Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 396850, 7 pages
http://dx.doi.org/10.1155/2013/396850

*Research Article*

# Fault Tolerance and Energy Consumption Scheme of a Wireless Sensor Network

**Doo-Soon Park**

*Department of Computer Software Engineering, Soonchunhyang University, Asan 336-745, Republic of Korea*

Correspondence should be addressed to Doo-Soon Park; parkds2013@gmail.com

Received 18 October 2013; Accepted 30 October 2013

Academic Editor: Hwa-Young Jeong



The commercialization of detection and surveillance systems by sensor networks has recently become more popular in various industries and in ordinary life. Wireless sensor networks (WSNs) may be utilized in areas such as military defense systems, u-care services, and disaster prevention systems. WSNs, which have three primary elements, should be able to communicate through a combination of network and sensor functions. When the size of a sensor network is small, the cost of building it accounts for a large fraction of the total cost. In large-scale sensor networks, however, maintenance costs to replace nodes when they are out of order, and batteries used for wireless communications, greatly increase, in addition to the cost of constructing them. Therefore, measures to detect and respond to sensor faults in a sensor network, as well as a mechanism to predict the energy consumption of sensors, are necessary. Accordingly, this paper proposes a measure to detect and deal with sensors by dividing WSN into fixed WSN and mobile WSN. This paper also suggests a method to predict and express the energy consumption of a sensor network.

## 1. Introduction

The application of WSN technologies has recently gained popularity in a number of different industries: industrial process monitoring and control, machine u-health monitoring [1–3], environment and habitat monitoring, healthcare, home automation, and traffic control [1–3]. A WSN consists of spatially distributed autonomous sensors that cooperatively monitor physical and environmental conditions such as temperature, sound, vibration, pressure, motion, and pollutants [1, 3]. WSN is divided into fixed WSN, in which sensors are fixed in a target area to play their role, and mobile WSN, in which moving sensors perform sensing. When the size of a sensor network is small, the cost of building it accounts for a large portion of the total cost. However, in a large-scale sensor network, the maintenance costs to replace nodes when they are out of order, and batteries used for wireless communication, greatly increase and add to the total cost of constructing it. Therefore, measures to detect and cope with sensor faults in a sensor network and a mechanism to predict the energy consumption of sensors are both necessary [4–8]. In order to construct a WSN environment, many sensors and appropriate protocols for communication among them are required. With regard to protocols for communication among sensors, sensors are diverse and their uses vary, and, as a result, establishing an optimal environment is not easy. In addition, to place sensors in a target area and to test them to verify theoretical content incur a high cost. In placing sensor nodes, large-scale sensor nodes should be used together with placement algorithms. Discovering and correcting errors in these sensor nodes require an enormous amount of time and effort, and checking each sensor node is virtually impossible [9–13]. Accordingly, this paper proposes a measure to detect and tackle sensor faults by dividing WSN into fixed WSN and mobile WSN. It also suggests a method for predicting and expressing the energy consumption of sensor networks.

## 2. Related Works

This section briefly examines topological reconfiguration and control in the existing sensors and actuators of wireless sensor networks (WSNs) and cyber physical systems (CPSs) according to the research shift from WSNs to CPSs as follows.

(i) Connected dominating set (CDS) [14]: the CDS guarantees connectivity to all nodes. In addition, the routing process is run quickly by simplifying the connected links.

(ii) Construction algorithms for reliable CDS (CAR-CODS) [15]: this is the method by which the performance of ad hoc topologies that consist of CDSs is improved. A CDS configuration method was suggested by making use of the broadcasting time delay of the neighboring configuration notification message that factors in the remaining battery capacity, the mobility of nodes, and the number of neighboring nodes.

(iii) Partial reconstruction of CDS (PRCDS) [16]: a CDS partial reconfiguration algorithm was suggested to effectively react to problems due to the occurrence of critical nodes when a CDS-based routing protocol was applied. In configuring the CDS nodes for load balancing, this method searches for the probability of connection, while topology reconfiguration, due to critical nodes, is expanded into two-hop nodes. Therefore, the nodes within a respective range show effectiveness in terms of their reconfiguration time.

