

$$\text{Codeword } 2 = \begin{pmatrix} d_{2,1} \\ d_{2,2} \\ d_{2,3} \\ p_{2,1} \\ p_{2,2} \end{pmatrix}, \tag{9}$$

$$\text{Codeword } 3 = \begin{pmatrix} d_{3,1} \\ d_{3,2} \\ d_{3,3} \\ p_{3,1} \\ p_{3,2} \end{pmatrix}.$$

Step 3 and 4 (encryption and routing). Depending on the required security level, encrypt any N_{enc} fragments, (7), for each codeword using any encryption algorithm and allocate fragments on np paths according to (4) and (6).

Scenario 1. For low security requirement, $N_{\text{enc}} = K + 1$, $N_{\text{enc}} = 3$ fragments:

$$\begin{gathered} \begin{pmatrix} d_{1,1} \\ d_{1,2} \\ d_{1,3} \\ p_{1,1} \\ p_{1,2} \end{pmatrix} \begin{pmatrix} d_{2,1} \\ d_{2,2} \\ d_{2,3} \\ p_{2,1} \\ p_{2,2} \end{pmatrix} \begin{pmatrix} d_{3,1} \\ d_{3,2} \\ d_{3,3} \\ p_{3,1} \\ p_{3,2} \end{pmatrix} \\ \text{path}_1 = d_{1,1}, p_{1,1}, d_{2,2}, p_{2,2}, d_{3,3} \\ \text{path}_2 = d_{1,2}, p_{1,2}, d_{2,3}, d_{3,1}, p_{3,1} \\ \text{path}_3 = d_{1,3}, d_{2,1}, p_{2,1}, d_{3,2}, p_{3,2}. \end{gathered} \tag{10}$$

In this scenario, the attacker must intercept at least two paths and decrypt six fragments to get the three codewords.

Scenario 2. For moderate security requirement, $N_{\text{enc}} = K+2$, $N_{\text{enc}} = 4$ fragments.

$$\begin{gathered} \begin{pmatrix} d_{1,1} \\ d_{1,2} \\ d_{1,3} \\ p_{1,1} \\ p_{1,2} \end{pmatrix} \begin{pmatrix} d_{2,1} \\ d_{2,2} \\ d_{2,3} \\ p_{2,1} \\ p_{2,2} \end{pmatrix} \begin{pmatrix} d_{3,1} \\ d_{3,2} \\ d_{3,3} \\ p_{3,1} \\ p_{3,2} \end{pmatrix} \\ \text{path}_1 = d_{1,1}, p_{1,1}, d_{2,2}, p_{2,2}, d_{3,3} \\ \text{path}_2 = d_{1,2}, p_{1,2}, d_{2,3}, d_{3,1}, p_{3,1} \\ \text{path}_3 = d_{1,3}, d_{2,1}, p_{2,1}, d_{3,2}, p_{3,2}. \end{gathered} \tag{11}$$

The attacker must intercept at least two paths and decrypt eight fragments to get the three codewords.

Scenario 3. For high security requirement, $N_{\text{enc}} = K + M$, $N_{\text{enc}} = 5$ fragments:

$$\begin{gathered} \begin{pmatrix} d_{1,1} \\ d_{1,2} \\ d_{1,3} \\ p_{1,1} \\ p_{1,2} \end{pmatrix} \begin{pmatrix} d_{2,1} \\ d_{2,2} \\ d_{2,3} \\ p_{2,1} \\ p_{2,2} \end{pmatrix} \begin{pmatrix} d_{3,1} \\ d_{3,2} \\ d_{3,3} \\ p_{3,1} \\ p_{3,2} \end{pmatrix} \\ \text{path}_1 = d_{1,1}, p_{1,1}, d_{2,2}, p_{2,2}, d_{3,3} \\ \text{path}_2 = d_{1,2}, p_{1,2}, d_{2,3}, d_{3,1}, p_{3,1} \\ \text{path}_3 = d_{1,3}, d_{2,1}, p_{2,1}, d_{3,2}, p_{3,2}. \end{gathered} \tag{12}$$

In this scenario, the attacker needs to intercept at least two paths and be able to encrypt a total of ten fragments to get the three codewords.

For all the above scenarios, an attacker needs to decode each codeword to be able to reconstruct the original data message and the allocation of fragments on the paths, allowing for

Table 1: Multipath routing protocols comparison.

Protocol	No. of transmitted packets	No. of redundant packets	No. of encrypted packets	Redundancy ratio
MVMP [9]	$\lceil S/M \rceil \times (M+K) = 15$	$\lceil S/M \rceil \times K = 6$	$\lceil S/M \rceil \times M + K = 15$	$K/(M+K) = 40\%$
Threshold secret sharing scheme	$S \times N = 27$	$(N-1) \times S = 18$	$S \times N = 27$	$(N-1)/N = 66.6\%$
Proposed scheme	$\lceil S/np \rceil \times (M+K) = 15$	$np \times K = 6$	$K + E = [3, 15]$	$K/(M+K) = 40\%$

Table 2: Simulation parameters.

Parameters	Value
Scenario 1	100% of nodes, $p_u = 0.14$
	10% of nodes, $p_u = 0.50$
Scenario 2	40% of nodes, $p_u = 0.20$
	50% of nodes, $p_u = 0.02$
S_{req}	$(1{-}10^{-1})$ to $(1{-}10^{-10})$ Lowest to highest

resilience to a failure of one path, which can be any path, since the three data fragments for each codeword can be obtained from the other two paths.

5.2. Multipath Protocol Performance Evaluation. In this section, we evaluate the proposed mechanism using the same scenario presented in Section 5.1 and compare it with the protocols that used the (T, N) threshold secret sharing scheme [6, 7] and RS coding technique, MVMP [9]. We present the comparison in Table 1 in terms of the total number of transmitted, redundant, and encrypted packets as well as the coding redundancy ratio.

Clearly, the number of encrypted packets in MVMP protocol is equal to the encrypted packet of our proposed protocol when the demanded security level is high. However, when the demanded security level is low, our proposed protocol encrypts only three packets while MVMP protocol has a fixed number of fifteen encrypted packets. Note that encrypted packets influence encryption time and energy consumption. We recognize that the encryption delay is related to the total amount of bits to be encrypted for each data packet (Section 3.4). Thus, the proposed security mechanism selects a minimum amount of data for encryption. In WSNs, if sensors run different encryption algorithms, like in MVMP protocol, it may lead to varying computational delays. For instance, the traditional RC4 algorithm takes 344 μsec to encrypt a block on the Atmega103 processor; however, it only takes 10 μSec on the StrongARM processor [22]. Also in [23], the experiment results show that the encryption process of RC5 algorithm consumes more energy than that of AES on MicaZ platform. Moreover, our proposed security mechanism uses one encryption algorithm while still maintaining a robust security protection unlike MVMP protocol where multiple versions of encryption algorithms are used to maintain the security.

We have conducted an extensive simulation study using C++ to evaluate the performance of our protocol. We adapted the same codes used in our previously published works [20, 24]. These papers illustrated the validity and comparability of our implementation, in which the validation tests cover the basic functionality of the on-demand routing protocol in WSNs. In WSNs, the likelihood of finding node-disjoint paths increases at higher node densities [25]. Thus, in order to increase the probability of finding these paths to evaluate the performance of our proposed protocol, we consider a network where 100 to 500 nodes are randomly scattered in a field of 500 m × 500 m area. We assume that all sensor nodes are static after deployment with transmission range of 100 m. The simulation parameters that we use are as follows. Source nodes are picked randomly, at least two hops away from the sink, to transmit a data packet at fixed generation rate of 1 packet/sec. The simulation time is 750 sec.

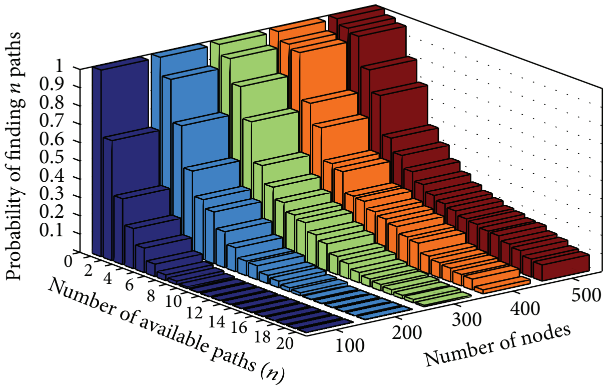


Figure 5: Probability of finding n node-disjoint paths.

We use two types of security scenarios in each simulation. In Scenario 1, each node is assumed equally likely to be compromised with probability, $p_u = 0.14$. In the second scenario and to evaluate the worst case where the probability that a sensor node is compromised, p_u, is changed suddenly at any transmission instant and is randomly distributed as presented in Table 2. Simulation results are obtained from different configurations to reduce the effect of the position of sensors. The results shown are averaged over 10 simulation runs.

The proposed mechanism depends on the availability of finding multiple node-disjoint paths and to justify the possibility of finding these paths in WSNs, the security requirements are not considered in this step. Figure 5 shows the probability of finding the maximal number of node-disjoint paths between the source nodes and the sink. As the number of paths found in both scenarios is equal, we only report one result in Figure 5, and this indicates that the process of finding the maximum number of paths depends on the network topology only.

Figures 6 and 7 illustrate the security performance and the number of used paths for various network sizes (500 and

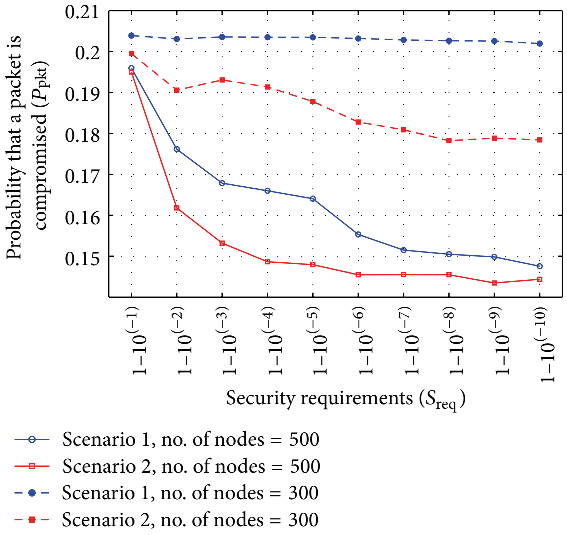


FIGURE 6: Security requirements (S_{req}) versus packet compromise probability (P_{pkt}),

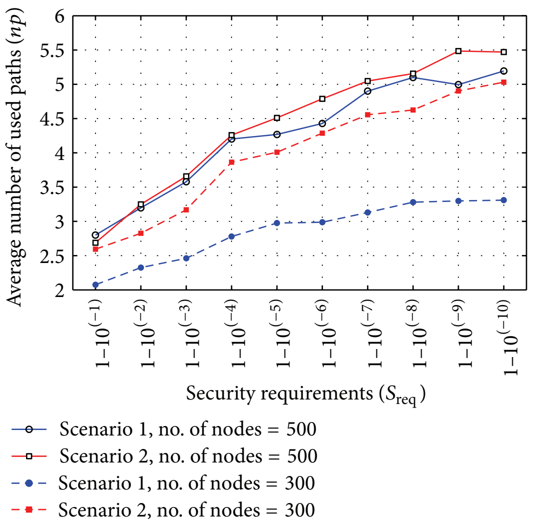


FIGURE 7: Security requirements (S_{req}) versus average number of used paths (np).

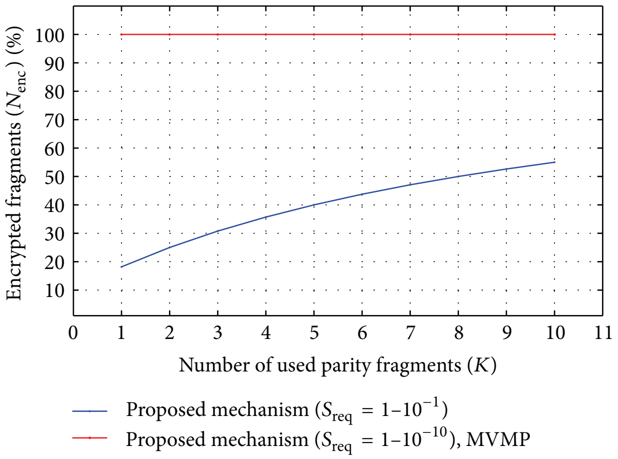


FIGURE 8: Percentage of encrypted fragments (N_{enc}) for a data packet of size M = 10 fragments.

300 nodes) as a function of the requested security. A message is compromised when at least M fragments are received and N_{enc} fragments are decrypted. It means $\lceil np/2 \rceil$ paths are intercepted out of the np used paths. It is clear that our mechanism is effective in increasing the security performance of a message according to the requested security. The probability that the message is compromised decreases with the increase of the security requirements since the number of paths used is related to these requirements. This result verifies the effectiveness of our mechanism. We also observe that when nodes are with different security levels (Scenario 2), our algorithm tends to select more secure paths compared to Scenario 1. However, in both scenarios, the probability that the message is compromised increases as the number of nodes increases. When the number of nodes increases, there are more sensor nodes available for forwarding packets.

In Figure 8, the number of encrypted fragments (N_{enc}) for different values of parity fragments ($K = 1, 2, \dots, K \leq M$) are presented. The data packet is set to $M = 10$ fragments. The number of encrypted fragments used in MVMP mechanism is compared with the lowest and the highest security requirements in our proposed protocol. The other S_{req}values show the same trend (between the two curves) and therefore are omitted. In MVMP mechanism all the fragments of the coded packet ($M + K$) are encrypted. Thus, the number of encrypted fragments using MVMP mechanism equals the number of encrypted fragments of the proposed mechanism at the highest security requirements. Clearly, the number of encrypted fragments is higher for the highest security requirement ($S_{\text{req}} = 1–10^{-10}$) to the encrypted fragments of the lowest security requirement ($S_{\text{req}} = 1–10^{-1}$); from 81.82% to 45% less fragments are encrypted for the lowest security requirement for $K = 1$ to 10, respectively. Obviously, when the demanded security level is high, our proposed protocol encrypts $K + M$ fragments similar to MVMP mechanism. However, when the demanded security level is low, $M + 1$ are encrypted. Note that encrypted packets influence encryption time and energy consumption; more encrypted fragments require more time and consume more energy.

6. Conclusions

In this paper, we propose and evaluate a secure and reliable routing protocol for WSNs that is designed to handle the application security requirements and reliable data transmission using coding and selective encryption scheme. In the proposed protocol, RS code is used to provide reliability and security. The proposed routing protocol is based on the node-disjoint multipath established depending on the link security parameters. The sink node decides on the paths selection process in order to satisfy the application requirements and the number of these paths is determined to enhance the security. Thus, different number of paths can be used for different security requirements. A novel security mechanism is proposed to support secure data transmission while respecting the network restrictions in terms of energy. The protocol reduces the energy consumption at sensor nodes by moving the path selection process to the sink node. Moreover, reducing the number of encrypted packets based on the required level of security limits energy consumption. Using different paths for different security requirements to route data and permitting the sink to be responsible for the path selection process, attacks such as the Sinkhole and Wormhole are no longer related, where in a Sinkhole attack the attacker tries to attract the traffic of surrounding neighbors by making itself look attractive to the surrounding neighbors with respect to the routing metric, and in a Wormhole attack two or more attackers may establish better communication tunnels between them in the path.

References

[1] E. Stavrou and A. Pitsillides, “A survey on secure multipath routing protocols in WSNs,” *Computer Networks*, vol. 54, no. 13, pp. 2215–2238, 2010.

[2] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, and S. Kotsopoulos, “Energy efficient and perceived QoS aware video routing over wireless multimedia sensor networks,” *Ad Hoc Networks*, vol. 9, no. 4, pp. 591–607, 2011.

[3] Y. Li, C. S. Chen, Y. Q. Song, and Z. Wang, “Real-time QoS support in wireless sensor networks: a survey,” in *Proceedings of the International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, pp. 373–380, Toulouse, France, November 2007.

[4] K. Akkaya and M. F. Younis, “Energy and QoS aware routing in wireless sensor networks,” *Cluster Computing*, vol. 8, no. 2-3, pp. 179–188, 2005.

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

[6] W. Lou and Y. Kwon, “H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1320–1330, 2006.

[7] W. Lou, W. Liu, and Y. Fang, “SPREAD: enhancing data confidentiality in mobile ad hoc networks,” in *Proceedings of the IEEE 23th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’04)*, pp. 2404–2413, Hong Kong, March 2004.

[8] C. H. Shih, Y. Y. Xu, and Y. T. Wang, “Secure and reliable IPTV multimedia transmission using forward error correction,” *International Journal of Digital Multimedia Broadcasting*, vol. 2012, Article ID 720791, 8 pages, 2012.

[9] M. Ruiping, L. Xing, and H. E. Michel, “A new mechanism for achieving secure and reliable data transmission in wireless sensor networks,” in *Proceedings of the IEEE Conference on Technologies for Homeland Security: Enhancing Critical Infrastructure Dependability*, pp. 274–279, Woburn, Mass, USA, May 2007.

[10] H. Alwan and A. Agarwal, “A survey on fault tolerant routing techniques in wireless sensor networks,” in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM ’09)*, pp. 366–371, Athens, Greece, June 2009.

[11] A. D. Wood and J. A. Stankovic, “Poster abstract: AMSecure: secure link-layer communication in TinyOS for IEEE 802.15.4-based wireless sensor networks,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys ’06)*, pp. 395–396, New York, NY, USA, November 2006.

[12] I. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *Journal of the Society For Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[13] S. K. Singh, M. P. Singh, and D. K. Singh, “A survey on network security and attack defense mechanism for wireless sensor networks,” *International Journal of Computer Trends and Technology*, vol. 1, no. 2, pp. 9–17, 2011.

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

[15] L. Chen and J. Leneutre, “On multipath routing in multihop wireless networks: security, performance and their Tradeoff,” *Journal of Wireless Communication and Networking, EURASIP*, vol. 2009, Article ID 946493, 13 pages, 2009.

[16] Y. Challal, A. Ouadjaout, N. Lasla, M. Bagaa, and A. Hadjidj, “Secure and efficient disjoint multipath construction for fault tolerant routing in wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1380–1397, 2011.

[17] N. Nasser and Y. Chen, “SEEM: secure and energy-efficient multipath routing protocol for wireless sensor networks,” *Computer Communications*, vol. 30, no. 11-12, pp. 2401–2412, 2007.

[18] J. Ben-Othman and L. Mokdad, “Enhancing data security in ad hoc networks based on multipath routing,” *Journal of Parallel and Distributed Computing*, vol. 70, no. 3, pp. 309–316, 2010.

