

Research Article

Energy-Efficient Fire Monitoring over Cluster-Based Wireless Sensor Networks

Young-guk Ha,¹ Heemin Kim,¹ and Yung-cheol Byun²

¹ Department of Computer Science and Engineering, Konkuk University, Seoul 143-701, Republic of Korea

² Department of Computer Engineering, Jeju National University, Jeju-si, Jeju 690-756, Republic of Korea

Correspondence should be addressed to Yung-cheol Byun, ycb@jejunu.ac.kr

Received 12 November 2011; Accepted 11 January 2012

Academic Editor: Tai Hoon Kim

Copyright © 2012 Young-guk Ha et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Uncontrolled fires occurring in wild areas cause significant damage to natural and human resources. Many countries are looking for ways to fight forest fires at an early stage using sensor networks, by integrating IT technologies. Studies in the fire-related sensor network field are broadly classified into efficient processing of fire data on sensor nodes and energy efficiency during communications among wireless sensor nodes in case of fire. Most studies of sensor network energy efficiency so far mainly focus on extending the connectivity of the entire network and minimizing isolated nodes by applying power evenly to each sensor node through efficient cluster-based routing. This paper proposes an energy-efficient fire monitoring protocol over cluster-based sensor networks. The proposed protocol dynamically creates and reorganizes the sensor network cluster hierarchy according to the direction of fire propagation over the sensor network clusters. This paper also presents experimental results to show that the proposed protocol is more energy efficient than fire monitoring with existing cluster-based sensor network protocols.

1. Introduction

The world is facing many risks caused by natural disasters such as forest fires, floods, and abnormal climate changes. IT scientists are studying ways of effectively solving such problems by looking at risks at an earlier stage. This study was conducted on a new network for detecting risk factors and rapidly responding to these problems. Wireless sensor networks are regarded as the best systems for applications in those environments. However, current wireless sensor networks have problems in fire monitoring over wide areas because of limited battery capacity and the short life spans of the sensor nodes [1]. Unlike general environmental data monitoring such as precipitation or climate monitoring, fire monitoring requires very frequent real-time data transmissions until the fire is put out. Consequently, the sensor nodes in fire monitoring applications consume much more energy in relatively short periods of time than do sensor nodes in general environmental data monitoring applications, and they finally have shorter life spans [2].

For fire monitoring applications, flat-based routing protocols for sensor networks, such as SPIN (Sensor Protocols

for Information via Negotiation), are not appropriate since all the sensor nodes that detect fire start sending fire data to the sink node individually and consume energy of the sensor network very quickly. The existing cluster-based routing protocols for sensor networks, such as LEACH (Low-Energy Adaptive Clustering Hierarchy) and TEEN (Threshold-sensitive Energy Efficient sensor Network protocol), are more energy-efficient for fire monitoring applications than the flat-based routing protocols since each cluster head can collect fire data from local sensor nodes, encapsulate the collected data into a single data packet, and transmit the packet to the sink node at a time. However, as the number of clusters that detect fire increases due to the wide propagation of the fire, the energy efficiency of fire monitoring with cluster-based routing protocols will decrease [2–6].

This paper proposes EFMP (Energy-efficient Fire monitoring Protocol), a fire monitoring protocol operating over cluster-based sensor networks. To further increase the energy efficiency of cluster-based sensor networks for fire monitoring, EFMP reduces the number of transmissions of fire data from the cluster heads to the sink node by dynamically creating and reorganizing the sensor network cluster hierarchy

according to the fire propagation over the sensor network clusters. Fire monitoring experiments showed that EFMP consumes 41% and 12% lesser energy on average per node than do LEACH and TEEN, respectively. Furthermore, the number of sensor nodes that survived in the EFMP-based experiment was 12% and 14% more than that in the LEACH- and TEEN-based experiments, respectively.

The rest of this paper is organized as follows. Section 2 analyzes the existing energy-efficient sensor network protocols and the problems faced when using these protocols in fire monitoring applications. Section 3 describes the design of the EFMP proposed in this study. Section 4 compares the performance of EFMP in fire monitoring environments with the performances of existing cluster-based sensor network protocols using NS-2. Finally, Section 5 summarizes the paper and future directions as conclusion.

2. Related Works

This section explains typical energy-efficient routing protocols for sensor networks such as SPIN, LEACH, and TEEN, which can be used in fire monitoring applications.

2.1. SPIN. SPIN is a method proposed for avoiding repetition of data transmission when the same data are sent to multiple nodes. SPIN sends ADV (Advertisement) messages from sensors collecting data. It asks for readiness to receive data by nearby sensors and the sensor receiving the ADV message sends a REQ (Request) message once it is ready. The sensor receiving the REQ message sends its own sensor data. However, the REQ is not sent if it is not ready. Data are, therefore, sent only to sensors that sent REQs since sensors not sending REQs are not ready. The drawback of SPIN in fire monitoring system environments is that the sensor only sends a data packet once it receives a REQ after sending an ADV packet. If a sensor receives four REQ packets, then the sensor sends four data packets individually. Packets are delivered to nodes that are not relevant to the actual transmission path in this case and this wastes battery power [2, 3]. The main drawback is high consumption of battery power by numerous transmissions from networks requiring many nodes, as in the case of fire monitoring system environments, since the sensors are connected un-hierarchically. In addition, transmission is not completed if there are isolated sensors, and battery power is consumed between nearby nodes and isolated nodes since ADV packets continue to be sent for connection. These problems may occur in the case of a forest fire where a large number of sensors are installed over wide area [7–9].

2.2. LEACH. LEACH is a protocol for forming efficient clustering; sensor nodes form local clusters by themselves and distribute energy evenly. One node-forming cluster acts as a cluster head, and the sensor node with the most energy operates as a cluster head; when its energy capacity becomes smaller than those of the other nodes, another node with the most energy then performs the role of cluster head. The battery drains quickly when a sensor node acts as a cluster

head; the burden on the node operating as the cluster head is, therefore, reduced by rotating the cluster heads. LEACH is currently one of the most popular methods for environmental monitoring systems. However, LEACH may waste a lot of energy since each individual cluster head operates independently in a fire monitoring system with a large number of sensor nodes. The number of related clusters increases as the fire spreads widely. Increases in the number of cluster heads sending information and the number of transmissions in the sensor network will result in a reduction in energy efficiency [2, 3, 10, 11].