(iv) Simple distributed approximation algorithm (SDAA) [17]: this algorithm was proposed to increase the overall lifetime of topologies by reducing the energy consumption of wireless nodes using the method of removing unnecessary CDS constituent nodes in order to approximate the number of gateway nodes that comprise the CDS to a minimum number.

As existing studies on topology reconfiguration and control are based on resources with batteries of limited capacity, they have attempted to configure topologies through minimum levels of energy consumption. However, the function for this is embedded in the resources and run, and thus the corresponding battery consumption rate cannot be excluded.

## 3. Fault Tolerance Scheme on WSN

In general, there are two types of WSN faults: node fault and network fault [1]. Nodes have several hardware and software components that can produce malfunctions. For example, the enclosure can suffer impacts and expose the hardware of the sensor node to the extreme conditions of the environment. Routing is one of the fundamental building blocks in a WSN. It is essential for collecting sensor data, for distributing software and configuration updates, and for coordination among nodes. WSN should follow two main steps. The first step is fault detection. This step detects whether a specific functionality is faulty and predicts whether it will continue to function properly in the near future. After the system detects a fault, fault recovery is the second step in enabling the system to recover. There are two basic types of detection techniques: *self-diagnosis* and *cooperative diagnosis*. Faults that can be determined by a sensor node itself can adopt self-diagnosis detection. For example, faults caused by battery depletion can be detected by a sensor node itself. The remaining battery of the sensor node can be predicted by measuring current battery voltage. Another example is the detection of failure links. A sensor node may detect that links to some of its neighbors are faulty if the node does not receive a message from a neighbor within a predetermined interval. However, there are some kinds of faults that require cooperative diagnosis among a set of sensor nodes.

*3.1. Adaptive Rerouting Scheme on Fixed WSN.* This paper hypothesizes that each sensor of a fixed WSN uses a multi-hop-based communication protocol. In the entire WSN, each sensor determines a parent node among neighboring sensor nodes based on the number of hops between sink nodes and decides on the parent node of sensor $S_i$ in consideration of the number of hops based on the selected parent node ($S_i$). Information sensed by the paths of repetitively selected nodes is ultimately transmitted to the sink node [3, 8].

The paths through which data sensed by each sensor are delivered to the sink node (routing) in fixed WSN should be reset (rerouted) when data traffic is excessive or when faults occur in the middle sensor nodes in paths. In the first case, it is slightly different according to the specifications and functions of each sensor used in an actual fixed WSN with the hardware characteristics of wireless communication bandwidth. In this study, it is presumed that rerouting occurs when data traffic delivered by each sensor node exceeds the bandwidth or when waiting time of data traffic that should be delivered exceeds reference time. It is only when data traffic takes place that a resetting algorithm routing to the sink nodes and a visual monitoring condition of changes in the paths can occur. Figure 1 shows an example of paths being rerouted when data traffic occurs. The dotted lines represent the paths through which each sensor node may transmit data. Routing from sensor node 1 to the sink node is colored blue, and routing to the sink node from sensor nodes 5 and 9 is colored violet and light green, respectively.

It is assumed that data sensed by sensor nodes 1, 5, and 9 are delivered to sink node $S$ via the middle sensor nodes of each routing path. When data traffic is excessive in sensor node 4, sensor node 4 determines sensor node 8, whose data transmission amount is small compared to its neighboring nodes, as a new parent node, and divides data and then transmits it to sensor node 8. At the same time, routing data that are transmitted to sensor node 5 are changed to sensor node 7, which delivers the data to the sink node. The number of hops may increase while delivering data, but overall, the time required to transmit data may be reduced, and information on rerouting is provided visually.

As shown in Figure 2, the fault should occur in the middle sensor nodes while sensed data are delivered to the sink node through the middle sensor nodes.