[19] W. Wang, D. Peng, H. Wang, and H. Sharif, “An adaptive approach for image encryption and secure transmission over multirate wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 9, no. 3, pp. 383–393, 2009.

[20] H. Alwan and A. Agarwal, “Multi-objective reliable multipath routing for wireless sensor networks,” in *Proceedings of IEEE Globecom Workshop on Ad Hoc and Sensor Networking (GC ’10)*, pp. 1227–1231, Florida, Fla, USA, December 2010.

[21] H. Alwan and A. Agarwal, “A secure mechanism for QoS routing in wireless sensor networks,” in *Proceedings of the 25th IEEE Canadian Conference on Electrical & Computer Engineering*, pp. 1–4, Montreal, Canada, April 2012.

[22] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, “Analyzing and modeling encryption overhead for sensor network nodes,” in *Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA ’03)*, pp. 151–159, San Diego, Calif, USA, September 2003.

[23] H. Wang, M. Hempel, D. Peng, W. Wang, H. Sharif, and H. H. Chen, “Index-based selective audio encryption for wireless multimedia sensor networks,” *IEEE Transactions on Multimedia*, vol. 12, no. 3, pp. 215–223, 2010.

[24] H. Alwan and A. Agarwal, “Multi-objective QoS routing for wireless sensor networks,” in *Proceedings of the International Conference on Computing, Networking and Communications*, pp. 1074–1079, San Diego, Calif, USA, January 2013.

[25] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, “Highly-resilient, energy-efficient multipath routing in wireless sensor networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 568580, 18 pages
http://dx.doi.org/10.1155/2013/568580

Research Article

A Cross-Layer Framework for Network Management in Wireless Sensor Networks Using Weighted Cognitive Maps

Amr El-Mougy and Mohamed Ibnkahla

Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada K7L 3N6

Correspondence should be addressed to Mohamed Ibnkahla; ibnkahla@post.queensu.ca

Received 30 September 2012; Accepted 3 January 2013

Academic Editor: Marc St-Hilaire

Achieving the end-to-end goals and objectives of Wireless Sensor Networks (WSN) is a highly challenging task. Such objectives include maximizing network lifetime, guaranteeing connectivity and coverage, and maximizing throughput. In addition, some of these goals are in conflict such as network lifetime and throughput. Cross-layer design can be efficient in proposing network management techniques that can consider different network objectives and conflicting constraints. This can be highly valuable in challenging applications where multiple Quality of Service (QoS) requirements may be demanded. In this paper, a novel cross-layer framework for network management is proposed that particularly targets WSN with challenging applications. The proposed framework is designed using the tool known as Weighted Cognitive Map (WCM). The inference properties of WCMs allow the system to consider multiple objectives and constraints while maintaining low complexity. Methods for achieving different objectives using WCMs are illustrated, as well as how system processes can operate coherently to achieve common end-to-end goals. Using extensive computer simulations, the proposed system is evaluated. The results show that it achieves good performance results in metrics of network lifetime, throughput, and Packet Loss Ratio (PLR).

1. Introduction

Wireless Sensor Networks (WSN) are an enabling paradigm for a wide variety of applications. Due to their low cost, flexibility, and ease of deployment, they have already been applied in many fields such as environmental monitoring, food safety, intelligent transportation, and smart grids. In order to unlock the vast potential of WSN, several known challenges have to be addressed, such as limited energy and processing capabilities, scalability, and fault-tolerance [1–4]. These challenges impose restrictions on network management protocols that can be used, thus hindering the possibility of using WSN in some applications, such as those with strict Quality of Service (QoS) requirements.

In addition to traditional challenges and QoS requirements, end-to-end goals of WSN are sometimes in conflict. For example, maximizing network lifetime might require reducing the frequency at which nodes transmit their data, which in turn reduces throughput. In another example, maximizing network lifetime may mean that a large number of nodes will have to be switched to sleep mode, which impacts connectivity and coverage. Some WSN also require the support of different applications. For example, a WSN for highway safety may periodically transmit information regarding road or traffic conditions. This information usually has low volume and requires low transmission frequency. However, if an accident occurs, the WSN may be required to transmit images or even low-resolution video from the scene of the accident to aid first responders. In another example, a WSN for animal tracking may occasionally need to transmit images of the location of the animal if any dangers are detected (which may be indicated by, e.g., an elevation in the animal's heart rate). WSN protocols have to adapt quickly to such changes in application requirements and guarantee the desired performance levels.

Performance assurance in WSN has been addressed in many ways by the research community. Unfortunately, the research is rather scattered and usually focuses on specific issues. For example, there is significant research on the areas of connectivity and coverage, routing, and congestion control (see Section 2). However, there is not as much research activity on networks where several conflicting objectives need to be considered. Cross-layer design provides means to consider different issues from multiple layers in order to improve network efficiency. Nevertheless, there are challenges that are typically associated with cross-layer design that need to be addressed. One important challenge that can have significant effects on network performance is adaptation loops, where improving a specific network issue may lead to deteriorating performance in other issues. For example, transmit power adaptation affects levels of interference in the network and thus may have an impact on routing, congestion, energy lost due to collisions, among other factors. In another example, sleep/awake scheduling directly impacts network connectivity and may affect load balancing between nodes (e.g., if a set of nodes are chosen to remain constantly active, they may become depleted, while other nodes remain unused). Furthermore, WSN protocols are required to have low complexity due to the limited capabilities of nodes. In order to consider complex dynamics within a network, and ensure that network elements are operating coherently towards end-to-end goals, new network management protocols are needed.

In this paper, a novel cross-layer framework for network management is proposed for WSN that run challenging applications. The mathematical tool known as Weighted Cognitive Maps (WCMs) is considered as a tool to provide a parameterized representation of conflicting system processes, in order to perform reasoning while considering multiple conflicting goals and constraints. In WCMs, each process, environment variable, or end-to-end goal is simply represented as a concept in the system, and edges of the map connect concepts that are causally related (an overview of WCMs is given in Section 3). The inference properties of WCMs enable conflicting interactions within the network to be represented as simple mathematical operations, requiring only information about causal relationships between processes. Thus, multiple objectives and constraints can be considered with low complexity, avoiding long processing times that may be associated with optimization problems.

To the best knowledge of the authors, WCMs have not been considered in the design of network management systems for WSN. Although WCMs have been considered in [5] to design cognitive nodes, those nodes operated independently and the design did not consider network interactions between nodes. The design proposed in this paper primarily targets the needs of WSN. WCMs are utilized to design a framework for WSN that monitors network interactions and achieves its end-to-end goals. We illustrate how multiple objectives, constraints, system processes, and environment variables can be translated into concepts of a WCM and how protocols can be implemented using their underlying causal relationships. The proposed system is evaluated through extensive computer simulations to illustrate its capabilities.

The remaining sections of this paper are organized as follows; in Section 2, some recent related research efforts are reviewed; Section 3 provides an overview of WCMs and some fundamental design concepts; Section 4 explains the details of our proposed cross-layer framework; Section 5 provides simulation results that illustrate the capabilities of our proposal; and finally, Section 6 offers concluding remarks.

2. Related Work

Several areas in WSN have been explored by the research community. These areas include coverage and connectivity, routing, topology management, transmit power, and data rate adaptation. For example, the network management protocol proposed in [6] attempted to improve energy-efficiency and network lifetime while considering routing and coverage constraints in a clustered network architecture. An optimization problem was formulated, labeled (OPT-ALL-RCC), which minimized energy consumption and achieved load balancing while guaranteeing coverage and connectivity. OPT-ALL-RCC was shown to be NP-complete, and a heuristic algorithm named TABU-RCC was proposed to achieve a compromise between efficient performance and processing time. Another network management protocol known as Energy-Efficient m-Coverage and n-Connectivity Routing (EECCR) was proposed in [7]. It considered the routing problem under coverage and connectivity constraints. EECCR has two main phases. In the first phase, the network was divided into mutually exclusive scheduling sets, and sets that can guarantee m-coverage were switched on. Then, routing paths were set up to achieve n-connectivity. The second phase in EECCR was the data transmission phase, where the set-up routing paths were utilized to relay data to the sink node. In [8], a framework called Topology-Aware Resource Adaptation (TARA) was proposed with the primary goal of alleviating congestion in WSN. The idea was to activate a larger number of nodes in periods of congestion in order to increase network resources and reduce congestion. Network topology and traffic patterns were considered in order to propose heuristics that can detect congestion, activate the correct number of nodes, and discover alternative routing paths that can relay packets away from congested spots. Another protocol for congestion control called Enhanced Congestion Detection and Avoidance (ECODA) was proposed in [9]. ECODA used two buffer thresholds, $Q_{\min}$ and $Q_{\max}$, to detect congestion. Once buffer capacity dropped below $Q_{\max}$, the protocol started to filter packets based on their delay requirements. If buffer capacity dropped below $Q_{\min}$, most packets were rejected. A dynamic scheduler was also proposed that ensured fairness in packet delivery between nodes that were close to the sink and nodes that were far away from the sink. In addition, source sending rate control was used to mitigate packet dropping due to congestion.

Guaranteeing connectivity and coverage was addressed in [10], where a distributed algorithm was proposed to schedule the activation of nodes in every time slot. The algorithm assumed that distances between adjacent nodes were known and used this information to find a schedule that considered the remaining battery power of nodes in order to maximize network lifetime. In another paper [11], the problem of finding the optimal transmit power to maximize network lifetime while guaranteeing connectivity was studied. A physical layer-oriented QoS constraint was considered, based on the maximum allowable Bit Error Rate (BER) at the end of any multihop path. However, the analytical framework in [11] was proposed for specific routing and Medium Access Control (MAC) protocols. In addition, it was assumed that all nodes used the same transmit power.

In [12], a protocol for adapting transmit power, data rate, and duty cycle was proposed with the goal of improving energy efficiency and throughput. In this protocol, nodes that observed good channel conditions transmitted at higher data rates and therefore could conclude their transmissions earlier and stay in sleep state for longer periods. Thus, by adapting the duty cycle according to the observed channel conditions, considerable energy savings were achieved. Another protocol named Symphony was proposed in [13] and targeted transmit power and data rate adaptation while considering throughput and energy efficiency. Algorithms were proposed to ensure that frequent changes in transmit power and data rate were avoided and that fairness in accessing the wireless medium was ensured among interfering links. In [14], a protocol named Throughput Plus Fairness Optimization (TPFO) was proposed. In this protocol, an optimization problem was formulated with the goal of maximizing throughput while ensuring that asymmetric channel access is avoided. Given a certain set of data rates used at different nodes, the optimization problem was solved to find the optimum packet lengths and contention window sizes that would maximize throughput and guarantee fairness.

To support QoS in WSN, the Multipath Multi-SPEED (MMSPEED) protocol was proposed in [15] to address delay and reliability metrics. If delay was the required QoS parameter, every node maintained an estimate of the delay, called “progress speed,” needed to transmit to each of its active neighbours. The neighbour with the greatest progress speed was chosen as the next hop. A threshold, *SetSpeed,* was defined. As long as the progress speed at every hop was greater than *SetSpeed*, the end-to-end delay across the network was bounded by *SetSpeed* and the distance between the source and the destination. On the other hand, MMSPEED supported reliability of packet delivery by allowing multiple paths to deliver packets. Intermediate nodes determined how many paths should be used based on an error metric. As the required reliability increased, more paths were utilized to ensure low Packet Loss Ratio (PLR). However, utilizing multiple paths can increase interference in the network and increase the chance of collisions. Alternatively, the Distributed Aggregate Routing (DARA) protocol [16] also supported latency and reliability metrics in a WSN using multiple sinks. Packets with strict delay requirements were delivered using the shortest paths to the sink, while packets with loose delay requirements used longer paths. Reliability was achieved by transmitting multiple copies of the packets to the sink nodes. Packet scheduling was performed at forwarding nodes by prioritizing packets with strict delay requirements. The disadvantage of DARA is that transmitting multiple copies of packets is not energy efficient and creates interference. To address the combined issue of energy efficiency and QoS support, the Optimized Energy-Delay Subnetwork Routing (OEDSR) was proposed in [17]. Routing in OEDSR was based on a metric that considered the available energy, average end-to-end delay, and distance between the source and the destination. Performance results showed that OEDSR achieves low energy consumption and lower average end-to-end delay. However, OEDSR was only compared to classic protocols such as Ad Hoc On-Demand Distance Vector (AODV) routing and Dynamic Source Routing (DSR), which are outdated and have no capabilities to support QoS.

Therefore, it can be concluded that new research directions that can consider multiple conflicting constraints with low complexity are required to achieve the desired end-to-end goals of WSN. In this paper, WCM was selected to design a cross-layer framework for WSN due to its versatility and wide range of capabilities. Even though there have been limited efforts on using WCMs in wireless networks [5], as mentioned before, there have been extensive analytical efforts on developing the theory of WCMs since its original proposal in [18]. For example, the work in [19] included comprehensive mathematical description as well as construction methods of WCMs. In addition, the use of WCMs in various applications such as precision agriculture, drought management, and modeling business processes was discussed. In another example [20], the analytical and modeling capabilities of WCMs were extended by proposing methods of transforming one cognitive map into another. In [21], the inference capabilities of WCMs were studied, and different inference methods were compared and analyzed. Furthermore, the idea of conditional edge weights, where edges are activated only under certain conditions, was proposed in [22]. It was shown that conditional edge weights can provide an increased degree of flexibility in system design.

3. Fundamental Design Concepts of WCM

To understand how WCMs can be used in the design of wireless networks, this section provides an overview of the main properties of WCMs. Afterwards, we propose some fundamental guidelines for using WCMs in the design of a cross-layer framework for any network.

3.1. Overview of WCM. A WCM (also known as fuzzy cognitive map) is a graphical model used to represent dynamic systems through their underlying causal relationships [18–20]. Each vertex in the WCM is called a concept and represents a particular process or event in the system being modeled. For example, in WSN, a concept in the WCM can represent the processes of transmit power adaptation or routing, or it can represent environment variables such as PLR or Expected Transmission Time (ETT) at a particular node. A concept can also represent end-to-end goals or constraints such as network lifetime or connectivity. Each concept C_i is characterized by a scalar A_i that represents the value of the concept or its activation level in the real system. These characterizing scalars can take values either in the range $[0, 1]$ or $[-1, 1]$. If the allowed interval is $[0, 1]$, then a concept can be inactive ($A_i = 0$), fully active ($A_i = 1$), or partially active. On the other hand, if the allowed interval is $[-1, 1]$, then C_i can be increasing ($0 < A_i \leq 1$) or decreasing ($-1 \leq A_i < 0$). Edges of the WCM connect concepts that are causally related. Edge weights can take on any value from the interval $[-1, 1]$. Negative edge weights

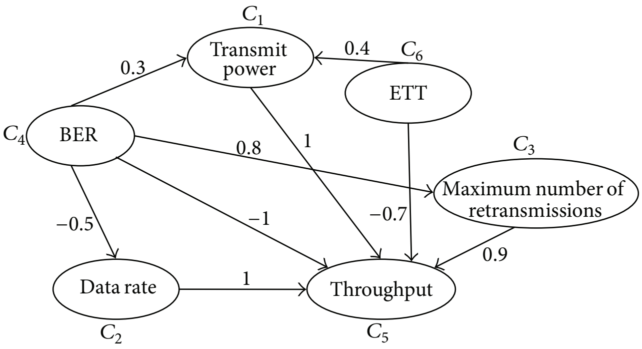


Figure 1: WCM representing processes of a wireless node.

imply negative causality and positive edge weights imply positive causality, while zero edge weights imply the absence of a causal relationship between concepts. WCMs can be qualitative or quantitative. Qualitative WCMs only represent causal relationships between concepts, while quantitative WCMs can also represent different levels of granularity in concepts.

One of the main advantages of WCMs lies in their inference capabilities. To illustrate, consider the following WCM representing processes of a wireless node, depicted in Figure 1.

The WCM of Figure 1 models the relationships between 6 processes (concepts) affecting a wireless node, namely, transmit power, data rate, maximum number of retransmissions allowed before a packet is dropped, BER, throughput, and ETT. The edge weights shown represent the strength of causality between concepts. We can classify concepts in any WCM into end-to-end goals, environment variables, and processes. All processes interact to achieve the end-to-end goals. On the other hand, environment variables cannot be manipulated directly, but only as a result of other actions. It can be seen from Figure 1 that the end-to-end goal is throughput (C_5), since all other concepts can cause changes in it. It can also be seen that there are two environment variables considered, namely, BER and ETT, since they cause changes in other processes but cannot be changed directly by them. They can only be affected by changes in the environment or as a result of actions taken by the WCM. The processes available to the WCM are transmit power and data rate adaptation and changing the maximum number of retransmissions. The WCM can be represented in matrix form as

$$W = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.9 & 0 \\ 0.3 & -0.5 & 0.8 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0 & -0.7 & 0 \end{pmatrix} \end{array}. \tag{1}$$

For a WCM with n concepts, its status at time t can be given by

$$A(t) = [A_1 \ A_2 \ A_3 \ \cdots \ A_n], \tag{2}$$

where A_i is a scalar value representing the activation level of concept C_i. The inference process of WCMs is the one by which the values of concepts change according to their underlying causal relationships. Thus, according to the inference properties of WCMs, the status at time $t + 1$ can be given by

$$A(t+1) = f(A(t)W), \tag{3}$$

where $f(x)$ is a threshold function that determines the type of WCM [21]. The inference process is initialized when a particular concept is triggered, causing its activation value to change, for example, due to a sudden increase in BER. Therefore, the input to the WCM at time t is $A(t)$, including the new value of the concept that was just triggered. The triggered concept influences other concepts according to W, thus producing an output to the WCM, $A(t + 1)$, as given by (3). This output provides the new activation values of the concepts, thus indicating the adaptations that need to take place. For example, if an increase in concept C_4 causes an increase in concept C_1 (see Figure 1), this means that the increase in BER requires that the system increases the transmit power.

WCMs have significant advantages over other tools, such as Bayesian and neural networks [23]. They allow for feedback loops, which are not present in Bayesian networks. Concepts in WCMs can also represent events or processes from the real system. This is not available in neural networks, which can be seen as a black box trained to model a particular system and may not faithfully reproduce its characteristics. The simple inference properties also make WCMs attractive for systems that require low complexity, such as WSN. It is also worth noting that WCMs have some disadvantages [23]. They rely on expert knowledge to design the system, which may be challenging, especially in quantitative WCMs. Also, there is no research on how to build a WCM with a centralized view of the network, where multiple WCMs may exist and have to work together to achieve end-to-end goals.