2.3. TEEN. TEEN is a cluster-based and reactive routing protocol that works with two threshold values: a hard threshold and a soft threshold. A hard threshold is a threshold value for the sensed attribute. Nodes sensing this value only turn their transceivers on if the sensed value is above the defined threshold. A soft threshold is a small change in the value of the sensed value; it triggers the node to switch on its transceiver to become active and transmit. The sensed value is stored as an internal variable in the node. Every time a new cluster head is selected, the threshold values can change. When the cluster node's value exceeds the soft threshold, then it starts sensing again. The advantage of this protocol is the reduced number of deliveries, making it highly efficient in terms of energy consumption and response time [3, 4, 12, 13]. Nevertheless, TEEN may also suffer from the drawback of typical cluster-based routing protocols in fire monitoring environments.

3. Design of EFMP

3.1. Protocol Stack for EFMP. The method of transmission from the sensor to the cluster head and the cluster head to the sink is based on the existing cluster structure. That is, the methods for partitioning the sensor network into clusters and election of cluster heads are done by the underlying cluster-based routing protocol.

Figure 1 shows the EFMP protocol stack designed in this study. The stack is broadly composed of the system layer and the protocol layer. The system layer includes the sensor node OS and hardware, and the protocol layer includes the MAC (Medium Access Control) protocol responsible for communication by the sensors. The protocol layer has cluster-based routing protocols, such as LEACH and TEEN, over the MAC protocol. The EFMP proposed in this paper is on top of the protocol stack and can improve energy efficiency by reducing the number of transmissions by dynamic hierarchical clustering, while maintaining the existing protocols; this is the main function of the EFMP since it controls cluster hierarchy by operating on existing cluster-based routing protocols.

3.2. Cluster Hierarchy and Roles of Cluster Heads. The major difference between the sensor network cluster organization of an EFMP and that of existing cluster-based routing protocols is that the EFMP has a hierarchical cluster structure that can be reorganized dynamically. As illustrated in Figure 2, in an EFMP, the cluster heads are layered and classified

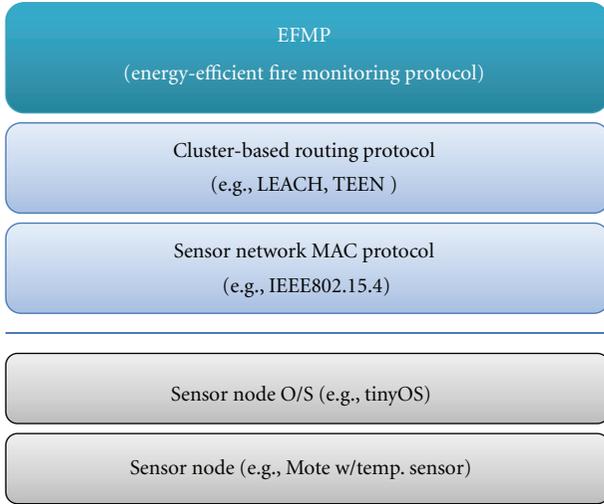


FIGURE 1: EFMP protocol stack.

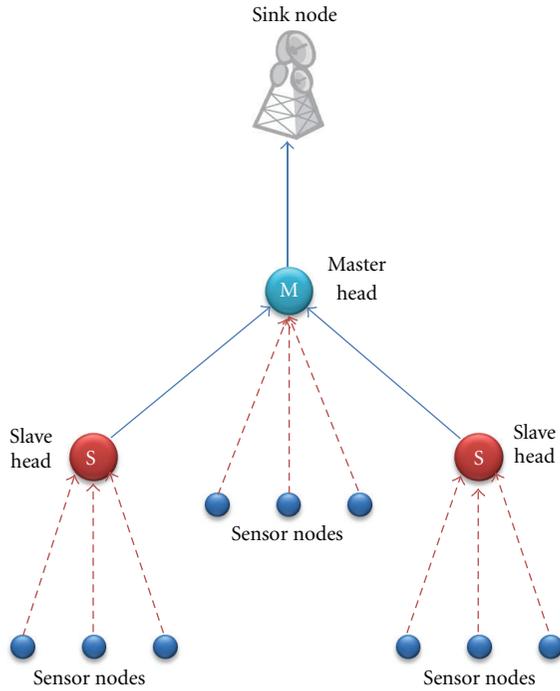


FIGURE 2: Cluster hierarchy of EFMP.

into master heads and slave heads. The structural difference between a general cluster structure and the proposed EFMP general cluster structure is the presence of a master head that collects and manages data from cluster heads. The master head collects information from the cluster heads and sends the collected information to the sink node.

Existing clusters only collect and send information from the cluster head. They use up large amounts of energy since each cluster head independently transmits data if there are many cluster heads. EFMP achieves better efficiency by reducing the number of transmissions compared with

cluster-based cluster head methods by electing a master head and transmitting information by collecting data, in the case of a fire, from sensor nodes, according to information on the fire, as shown in Figure 2. Sensor nodes are the smallest unit forming a sensor network. All the sensor nodes can be slave heads or master heads, depending on the environment. If a master head is elected among the cluster heads, all the cluster heads except the master head become slave heads.

Slave heads send information to the master head. The master head sends information to the sink node in batches by collecting data received from the slave heads. From among the existing master head candidates, the cluster head with the least number of transmissions to the sink node is selected as the master head. If the number of transmissions is the same, then the one that is closest to the sink node, with the least number of sensors detecting fire, or the cluster head with the most battery power, is selected as the master head.

Figure 3 shows the changes in the roles of the nodes according to changes, in the case of a fire, and all the nodes are initialized while in watch mode.

Watch mode refers to the initial state, in which the sensors did not detect a fire. The cluster head that first detects a fire changes from watch mode to master mode. It transforms the nearby cluster heads in watch mode to slave mode since it is in master mode itself, that is, the EFMP system displays much better energy efficiency than existing cluster networks do since the system makes hierarchical networks dynamic by changing the cluster head into a cluster head in watch mode, slave mode, or master mode, according to the direction of the fire.

3.3. Types of Packets. The features of EFMP are electing a master head that collects and manages information from slave heads managing clusters. The location of the master head changes according to information on the fire (direction of the fire). The passage defines the structural design of the EFMP and the algorithm and packet type for deciding a master head.