This occurs when faults take place in middle sensor nodes 3 and 6 while sensed data are being delivered. Therefore, each reroute is applied. In addition, this example shows that data traffic occurs in sensor node 4 so that data is delivered to the sink node using sensor node 7.

Faulty sensor nodes can either do harm to the entire network or trigger loss of data when being delivered from the sensor nodes to the sink node. A parent node sends acknowledged (ACK) signals after receiving data from its child node, and when the ACK signals do not arrive within a certain

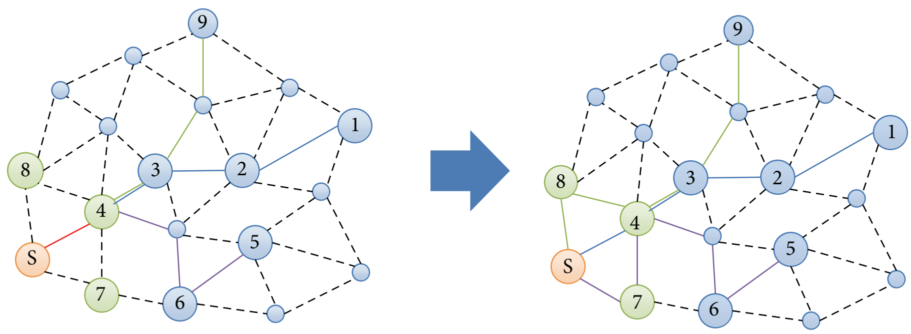



Figure 1: Example of adaptive rerouting with data traffic.

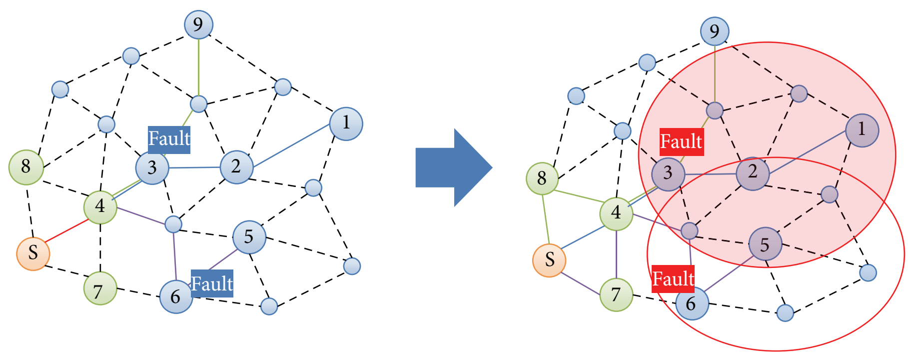



Figure 2: Example of adaptive rerouting with data traffic.

period, the parent node is considered to be out of order. When sensor nodes are out of order, nodes which have such sensor nodes as their parent node search for sensor nodes appropriate as parent nodes among their neighboring nodes. In other words, sensor nodes with a small number of hops and a small amount of transmitted data are established as new parent nodes. Regarding faults in sensor nodes, both cases in which only one sensor node is out of order and those where multiple sensor nodes are out of order, are all considered.

An adaptive rerouting algorithm for excessive data traffic in certain sensors and the occurrence of faults is in Algorithm 1.

*3.2. Covering Scheme of Adjacent Sensor on Mobile WSN.* First, what is important in mobile WSNs is the embodiment of moving mobile sensors. Mobile sensors are required to maintain a given connectivity, avoid obstacles, and maximize coverage in the target area. Therefore, this paper uses the coverage method in addition to the obstacle avoidance method and the constrained coverage method to maximize coverage while maintaining the given connectivity [5, 8]. In mobile sensor networks, the mobile sensor that senses the information (and that requires observation in the target area) uses the connection between mobile sensors in order to send the collected information to the sink node. In this paper, mobile sensors move due to $F_{\text{cover}}$, $F_{\text{degree}}$, and $F_{\text{obstacle}}$. In addition, there exists a moment when such forces reach the static equilibrium and become zero [8]. $F_{\text{cover}}$ is the force with which mobile sensors push against one another to maximize the sensing range in the target area. $F_{\text{degree}}$ is the force that