3.2. Design Guidelines for WCM. In order to design a reasoning machine based on WCM, we have to identify the concepts to be considered and the causal links between them. As mentioned in Section 3.1, concepts of the WCM can be classified into end-to-end goals, processes, and environment variables. Thus, the first step is to identify the end-to-end goal(s) of the WCM. These are the concepts that will be constantly monitored by the WCM in order to ensure that they are achieved. Thus, all concepts will interact to achieve these goals. The goals of the WCM must relate directly to the end-to-end goals of the application and the requirements of the network. In addition, it is desirable to design the goals in a way that facilitates monitoring by the WCM. For example, if the network is required to remain operational for a specific period of time, then a suitable goal of the WCM might be energy consumption, which the WCM monitors and takes actions when it is above an unacceptable threshold. On the other hand, if the network is expected to support QoS applications that require the guarantee of certain parameters such as bandwidth, reliability, and delay, these parameters can be directly incorporated as end-to-end goals of the WCM.

Thus, the WCM will monitor these parameters and take actions at appropriate nodes when required. Furthermore, a goal of the WCM can incorporate several parameters simultaneously. For example, a goal can be the ratio of source load to transmission rate, which can be used to avoid congestion.

After determining the goals of the WCM, the next step is to identify the processes that will be used to achieve these goals. Any process can be incorporated in the WCM, depending on the capabilities of the nodes in the network. For example, some of the processes can be transmit power and data rate control, adjusting the size of the contention window, adjusting the sleep/active schedule of devices, or routing. The task of the WCM is to determine when to activate these processes in response to environmental changes. Thus, after determining the processes, the environment variables that trigger the WCM to operate have to be specified. For example, if transmit power and data rate control are considered, a suitable environment variable that would trigger these processes can be PLR, BER, ETT, or others depending on the system and the application requirements. Choosing the processes and environment variables ultimately depends on the issues that the WCM must take into account. Thus, if congestion is a probable event that must be addressed, processes such as adjusting the source loading rate and routing might be incorporated, and the environment variables in this case might be remaining buffer capacity and channel utilization.

After determining all the concepts of the WCM, the next step is to identify the edge weights and causal relationships between concepts. These edge weights depend on the nature of the processes (or the protocols) and design principles utilized. For example, the edge between the concept of PLR and the concept of transmit power control should have a positive weight. This is because an increase in PLR requires an increase in transmit power, and vice versa. Conversely, the edge between PLR and data rate should have a negative edge weight because an increase in PLR requires a decrease in data rate, and vice versa. On the other hand, causal relationships may also be affected by the system design itself. For example, if the system considers both transmit power and data rate control, it may not be desirable to activate both concepts every time there is a change in PLR. To address this issue, the idea of conditional edge weights can be used [22], where only specific edges will be activated depending on the status of the system. For example, if there is an increase in PLR and the WCM has the option of either increasing the transmit power or decreasing data rate, the WCM can make the decision based on the current value of energy consumption and throughput. Thus, if energy consumption is too high, the WCM may opt to decrease data rate to avoid excess drainage in battery power. On the other hand, if throughput is too low, then the WCM may choose to increase transmit power to avoid any impact on the speed of data delivery. This way, multiple objectives and conflicting constraints can be considered by the WCM when executing every action.

An important issue arises in WCM design, which is how to determine the specific weights of the edges. As explained in Section 3.1, a WCM can be qualitative or quantitative. A quantitative WCM (where edge weights may be fractional) may provide an accurate representation of system interactions but will ultimately require a continuous supervised learning operation to determine the exact value of the edge weights depending on network conditions [19]. This imposes a heavy processing burden on the system, which may not be acceptable in networks with limited capabilities such as WSN. Moreover, the efficiency of the overall system will be highly affected by the accuracy of the learning process. Thus, in this paper, focus will be given to qualitative WCM (where edge weights are either −1, 0, or 1), which do not require supervised learning processes. For example, the edge weight between PLR and transmit power would simply be equal to 1 because an increase in PLR causes the system to use the next higher transmit power. The choice of edge weights may also depend on the situations where the system designer specifies that these edges must be activated. For example, the designer may specify that routing should be activated when congestion is detected at a certain node. Thus, the edge weight between the congestion detection concept and the routing concept should be set to 1. Nevertheless, in order to incorporate some of the advantages of quantitative WCM, some design methods will be proposed in the following sections that illustrate how to give the system some quantitative traits while maintaining the qualitative nature of the WCM.

The final issue to be considered in the design of a WCM framework is implementation. Particularly, distributed versus centralized implementation has to be studied. The advantages and disadvantages of each of these options are well known and will not be repeated here. A hybrid implementation can also be adopted with WCMs, where concepts that require centralized operation are implemented at a central node, while other concepts are distributed over other nodes. In Section 4, we propose a framework for challenging WSN applications based on WCMs, in order to illustrate how the design guidelines that were explained in this section can be implemented. It is important to stress that WCM design is not restricted to the concepts included in the system proposed in Section 4. They are only being used as examples of how WCMs can be used to achieve multiple conflicting goals with low complexity.

4. Designing the Cross-Layer Framework Based on WCM

This section illustrates how a cross-layer framework for WSN can be designed using WCMs. First, the system model is illustrated, and then the design of WCMs to achieve different objectives of the system is explained.

4.1. System Model. The WCM system can be implemented using any WSN architecture. Without loss of generality, and to simplify the concepts to the reader, a clustered hierarchy is adopted in this paper. Nodes where the WCM is implemented are labeled "intelligent nodes," and they naturally will require higher energy and processing capabilities than regular sensor nodes. However, not all nodes in the network are required to be intelligent. Intelligent nodes can make decisions that are executed by regular nodes.

As shown in Figure 2, the WCM is implemented at the sink node and cluster heads (CH). Each CH will gather

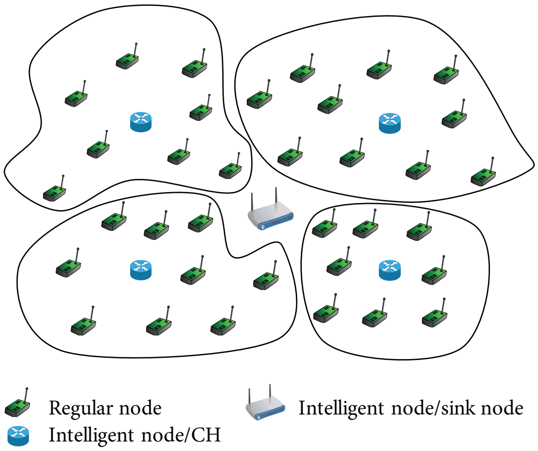


Figure 2: Proposed WSN topology.

information from its cluster (such as PLR and residual battery power of nodes) and use the WCM to make decisions about different system parameters, such as transmit power and data rate, which will then be applied by regular nodes. The sink node will be responsible for monitoring WCM concepts that require a centralized view of the network, such as connectivity and coverage. It is also responsible for ensuring that end-to-end goals of the network are achieved. The sink node and CHs are placed in strategic geographic positions in the network to maximize the performance gain from their deployment. The regular sensor nodes can be deployed at random or in predetermined (specific) patterns, depending on the application requirements.

Figure 2 shows regular sensor nodes randomly deployed throughout the area to be covered, with the sink node at the center of the network. There are four clusters with four CHs strategically deployed in positions such that the number of regular sensor nodes managed by each CH is fairly even. It is assumed that nodes are synchronized and that time is slotted. Every time slot has the same duration as one duty cycle. Thus, in one time slot, a node may wake up, sense the medium, transmit one packet, and go back to sleep mode. Moreover, it is assumed that the active portion of the duty cycle is adaptive according to the data rate utilized, similar to the algorithm proposed in [12], which has been previously reviewed in Section 2. If there are redundant nodes in the network, which is probable WSN, not all nodes will wake up in every duty cycle. Thus, scheduling which nodes to wake up takes place every M time slot. This means that, for every M consecutive time slots, the same group of nodes will wake up in every duty cycle. If some nodes fail during this period or need to be switched off for any reason, the WCM will be responsible for making decisions that guarantee connectivity and coverage in every time slot. We also assume that the locations of nodes are known to the WCM. This can be done using GPS devices or any other method for node localization.

In addition, it is assumed that every node transmits using one transmit power from the set $\mathrm{Tx} = \{\mathrm{Tx}_1, \mathrm{Tx}_2, \ldots, \mathrm{Tx}_L\}$, and one data rate from the set $\mathrm{Rate} = \{R_1, R_2, \ldots, R_K\}$. Also, every node can choose from a discrete set of duty cycles, $\mathrm{DC} = \{D_1, D_2, \ldots, D_K\}$. Furthermore, nodes have limited buffer capacity of B packets. Thus, the set of occupied buffer slots is given by $\mathrm{BF} = \{\mathrm{BF}_1, \mathrm{BF}_2, \ldots, \mathrm{BF}_B\}$. A simple routing protocol is assumed, where nodes formulate a tree that identifies the shortest path to the associated CH. Thus, multiple hops are allowed within a cluster.

In the following sections, the design of different parts of our WCM will be illustrated. The objective is to design a WSN that considers multiple objectives and conflicting constraints, for example, the energy efficiency versus throughput tradeoff, while maintaining low complexity. Therefore, without loss of generalization, the proposed WCM will include the processes of transmit power, data rate, and duty cycle control; routing; and congestion control. The constraints to be satisfied are connectivity and coverage, and we define an end-to-end goal that considers energy consumption as well as traffic load. The authors emphasize that the proposed WCM is not restricted to the processes or protocols being adopted, but the goal is to illustrate the capabilities of WCMs and provide examples of how they can be designed.

4.2. End-to-End Goal of the WCM. As mentioned before, the first step in designing the WCM is to determine the end-to-end goal. Typical choices of end-to-end goals in WSN include maximizing network lifetime, minimizing energy consumption, and maximizing the amount of data that the network can collect before it reaches its lifetime. In this paper, "Network Efficiency" was chosen as the end-to-end goal of the WCM, which is defined as traffic load divided by energy consumption. The reason behind this choice is to give the network user the flexibility to request higher volumes of data throughout the lifetime of the network. Higher volumes of data will ultimately require higher energy consumption, which has to be considered. Choosing energy consumption only or network lifetime only may cause the network to deny the user higher volumes of data in order to achieve the end-to-end goal. The energy consumption of a node is the amount of battery declination within a given time window (M time slots) divided by its residual battery power. Every node calculates how much the battery declines within the time window and then transmits this value along with its remaining battery power in every packet. Thus, the energy consumption of node i can be given by

$$\mathrm{Eng\_consump}_i = \frac{E_{\mathrm{start\_window}} - E_{\mathrm{end\_window}}}{E_{r_i}}, \qquad (4)$$

where E_{r_i} is the residual energy of node i, $E_{\mathrm{start\_window}}$ is the remaining energy of node i at the beginning of the time window, and $E_{\mathrm{end\_window}}$ is the remaining energy at the end of the window. Equation (4) considers remaining battery power as well as load balancing. In particular, if the residual battery power of a node is low, then its efficiency will decrease, prompting the WCM to take action. On the other hand, if its energy consumption increases, say from being involved in excess traffic, its efficiency will also decrease, again prompting the WCM to take action (specific actions will be explained in later sections). This ensures that nodes will not be overused and will be avoided if their battery power is low.

Traffic load is defined as the amount of traffic dispensed by any node within a given time window (M time slots). Thus, the traffic load of node i can be expressed as

$$\begin{aligned} &\text{Load}_i \\ &= \text{Number of packets dispensed within } M \text{ time slots} \\ &\quad /\text{Total number of packets generated at } i \text{ within } M \text{ time slots} \\ &= \frac{P_{\text{Trans}(i)}}{P_{\text{Gen}(i)}}, \end{aligned} \tag{5}$$

where $P_{\text{Trans}(i)}$ is the number of packets transmitted at i within M time slots and $P_{\text{Gen}(i)}$ is the number of packets generated at i within M time slots. Thus, if the number of transmitted packets is significantly lower than the number of generated packets, the WCM will be prompted to take action. Therefore, the efficiency of a cluster with number of nodes G is given by

$$\text{Efficiency} = \sum_{i=1}^{G} \frac{\text{Load}_i}{\text{Eng\_consump}_i}. \tag{6}$$

Cluster efficiency, as calculated in (6), is the end-to-end goal that will be considered by the CHs. Since decisions made by one CH can affect the efficiency of another cluster (e.g., routing decisions), the sink node will make sure that the efficiency of all CHs is above the required threshold. Proper actions will be taken if the efficiency of one cluster drops below the threshold, as will be shown in the following sections.

In order to determine the appropriate threshold for efficiency, $\text{Eff}_{\text{Thresh}}$, that would trigger the WCM to take action, application requirements have to be considered. For example, if the initial battery power of nodes is E mAhr and the target lifetime of the network is X time slots, the task of the WCM is to make sure that energy consumption does not exceed E/X mAhr per time slot per node or MG$E/X$ mAhr per M time slots per cluster. On the other hand, the average value of Load_i should be close to 1 so that the memory of nodes is not consumed and nodes are not overused in relaying packets for other nodes. Thus, the WCM will be prompted to take action if Load_i is less than 0.9 (10% of the nominal value).

4.3. Designing a WCM for Transmit Power, Data Rate, and Duty Cycle Control. Controlling transmit power, data rate, and duty cycle are powerful tools for improving performance. In order to design a WCM that performs these adaptations, the first step is to identify the environment variables, processes, and end-to-end goal that will make up the concepts of the WCM. The second step is to identify the protocols that will define the causality between the considered concepts.

The processes used in this WCM are transmit power, data rate, and duty cycle control, while the end-to-end goal is network efficiency, given by (6), which is the goal of the overall system. The environment variables are PLR and ETT. PLR was chosen because it is affected by channel conditions and interference and can thus give a clear indication about the quality of the wireless link. ETT is the expected amount of time needed to successfully transmit a packet. It accounts for interference between links, number of retransmissions needed to successfully transmit a packet, and the data rate used at interfering links. Therefore, it can be used to improve throughput and ensure fairness between interfering links. For example, when the ETT of one node is significantly larger than that of another, this means that the node is not getting fair access to the wireless medium. By taking actions to keep ETT close among interfering nodes, fairness can be achieved.

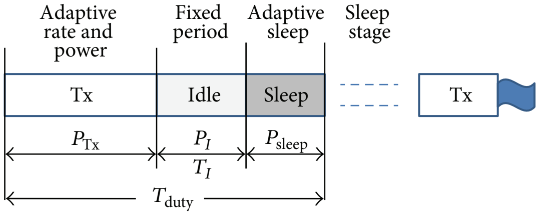


Figure 3: Duty cycle with adaptive sleep period.

The next step is to determine the set of rules and interactions that will determine the causality between concepts of the WCM. Whenever there are changes in the environment variables that require intervention by the CH, the WCM is triggered to take action. This action can be to adapt transmit power, data rate, or both. Duty cycle adaptation directly follows data rate adaptation, similar to the idea proposed in [12]. If the data rate is increased, the duty cycle is decreased, and vice versa. The WCM decides whether to adapt transmit power or data rate by considering the energy consumption resulting from such adaptations. To illustrate, consider the duty cycle shown in Figure 3.

Figure 3 shows a duty cycle with length T_{Duty}. It is divided into an active period, where packet transmissions take place; a fixed idle period, with length T_I; and an adaptive sleep period. P_{Tx} is the energy consumed during the active transmission period in Watts, P_{idle} is the energy consumed during the idle period in Watts, and P_{sleep} is the energy consumed during sleep period. Note that P_{Tx}, P_{idle}, and P_{sleep} are determined by hardware characteristics. Also P_{idle} and P_{sleep} are constant parameters, but P_{Tx} is directly proportional to the transmission power used. Adapting transmit power and data rate takes place during the active period. The energy consumed at a single node during one duty cycle can be expressed as

$$E = P_{\text{Tx}} \times N\frac{L}{R} + P_{\text{idle}} \times T_I + P_{\text{sleep}} \left(T_{\text{Duty}} - T_I - N\frac{L}{R}\right), \tag{7}$$

where N is the number of packets transmitted during the active period, L is the packet length in bits, and R is the data rate used. The WCM uses (7) to compare between the energy consumption resulting from adapting transmit power or data rate. For example, if there is an increase in PLR that affects link reliability, the WCM needs to compare between increasing transmit power and decreasing data rate. After

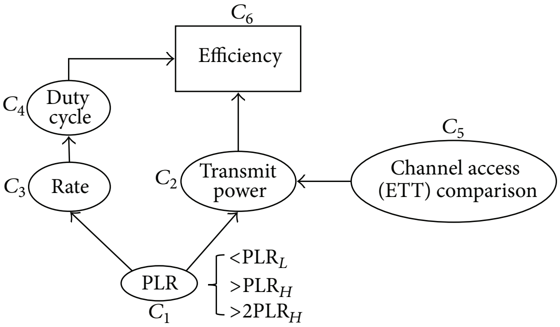


FIGURE 4: WCM for transmit power, data rate, and duty cycle adaptation. The concepts are labelled C_1–C_6.

removing the constant parameters from (7), this comparison can be expressed as

$$\frac{P_{\text{Tx-Old}}L}{R_{\text{New}}} - P_{\text{sleep}}\frac{L}{R_{\text{New}}} \geq \frac{P_{\text{Tx-New}}L}{R_{\text{Old}}} - P_{\text{sleep}}\frac{L}{R_{\text{Old}}}, \quad (8)$$

where $P_{\text{Tx-Old}}$ and R_{Old} are the current transmit power and data rate values, while $P_{\text{Tx-New}}$ is the value of transmit power if it will be increased and R_{New} is the value of data rate if it will be decreased. If the comparison in (8) is "True," then this means that decreasing the data rate will result in higher energy consumption, and thus the transmit power will be increased. On the other hand, if it is "False," then the data rate will be decreased instead. A similar process takes place if there is a decrease in PLR. Also, if there is a sharp decrease in PLR, the WCM has to increase transmit power and decrease data rate simultaneously in order to address the problem. After determining the end-to-end goal, environment variables, processes, and causality between concepts, the resulting WCM is shown Figure 4.

In order to implement the adaptation algorithm, conditional edge weights are used. Thus, when the PLR crosses a high or low threshold, a comparison according to (8) is performed, and a weight matrix, W, is formed similar to the one in (1). Edges $W_{(1,2)}$ or $W_{(1,3)}$ are activated according to the following rule:

$$\begin{aligned}
&\text{If PLR} < \text{PLR}_L \text{ Or PLR} > \text{PLR}_H \text{ and (8) is} \\
&\quad \begin{array}{ll} \text{True,} & W_{(1,2)} = 1 \text{ and } W_{(1,3)} = 0 \\ \text{False,} & W_{(1,2)} = 0 \text{ and } W_{(1,3)} = -1 \end{array} \\
&\text{If PLR} > \text{PLR}_{2H} \\
&W_{(1,2)} = 1 \text{ and } W_{(1,3)} = -1,
\end{aligned} \quad (9)$$

where PLR_H, PLR_L, and PLR_{2H} are the values chosen for high, low, and too high PLR thresholds, respectively, and their specific values are determined according to the reliability requirements of the application. For example, if the application requires a reliability level of 98%, then $\text{PLR}_H = 0.02$, $\text{PLR}_L = 0.02 - 10\% = 0.018$, and $\text{PLR}_{2H} = 0.02 + 10\% = 0.022$. The value 10% was chosen to avoid the WCM from triggering frequent adaptations.