(1) *SIG_FIRE Packet.* If a sensor node detects fire or fire data, it immediately sends a SIG_FIRE packet to its cluster head. The fire is detected if the following condition is met, where $TEMP(t_n)$ represents the temperature measured by a sensor node at the current time t_n , $(\sum_{t=t_1}^{t_n-1} TEMP(t))/(n-1)$, represents the average temperature measured from time t_1 to t_{n-1} (i.e., the average of all temperatures measured before t_n):

$$\left| TEMP(t_n) - \frac{\left(\sum_{t=t_1}^{t_n-1} TEMP(t) \right)}{(n-1)} \right| > \Delta TEMP_{MAX}. \quad (1)$$

Thus, if the difference between the currently measured temperature and the average of the previously measured temperatures is greater than a specific limit $\Delta TEMP_{MAX}$, the sensor node transmits a SIG_FIRE packet to its cluster head.

Figure 4 shows the structure of the SIG_FIRE packet. As shown in the figure, the SIG_FIRE packet consists of Sensor_ID and Sensor_data fields. The Sensor_ID field includes

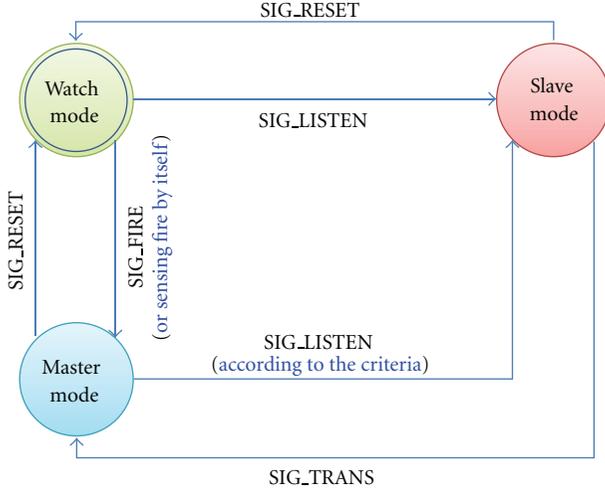


FIGURE 3: Role-transition diagram for cluster heads.

Type	Content	
SIG_FIRE	sensor_ID	sensor_data

FIGURE 4: SIG_FIRE packet format.

the unique identifier of a sensor node (e.g., the MAC address or IP address of a Zigbee node), and the Sensor_data field includes currently measured temperature data.

(2) *SIG_DATA Packet*. SIG_DATA is gathered at each cluster head to be transmitted to the sink node. The sensor monitors the fire and sends information to the slave heads by detecting fire data. The sensor node receiving information relays the fire data (SIG_FIRE) collected to a master head, and SIG_DATA is used when sending information from slave heads to the master head. Figure 5 shows the structure of the SIG_DATA packet. The SIG_DATA packet consists of Head_ID, Sensor_ID_N, and Sensor_data_N fields. The Head_ID field is a unique identifier of the cluster ID and the Sensor_ID_N is the sensor node's ID in the cluster header. Sensor_data_N indicates the fire data collected from all the sensor nodes in the cluster head.

(3) *SIG_INFORM Packet*. The SIG_INFORM packet is used when a new master head provides its own information to a slave head. It is used when the first slave head within a cluster detects a fire and is changed to the master head and provides master head information to nearby sensors as well as information about the new master head when the master head is replaced. Figure 6 shows structures of SIG_INFORM packet. The SIG_INFORM packet consists of Head_ID, Bat_capacity, Num_sensors, Sensor_ID_N, and Sensor_data_N fields.

The Head_ID field is a unique identifier of the cluster ID and Bat_capacity is the remaining battery capacity.

Num_sensor is the number of sensor nodes, and Sensor_ID_N is the sensor node's ID in the cluster head. Sensor_data_N is the fire data collected from all the sensor nodes in the cluster head.

(4) *SIG_QUERY and SIG_RESP Packets*. The slave head within the cluster is transformed to the master head once the sensor detects a fire within its own cluster. The slave head becomes a master head candidate and the previous master head asks nearby master head candidates whether it is the appropriate master head. The packet format sending this information is called SIG_QUERY and the response is called SIG_RESP. Figures 7 and 8 show the structures of the SIG_QUERY packet and the SIG_RESP packet. The SIG_QUERY packet has a unique identifier, Master_ID, and Num_candidate is the number of master candidates. SIG_RESP has a unique identifier, Slave_ID. The Hop_to_master field is the number of hops between the master heads. Candidate_ID_N is the master head candidate.

(5) *SIG_LISTEN Packet*. Address information on the new master head should be sent to nearby slave heads once a new master head has been elected. Figure 9 shows SIG_LISTEN packet. The SIG_LISTEN packet provides the address information of the newly elected master head.

(6) *SIG_TRANS and SIG_RESET Packets*. Figure 10 shows structures of the SIG_TRANS, SIG_RESET packet. SIG_TRANS is used for registering information on the new master head by receiving the SIG_LISTEN packet from the new master head. It is also used when a slave head registers its own address information. The SIG_RESET packet is used when the previous master head again becomes a slave head by handing over its authority as a master head to the newly elected master head, and when the reverted previous master head registers its information with the new master head.

3.4. *EFMP Operating Procedures*. EFMP not only sends information acquired by fire detection but also provides hierarchical clusters by selecting a master head according to the direction of the fire and reduces energy consumption by sensors. EFMP is composed of a FIRE_Detection part for detecting a fire and a Startup_Monitoring part for sending information on fires. This section describes the detailed operating procedures of EFMP in two parts.

(1) *Fire Detection in Sensor Nodes*. Fire detection procedures refer to sending information to cluster heads by detecting fire for the first time from the information collected by sensors. A sensor node has an initial value of 0 for $TEMP_{SUM}$, the current temperature as the initial value of $TEMP_{VEA}$, and 1 for the initial value of n , as shown in Figure 11. The sensor node sends the SIG_FIRE packet to the cluster head if the current temperature value is inserted into $TEMP_{CUR}$ and $|TEMP_{CUR} - TEMP_{AVE}|$ is greater than $\Delta TEMP_{MAX}$.

(2) *Fire Monitoring Startup*. The cluster head among the sensor nodes receiving the SIG_FIRE packet becomes aware

Type	Content								
SIG_DATA	head_ID	num_sensors	sensor_ID_1	sensor_data_1	sensor_ID_2	sensor_data_2	...	sensor_ID_N	sensor_data_N

FIGURE 5: SIG_DATA packet format.