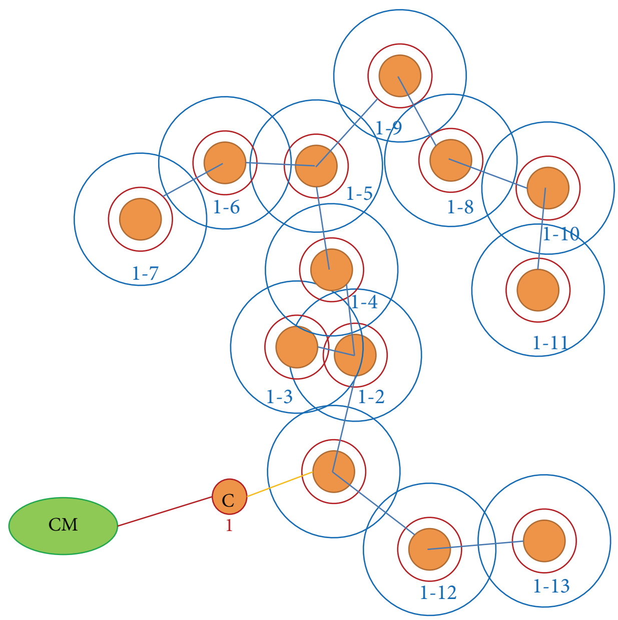



Figure 3: Control manager view and single control view.

is exerted by mobile sensors to keep the number of given neighboring sensors at the degree K. $F_{\text{obstacle}}$ maximizes coverage, maintains the given connectivity, and avoids obstacles. Therefore, $\|F_{\text{cover}} + F_{\text{degree}} + F_{\text{obstacle}} - F_{\text{damper}}\| = 0$ is used for the moment of mobile sensors in this research. $F_{\text{damper}}$ is calculated based on the damper constant and the moving

```
while (true)
create data of current node;
for each neighbor node do
if neighbor node == parent node then
        send recent data of current node to parent node;
        send recent data of children node to parent node;
end if
end for
if occur data overhead for current node then
        if not children node && not parent node then
            if lowest packet neighbor node then
                select new parent node & re-routing for parent node;
            end if
        end if
end if
if occur fault for current node then
        if not children node && not parent node then
            if lowest packet neighbor node then
                select new parent node & re-routing for parent node;
            end if
        end if
end if
end while
```

Algorithm 1: Adaptive rerouting algorithm.

speed of the current node [5, 8]. In mobile MSNs, a sensor is moved by $F$, which is the sum of the three forces: $F = F_{\text{cover}} + F_{\text{degree}} + F_{\text{obstacle}}$. This force ($F$) is called the *centroid-directed virtual force* [3, 5, 8].

Basically, the state under which mobile sensors sense the area to a maximal extent in the target area is presumed to be an equilibrium state since there are no longer any large movements. In actuality, each sensor covers its sensing area with small movements with the coordinate as the center. This section proposes how to cope when each mobile sensor transmits sensed data based on its communication range to the sink node and when faults occur in the middle sensors at the equilibrium state.

Mobile sensors' sensing of faults is revealed by beacon signals. In other words, it may be sensed that when there is no acknowledgement by the "hello" message, the relevant sensor is out of order. At an actual equilibrium state, each mobile sensor moves minutely in all directions and communicates information about the conditions of sensors connected to it within the communication range. The scheme about sensing faults at an equilibrium state is presented by the movement of mobile sensors in the target area.

For considering control manager and single control, sensors engage in their transmission of information when a server manages sensors as a single control manager within the target area in Figure 3. The index in the lower part means that the control server number and the indexes in the lower part of the resource consist of 1-1. The first "1" signifies the index of the connected control server; the second "1" signifies the index of a sensor. For example, 1-3 means the third sensor connected to the control server 1.