The rules in Expression (9) simply implement the algorithm detailed above, with the precaution that if $\text{PLR} > \text{PLR}_{2H}$, then edges $W_{(1,2)}$ and $W_{(1,3)}$ are activated simultaneously to address the deterioration in channel conditions. Note that the WCM is triggered to take action if PLR changes considerably, or if the ETT of one node is 10% larger than that of an interfering node. The imbalance in ETT is addressed by activating $W_{(5,2)} = 1$, which leads to an increase in transmit power to give the node a better chance in accessing the medium. In addition, an increase in data rate will lead to the activation of $W_{(3,4)} = -1$, so that the duty cycle will be decreased. Also, an increase in transmit power or duty cycle will activate $W_{(2,6)} = -1$ and $W_{(4,6)} = -1$, respectively, prompting the WCM to recalculate the new value of efficiency, in order to ensure that the adaptations made do not violate the end-to-end goal.

After formulating the weight matrix, the CH formulates an array $A(t)$ similar to (2), reflecting the current activation levels of the concepts in Figure 4. For example, if $\text{PLR} > \text{PLR}_H$, then $A_1 = 1$, and if $\text{PLR} < \text{PLR}_L$, then $A_1 = -1$. According to (3), a multiplication is performed to determine $A(t + 1)$. The new values of concepts are then used to determine the actions needed to be taken. It is important to stress that the result of this multiplication process will not specify the exact level of transmit power or data rate to be used, but only if they should be increased or decreased.

4.4. Designing a WCM to Guarantee Connectivity and Coverage. This section illustrates how to design a WCM to satisfy connectivity and coverage constraints. Typical WSN have redundant nodes in order to extend their lifetime. Nodes in the network wake up every duty cycle with probability P. This value should ensure that every point in the area is within the sensing range of at least k sensors, and every node can find a routing path to the sink node. We assume that every node has a circular sensing range with radius r_s. The value P is lower as the number of redundant nodes increases and should increase to 1 as nodes die throughout the network's lifetime. This is because, as nodes die, the number of redundant nodes in the network decreases, and thus the remaining nodes need to wake up with higher probability to guarantee connectivity and coverage.

The objective of the WCM in this section is to adapt P in every duty cycle in order to ensure connectivity and coverage. The environment variable that triggers the activity of this WCM is node failure. Thus, the CH will gather information from its cluster to determine how many nodes are active in every duty cycle. This can be done by monitoring traffic from regular sensor nodes in every duty cycle, since every node that wakes up senses and transmits information. Once the CH detects that the number of active nodes is close to violating the conditions of connectivity and coverage, the WCM will be triggered to increase P in order to ensure that more nodes will wake up in the following duty cycle.

In order to perform this task, we use the theorem derived in [24], which states that for sensing range r_s and communication range r_c at every node. When $\alpha = r_s/r_c \leq 1$, the area (D) is almost surely connected-k-covered if, for some growing function $\varphi(nP)$, P and r_s satisfy

$$nP\pi r_s^2 \geq \log(nP) + k\log\log(nP) + \varphi(nP), \quad (10)$$

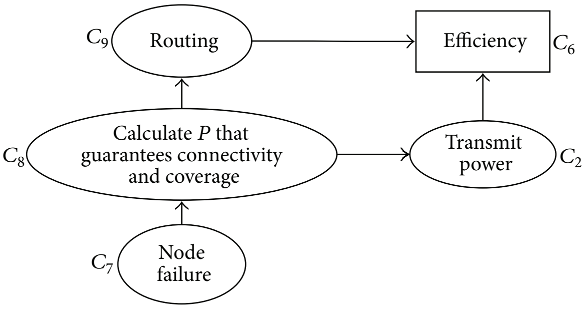


Figure 5: WCM to guarantee connectivity and coverage.

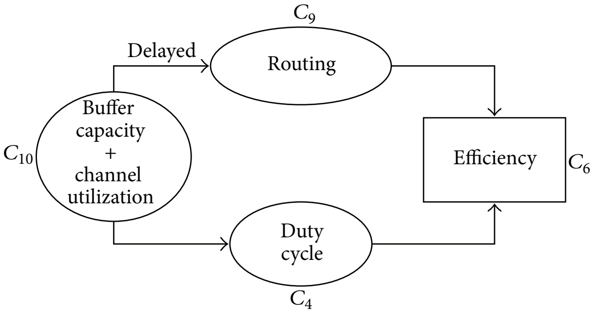


Figure 6: WCM for congestion control.

where n is the number of nodes in the network. The expression "connected-k-covered" means that every point in the network is covered by at least k sensors and can find a path to the sink node. Almost surely connected-k-covered means that as n increases to infinity, the probability of connected-k-coverage goes to 1.

Note that as nodes die in the network, n will decrease accordingly. Thus, the WCM will attempt to find P that satisfies Inequality (10) in every duty cycle according to the current value of n. Section 5 will illustrate how an appropriate activation function $\varphi(nP)$ is chosen to ensure high probability of connectivity and coverage when Inequality (10) is satisfied. The selected theorem achieves connectivity and coverage in a dynamic way, by adjusting P using simple mathematical operations, thus avoiding intensive processing operations typically associated with optimization problems used to address this issue. The WCM for guaranteeing connectivity and coverage is depicted in Figure 5.

As this figure shows, the WCM is triggered when node failure (C_7) occurs, causing a reduction in the number of active nodes. The edge $W_{(7,8)} = 1$ is activated, and the CH formulates an input array $A(t)$ as in (2), including the current activation values of the concepts ($A_7 = 1$). New activation values of concepts $A(t + 1)$ are then calculated using (3), causing concept C_8 to be activated. This will prompt the CH to calculate P that satisfies Inequality (10) according to the new value of n (after node failure). Based on the new value of P, rerouting (C_9) may need to be activated ($W_{(8,9)} = 1$) to find paths for the newly activated nodes. Note that the end-to-end goal of this WCM is again efficiency, since this is the goal of the overall system. Figure 5 also shows that an edge from C_8–C_2 is added, so that the transmit power of nodes can be increased ($W_{(8,2)} = 1$) if the current values do not guarantee connectivity. Similar to the WCM in Figure 4, changes in transmit power or routing will activate $W_{(2,6)} = -1$ and $W_{(9,6)} = -1$, respectively, in order to ensure that adaptations do not violate the end-to-end goal.

4.5. Designing a WCM for Congestion Control. Congestion typically occurs when a particular set of nodes in the network becomes exposed to an amount of traffic larger than what the nodes can handle. This can cause queue buildups and significant transmission delays. The traditional way of dealing with congestion is to instruct the source node to reduce its loading rate, which is the number of bits inserted in the transmission queue per second. Note that this is different from the node's data rate, which corresponds to the rate at which bits are transmitted from the transmission queue. Although reducing source loading rate may reduce congestion, it has a significant disadvantage from the point of view of the QoS observed by the user. Decreasing the loading rate reduces the resources given to the user at the specific time when she/he is requesting more resources [8]. In WSN, there is a possibility for another solution to this problem, which is to increase the amount of resources available in the network. This approach was explored in [8], where topology-aware resource adaptation (TARA) was proposed. In TARA, sleeping nodes are instructed to wake up around congested areas, and data is routed through these nodes to alleviate the problem. This approach may not always work over the entire network lifetime if nodes die and no more redundant nodes are available to be activated. It also requires significant knowledge of network topology.

However, redundant nodes are not the only way of increasing network resources. Since duty cycle control is already being utilized, as previously explained, it can be exploited to keep the nodes active for longer periods of time when congestion occurs in order to dispense larger volumes of traffic. In the proposed WCM system, nodes will be instructed to extend their duty cycles to accommodate extra traffic. However, route maintenance will still be needed to disperse paths that are causing congestion but without the need for significant topology knowledge to learn which nodes need to be woken up. Using adaptive duty cycle will also work throughout network lifetime, since it does not depend on redundant nodes.

In order to implement this congestion resolution algorithm in a WCM, the first step is to identify the appropriate parameters to detect congestion. The parameters of channel utilization and buffer capacity are chosen, since they can be easily measured in real networks. Channel utilization is the fraction of time that a node detects the channel to be busy during a predefined interval, while buffer capacity is the remaining slots available in the buffer. Both parameters are required for an accurate indication of congestion. The WCM implementation is shown in Figure 6.

Once buffer capacity drops below a specific threshold "AND" channel utilization is above a specific threshold,

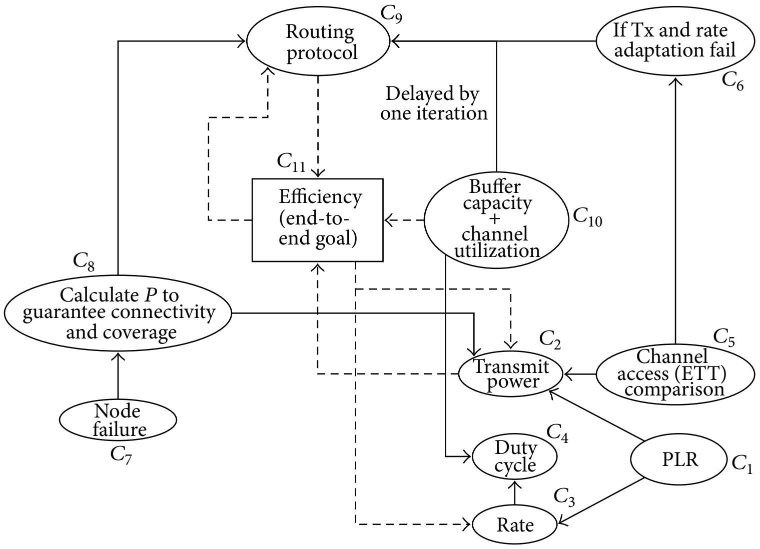

Routing protocol
C_9
If Tx and rate adaptation fail
C_6
Delayed by one iteration
C_{11}
Efficiency (end-to-end goal)
Buffer capacity + channel utilization
C_{10}
C_8
Calculate P to guarantee connectivity and coverage
C_2
Transmit power
C_5
Channel access (ETT) comparison
Node failure
C_7
C_4
Duty cycle
PLR
C_1
C_3
Rate

FIGURE 7: Overall WCM of the network.

concept C_{10} and edges $W_{(10,9)} = 1$ and $W_{(10,4)} = 1$ will be activated. These thresholds are determined according to application requirements. For example, the application may determine that a critical buffer threshold is 10% of the buffer capacity, and the critical value of channel utilization is 90% of M consecutive time slots. Activating C_{10} prompts an increase in duty cycle and triggers route maintenance operation. Again it should be stressed that the output of the WCM only indicates if duty cycle needs to be increased or decreased, not the actual duty cycle value. Note that route maintenance is delayed by one iteration, since it has to be done after the increase in duty cycle. Edges $W_{(9,6)} = -1$ and $W_{(4,6)} = -1$ are activated if there are changes in routing or duty cycle, respectively, in order to ensure that the end-to-end goal of efficiency is not violated.

4.6. The Overall WCM. In previous sections, methods to design WCMs for transmit power, data rate, and duty cycle control; guaranteeing connectivity and coverage; and congestion control were shown. In addition, the end-to-end goal, environment variables, and processes of the proposed WCM system were explained. This section illustrates how these WCMs can be combined to form a reasoning machine capable of considering all the goals, constraints, and variables in the individual WCMs. The overall WCM is shown in Figure 7.

This WCM combines the WCMs explained in previous sections. The overall WCM is activated when there is a change in one of the environmental variables, namely, PLR, ETT, node failure, and buffer capacity and channel utilization. An array $A(t)$ is formulated as in (2), and a weight matrix W is formed with the proper edges activated according to the status of nodes and the protocols implemented by the WCM. The process for formulating $A(t)$ and W is illustrated by the flowchart in Figure 8. After formulating $A(t)$ and W, a multiplication process according to (3) takes place to determine the new values of concepts, which will determine if parameters will change or if certain modules need to be activated, such as increasing transmit power or invoking routing.

In addition to the aforementioned environment variables, another concept is added that is activated if transmit power and data rate control are not able to repair a link, as shown in Figures 7 and 8. In this case, route maintenance is invoked to replace the failed link. This is particularly important in ensuring fairness, detected by ETT imbalance between interfering links, since adapting transmit power may not solve the problem or may cause deterioration in network efficiency. As Figure 7 shows, all dashed edges are those going in and out of the end-to-end goal of efficiency. These edges are activated only if the end-to-end goal of efficiency violates its threshold, $\mathrm{Eff}_{\mathrm{Thresh}}$. Once this happens, the actions taken are determined by the process shown in Figure 8. Note that other end-to-end goals, such as delay, may be incorporated as well. In this paper, we have opted to consider only one end-to-end goal to keep this illustrative example simple. However, in [25], we have illustrated how a WCM machine can consider multiple end-to-end goals, including delay. It is also worth noting that the edges determined by the processes in Figure 8 are only the conditional edges. Other edges are fixed and are activated every time the WCM is triggered. The weights of these edges were specified in the previous sections.

The implementation of the proposed WCM is neither fully quantitative nor qualitative but can rather be regarded as an intelligent decision support system that can produce qualitative decisions, which can be used to determine specific values of different system parameters. Therefore, the advantages of a quantitative WCM are achieved while maintaining the simplicity of a qualitative WCM and avoiding the need for extensive training of the map.

5. Simulation Results

In this section, the performance of the proposed cross-layer framework is evaluated through computer simulations. A

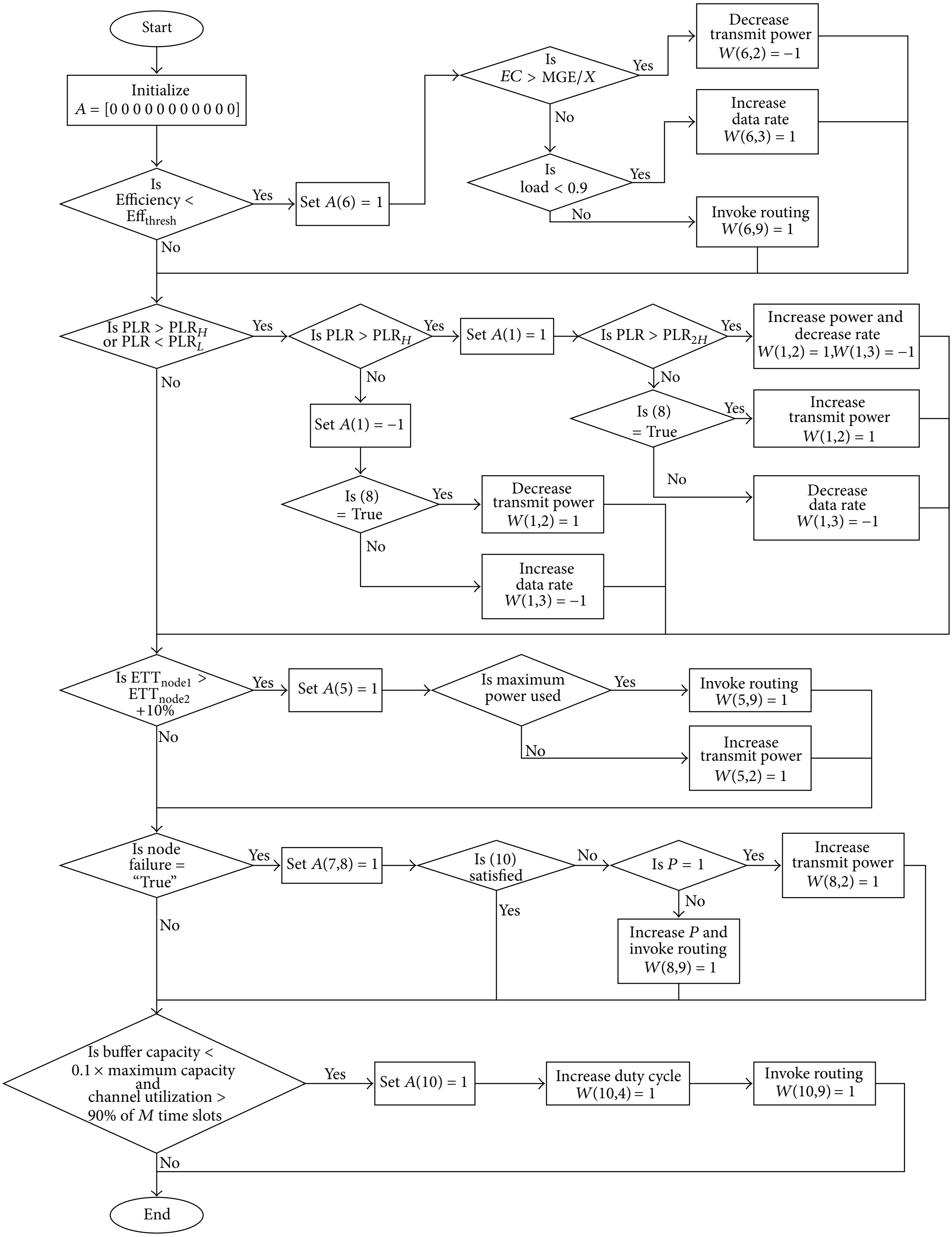

Start
Initialize
$A = [0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0]$
Is Efficiency < $\text{Eff}_{\text{thresh}}$
Yes
Set $A(6) = 1$
Is $EC > \text{MGE}/X$
Yes
Decrease transmit power $W(6,2) = -1$
No
Is load < 0.9
Yes
Increase data rate $W(6,3) = 1$
No
Invoke routing $W(6,9) = 1$
No
Is PLR > PLR_H or PLR < PLR_L
Yes
Is PLR > PLR_H
Yes
Set $A(1) = 1$
Is PLR > PLR_{2H}
Yes
Increase power and decrease rate $W(1,2) = 1, W(1,3) = -1$
No
Is (8) = True
Yes
Increase transmit power $W(1,2) = 1$
No
Decrease data rate $W(1,3) = -1$
No
Set $A(1) = -1$
Is (8) = True
Yes
Decrease transmit power $W(1,2) = 1$
No
Increase data rate $W(1,3) = -1$
No
Is $\text{ETT}_{\text{node1}} > \text{ETT}_{\text{node2}}$ +10%
Yes
Set $A(5) = 1$
Is maximum power used
Yes
Invoke routing $W(5,9) = 1$
No
Increase transmit power $W(5,2) = 1$
No
Is node failure = "True"
Yes
Set $A(7,8) = 1$
Is (10) satisfied
No
Is $P = 1$
Yes
Increase transmit power $W(8,2) = 1$
No
Increase P and invoke routing $W(8,9) = 1$
Yes
No
Is buffer capacity < 0.1 × maximum capacity and channel utilization > 90% of M time slots
Yes
Set $A(10) = 1$
Increase duty cycle $W(10,4) = 1$
Invoke routing $W(10,9) = 1$
No
End

FIGURE 8: Process to determine the edge weights when the WCM is triggered.

