Agent-based Framework for Energy Efficiency in Wireless Sensor Networks

Hongjoong Sin, Jangsoo Lee, Sungju Lee, Seunghwan Yoo, Sanghyuck Lee, Jaesik Lee, Yongjun Lee, and Sungchun Kim

Abstract**—** Wireless sensor networks are consisted of hundreds or thousands of small sensors that have limited resources. Energy-efficient techniques are the main issue of wireless sensor networks. This paper proposes an energy efficient agent-based framework in wireless sensor networks. We adopt biologically inspired approaches for wireless sensor networks. Agent operates automatically with their behavior policies as a gene. Agent aggregates other agents to reduce communication and gives high priority to nodes that have enough energy to communicate. Agent behavior policies are optimized by genetic operation at the base station. Simulation results show that our proposed framework increases the lifetime of each node. Each agent selects a next-hop node with neighbor information and behavior policies. Our proposed framework provides self-healing, self-configuration, self-optimization properties to sensor nodes.

Keywords**—**Agent, Energy Efficiency, Genetic algorithm, Wireless Sensor Networks.

I. Introduction

Wireless sensor networks are consisted of thousands of small sensors that span a large geographical region. These sensors are able to communicate with each other to collaboratively detect objects, collect information and transmit messages. However, as sensors are usually small in size, they have many physical limitations – such as battery, computational power and memory. Because of those limitations, energy-efficient techniques are main research challenges in wireless sensor networks. A number of techniques have been proposed to solve these challenges.

LEACH (Low-Energy Adaptive Clustering Hierarchy) [1] is one of the famous techniques in wireless sensor networks. This is a cluster-based protocol that utilizes randomized rotation of local cluster base stations (cluster-heads) to evenly distribute the energy load among the sensors in the network. This technique can reduce a number of transmissions in clusters. However, the intense data flow can be harmful, especially in wireless sensor networks, since congestions and collisions may be occurred. Traditional server/client-based techniques like a LEACH cannot utilize autonomous-repair and scalability and so on. And also it gives too much burden on base-station. Therefore it is need to use the distributed approach to improve this critical issue.

Mobile agents are software component related to code mobility that it allows the management intelligence to migrate the network elements and executive locality [2]. The agent-based technique allows WSN applications to exhibit self-healing, self-configuration and self-optimization properties. To achieve these properties, agent behavior policies have to be optimized automatically.

Market-oriented approach [3, 4] is motivated on e-commerce business. A sensor management system recommends a new information product (such as a target track) based on what previous users in similar situations selected. Biologically inspired approaches are based on an observation to the various biological systems. MONSOON [5, 6, 7] is designed to support data collective applications, event detection applications and hybrid applications. Each application is implemented as a decentralized group of software agents, analogous to a bee colony (application) consisted of bees (agents).

In this paper, we propose a framework that it includes energy-efficient agent behavior policies and optimization schemes to prolong a network lifetime. The rest of this paper is organized as follows. Section II presents related work. We describe our proposed framework in Section III. Simulation results are presented in Section IV. Section V concludes the paper.

II. Related work

Recently, agent-based techniques have been proposed for efficient data dissemination in wireless sensor networks. In [8], the rule for deciding behavior of each agent was specified to optimize its behavior. Agilla [9] is proposed to middleware adopted agent-based paradigm. Agents in these techniques don’t consider optimization of agent behavior policy, so they cannot work efficiently in wireless sensor networks.

In [3, 4], the authors model the system as a market and explore the advantages of incorporating e-commerce concepts to agent management. At first sensor produces information “good”, and finally they maximize the amount of goods delivered without exceeding the sensor’s budget. This is a centralized system and it is very difficult to implement in a distributed way because of the complexity of these models and the variable decision problem.

BiSNET [5] allows agents to autonomously adapt to dynamic network conditions. However, it does not investigate evolutionary adaptation. Agent behavior policies are manually

All authors are with the Computer Science & Engineering Department, Sogang University, Seoul, South Korea (e-mail: siceman00@gmail.com, jangso417@hotmail.com, ksc@sogang.ac.kr).

configured through trial-and-error and fixed at runtime.