The conditions of control servers connected with sink nodes performing the role of multiple control servers are shown Figure 4. Whether or not sensors connected to each control server which are out of order may be identified through each control manager. Each control manager internally creates a sensor information table on each control. The table is comprised of the current location of a sensor, the amount of battery remaining, and the sensor ID. Using these tables, resources that do not operate despite a large amount of remaining battery may be found.

The failure of the sensor 4-2 occurred. As in Figure 5, when failure of the sensor 4-2 takes place, the lower level sensors that are connected are affected. The replacing sensor to perform the function of the sensor that became out of order sends a "hello" to neighboring sensors and moves data for a short while to sensors that send acknowledgement and then recover them.

The process of fault tolerance is shown in Figure 6 with a sensor's failure. The range to find a replacement sensor is twice as far as the sensor's communication distance. When there is no replacement sensor, the range is expanded by the same distance. Here, the range is expanded as far as the communication distance of the sensor. When a replacement sensor is found, it is induced to the location of the failed sensor through the control manager. The induced sensor covers fault tolerance by performing the functions of a failed sensor and reconnecting lost sensors.

## 4. Energy Consumption Strategy on WSN

This section deals with predicting the life of a sensor node based on how much electric current each node consumes.

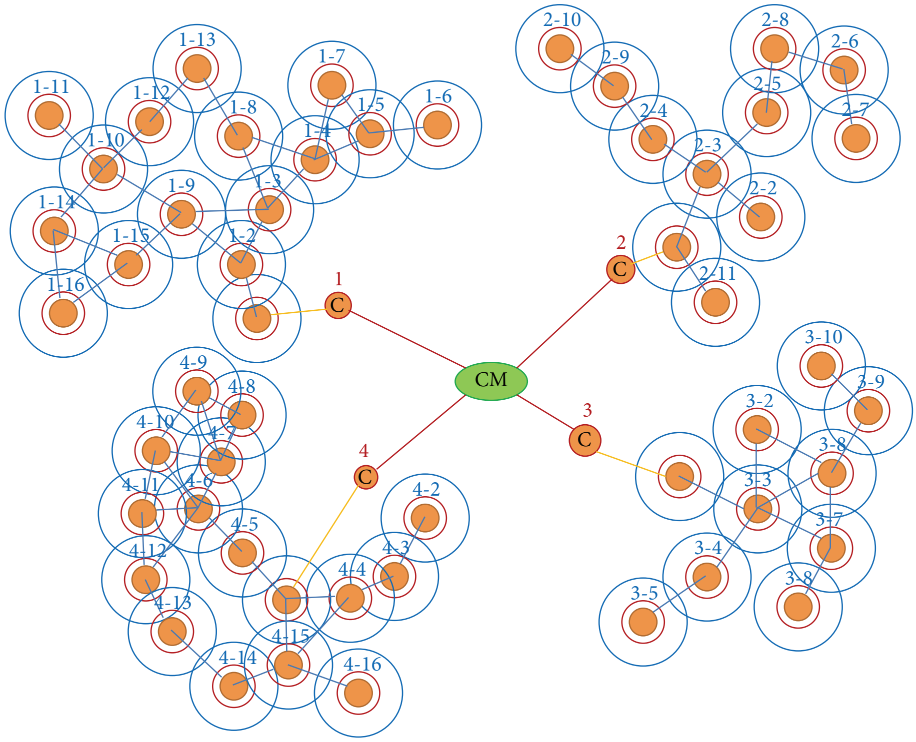



Figure 4: Control manager view and multi-control view.