Type	Content								
SIG_INFORM	head_ID	bat_capacity	num_sensors	sensor_ID_1	sensor_data_1	...	sensor_ID_N	sensor_data_N	

FIGURE 6: SIG_INFORM packet format.

Type	Content					
SIG_QUERY	master_ID	num_candidate	candidate_ID_1	candidate_ID_2	...	candidate_ID_N

FIGURE 7: SIG_QUERY packet format.

Type	Content					
SIG_RESP	slave_ID	hops_to_master	num_Candidate	hops_to_Candidate_1	...	hops_to_Candidate_N

FIGURE 8: SIG_RESP packet format.

Type	Content
SIG_LISTEN	master_addr

FIGURE 9: SIG_LISTEN packet format.

Type	Type
SIG_TRANS	SIG_RESET

FIGURE 10: SIG_TRANS and SIG_RESET packet format.

of a fire breaking out from its own cluster. Figure 12 shows procedure of fire monitoring startup. The cluster head recognizing the fire transforms to master mode from watch mode, and it sends a SIG_LISTEN packet containing detection of fire to the other cluster heads and sends the SIG_DATA packet to the sink node.

(3) *Monitoring Procedure.* Figure 13 shows monitoring procedure. The sensor node for detecting fire has the function of detecting fire continuously. Detected information is sent to slave heads and the master head through a SIG_FIRE packet, and each slave head collects and relays sensor information. The slave head in a fire zone, on detecting a fire within its own cluster of slave heads, transmits that it is a candidate for master in the case of a fire through the SIG_INFORM packet. Conversely, slave heads send the SIG_DATA packet

to the master head, thinking that they are not in a fire zone when they do not detect a fire. The master head sends all the collected SIG_DATA received from the multiple slave heads to the sink node in the batch.

(4) *Master Election.* The master head of an EFMP is not a fixed type but can change continuously, depending on the fire. In other words, a new cluster will participate in fire monitoring when the fire spreads to a new cluster head. The following procedures are used to elect the new master head.

Suppose that M represents a set of master candidates that includes the current master head and all the other heads newly involved in fire monitoring, and the element of M are represented as m_1, m_2, \dots, m_L . In addition, existing slave heads are represented by s_1, s_2, \dots, s_N and the sink node is represented by sink. The master election criteria are represented by the following formula, where $\text{DIST}(a, b)$ represents the number of transmissions from node a to node b :

$$\begin{aligned} \text{DIST}_{\text{tot}}(m_x) = & \sum_{i=1}^N \text{DIST}(s_i, m_x) \\ & + \left(\sum_{j=1}^{L(\text{excl. } x)} \text{DIST}(m_j, m_x) \right) \\ & + \text{DIST}(m_x, \text{sink}). \end{aligned} \quad (2)$$

$\text{DIST}_{\text{tot}}(m_x)$ represents the total number of transmissions from each slave head to the sink node via the newly elected

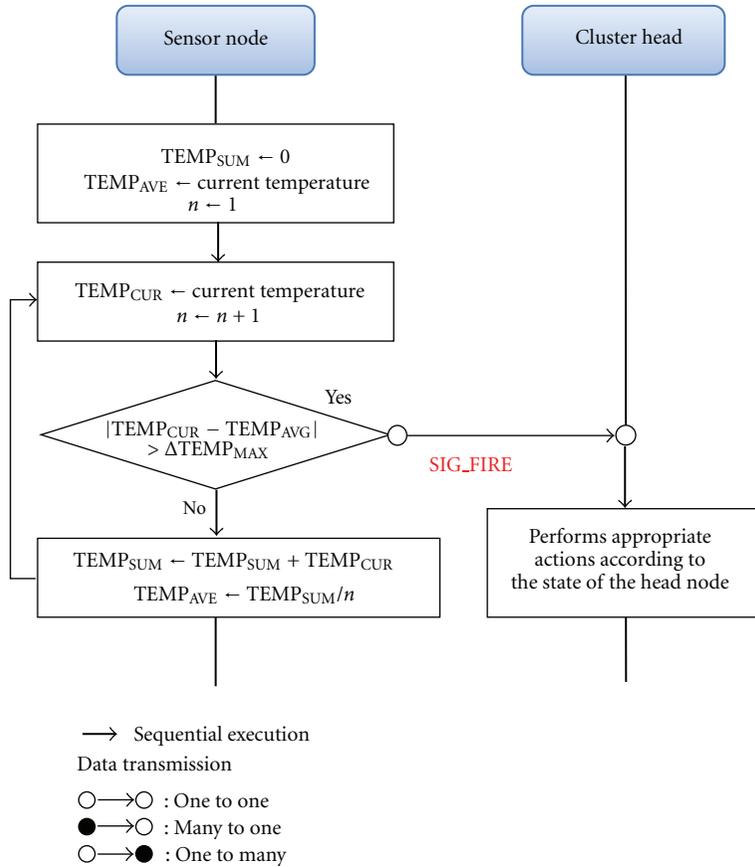


FIGURE 11: Fire detection procedure.

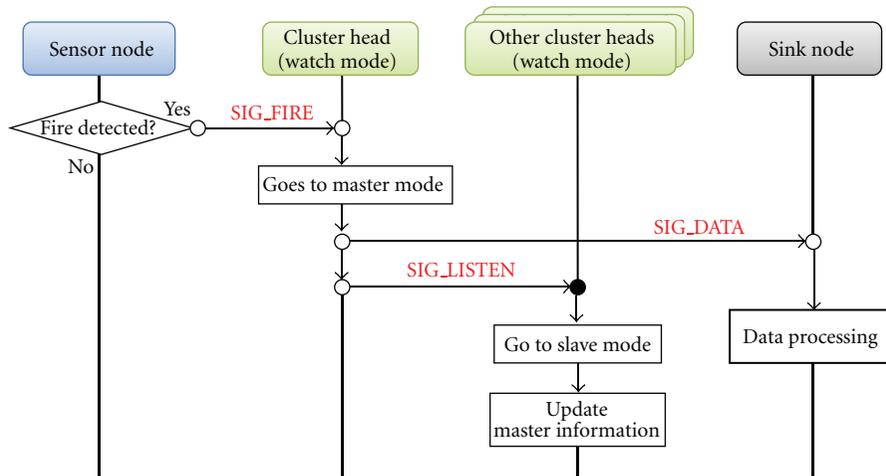


FIGURE 12: Fire monitoring startup procedure.

master head m_x . Thus, for all $m_x \in M$, the m_x that has the minimum $DIST_{tot}(m_x)$ value is elected as the new master head.