Table 1: Simulation parameters.

Parameter	Value
Grid size	500 × 500 m
Number of nodes	169, 256, 400, 625, 900, 1225, and 1600
Transmit power levels	[−12, −6, 0, 2, 4] dBm
Data rates	[20, 40, 250] kbps
Cycle period	100 ms
Duty cycle	[0.25, 0.5, 0.75, 1] × 100 ms
Packet size	80 bytes
Packet generation rates (λ)	[0.007, 0.009, 0.02] pckts/sec
Initial battery power	5 Ahr
Power consumption	Rx: 26 mAhr Tx: 26 mAhr + (Tx power × packet size/data rate) Sleep: 0.3 μAhr × sleep time
Initial buffer capacity	100 packets

network simulator was created in MATLAB, in which we consider different network topologies to study the performance of the system under different scenarios. The lognormal propagation model with shadowing and multipath fading is used, and correlated shadowing is also considered to ensure a realistic propagation environment. Packet generation at every node is done according to a random Poisson process with rate λ pckts/sec. The main simulation parameters are shown in Table 1. As this table illustrates, the transmit power and data rate parameters of IEEE Standard 802.15.4 have been utilized, since it is the standard generally used in WSN.

The WCM system is compared to several known protocols. Particularly, the network management protocols TABU-RCC [6] and EECCR [7], and the transmit power and data rate adaptation protocol known as Symphony [13] are chosen. TABU-RCC and EECCR were chosen because their end-to-end goal is to maximize network lifetime in the presence of coverage and connectivity constraints, which is similar to the goals and constraints considered by our WCM system. Since TABU-RCC and EECCR do not consider transmit power or data rate adaptation, the WCM system is also compared against a network with the Symphony adaptation protocol. In addition, comparisons of the WCM system with a reduced set of features are included. Particularly, the comparisons include a WCM with no transmit power, data rate, or duty cycle adaptation, designated as "WCM-no adaptation," and a WCM where the capabilities to adjust the value of P are switched off, designated as "WCM-no management." As a baseline for the comparisons, a regular network with no transmit power or data rate adaptations and no network management protocols is included. Note that all systems are simulated with the MAC layer of the IEEE Standard 802.15.4. Also, all simulations are repeated enough times to ensure a confidence level of 95%.

5.1. Evaluation Using a Uniform Random Topology. In this section, the performance of the proposed system is evaluated using a uniform random topology, similar to the one that was previously illustrated in Figure 2. The metrics of network lifetime, throughput, and PLR and the amount of information that the network is able to dissipate before it reaches its lifetime are considered. Note that the amount of information dissipated before the network expires reflects the efficiency of the system in utilizing the available resources in transmitting packets. Network lifetime is defined as the time spanning from the start of network operation until the time when the remaining nodes can no longer guarantee full coverage and connectivity of the network. This particular definition was chosen as it reflects the capability of the systems to guarantee the QoS constraints of the network.

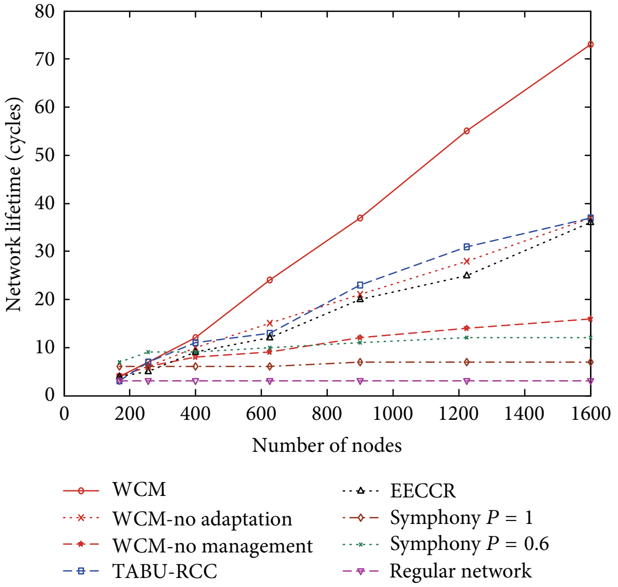


Figure 9: Network lifetime of WCM compared to other systems.

In all simulations conducted, time is divided into cycles. In every cycle, nodes wake up with probability P. In WCM, WCM-no adaptation, TABU-RCC, and EECCR, P changes throughout network lifetime to maximize energy efficiency. In Symphony, WCM-no management, and the regular network, there will be a predefined value for P that will not change throughout the lifetime of the network, since these protocols do not define methods for adapting P.

Nodes will transmit packets according to the packet generation rate, routing path, transmit power, and data rate specified by the WCM or the systems under comparison. The same packet generation rate, initial buffer capacity, initial battery power, packet size, and cycle period are used in all simulations to ensure fair comparisons between different protocols. The basic "shortest path" routing protocol is also used in all systems. In the first experiment, the lifetime of the proposed system is evaluated under different numbers of nodes. The results are shown in Figure 9.

As Figure 9 shows, the proposed WCM system achieves the highest lifetime results compared to other systems, especially in larger networks. In smaller networks, the WCM is forced to switch most nodes in the network to active mode in every duty cycle to guarantee connectivity and coverage. Therefore, there are not as many redundant nodes to be managed by the WCM. In larger networks the capabilities

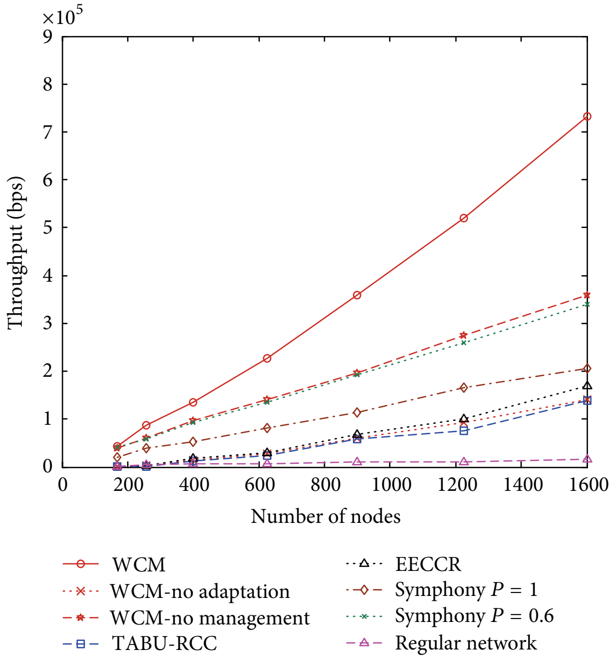


Figure 10: Throughput of WCM compared to other systems.

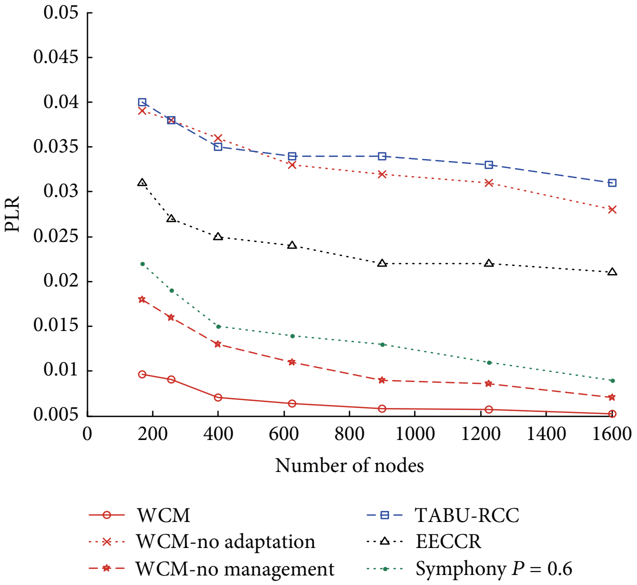


Figure 11: PLR of WCM compared to other systems.

of management systems become more apparent. TABU-RCC comes second as it has capabilities to choose a suitable number of nodes to be active in every duty cycle. Figure 9 also shows that when transmit power, data rate, and duty cycle adaptation capabilities of the WCM system are switched off (WCM-no adaptation), the lifetime results approach those of TABU-RCC and EECCR, since the processes considered within those systems are similar to WCM-no adaptation. The Symphony protocol with $P = 0.6$ achieves longer lifetime than Symphony with $P = 1$. This is because, with $P = 1$, all nodes are switched on in every duty cycle, causing network resources to become depleted quickly. Note that $P = 0.6$ was chosen because it is the minimum value that can guarantee connectivity and coverage in the smallest network (169 nodes). Thus, $P = 0.6$ and $P = 1$ are included since they account for the minimum and maximum values that can be used. When the capabilities of the WCM to adjust the value of P are switched off (WCM-no management), the results are close to the Symphony system, since they both consider similar processes. WCM-no management was also simulated using $P = 0.6$, and it achieves slightly better results than Symphony, since it considers congestion. In all systems with no management protocols, the performance does not improve significantly as the number of nodes increases. This is because, with constant P, a larger number of nodes are switched on in every duty cycle as the total number of nodes in the network increase. Thus, redundant nodes are not being utilized efficiently.

In the next experiment, the throughput and PLR of the proposed WCM system are compared to other protocols. The results are shown in Figures 10 and 11. Note that the PLR results of the regular network and Symphony with $P = 1$ are not shown, since their PLR was significantly larger than other protocols.

As Figures 10 and 11 show, the proposed WCM system achieves the highest throughput and the lowest PLR results. In larger networks, the difference in throughput is larger in the favor of WCM, showing its ability to adapt system parameters efficiently. The WCM system considers data rate in the end-to-end goal (6) and tries to maximize it whenever possible as long as other constraints, such as PLR, are not violated. In addition, transmit power cannot be decreased unless $\text{PLR} \leq \text{PLR}_L$. This ensures that decreasing transmit power will not lead to packet failures, which leads to low PLR and high throughput. WCM-no management and Symphony with $P = 0.6$ also achieve high throughput, showing the impact of transmit power and data rate adaptation in achieving good throughput results. Figures 10 and 11 show that Symphony with $P = 0.6$ achieves similar throughput and PLR results to WCM-no management, and WCM-no adaptation achieves similar results to EECCR, confirming the results in Figure 9.

In the next experiment, the capability of the proposed system to utilize resources in disseminating information is evaluated. This is done by counting the total number of bits transmitted by all nodes in the network throughout its lifetime. The results are shown in Figure 12.

The results in this figure show that the WCM system is able to transmit the largest number of bits before the network reaches its lifetime. If this observation is combined with those of Figures 10 and 11, we can see that WCM is able to utilize network resources efficiently, since a large number of packets are transmitted with high throughput and low PLR. The networks employing Symphony also transmit large numbers of packets, especially in larger networks. This is because the fixed value of P forces a larger number of nodes to be switched on and transmit packets. However, from Figure 10, we have

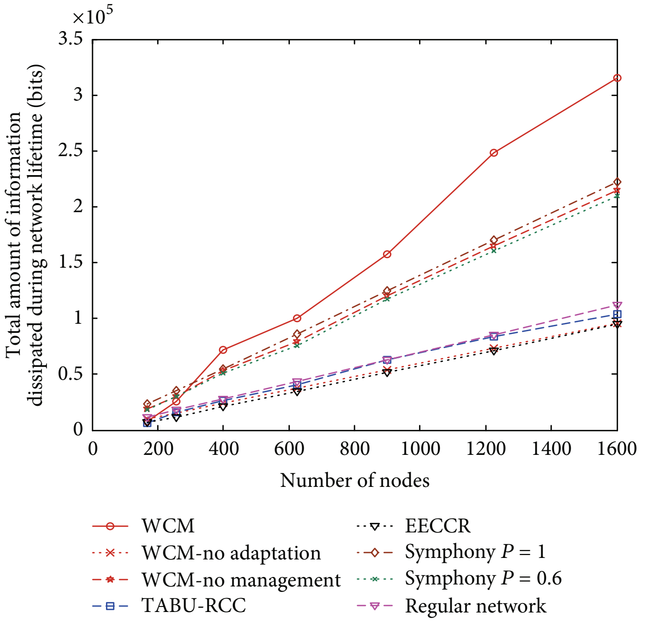


Figure 12: Ability of WCM to dissipate information compared to other systems.

seen that Symphony with $P = 1$ does not achieve good throughput results, due to high interference and collisions. As with previous experiments, WCM-no adaptation achieves similar results to TABU-RCC and EECCR, and WCM-no management achieves similar results to Symphony.

5.2. Evaluation Using a Grid Topology. In this section, the performance of the proposed system is evaluated using a grid topology, in order to test the system operation in different scenarios. As in random topology simulations, the same routing protocol, packet generation rates, initial buffer capacity, initial battery power, packet size, and cycle period are used for all systems under comparison. The topology used in this section is illustrated in Figure 13.

The same experiments that were performed for the random topology simulations are repeated in this section. The simulation results for network lifetime, throughput, and PLR are shown in Figures 14, 15, and 16, respectively. Note that the results for the total amount of information dissipated during network lifetime were not shown due to its similarity with the results of the random topology (Figure 12).

Figures 14–16 confirm the results of the random topology experiments. The proposed WCM system achieves good results in terms of network lifetime, throughput, and PLR. The performance also improves significantly in larger networks, since management capabilities become more apparent as redundant nodes increase. This clearly illustrates the efficiency of the proposed protocol in utilizing network resources in different network scenarios. Also, WCM-no adaptation achieves similar results to EECCR, and WCM-no management achieves similar results to Symphony. Note that TABU-RCC does not achieve good lifetime results in the grid topology because it was primarily designed for random topologies.

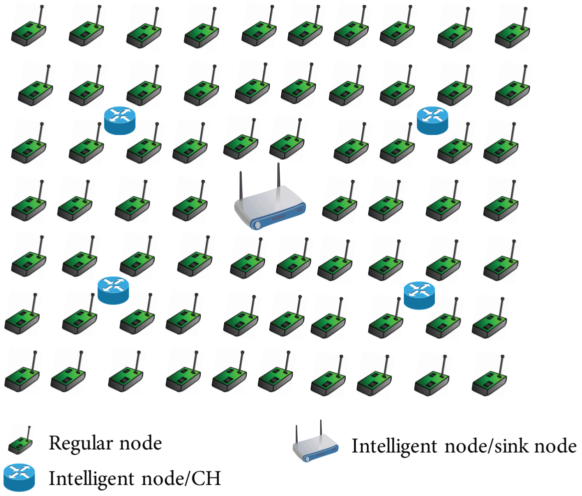


Figure 13: Grid topology used in simulations.

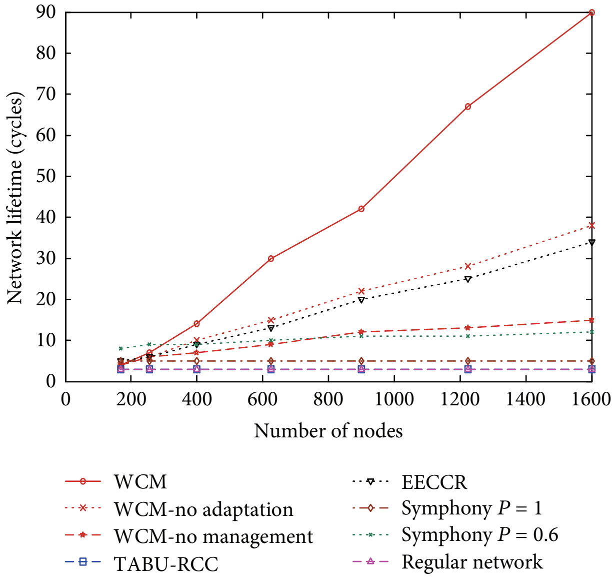


Figure 14: Network lifetime of WCM and other systems in grid topology.

It is important to note that the efficient performance of WCM is achieved while maintaining low computational complexity. The execution of WCM is simply a matrix multiplication operation. The activated concepts determine the input array, and the status of the nodes determines the WCM matrix. On the other hand, in TABU-RCC, the best network configuration is discovered in every duty cycle through an extensive search operation. The algorithm starts with the configuration where all nodes are active, and then nodes are randomly switched off to find new energy-efficient configurations. The algorithm ends when switching off more nodes will no longer guarantee connectivity and coverage. Also, EECCR needs to determine new scheduling sets every

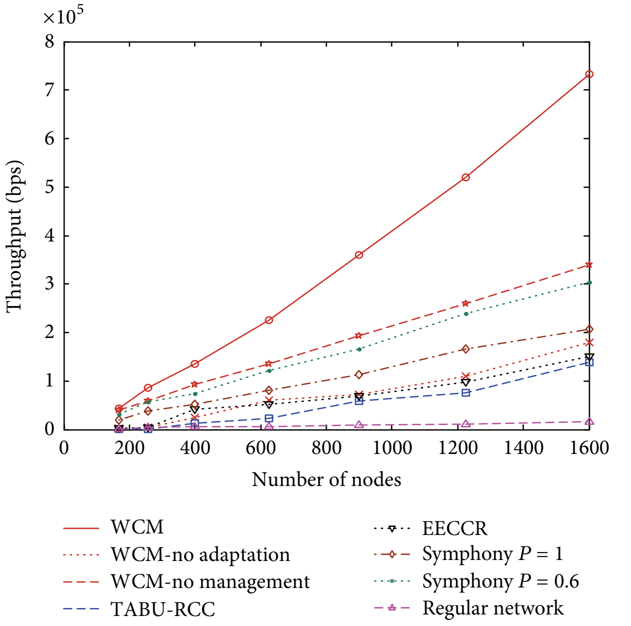


Figure 15: Throughput results of WCM and other systems in grid topology.