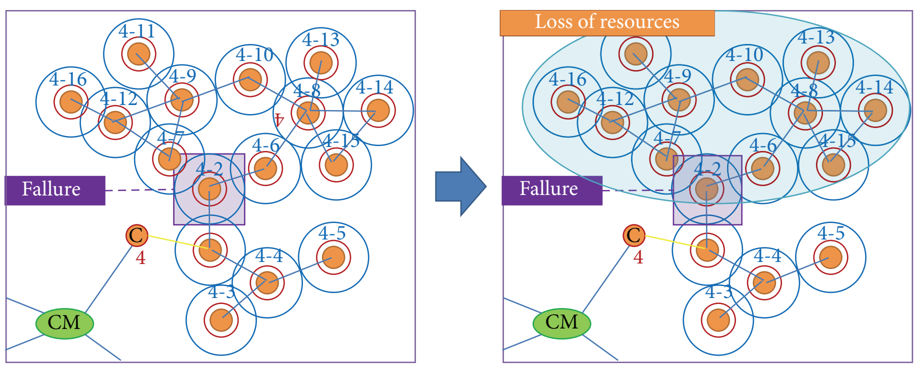



Figure 5: Information on loss sensors in the case of sensor 4-2 failure.

There are three elements that influence the life of a sensor node consuming an electrical current—a wireless transceiver, a sensor module, and a microprocessor. In a wireless transceiver, the amount of electrical current consumed differs between transmission and reception. This is due to the influence of the microprocessor, which operates within the module of a wireless transceiver. More electrical current is consumed by about 5 mA during reception. A sensor module consumes far less electrical current than a wireless transceiver. However, in an area where detection of sensors occurs frequently, consumption of electrical current increases by the accumulation of amounts, thereby affecting the life of the sensor node. A sensor node requires a microprocessor on which programs may be mounted. There are different kinds of microprocessors according to usage and calculation capability. A microprocessor mounted on a sensor node is generally characterized as operating with low electrical power. In general, a microprocessor is composed of similar kinds of processors called families, and if it is not a kind whose design is disclosed and expanded from a basic processor like the ARM processor, it is possible that the Atmel processor will get greater use. In this section, a node is formed based on the three kinds of modules explained earlier. It is a wireless transceiver that consumes the largest amount of electrical current among the three kinds of modules, and the model of electrical current consumption becomes different according to which wireless transceiver is used. This paper assumed the three most widely used kinds of modules as a basic model in consideration of general use. A node is formed using CMOS Transceiver CC2420 in wireless

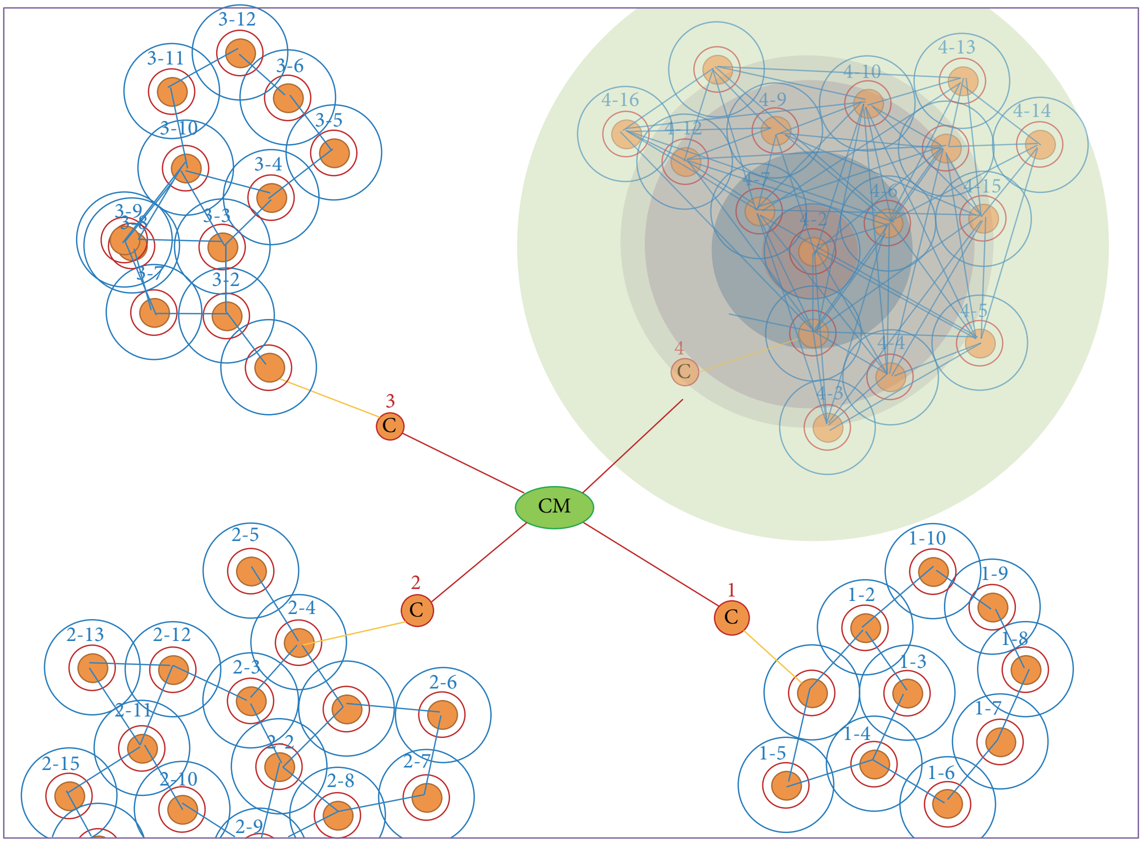
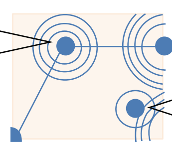