The current master head asks each master head candidate, using the SIG_QUERY packet, for the information needed to evaluate the master election criteria, and each

master head candidate replies with the SIG_RESP packet, as shown in Figure 14. If a new master head is elected according to the master head criteria, the current master head notifies the new master head with the SIG_TRANS packet, and then, the new master head sends the other slave heads information about itself through the SIG_LISTEN packet.

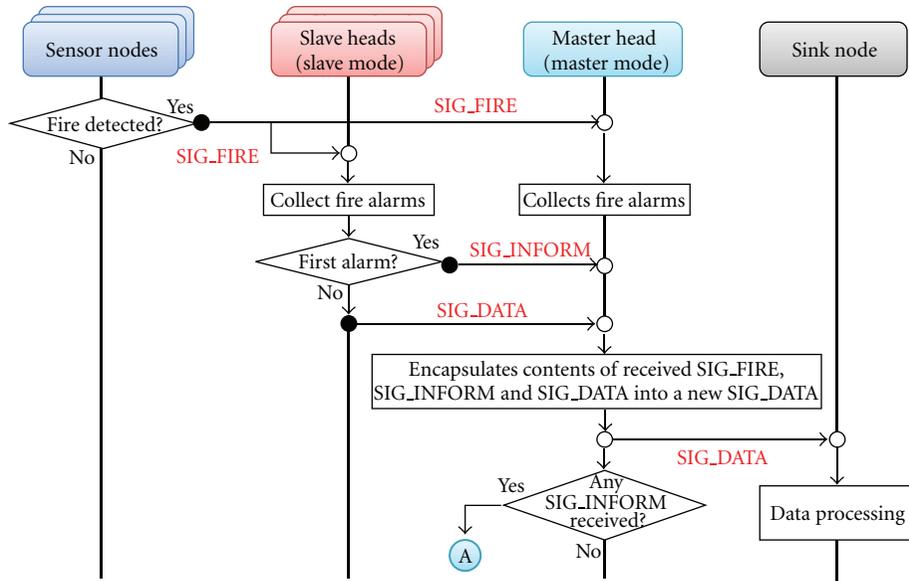


FIGURE 13: Monitoring procedure.

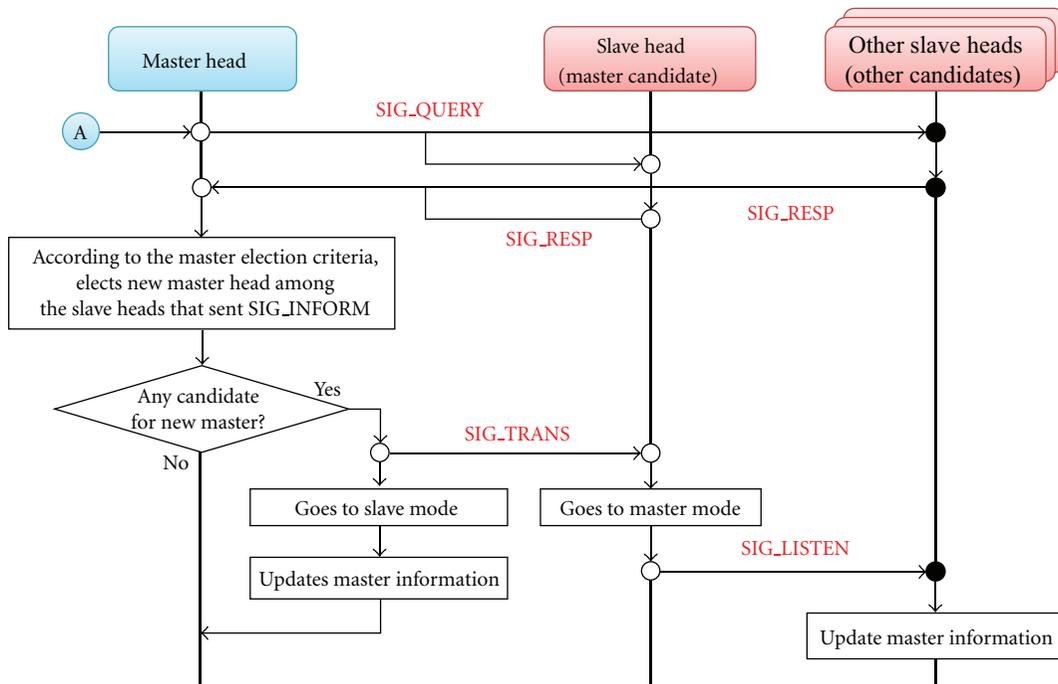


FIGURE 14: Master election procedure.

3.5. *Fire Monitoring with EFMP.* The cluster head in EFMP operates in watch mode before detection of a fire, monitors fires from its own zone, and collects sensor information. The sensor of a cluster head in watch mode, as shown in Figure 15(a), sends fire information to its own cluster head once a fire is detected within its own cluster.

The cluster head receiving information on a fire from the sensor recognizes a fire within its own cluster for the first time and changes into a master head from watch mode. First, the elected master head sends information on the fire

within its own cluster to the sink node (Figure 15(b)). It then sends the information about itself to nearby cluster heads in the slave mode through a SIG_LISTEN packet (Figure 15(c)) and the slave heads update their master head information. The cluster head where the fire broke out is elected as the first master head. A new master head, elected according to the progress of the fire, sends collected data to the sink node, as shown in Figure 15(d). It acts as the master head until another appropriate master head candidate is elected.