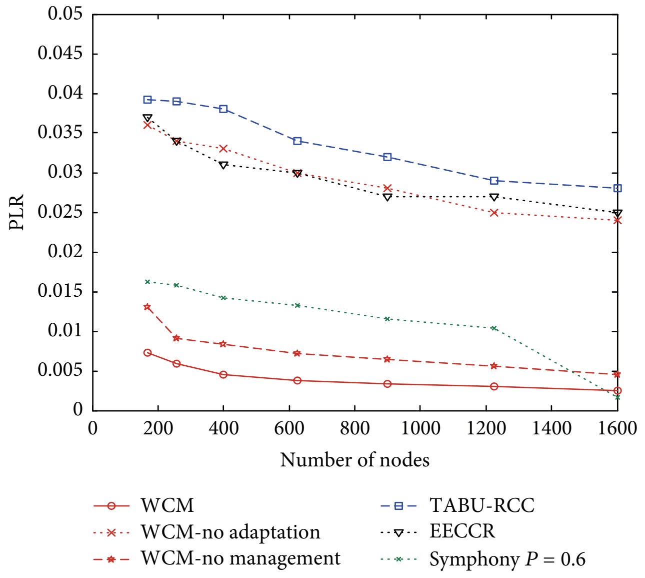


Figure 16: PLR of WCM and other systems in grid topology.

time there is a change in network topology. Thus, the processing time of TABU-RCC and EECCR are significantly longer than WCM. If TABU-RCC and EECCR are executed every few duty cycles, they will not be able to react quickly to network changes.

Note that WCM, TABU-RCC, and EECCR are network management protocols that may be implemented on top of existing network functionalities. For example, they all require the existence of routing protocols that can establish paths between nodes. The complexity of WCM is adjustable according to the concepts required. For example, if network load is light, concepts that deal with congestion can be removed. This leads to a smaller WCM matrix and shorter processing time. The qualitative nature of WCM enables it to avoid long training processes. However, the proposed design ensures that it is able to produce some decisions that can be implemented quantitatively, such as transmit power and data rate adaptations. For concepts where quantitative decisions are produced, there is no need to run parallel protocols. The proposed WCM can also be implemented in a distributed way between the CHs and the sink node, thus avoiding the need for heavy processing at the sink node.

Furthermore, it is important to note that the complexity of the WCM system grows with the size of the network. An additional WCM machine needs to be implemented in every new CH. Also, if the number of nodes within a cluster grows, there will be a higher probability that the WCM will be triggered to operate. However, this is true of most network management systems that are implemented in a clustered hierarchy. Since the WCM system avoids long search processes, it will be quite scalable with larger networks and will respond faster than systems that employ optimization problems. In addition, the WCM system can be programmed to avoid frequent triggers simply by adjusting the thresholds that cause the WCM to operate. Moreover, the operation of the WCM can be delayed for a specific time after a threshold has been violated in case the problem is temporary and can resolve itself.

WCM does not require a significant increase in communication overhead. Information necessary for the execution of WCM, such as PLR levels and battery consumption levels, can be piggybacked on transmitted data packets. Occasional overhead packets can be sent if certain nodes require fast intervention. Note that in WSN nodes transmit sensed information regularly, so there will always be data packets on which information can be piggybacked.

5.3. On Choosing an Appropriate Activation Function for Guaranteeing Connectivity and Coverage. In Section 4.4, it was mentioned that an appropriate choice for the activation function $\varphi(nP)$ is necessary for Inequality (10) to be used efficiently in guaranteeing connectivity and coverage. The choice of this function depends on the area to be covered as well as sensing and communication ranges of nodes, and it is done offline. This section illustrates how $\varphi(nP)$ is chosen and the impact of this choice on network performance.

An experiment that is divided into two parts is conducted. In the first part, computer simulations are performed with parameters $k = 1$, $r_s = 100$ m, and $r_c = 200$ m. The area to be covered is a square of size 500 m × 500 m. The number of nodes $n = [100, 110, 120, \ldots, 1600]$ nodes. For every value of n, 1000 simulation runs are performed, whereby every run consists of a new random deployment of nodes. In every run, the current deployment of nodes is checked to find if it achieves connectivity and coverage. After 1000 runs, the probability that the network is connected and covered using this value of n is calculated. This is repeated for values of $P = [0.1, 0.15, \ldots, 0.3]$. Simulation results are plotted in Figure 17.

In the second part of this experiment, Inequality (10) is used with the same values of k, r_s, r_c, and P that

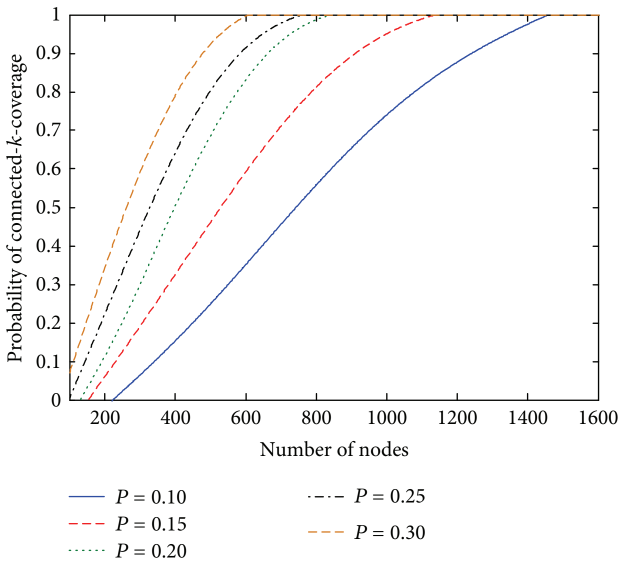


Figure 17: Probability of connectivity and coverage using different values of P.

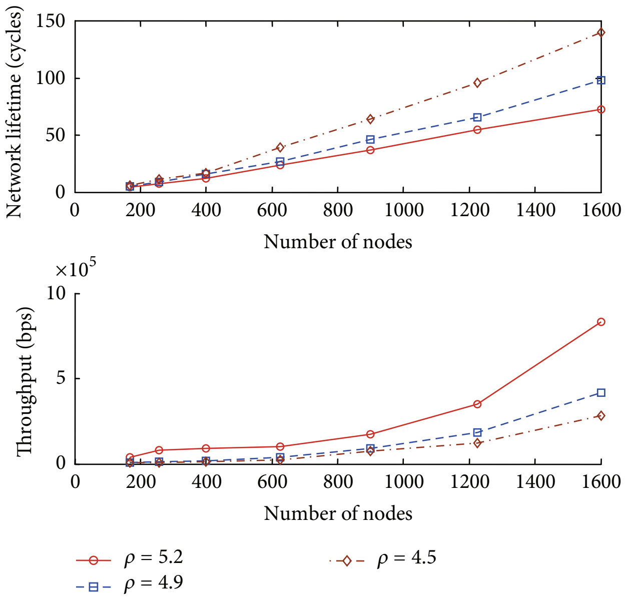


Figure 18: Network lifetime and throughput results at different values of ρ.

were used in the simulations. A slowly growing function $\varphi(nP) = (\log\log(nP))^{\rho}$ is chosen to provide fine granularity for choosing P that satisfies Inequality (10), and the value of ρ provides some flexibility in the speed at which this function grows. However, a different $\varphi(nP)$ can be chosen provided that it is slowly growing. A fast growing function may lead to values of P that are either too conservative (activates too many nodes) or too low to guarantee connectivity and coverage. We use values of $\rho = [4.0, 4.1, 4.2, \ldots, 5.5]$ for experimentation. For every value of ρ, the minimum number of nodes n that would satisfy Inequality (10) is evaluated. Then, simulation results from Figure 17 are used to check the probability of connectivity and coverage using this value of n. This is repeated for every value of P used in the simulations. The results are shown in Table 2.

As Table 2 shows, with $\rho = 4.0$, the minimum values of n that satisfy Inequality (10) for different values of P are 876, 584, 438, 351, and 292 nodes, respectively. However, the simulation results specify that the probability of connected-k-coverage is less than 0.63 using this value of n for all values of P. Therefore, $\rho = 4.0$ does not guarantee connectivity and coverage. On the other hand, with $\rho = 5.4$, the minimum values of n that satisfy Inequality (10) for different values of P are 1652, 1102, 826, 661, and 551 nodes, respectively. Here, the simulation results specify that the network is connected-k-covered with probability greater than 0.96 for all values of P. Thus, this value of ρ provides a better guarantee for connectivity and coverage. We have used $\rho = 5.4$ in all simulations in Sections 5.1 and 5.2. Thus, in order to use Inequality (10) in the WCM system, a small offline experiment can be performed using the given target area to be covered and the sensing ranges of nodes to determine the appropriate value of ρ. It is important to note that $\varphi(nP)$ was chosen in this way so that there is no need to change it for different areas and sensing ranges of nodes. Only the appropriate value of ρ needs to be determined for every network once before the network becomes operational.

To illustrate the impact of varying ρ on network performance, network simulations are performed similar to the ones in Sections 5.1 and 5.2, using the WCM system. A uniform random topology with the same simulation parameters as the ones used in Section 5.1 is utilized, and system performance is evaluated under values of $\rho = \{4.5, 4.9, 5.4\}$. Values of ρ lower than 4.5 provide poor guarantees for connectivity and coverage, as shown in Table 2, and thus were not considered. Metrics of network lifetime and throughput are considered, and the results are illustrated in Figure 18.

This figure shows that network lifetime increases with lower values of ρ, while throughput decreases. This is because, with lower values of ρ, a smaller number of nodes are switched on every duty cycle. Thus, the rate of consumption of network resources decreases causing lifetime to increase. However, switching on a smaller number of nodes with lower values of ρ means that the network is less dense. Therefore, the total number of transmitted packets decreases and connectivity decreases, which may cause higher probability of transmission failure and lower throughput.

In order to test the efficiency of the chosen function in guaranteeing connectivity and coverage, an experiment is performed, where the average percentage of the total area that is covered by at least one sensor node is measured in every time slot during the lifetime of the network. This percentage is calculated by measuring the fraction of the area that is within the sensing range (r_s in Section 4.4) of at least one sensor node and then multiplying by 100. In this experiment, a maximum number of nodes of 625 is considered, since connectivity and coverage problems are more probable in networks with smaller node densities. In addition, node deployment is done in a uniform random fashion, since

Table 2: Values of ρ and their corresponding probability of connected-k-coverage.

ρ	$P = 0.1$		$P = 0.15$		$P = 0.2$		$P = 0.25$		$P = 0.3$	
	Min. n	Probability	Min. n	Probability	Min. n	Probability	Min. n	Probability	Min. n	Probability
4.0	876	0.63	584	0.57	438	0.57	351	0.55	292	0.57
4.1	904	0.66	603	0.60	452	0.60	362	0.57	302	0.60
4.2	934	0.68	623	0.62	467	0.63	374	0.59	312	0.62
4.3	966	0.71	644	0.65	483	0.66	387	0.62	322	0.64
4.4	1002	0.74	668	0.67	501	0.69	401	0.64	334	0.67
4.5	1041	0.77	694	0.70	521	0.72	417	0.67	347	0.69
4.6	1084	0.80	723	0.74	542	0.75	434	0.70	362	0.72
4.7	1131	0.83	754	0.77	566	0.79	453	0.73	377	0.75
4.8	1184	0.87	789	0.80	592	0.82	474	0.77	395	0.78
4.9	1242	0.90	828	0.84	621	0.86	497	0.80	414	0.81
5.0	1307	0.93	871	0.87	654	0.89	523	0.83	436	0.85
5.1	1379	0.97	919	0.92	690	0.92	552	0.87	460	0.88
5.2	1460	1.00	973	0.95	730	0.95	584	0.90	487	0.91
5.3	1550	1.00	1034	0.97	775	0.97	620	0.93	517	0.94
5.4	1652	1.00	1102	0.99	826	1.00	661	0.96	551	0.97
5.5	1767	1.00	1178	1.00	884	1.00	707	1.00	589	1.00

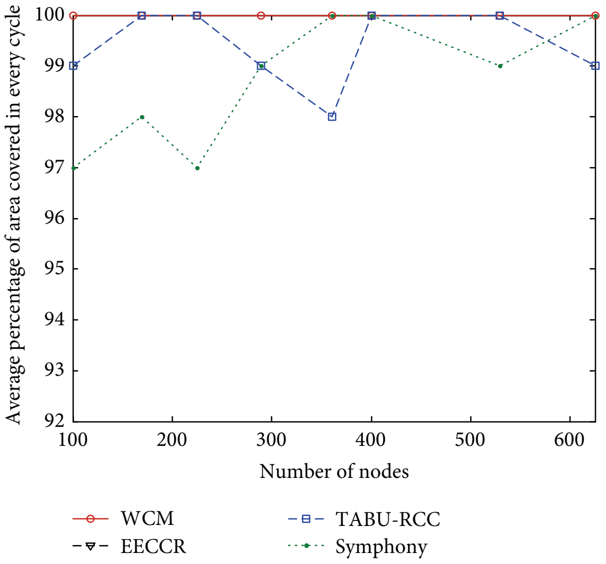


Figure 19: Average coverage results of WCM compared to other protocol.

this is the more challenging scenario for connectivity and coverage. The performance of WCM is compared to EECCR, TABU-RCC, and Symphony. Simulation results are shown in Figure 19.

The results in this figure show that the WCM system achieves 100% connectivity and coverage in every duty cycle for all network sizes. This proves that the chosen function performs efficiently and is able to switch an appropriate number of nodes to active mode during network lifetime, without the need for complicated optimization operations. Furthermore, if we combine these results with the ones in Figure 9, it can be deduced that the number of activated nodes in every cycle with WCM achieves good energy efficiency that leads to the maximization of network lifetime. Moreover, Figure 19 shows that EECCR also achieves 100% connectivity and coverage, while TABU-RCC and Symphony do not. This is because Symphony has no means of guaranteeing connectivity and coverage, while TABU-RCC is a heuristic algorithm that may not achieve 100% coverage and connectivity in every cycle.

6. Conclusions

In this paper, a cross-layer framework for network management in WSN based on the WCM tool was presented. The proposed system is able to perform efficient reasoning while considering multiple objectives and constraints. By maintaining an overview of all network elements, the WCM is able to ensure that they operate coherently. The WCM continuously monitors the required QoS levels specified by the user and takes fast and efficient actions whenever those levels are violated. This is achieved while avoiding high complexity typically required by optimization problems or long search operations.

To evaluate the performance of the proposed system, extensive computer simulations were conducted, and the WCM was compared against other well-known protocols. The WCM system showed ability to utilize network resources efficiently and adapt to different network scenarios and conditions by adjusting system parameters accurately. For these reasons, the WCM outperforms other protocols in metrics of lifetime, throughput, and PLR.

In future work, the authors will present a theoretical framework that models the behavior of the proposed system. This model will be used to further analyze system functionality and discover new methods to improve system performance.

Acknowledgments

The authors would like to thank the NSERC DIVA Network and the Ontario Research Fund (ORF) WiSense project for supporting this work.

References

[1] M. Ibnkahla, *Adaptation and Cross-Layer Design in Wireless Networks*, CRC Press, Boca Raton, Fla, USA, 2008.

[2] N. M. Freris, H. Kowshik, and P. R. Kumar, "Fundamentals of large sensor networks: connectivity, capacity, clocks, and computation," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1828–1846, 2010.

[3] A. Ananda, M. Chan, and W. Ooi, *Mobile, Wireless, and Sensor Networks: Technology, Applications, and Future Directions*, John Wiley & Sons, Hoboken, NJ, USA, 2006.

[4] M. Ibnkahla, *Wireless Sensor Networks: A Cognitive Perspective*, CRC Press, Boca Raton, Fla, USA, 2012.

[5] C. Facchini and F. Granelli, "Towards a model for quantitative reasoning in cognitive nodes," in *Proceedings of the IEEE Globecom Workshops*, pp. 1–6, Honolulu, Hawaii, December 2009.

[6] A. Chamam and S. Pierre, "On the planning of wireless sensor networks: Energy-efficient clustering under the joint routing and coverage constraint," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1077–1086, 2009.

[7] Y. Jin, L. Wang, J.-Y. Jo, Y. Kim, M. Yang, and Y. Jiang, "EECCR: an energy-efficient m-Coverage and n-Connectivity routing algorithm under border effects in heterogeneous sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1429–1442, 2009.

[8] J. Kang, Y. Zhang, and B. Nath, "TARA: topology-aware resource adaptation to alleviate congestion in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 919–931, 2007.

[9] L. Q. Tao and F. Q. Yu, "ECODA: enhanced congestion detection and avoidance for multiple class of traffic in sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1387–1394, 2010.

[10] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and coverage guarantees through distributed coordinate-free sensor activation," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 470–483, 2011.

[11] S. Panichpapiboon, G. Ferrari, and O. K. Tonguz, "Optimal transmit power in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1432–1446, 2006.

[12] X. Zhao, E. Bdira, and M. Ibnkahla, "Joint adaptive modulation and adaptive MAC protocols for wireless sensor networks," Internal Report, Queen's University, 2012.

[13] K. Ramachandran, R. Kokku, H. Zhang, and M. Gruteser, "Symphony: synchronous two-phase rate and power control in 802.11 WLANs," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1289–1302, 2010.

[14] Y.-L. Kuo, K.-W. Lai, F. Y.-S. Lin, Y.-F. Wen, E. H.-K. Wu, and G.-H. Chen, "Multirate throughput optimization with fairness constraints in wireless local area networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2417–2425, 2009.

[15] 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. 738–754, 2006.

[16] M. A. Razzaque, M. M. Alam, M. Mamun-Or-Rashid, and C. S. Hong, "Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 157–162, Las Vegas, Nev, USA, January 2008.

[17] S. Ratnaraj, S. Jagannathan, and V. Rao, "OEDSR: Optimized energy-delay sub-network routing in wireless sensor network," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control, (ICNSC'06)*, pp. 330–335, April 2006.

[18] J. A. Dickerson and B. Kosko, "Virtual worlds as fuzzy cognitive maps," in *Proceedings of the IEEE Annual Virtual Reality International Symposium*, pp. 471–477, September 1993.

[19] M. Glykas, *Fuzzy Cognitive Maps: Advances in Theory, Methodologies, Tools, and Applications*, Springer, Berlin, Germany, 2010.

[20] Y. Miao, C. Miao, X. Tao, Z. Shen, and Z. Liu, "Transformation of cognitive maps," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 114–124, 2010.

[21] A. K. Tsadiras, "Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps," *Information Sciences*, vol. 178, no. 20, pp. 3880–3894, 2008.

[22] D. E. Koulouriotis, I. E. Diakoulakis, and D. M. Emiris, "Realism in fuzzy cognitive maps: Incorporating synergies and conditional effects," in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, pp. 1179–1182, Melbourne, Australia, December 2001.

[23] X. Luo, X. Wei, and J. Zhang, "Guided game-based learning using fuzzy cognitive maps," *IEEE Transactions on Learning Technologies*, 2008.

[24] G. Yang and D. Qiao, "Critical conditions for connected-k-coverage in sensor networks," *IEEE Communications Letters*, vol. 12, no. 9, pp. 651–653, 2008.