Figure 6: Processing of fault tolerance.

| Node information | |
|---|---|
| (Node type) | Sensor node |
| Node ID | SID00017 |
| Node name | Sensor node 17 |
| Interval (ms) | 1000 |
| Battery (mAh) | 124.23514423 |
| Event | |
| Message send | 1076596 |
| Message receive | 1076596 |
| Sensing event | 336430 |



| Node information | |
|---|---|
| (Node type) | Sensor node |
| Node ID | SID00018 |
| Node name | Sensor node 18 |
| Interval (ms) | 1000 |
| Battery (mAh) | 174.70114423 |
| Event | |
| Message send | 807430 |
| Message receive | 807430 |
| Sensing event | 121117 |

Figure 7: Conditions of energy consumption according to the frequency of detection by a sensor.

transceiver module and ATmega128L[19] designed with low electrical power among Atmel processors in microprocessor. Regarding sensors, four types of sensors were designed with low electrical power: a passive infrared sensor, a vibration sensor, a microphone, and a magnetic sensor.

To predict the life of a node in a sensor network, a life calculator was designed using the node's remaining energy and electrical current consumption. A sensor node has a wake-up cycle for how frequently to perform in active mode and sleep mode; one after another. Active mode and sleep mode have different characteristics in consuming electrical currents. Based on these characteristics, a desired/expected life is calculated with the unit of life aiming to calculate time. When a sensor has barely any life remaining, the remaining energy decreases, and this is expressed as a pie graph on the sensor. According to the results of CROWD simulation, the more frequent the detection in a sensor node takes place, the shorter its life becomes. Figure 7 displays the condition of energy consumption in a sensor node according to the frequency of events detected by the sensor.

When the battery is discharged and the sensor node's life completely ends, the heat map and pie graph that were expressed on the sensor node disappear from the screen and connect with the node whose life has ended. They are then expressed as red dotted lines.

## 5. Conclusion

A measure to detect and cope with the failure of a sensor in a sensor network and a mechanism to predict the energy consumption of sensors are necessary. For the construction of a WSN environment, several sensors—and appropriate protocols for communication among them—are required. With regard to protocols for communication among sensors, sensors are diverse and their uses vary. For this reason, the establishment of an optimal environment is not easy. In addition, to place and test sensors in a target area for verification of theoretical content incurs a high cost.

In placing sensor nodes, large-scale sensor nodes should be used together with placement algorithms. To discover and correct errors with these sensor nodes, an enormous amount of time and effort are consumed, and checking each sensor node is virtually impossible. Accordingly, this paper proposed a measure to detect and deal with sensors by dividing WSN into fixed WSN and mobile WSN. This paper also suggested a method to predict and express the energy consumption of a sensor network.