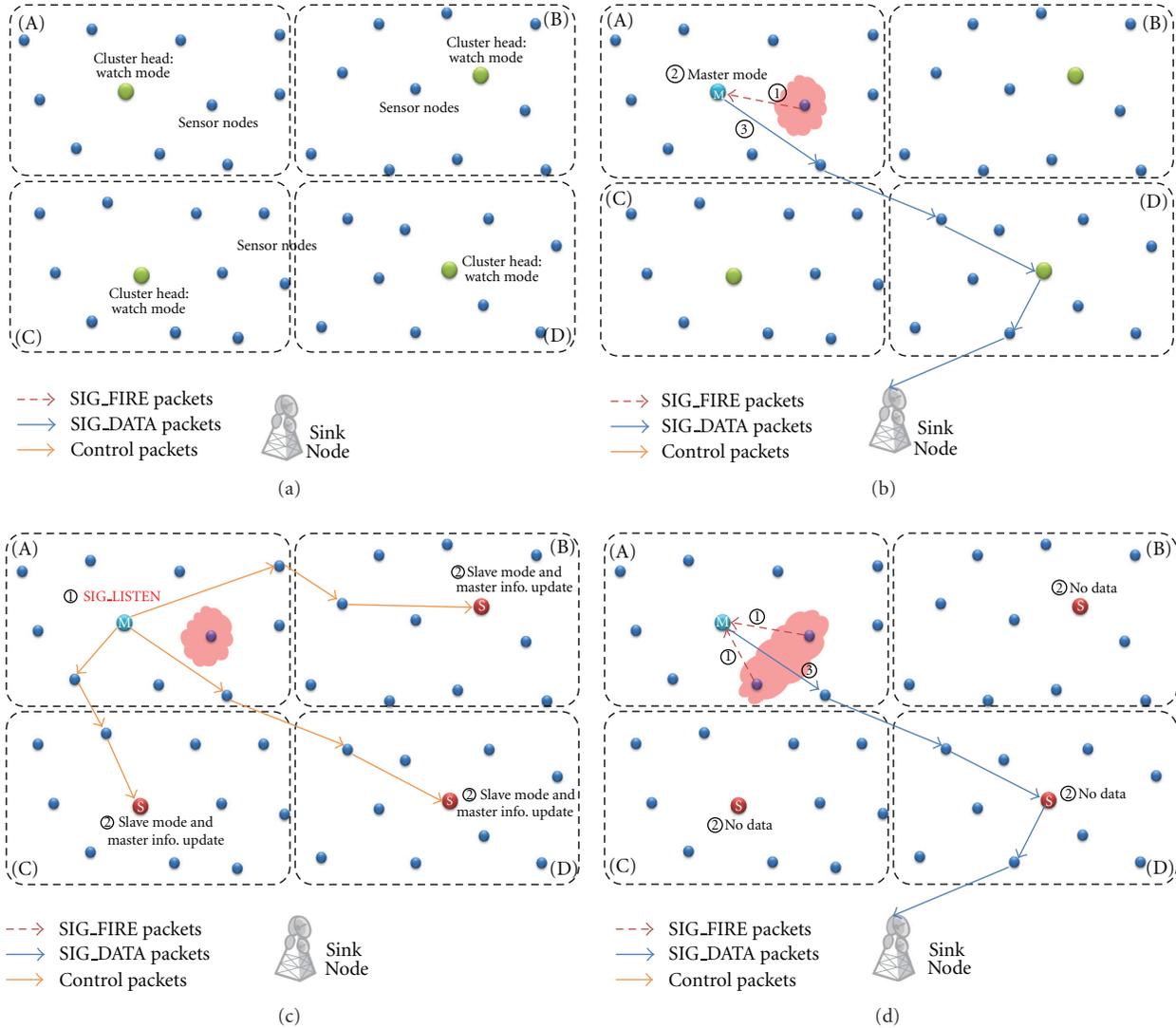


FIGURE 15: Method of electing master head from watch mode.

The number of cluster heads detecting fire within their own cluster increases as the fire spreads (Figure 16(a)). The cluster head detecting the fire can become a master head candidate, and it uses a SIG_INFORM packet to notify the current master head that it is a new master head candidate.

Figure 16(b) shows that the master head receiving the SIG_INFORM message sends each master head candidate a SIG_QUERY message asking for the total number of hops, and the master head candidates calculate the total number of hops to the sink and send the information to the master through a SIG_RESP message. The master head receiving the information compares the total number of hops for sending messages to the sink and the number of hops from each master head. It then asks the candidates having smaller numbers of hops than itself if they are available as the next master head, sends a message that it will maintain the role of the current master if the total numbers of hops from the master head candidates are greater, and maintains the position of the current master head. As shown in Figure 16, the master head

changes from cluster head (A) to cluster head (B) since the fire spreads more widely than in (C). When it becomes the master head, cluster head b sends a SIG_LISTEN packet requesting updating of the new master head information to each master head candidate, and the master head candidates receiving the messages update the master head information.

The previous master head updates the information on the next master head itself but does not update after receiving SIG_LISTEN since it knows which one is the next master head; it sends a SIG_TRANS packet indicating completion of the update and thus elects a new master head. The elected master head relays information to the sink node in batches on receiving information from each cluster head, as the previous master head had been doing.

Each cluster head sends information collected from its own sensors to the sink node in the previous cluster head method, but there are problems with different energy efficiencies in transmission by cluster according to the location of the cluster head, so the location of the cluster head needs