[25] A. El Mougy, *A cognitive framework for WSN based on WCM and Q-learning, [Ph.D. thesis]*, Queen's University, Kingston, Canada, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 190813, 9 pages
http://dx.doi.org/10.1155/2013/190813

Research Article

An Internet of Things Approach for Managing Smart Services Provided by Wearable Devices

Pedro Castillejo, José-Fernán Martínez, Lourdes López, and Gregorio Rubio

Next-Generation Networks and Services (GRYS) Research Group, Research Center on Software Technologies and Multimedia Systems for Sustainability (CITSEM), Technical University of Madrid (UPM), "La Arboleda-Campus Sur" Building, Crt. Valencia, Km. 7, 28031 Madrid, Spain

Correspondence should be addressed to Pedro Castillejo; pcastillejo@diatel.upm.es

Received 28 September 2012; Revised 22 January 2013; Accepted 1 February 2013

Academic Editor: Marc St-Hilaire

The Internet of Things (IoT) is growing at a fast pace with new devices getting connected all the time. A new emerging group of these devices is the wearable devices, and the wireless sensor networks are a good way to integrate them in the IoT concept and bring new experiences to the daily life activities. In this paper, we present an everyday life application involving a WSN as the base of a novel context-awareness sports scenario, where physiological parameters are measured and sent to the WSN by wearable devices. Applications with several hardware components introduce the problem of heterogeneity in the network. In order to integrate different hardware platforms and to introduce a service-oriented semantic middleware solution into a single application, we propose the use of an enterprise service bus (ESB) as a bridge for guaranteeing interoperability and integration of the different environments, thus introducing a semantic added value needed in the world of IoT-based systems. This approach places all the data acquired (e.g., via internet data access) at application developers disposal, opening the system to new user applications. The user can then access the data through a wide variety of devices (smartphones, tablets, and computers) and operating systems (Android, iOS, Windows, Linux, etc.).

1. Introduction

Modern society is looking for user-friendly aiding systems, able not only to remotely monitor the health of the elderly and people suffering from chronic diseases, but also to find a safe and efficient routine to practice some sport or single exercises in an outdoor or indoor environment (such as gymnasium), in order to improve each person's level of fitness and health. In this context, Internet of Things (IoT)-related systems are able to bring solutions.

The IoT paradigm can be built using wireless sensor networks (WSNs) as the leading technology to acquire and manage data. Connecting other smart elements to a WSN (smartphones, watches, tablets, etc.) may also improve the user experience in the IoT, and it could act as a starting point for the use of the technology. If the smart devices are wearable, the first technology-access barrier is broken: the user just has to "wear" the technology as a daily-life garment.

A differential value of a WSN node is the fact that any external sensor can be connected in an easy way to itself, and the data sensing does not depend on the network management. For example, if the application needs biometric or human physiological parameters (blood pressure, heart rate, breathing rate, etc.) an external sensor must be connected in some way to the node. A very easy and fast solution is using wireless communications protocols, such as Bluetooth. However, in this scenario a new challenge springs up: the existence of different types of devices or platforms (as there is no standardization in this kind of sensors), so it is desirable to abstract the hardware features and protocols from high-level layers. This can be done with an intermediate level, called middleware [1]. It would be very useful if this middleware level is able both to process environmental measures, but, and receiving user parameters with several sensors: localization, speed, health status, preferences, and so forth. In this way, a context-awareness system could be developed, and new services are able to be included in the applications without any other external intervention.

In this paper, we present a physical application involving a WSN as the foundation of a sports-related scenario. Gathering environmental and human physiological data and storing a user's profile can lead into an autonomous physical condition performance system, where the preferences and needs of every single user are evaluated to obtain safe and optimum exercise routines. Our proposal has three key novel components. The first one is the integration of several wearable devices in the Internet of Things world. In order to integrate these devices (provided with only Bluetooth connection), we have implemented a dual-protocol WSN/Bluetooth node. In our proposal, we use two of these nodes: one connected to the wearable health-data monitor and another node connected to the smartphone or smartwatch. With these nodes, all the information from/to the wearable devices can be managed in the same way as the information from other WSN nodes. Any new wearable device can be included in our system, with the only limitation of being Bluetooth compliant, and the services offered by this new device can be discovered and used as well. The second contribution is the development of an ontology, included within the service-oriented semantic middleware, to model the services provided by the WSN. With the semantically annotated services it is possible to compose new services based on the existing single services, widening the platform for future applications. The third key contribution is the integration of an enterprise service bus (ESB) in a WSN for an IoT-based application. With the ESB, all the services published by the WSN nodes (including the services available at the wearable devices, such as heart rate and body temperature) are available for third-party applications. If the ESB is connected to the internet, any external application can use our RESTful API to make requests to the published services.

This paper is divided as follows. In Section 1, we introduce wireless sensor networks features and applications as the foundations of an Internet of Things-designed application. In Section 2, we present a series of works related with healthcare and smart spaces using WSNs. We present the contextualization of the requirements for a sports monitoring system with wearable devices in Section 3. In Sections 4 and 5, an ESB platform for middleware architectures and ontologies interoperability is described. Section 6 presents the testing scenario description. Section 7 includes the conclusions and future work to develop from this paper.

2. Related Work

The Internet of Things is gaining the attention of researchers worldwide. In [2], Miorandi et al. present a survey of technologies, applications, and research challenges for the IoT. The most relevant challenges are related with distributed systems technology and distributed intelligence. Related to applications, the survey remarks the idea of RFID as the base of the IoT paradigm, but other technologies are expected to emerge in order to bring the paradigm to real-life applications, identifying six ones which they believe can play a leading role in the adoption of IoT technologies: environmental monitoring, smart cities, smart business/inventory and product management, smart homes/smart building management, healthcare, and security and surveillance.

Due to the characteristics of wireless sensor networks, they are getting an important place in e-Health applications and assisted living. WSNs are flexible to integrate into health environments, not intrusive, not high priced, small, easily portable, and, in some cases, wearable.

In [3], Mileo et al. present an intelligent home environment to monitor a patient in a context-aware setting. The goal is to control the evolution of patient's health and home environment. They propose a middleware environment to provide routing, topology control, and data aggregation, so they can carry the data collected by the sensors to feed the reasoning system. The system is not detailed in the paper, and the latter does not clarify whether the system has been deployed in a real WSN or not, though.

An already implemented solution is presented by Wood et al. in AlarmNet [4]. It is an assisted living and residential monitoring network for pervasive adaptive healthcare in assisted living communities with residents or patients with diverse needs. The system is composed by an extensible heterogeneous network middleware, bidirectional network information flow protocols for environmental systems, resident data flow analysis, and a query protocol for efficient streaming. The researchers built a test bed with MicaZ and Telos Sky nodes, where the system was deployed and tested.

Philips Research Europe and the Department of Biomedical Engineering at the Imperial College London have developed an end-to-end and generic infrastructure [5] that allows the application optimization by developing services distributed over a "Wireless Accessible Sensor Populations" (WASPs) network. Their goal is optimizing the battery life of the nodes monitoring needs of individual persons. They validated the system with realistic elderly care scenarios: monitoring activities of daily living, hypothermia, and ECG arrhythmia. This project is focused overall on controlling physical parameters from elderly people, acting in case of danger, taking into account the different vital data from each person. For this end, the person must bring along a node, and it might not be accepted by everybody due to its size.

In [6], the design problems of an ambient home care system relying on WSN are addressed. The approach is composed by traditional WSN nodes and also includes personal mobile devices, deploying a heterogeneous sensor network, integrated into an OSGi framework. The most notorious weakness is that this proposal was based on the idea that personal mobile phones and medical equipments would have integrated, in a near future, ZigBee protocol.

Qixin et al. present in [7] an architecture for assisted living that allows independent parties work together in a dependable, secure, and low-cost fashion with predictable properties. This approach is based on an Assisted Living Service Provider (ALSP) who provides a server that collects and maintains encrypted assisted persons' (APs) records. ALSP can be a third party distinct from APs, communication providers, and clinicians; or it can be part of a hospital or a similar facility by using standards such as XML and Java technology in WLAN networks.

In [8], the EU funded a project called ANGEL (Advanced Networked embedded platform as a Gateway to Enhance quality of Life) focuses on the development and deployment of wireless sensor networks building ambient intelligence systems for assisted living and personal health monitoring. The provision of security services such as confidentiality and authentication is a fundamental requirement for ANGEL in order to ensure the safety and privacy of the users; the development is challenging due to the high mobility of users accessing a multitude of wireless sensor networks. Based on keying material distribution, they provide a solution for the secure configuration of sensors and gateways. This ensures that users can interact with their system in an easy, transparent, and safe way. The proposal architecture is based on the ZigBee standard and consists of body and home sensor networks and gateway nodes. Although the ANGEL system is very concerned about security issues, they do not perform a full ambient assisted living platform.

Smart devices (phones, watches, etc.) have become common life elements. Their capability to acquire and to process complex data is growing with new devices and new operating systems (iOS, Android, etc.). Wireless devices can monitor not only environmental parameters (temperature, humidity, etc.) but also several human body parameters, like blood pressure, heart rate, breathing rate, body temperature, and so forth. The communication between these devices, conforming a body area network (BAN), and internet-capable devices (as the already mentioned smart ones) produce a wide new range of protocols and applications, as presented in several proposals [9–15].

Moreover, several sensor-specific ontologies have been proposed. For example, sensorML [16] specifies models and XML encoding providing a framework within which the geometric, dynamic, and observational characteristics of sensors and sensor systems can be defined. Based in sensorML appeared ontoSensor [17], a prototype sensor knowledge repository compatible with Semantic Web infrastructure.

In this paper, we propose a sports environment monitoring system, where environmental parameters (i.e., room temperature), physiological data (breathing rate, heart rate, etc.), and user's profile (weight, height, goals, thresholds, etc.) are combined to offer the user safe and efficient suggestions of sport routines in order to improve his/her physical condition. Also, the system will send the user an alarm from an available set if any hazardous condition appears (i.e., a high heart rate value) or any of the defined user upper or lower thresholds are surpassed. The user devices are noninvasive wearable ones: smartphone, pulse meter, smartwatch, and so forth.

The system foundations are a service-oriented semantic middleware and an ontology to model the services provided by the WSN. Furthermore, it is possible to compose new services based on the existing single services, using the context-awareness provided by the middleware. In order to integrate different middleware architectures or ontologies in a single user application, an enterprise service bus (ESB) has been included in the system. This element is an efficient solution to abstract the application layer of the sensors heterogeneity related issues and an easy way to manage the introduction of new middlewares or WSN platforms in the system. The user application just needs to access a defined URL to retrieve the desired parameter, abstracting the underlying WSN platform. If there is a need of integrating a new ontology or sensor, the changes are minimal and only involve the creation of a new bundle inside the ESB and a new URL definition.

3. A Real Sportsman Monitoring System: Contextualization and Requirements

The sports environment monitoring system proposed in this paper is based on a sport area (gymnasium, stadium, etc.) with a sensor network (WSN) spread all over it. The WSN is sensing environmental data, such as light, temperature, humidity. What is more, the user will be provided with a device compatible with the installed network, with the tasks of identifying the user inside the mentioned network and retrieving data. This functionality can be included in a device previously carried by the user (e.g., a smartphone or a smartwatch). These devices send the data to a WSN node equipped with a Bluetooth interface, so no cables are needed to establish the communication. A unique user ID is assigned and represents the user's profile in the system. This profile includes several user parameters: sports routines, fitness, thresholds, and so forth.

Once the user enters the network coverage range, the device transmits the identifier over the network. This way the system always knows which users are in which area. With this information, as well as the environmental data sampled by the sensors (humidity, temperature, lightness. . .) and the specification stored for each profile, the network makes decisions concerning the performance of the sport practice and suggests the user a set of exercises (user context). Then, the network starts monitoring the user's physiological data (heart rate, breath rate, ECG, etc.) and compares them with the thresholds stored. A general scenario overview is shown in Figure 1. The components are ESB, sink, sensor nodes, and Bluetooth devices. The user can carry a communication device (i.e., a smartphone or a smartwatch) to interact with the system and receive alarms.

4. Service-Oriented Semantic Middleware and Service Ontology Description

As introduced in the first section, middleware is an abstraction layer between the wireless sensor and the application layer. The middleware architecture used in this proposal is a service oriented semantic middleware developed in our researching group, named nSOM [18], a service-oriented framework designed for WSNs. Based on a dynamic service composition model and a semantic knowledge management scheme, it provides an agent-based virtual sensor service approach that is able to create new composed services combining the existing ones. nSOM is being extended to include more semantic and context-awareness services.

The services that the WSN offers to the applications rely on measures obtained by the sensors (environmental or physiological). These measures are delivered to a broker that compounds and offers the service. Moreover, this semantic

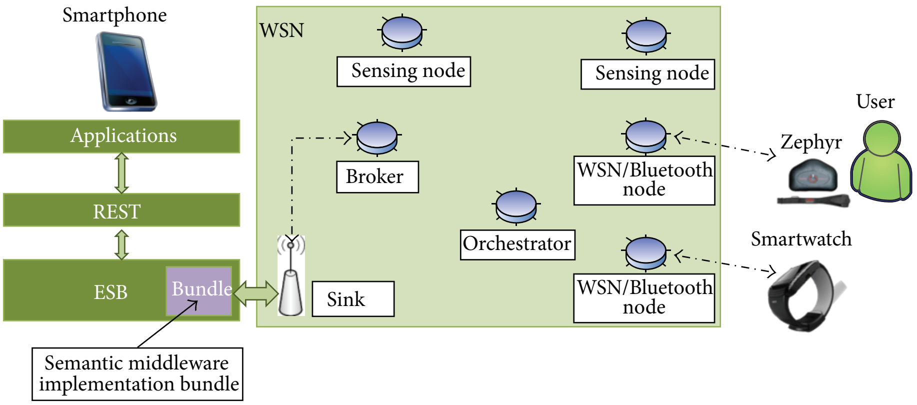


Figure 1: Scenario overview, including key elements: ESB (to integrate different middleware implementations), WSN (with two Bluetooth-enabled nodes), and user's devices (a smartphone, a Zephyr body sensor, and a smartwatch).

middleware architecture proposes the semantic annotation of the service provided by WSN to get a couple of advantages. The annotation of the services is based on an ontology developed specifically for these services. Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. The ontology proposed here is not sensor related, as mentioned ontologies (sensorML or ontoSensor) are, but it is related with the services provided by low capabilities devices integrated in a WSN. Thus, our ontology provides several advantages provided by semantically annotated services.

(i) Applications that use these services can retrieve from the ontology repository the profile and the features of the services, being able to choose the most appropriate according to its function.

(ii) The service composition can be done within the WSN, so the generation of new services is a new feature provided by the WSN.

(iii) Accurate knowledge of the environment in which services are provided.

(iv) Controlling the function of the service.

(v) Adequate the service to the current state and capabilities of the wireless sensor networks.

The ontology description is divided into three parts: profile, process, and context. Profile is the description of the features of the service, and it must be published in an ontology repository in order to be queried by applications or other users before the use of the service. The profile class is composed of service identification, service functionality, security profile, and grounding. Process description of the service is the most important part of the ontology, as it contains the classes related with the process that is behind of the service delivery. The process class is composed of operation, atomic process, aggregated process, and input and output. An important contribution of this ontology is the specification of the context condition under which the service is provided. There are several differences when providing a temperature service indoors or outdoors or a heart rate service at sea level or on the top of a mountain. The context information can be written in the ontology repository and can be used by processes to provide the service. An example of semantic annotation in JSON is provided in Algorithm 1.

Service composition can be done using the base of the already existing services. The new service will be published as a "virtual" service, by a special node called orchestrator. The virtual service is, for all the nodes in the network but the orchestrator, a similar service to the noncomposed ones. When a request for a service reaches the network, the broker will deliver the request to the node that has previously published that service: a sensing node if it is a single service or the orchestrator node if it is a composed service. Once the request is received by the orchestrator, it will be split into single service requests. All of these requests will be sent to the corresponding nodes, and when the orchestrator has all the data needed, it will reply to the composed service request. The only noticeable difference will be the time needed to process the request. An example to illustrate the service composition could be a service called "injury prevention", which is composed of three single services: body temperature, breathing rate, and heart rate. This composed service, according to the values of the three single services, will return a "low", "medium", or "high" injury risk.

5. Enterprise Service Bus Platform for Middleware and Ontology Interoperability

In order to obtain a generic integrative and scalable solution, the use of an enterprise service bus (ESB) was decided.

An ESB is a software architecture model based on the service-oriented architecture (SOA) paradigm used for designing and implementing the interaction and communication between interacting software applications. It provides an asynchronous message oriented communication between the applications inside the bus. The strength of the ESB solutions is the integration of heterogeneous solutions inside a unique communication bus.

Following the specification of the generic client server architecture, the client requests are routed (and adapted if

```
{/*service*/
    "profile": {/*start profile*/
        "identification": {
            "serviceName": {
                "identif": "Real-Time Pulse",
                "idService": "idInto SmartSpaceLifewear"
                },
            "serviceDescription": "Real-Time pulse",
            "owner": "lifewear"
                },
        "functionality": {
                "preconditionDescription": "empty",
                "inputDescription": "empty",
                "outputDescription": "pulse rate"
                },
        "security": {
                 "policy": "security policy lifewear",
                "dataProtection": "integrity"
                },
        "grounding": {
                "inputMessage": "empty",
                "outputMessage": "control, integer",
                "endPoint": "/lifeware/pulse/"
                }
    }/*end profile*/
}/*end service*/
```

Algorithm 1: Ontology-based example of a simple service semantic annotation in JSON: the "real-time pulse" service.

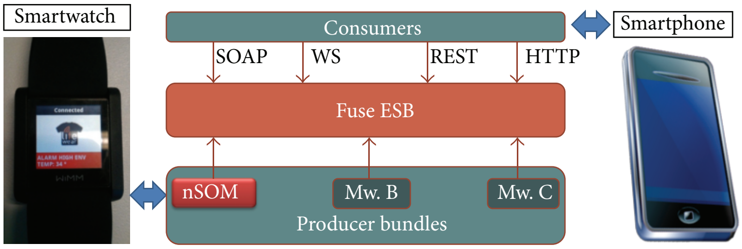


Figure 2: An ESB for different consumers and middleware implementations (center), with a smartwatch attached (left). The services can be consumed by a wide range of applications, for example, a mobile application (right).

necessary) to the corresponding service, which will answer the petition. In fact, the client sends the request to the ESB (instead of sending it to the application server), and the ESB is responsible for routing the request to the server, monitoring and logging the traffic. This approach exposes a unique external interface to access different applications lying behind the bus (or, in this approach, different middleware architectures). Thus, the updating and migration issues are easier to manage.

A general overview of the integration provided by the ESB to external applications is shown in Figure 2.