## Acknowledgment

This work was supported by the Soonchunhyang University Research Fund.

## References

[1] L. M. S'a de Souza, H. Vogt, and M. Beigl, *A Survey on Fault Tolerance in Wireless Sensor Networks*, Interner Bericht, Fakultät für Informatik, Universität Karlsruhe, Karlsruhe, Germany, 2007.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] Y. S. Jeong, Y. J. Chung, and J. H. Park, "Visualisation of efficiency coverage and energy consumption of sensors in wireless sensor networks using heat map," *IET Communications*, vol. 5, no. 8, pp. 1129–1137, 2011.

[4] B. Musznicki and P. Zwierzykowski, "Survey of simulators for wireless sensor networks," *International Journal of Grid and Distributed Computing*, vol. 5, no. 3, pp. 23–50, 2012.

[5] Y.-H. Han, Y.-H. Kim, W. Kim, and Y.-S. Jeong, "An energyefficient self-deployment with the centroid-directed virtual force in mobile sensor networks," *Simulation Transactions of the Society for Modeling and Simulation InternationalIssues*, vol. 88, no. 10, pp. 1152–1165, 2012.

[6] J. Chen, M. B. Salim, and M. Matsumoto, "A single mobile target tracking in Voronoi-based clustered wireless sensor network," *Journal of Information Processing Systems*, vol. 6, no. 4, pp. 17–28, 2010.

[7] A. U. Bandaranayake, V. Pandit, and D. P. Agrawal, "Indoor link quality comparison of IEEE 802. 11a channels in a multi-radio mesh network testbed," *Journal of Information Processing Systems*, vol. 8, no. 1, pp. 1–20, 2012.

[8] Y.-S. Jeong, Y.-H. Han, J. J. Park, and S. Y. Lee, "MSNS: mobile sensor network simulator for area coverage and obstacle avoidance based on GML," *EURASIP Journal on Wireless Communications and Networking*, vol. 95, no. 1, pp. 1–15, 2012.

[9] S. Silas, K. Ezra, and E. B. Rajsingh, "A novel fault tolerant service selection framework for pervasive computing," *Human-centric Computing and Information Sciences*, vol. 2, no. 5, pp. 1–14, 2012.

[10] X. Zhou, Y. Ge, X. Chen, Y. Jing, and W. Sun, "A distributed cache based reliable service execution and recovery approach in MANETs," *Journal of Convergence*, vol. 3, no. 1, pp. 5–12, 2012.

[11] "Java in Simulation Time/Scalable Wireless Ad hoc Network Simulator," http://jist.ece.cornell.edu/.

[12] L. Shu, C. Wu, and M. Hauswirth, "NetTopo: Beyond Simulator and Visualizor for Wireless Sensor Networks," Tech. Rep., Digital Enterprise Research Institute (DERI), Galway, Ireland, 2008.

[13] O. C. Ghica, "SIDnet-SWANS Manual," 2010, http://users.eecs.northwestern.edu/~ocg474/SIDnet/SIDnet-SWANS%20manual.pdf.

[14] J. Blum, M. Ding, A. Thaeler, and X. Cheng, "Connected dominating set in sensor networks and MANETs," in *Handbook of Combinatiorial Optimization*, pp. 329–369, 2004.

[15] H.-S. Cho and S.-J. Yoo, "Power, mobility and wireless channel condition aware connected dominating set construction algorithm in the wireless ad-hoc network," *The Journal of Korea Information and Communications Society B*, vol. 30, no. 5, 2005.

[16] Y.-S. Hong and H.-S. Lim, "A load-balanced topology maintenance with partial topology reconstruction," *The Journal of Korea Information and Communications Society B*, vol. 35, no. 12, pp. 1188–1197, 2010.

[17] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," *Journal of Communications and Networks*, vol. 4, no. 1, pp. 59–70, 2002.