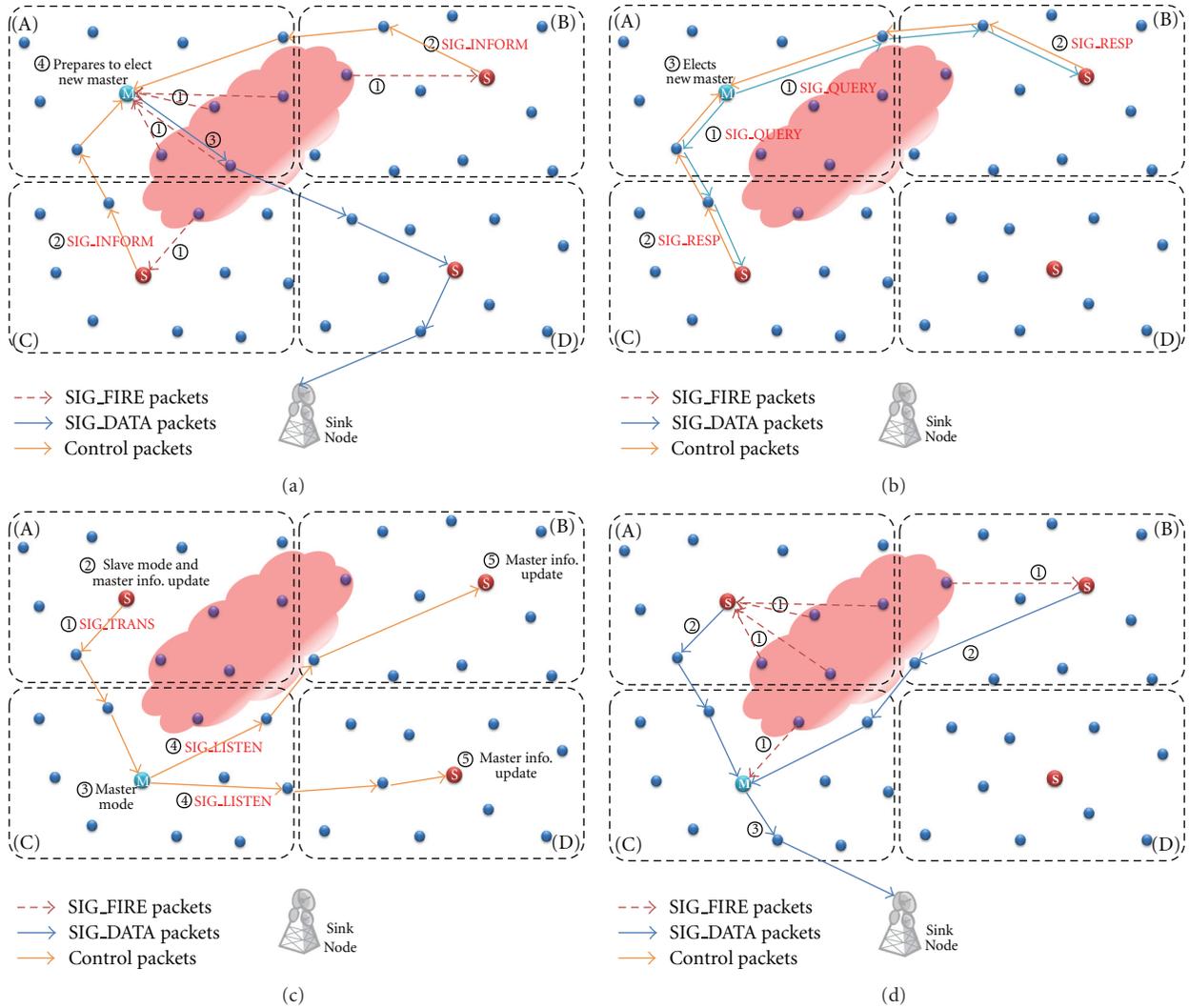


FIGURE 16: Method of electing new master head.

to be regularly changed to solve this problem. Otherwise, the overall energy efficiency of the sensor network worsens because the locations of several cluster heads are wrong. However, EFMP elects the optimal cluster head according to changes in the surroundings, and thus solves the problems of the cluster head method and increases the energy efficiency accordingly.

4. Experiments and Performance Evaluation

4.1. *Experimental Environment.* NS-2 [14] was used to evaluate the performance of EFMP in this study. A Mannasim module was used to implement the sensor network. The goal of Mannasim is to develop a detailed simulation framework that can accurately model different sensor nodes and applications, while providing a versatile testbed for algorithms and protocols. NS-2 supports modeling and simulations such as mobile and fixed wireless networks, wireless sensor networks, body area networks, ad-hoc networks, and vehicular networks for Mannasim [14, 15]. This study simulates

a wireless sensor network using IEEE 802.15.4 (Zigbee) MAC. For our experimental environment, we simulated a fire monitoring sensor network by first placing 300 nodes that are clustered into 10 clusters consisting of 30 nodes per cluster in a bounding area of $100 \times 100 \text{ m}^2$. All the nodes start with an initial energy of 10 J. The performance evaluation compares LEACH with a LEACH-based EFMP (LEACH_EFMP) and TEEN with a TEEN-based EFMP (TEEN_EFMP) in terms of the number of surviving nodes and average power consumption per sensor node of the sensor network. Each round of experiments was performed for 180 min and a total of 10 rounds of experiments were performed under the same environment. It was assumed that the fire was propagated to one adjacent cluster at a time and the direction of propagation was random.

4.2. *Experimental Results.* Figure 17 shows the number of surviving nodes during the experiments. The proposed protocols, LEACH_EFMP and TEEN_EFMP, had larger numbers of nodes alive than did the original LEACH and TEEN

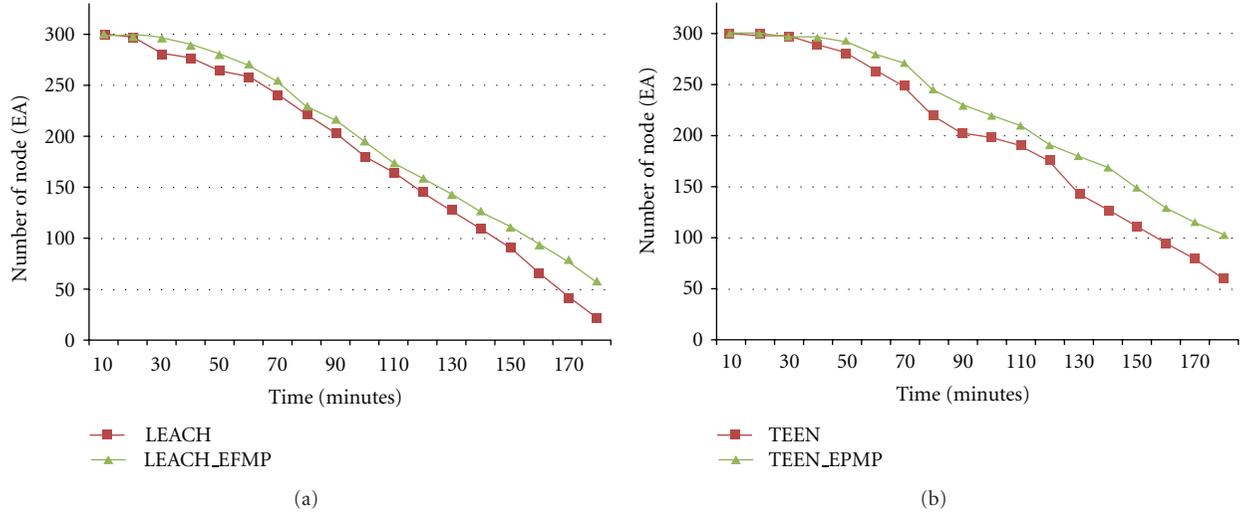


FIGURE 17: Number of surviving nodes.