Components use the bus to communicate between them, reducing the underlying number of connections. By doing this, debugging errors or changing components is easier. The ESB can be distributed, improving the scalability and resilience features of our system. The user application would be able to communicate with the ESB by means of the external interfaces. The ESB solution contributes to integrate all the system components, but it does not solve by itself the semantic and context-awareness issues. Here is where our proposal gives the solution with the nSOM middleware, integrated inside the ESB as a producer bundle.

5.1. External ESB Interfaces. Several interfaces for clients to access the ESB published services have been defined. In order to simplify the system and have lightweight clients, Representational State Transfer (REST) protocol and JavaScript Object Notation (JSON) [19] messages are used. Both are well-known and widely used technologies in mobile and smart devices, giving the opportunity to external developers to create new applications using the existing smart wearable network interfaces.

In order to provide the services needed for the sportsman scenario, the already mentioned nSOM middleware has been

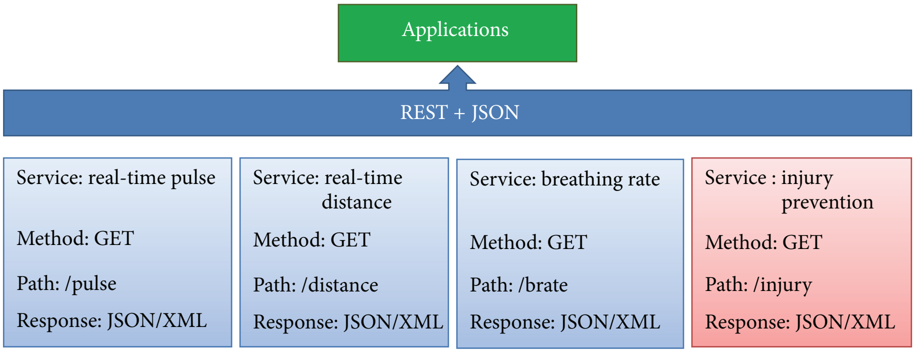


FIGURE 3: External interfaces for third-party user applications.

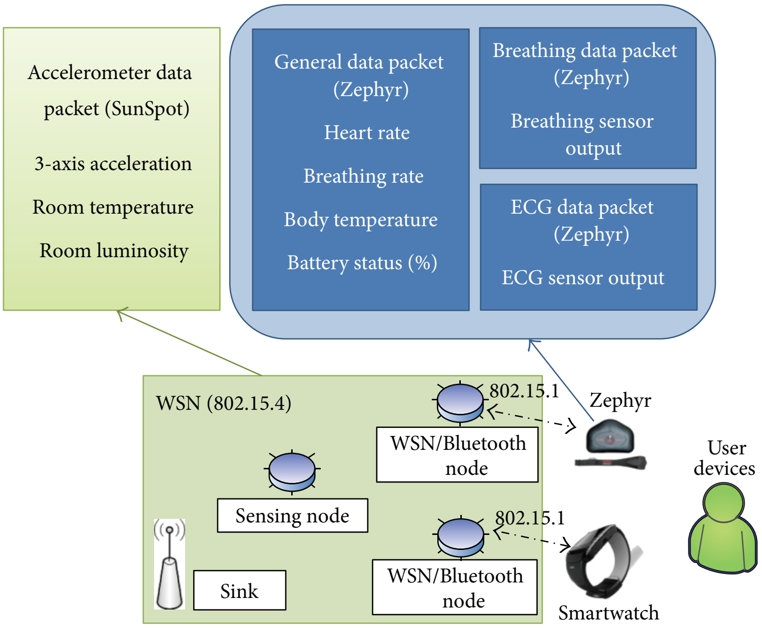


FIGURE 4: Lifewear testing scenario components and communication protocols (down) and data provided by the Zephyr body sensor and SunSpot nodes (up).

extended to include semantic annotations to the services published by the WSN. The definition of the sports scenario includes several interfaces to access the services published by the network. In Figure 3, there are some of these interfaces represented: pulse, distance, breathing rate, and injury prevention.

If a client wants to access any of those services, it needs to send a REST request (in a URL formatted HTTP packet) to the endpoint of the application and the path of the service. The ESB will receive the request, route to the corresponding application, and return the response to the client in the specified format. In this case, all the service responses have the same format (JSON/XML inside an HTTP body response).

6. Testing the System in a Real Scenario and Results

In order to test the functionality of the proposal, a scenario defined inside the research project "LifeWear-Mobilized Lifestyle with Wearables" [20] was used. The system was deployed in a real sportsman scenario: the Technical University of Madrid (UPM) student's gymnasium.

The user's body parameters (heart and breath rate, body temperature, etc.) were measured by a wearable sensor and sent to a special WSN node via Bluetooth. A general overview of the Lifewear testing scenario is presented in Figure 4.

The architecture was deployed over a commercial WSN node solution: SunSpot platform [21], manufactured by Oracle. Main characteristics of SunSpot hardware platform are ARM 920T CPU (180 MHz—32-bit); 512 Kb RAM and 4 Mb FLASH; and a 3.6 V rechargeable 750 mAh Li-Ion battery. In terms of security and privacy, the platform used to implement the proposed system including mechanisms to cipher all the information inside the WSN. The SDK includes libraries providing booth symmetric and asymmetric cryptography mechanisms. For symmetric cryptography, the ciphering algorithm used is AES (Advanced Encryption Standard), and RSA is used for asymmetric cryptography. Moreover, it is also possible to use elliptic curve cryptography (ECC) algorithms in the last update of the platform SDK. A secure communication can be established using secure radio streams, which reuse Secure Sockets Layer (SSL) protocol underneath with ECC.

The ESB implementation used was Fuse ESB [22], an open-source implementation based on Apache ServiceMix. It supports JBI and OSGi for use in enterprise IT organizations. Any standard JBI or OSGi-compliant service engine or binding component—including BPEL, XSLT, or JMX engines—may be deployed to a Fuse ESB container, and Fuse ESB components may be deployed to other ESBs.

The smartwatch used was WIMM, a commercial watch developed by WIMM Labs [23]. It is an Android-based watch, so it was necessary to develop Android applications for the project. This application is able to show alarms to the user in real time if any hazardous value is reached. As an open platform, Android gives the opportunity to any developer to create a new application for our WSN, using the interfaces provided to access to our services. A smartphone application was also developed within the project, and it provides the user a full access to all the system services: exercise routines, alarms, profiling, historical data record, and so forth.

In order to obtain user data (pulse rate, breathing rate, temperature, position, etc.) a wearable device is the most suitable solution. For evaluation purposes, a commercial solution was integrated in the platform: the Zephyr Bio-Harness Bluetooth device [24]. BioHarness BT enables the capture and transmission of comprehensive physiological data via Bluetooth, providing remote monitoring of human performance and condition in the real world. This sensor can monitor heart rate, breathing rate, body temperature, body posture, and activity levels. The sensor sends data every second by default, but it can be configured if the application needs another data sampling rate.

6.1. Results. With all the elements connected and the network deployed, several time measurements were obtained: startup time of the network elements, service response time, and nodes lifetime. Also, the amount of memory used in the nodes to run the applications was measured.

The full system startup time was tested in order to determine the time that the user needs to wait before starting the usage of the platform. The results for the most relevant elements are shown in Figure 5. The ESB startup time is long (13000 milliseconds), since it is the time period for an ESB Linux-based machine to bootup. The Zephyr device boot-up time is longer than a regular node because it needs to set up the Bluetooth link with the WSN node.

The reading data delay introduced by the ESB is also high (4 seconds), and it must be improved in order to avoid bottlenecks in the network. The physiological data is sampled by the Bluetooth device every second and remains stored until the ESB performs a data query or an alarm is issued. Although the use of an ESB in the network introduces a delay, the integration and scalability facilities offered by the ESB justifies the use of this element. Moreover, in our testing scenario the alarms can reach the smartwatch in a direct Bluetooth connection, so real-time alarms delivery is guaranteed.

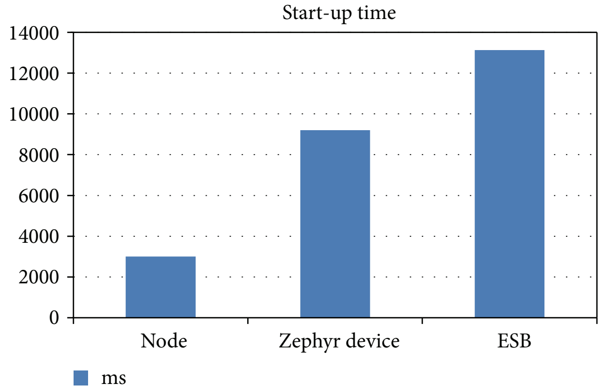


Figure 5: Network elements startup time.

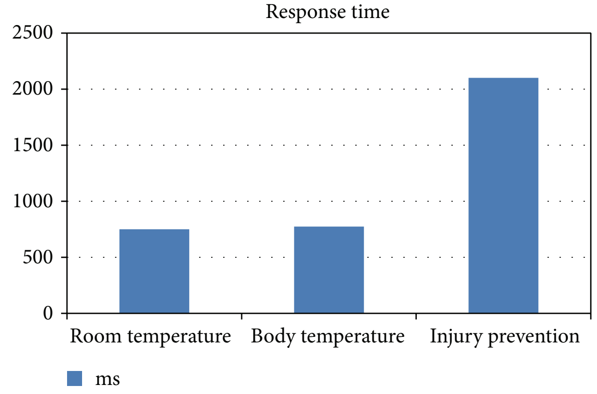


Figure 6: Response time for different service queries: simple services (temperature 1, 2, and 3 and body temperature) and composed service (injury prevention).

The response time for simple and composed service queries is shown in Figure 6. It is remarkable that the injury prevention composed service request takes longer, since it has to be received by the broker, and then wait for the composing request to be processed (as explained in Section 4).

Node lifetime is an important factor in the application. Depending on the role assigned, the same node battery would last different time. For example, the most active nodes are the ones carrying on the Bluetooth connection. Thus, these nodes need to be charged every 6 hours of continuous activity. Other nodes lifetime are shown in Figure 7. In a sportsman scenario, it is admissible that the user devices can be recharged when the activity has ended, and the WSN nodes can be recharged every day when the activity center closes (e.g., in a gymnasium every night). If the application requires a longer node lifetime, the duty cycle and the data sampling frequency can be modified in order to reduce the energy consumption.

Also the amount of memory used by the source code was measured inside the nodes. The most important data is shown in Table 1. The amount of memory used is low, and there is

Table 1: Memory usage in the SunSpot hardware platform.

	Storage (KB)	RAM (KB)
Free	3200	512
Used	83	4
% used	2.59%	0.78%

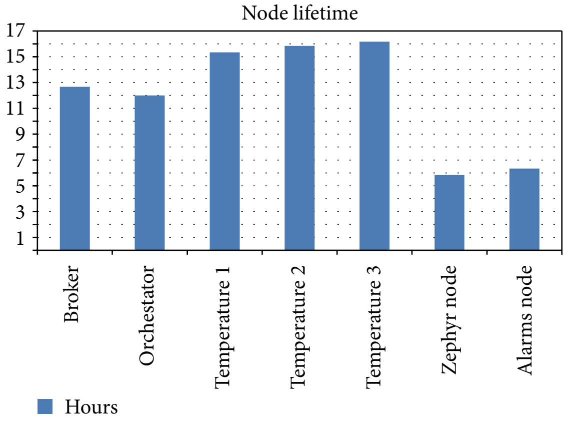


Figure 7: Power consumption of the different nodes, measured in lifetime hours.

enough free space to store application data or to scale up the system.

7. Conclusions and Future Work

In this paper, we have presented an autonomous physical condition performance system, based on a WSN, bringing the possibility to include several elements in an Internet of Things scenario: a smartwatch, a physiological monitoring device, and a smartphone. The integration of these wearable devices has been accomplished using Bluetooth, wireless sensor networks (to connect all the system components), and smart services (in order to publish all the facilities offered for each of the devices).

Also, the proposed solution includes a novel element in a WSN: an enterprise service bus (ESB) as an integration element for different middleware implementations and platforms. The ESB introduces a network delay, but, on the other hand, provides the system with middleware integration and scalability features. The middleware used (nSOM) also provides context-awareness and service composition features, creating a fully deployed real-life application for a sportsman scenario.

The system acquires the physiological data from a Bluetooth commercial device. With these data and the user's profile, the application suggests to the user a series of exercises to improve his or her fitness condition. If a hazardous level of any vital parameter is reached (e.g., heart rate), an alarm is issued and alerts the user to stop doing the workout. This alarm can reach a smartphone or a wearable smartwatch and, if configured, the emergency services through the ESB. All the tests and the measures obtained were carried out in a university campus gymnasium with satisfactory results for the users.

Although our proposal is a generic framework for applications based in services provided by wearable devices, we have included an application scenario for testing purposes. This is an indoor scenario, because it is fairly complex to cover an outdoor sports scenario with a WSN. Furthermore, in an outdoor scenario with a moving user, the issues of tracking, mobility, and localization must be addressed. The service-oriented semantic middleware and service ontology used were designed to be fully scalable. The nSOM agent-based virtual sensor service was implemented to deal with all the scalability and upgrade issues. Thus, new nodes and/or agents managing the user's mobility can be deployed and registered. Since the services are dynamically composed, new services can be added with no issues. Another limitation is that the enterprise service bus is now deployed in a PC machine. Small-sized equipment (such as mini computers with limited resources, e.g., Raspberry Pi, Pandaboard, BeagleBoard, or any other open-hardware platform) with the ESB implemented could be tested in order to include it in our proposal.

Future work will consider including new Bluetooth devices (bathroom scale, GPS tracking device, etc.) in order to improve the accuracy and efficiency of the suggested exercises. Delay times can be improved by using smart routing algorithms. We are working to migrate the network infrastructure to the 6LoWPAN RFC of the IETF [25].

Conflict of Interests

The authors declare that they have no conflict of interests to disclose.

Acknowledgments

The work presented in this paper has been partially funded by the Spanish Ministry of Industry, Tourism and Trade in the framework of the European Research Project "LifeWear-Mobilized Lifestyle with Wearables" (TSI-020400-2010-100), which belongs to the ITEA 2 (Information Technology for European Advancement 2) program. The user's mobile application has been developed in collaboration with SAI Wireless, within the Lifewear project framework.

References

[1] K. Römer, O. Kasten, and F. Mattern, "Middleware challenges for wireless sensor networks," *Mobile Computing and Communications Review*, vol. 6, no. 2, 2002.

[2] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[3] A. Mileo, D. Merico, and R. Bisiani, "Support for context-aware monitoring in home healthcare," *Journal of Ambient Intelligence and Smart Environments*, vol. 2, no. 1, pp. 49–66, 2010.

[4] A. D. Wood, J. A. Stankovic, G. Virone et al., "Context-aware wireless sensor networks for assisted living and residential monitoring," *IEEE Network*, vol. 22, no. 4, pp. 26–33, 2008.

[5] M. Bennebroek, A. Barroso, L. Atallah, B. Lo, and G. Yang, "Deployment of wireless sensors for remote elderly monitoring," in *Proceedings of the 12th IEEE International Conference on*

e-Health Networking, Application and Services (Healthcom '10), pp. 1–5, July 2010.

[6] H. Martín, A. M. Bernardos, L. Bergesio, and P. Tarrío, "Analysis of key aspects to manage wireless sensor networks in ambient assisted living environments," in *Proceedings of the 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies (SABEL '09)*, pp. 1–8, November 2009.

[7] W. Qixin, S. Wook, L. Xue et al., "I-living: an open system architecture for assisted living," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '06)*, pp. 4268–4275, October 2006.

[8] O. Garcia-Morchon and H. Baldus, "The ANGEL WSN security architecture," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM '09)*, pp. 430–435, June 2009.

[9] P. M. Butala, Y. Zhang, T. D. C. Little, and R. C. Wagenaar, "Wireless system for monitoring and real-time classification of functional activity," in *Proceedings of the 4th International Conference on Communication Systems and Networks (COMSNETS '12)*, pp. 1–5, January 2012.

[10] K. Yamasue, K. Takizawa, K. Sodeyama, C. Sugimoto, and R. Kohno, "Vital sign monitoring by using UWB Body Area Networks in hospital and home environments," in *Proceedings of the 6th International Symposium on Medical Information and Communication Technology (ISMICT '12)*, pp. 1–4, March 2012.

[11] S. Jung, J. Y. Ahn, D.-J. Hwang, and S. Kim, "An optimization scheme for M2M-based patient monitoring in ubiquitous healthcare domain," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 708762, 9 pages, 2012.

[12] G. B. Gil, A. Berlanga, and J. M. Molina, "InContexto: multisensor architecture to obtain people context from smartphones," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 758789, 15 pages, 2012.

[13] Q.-D. Ho, T.-N. Tran, G. Rajalingham, and T. Le-Ngoc, "A distributed and adaptive routing protocol designed for wireless sensor networks deployed in clinical environments," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '12)*, pp. 2746–2750, April 2012.

[14] K. S. Prabh, F. Royo, S. Tennina, and T. Olivares, "BANMAC: an opportunistic MAC protocol for reliable communications in body area networks," in *Proceedings of the IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS '12)*, pp. 166–175, May 2012.

[15] M. Nabi, M. Blagojevic, M. Geilen, and T. Basten, "Demonstrating on-demand listening and data forwarding in wireless body area networks," in *Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '12)*, pp. 82–84, June 2012.

[16] "SensorML core specification," http://www.opengeospatial.org/standards/sensorml.

[17] D. J. Russomanno, C. R. Kothari, and O. A. Thomas, "Building a sensor ontology: a practical approach leveraging ISO and OGC models," in *Proceedings of the International Conference on Artificial Intelligence (ICAI '05)*, pp. 637–643, Las Vegas, Nev, USA, June 2005.

[18] M. S. Familiar, J. F. Martinez, and L. Lopez, "Pervasive smart spaces and environments: a service-oriented middleware architecture for wireless Ad Hoc and sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 725190, 11 pages, 2012.

[19] "The Internet Engineering Task Force (IETF) RFC4627," http://tools.ietf.org/html/rfc4627.

[20] "LifeWear project main page. (Sept'2012)," http://lifewear.diatel.upm.es/index_en.htm.

[21] "SunSpot Reference Guide. Oracle Inc. (Sept'2012)," http://sunspotworld.com/docs/Yellow/SunSPOT-Programmers-Manual.pdf.

[22] "Fuse ESB reference. FuseSource. (Sept'2012)," http://fusesource.com/products/enterprise-servicemix/.

[23] "Wimm Labs web page. (Sept'2012)," http://www.wimm.com/.

[24] "Zephyr Technology Corporation. Zephyr BioHarness-BT data sheet (May'2012)," http://www.zephyr-technology.com/bioharness-bt.

[25] "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," (IETF RFC4919), http://tools.ietf.org/html/rfc4919.