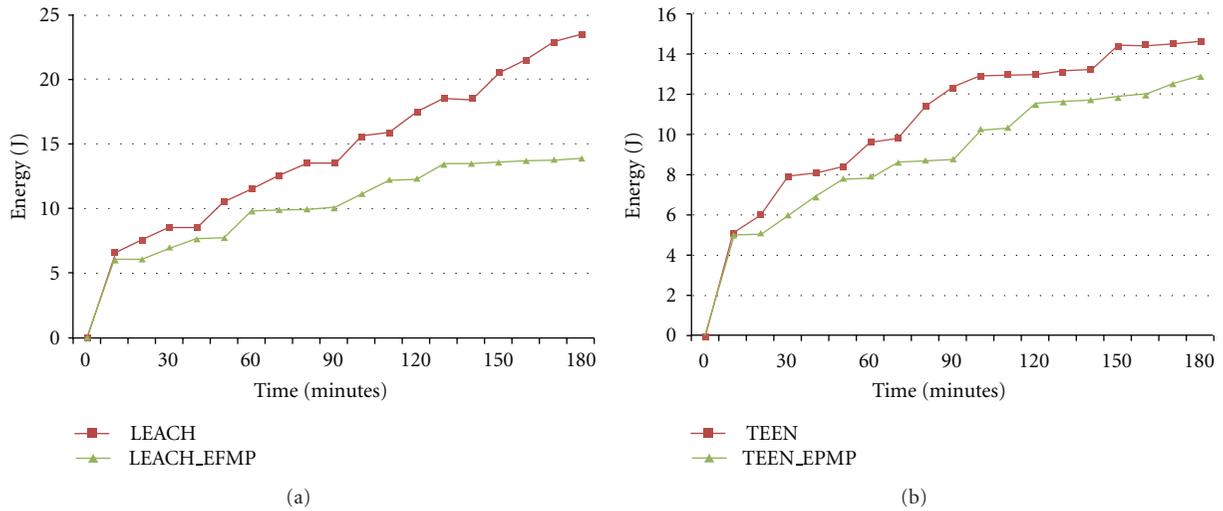


FIGURE 18: Average energy consumption over all sensor nodes.

protocols. When using LEACH_EFMP, 58 out of 300 nodes were still alive (i.e., approx. 19% of the entire nodes) after 180 min. However, only 22 nodes (i.e., 7% of the entire nodes) were alive when using the original LEACH protocol under the same condition. TEEN_EFMP also gave better results than did the original TEEN. TEEN_EFMP had 103 nodes (i.e., 34%) alive, as opposed to the case of the original TEEN protocol, where only 60 nodes (i.e., 20%) were alive.

Figure 18 shows the average energy consumption (including energy consumed for initial cluster formation of the underlying routing protocol) over all the sensor nodes during the experiments; this was calculated using the following formula, where $C(S_i, t)$ represents the energy consumed by a sensor node S_i at time t , and N represents the total number of sensor nodes:

$$\frac{\sum_{i=1}^N C(S_i, t)}{N}. \quad (3)$$

As shown in Figure 18, the average energy consumptions at the end of the experiments with LEACH and LEACH_EFMP were 23.5 J and 13.9 J respectively. The simulation results show that LEACH_EFMP used 41% lesser energy per sensor node on average than did LEACH under the same fire monitoring condition. The average energy consumptions of TEEN and TEEN_EFMP were 14.6 J and 12.9 J, respectively. The simulation results show that TEEN_EFMP used 12% lesser energy than did TEEN.

5. Conclusion

Sensor networks are the most appropriate systems in fire monitoring environments. However, sensor networks have problems in fire monitoring environments because of their limited battery capacity. The energy efficiency of a typical cluster-based sensor network drops when a large number of

sensor clusters simultaneously monitor a fire over a wide area, because of the fire monitoring characteristics. In this study, we proposed the EFMP that reduces overall energy consumption of the sensor network by dynamically forming a multilayer cluster hierarchy based on the propagation of the fire and efficiently transmitting data over the hierarchical cluster-based sensor network appropriate for fire monitoring.

The performance evaluation results showed that EFMP-based fire monitoring consumes 41% lesser energy on average per sensor node than does LEACH-based fire monitoring and 12% lesser energy on average per node than does TEEN-based fire monitoring. Furthermore, the number of sensor nodes surviving in the EFMP-based experiments was 12% and 14% more than that in the LEACH- and TEEN-based experiments, respectively.

Finally, future work will focus on estimating the size, speed, and direction of a fire by extending EFMP in general fire monitoring environments. We hope that this study will contribute to protecting more lives and properties from fire disasters.

Acknowledgment

This paper was supported by Konkuk University in 2010.

References

- [1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless micro-sensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 8, pp. 1–10, January 2000.
- [2] F. Xiangning and Y. Song, "Improvement on LEACH protocol of wireless sensor network," in *Proceedings of the International Conference on Sensor Technologies and Applications (SENSOR-COMM '07)*, pp. 260–264, October 2007.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.
- [4] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS '01)*, vol. 3, pp. 2009–2015, 2011.
- [5] 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 Communications Networks*, pp. 368–372, 2002.
- [6] L. Zou, M. Lu, and Z. Xiong, "A distributed algorithm for the dead end problem of location based routing in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 4, pp. 1509–1522, 2005.
- [7] D. E. Heim, R. E. Fontana, C. Tsang, V. S. Speriosu, B. A. Gurney, and M. L. Williams, "Design and operation of spin valve sensors," *IEEE Transactions on Magnetics*, vol. 30, no. 2, pp. 316–321, 1994.
- [8] J. Qiangfeng and D. Manivannan, "Routing protocols for sensor networks," in *Proceedings of the 1st IEEE Consumer Communications and Networking Conference (CCNC '04)*, pp. 93–98, January 2004.
- [9] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.
- [10] A. A. Ahmed and H. Shi, "A survey on network protocols for wireless sensor networks," in *Proceedings of the of Information Technology: Research and Education (ITRE '03)*, pp. 301–305, 2003.
- [11] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S8–S13, 2005.
- [12] C. Patra, A. Guha Roy, S. Chattopadhyay, and P. Bhaumik, "Designing energy-efficient topologies for wireless sensor network: neural approach," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 216716, 7 pages, 2010.
- [13] Y. Gu, Q. Wu, and N. S. V. Rao, "Optimizing cluster heads for energy efficiency in large-scale heterogeneous wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 961591, 9 pages, 2010.
- [14] "Network Simulator-2," <http://isi.edu/nsnam/ns/>.
- [15] "Mannasim for Wireless Sensor Network Simulation," <http://www.mannasim.dcc.ufmg.br/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