MONSOON [6] proposed an evolutionary multi-objective adaptation framework, in biologically inspired application architecture, called BiSNET/e. The architecture of MONSOON is shown in Fig. 1. The MONSOON consists of two types of software components: agents and middleware platforms. Each application is implemented as a decentralized group of software agents, analogous to a bee colony consisting of bees. Agents collect sensing data and/or detect an event on individual nodes, and carry sensing data to the base station. They perform these data collection and event detection functionalities by sensing their surrounding environment conditions and adaptively invoking biologically inspired behaviors such as pheromone emission, reproduction and migration. Each agent has its own behavior policy, as a gene, which defines how to invoke its behaviors. At the end of each round, MONSOON servers collect agents which are arrived at BS and elects elite agents for heuristic optimization using genetic algorithm. MONSOON allows agents to evolve their behavior policies (genes) and adapts their operations to given objectives. But each agent decides their behavior based on “logical energy”. It is a logical concept and does not represent the amount of physical battery in a node. It is not helpful for load balancing of sensor nodes. In our proposed framework, we use physical battery for load balancing of routing.

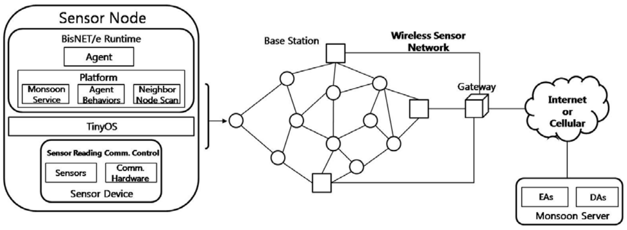


Fig. 1 The Architecture of MONSOON

III. Proposed Framework

Our proposed framework is designed to implement sensing data collection applications. Each node deploys a Sensing Agent(SA) with a randomly-generated behavior policy. When a SA has enough data to transmit, a SA generates a Delivery Agent(DA). A DA contains the sensing data that a SA collected. Each DA carries the data toward the base station using 1-hop neighbor information or waits one or more duty-cycles to aggregate other DA’s data. Several duty-cycles are included in one-round. After each round, the base station elects elite agents through fitness function. The base station propagates elected elite agents to each node in the network. Each node selects one of the propagated elite agents and performs genetic operations.

A. Agent Aggregation Policy

Agents may aggregate(merge) with others at intermediate nodes on their ways to the base station. On each intermediate node, each agent decides whether it migrates to a neighbor-node or waits for other agents to arrive at the current node and aggregate with them. This decision is made by the probability of wait for aggregation(P_w). When an agent decides to wait for other agents, it stays at the current node during one duty-cycle and decides whether it migrates or waits. If an agent meets other agents during a wait period, they compare their policies with SA of the current node and elect most similar agent to aggregate other agents. The elected agent contains the sensing data that other agent carries. The aggregation behavior is efficient to save power consumption for transmission. It can reduce the number of data transmissions. But all sensing data have to arrive at the base stations in a time limit. Limitless waits on the way to the base stations make sensing data violate their time limits. So we setup the threshold value to limit the number of waits. Maximum waits ratio(WR_{max}) is allocated randomly at the initialization phase as a gene of agents. When a SA decides to generate a DA, it calculates the number of maximum waits based on the WR_{max} value. When a DA waits more than maximum waits, it doesn’t wait for aggregation on its way to the base station.

Fig. 2 shows aggregation example of our proposed framework. DA2 is generated at node F and migrate to node C. DA2 decides to wait for aggregation. When DA1 arrived at node C, DA2 and DA1 choose a leader and leader agent aggregate other agent.

B. Agent Migration Policy

On each intermediate node toward a base station, each agent chooses migration of a destination node by 1-hop neighbor information. At the beginning of each round, a base station disseminates a message with elite agents to make each node measure its distance from a base station. Agents can estimate where a base station exists approximately using this distance. When an agent fails to migrate to destination, it counts the number of errors and writes it down during each duty-cycle. At the beginning of each duty-cycle, error counter becomes half and accumulates the number of errors in this duty-cycle.

To decide which node is the best to migrate, each agent uses 1-hop neighbor information which includes distance from base station, remaining energy, error rate and existence of agents at each node. All of these variables are calculated by the weighted sums. Equation (1) is examined to determine which next-hop it migrates to.

$$WS_j = \frac{E_j}{E_{Max}} \left(W_1 \frac{H_{cur} - H_j}{H_{cur} - H_{Min}} + W_2 \times ERR_j + W_3 \times A_j \right) \quad (1)$$

An agent calculates this weighted sum (WSj) for each neighboring node j, and migrates to a node that generates the highest weighted sum. E_j represents the ratio of remaining energy of a neighboring node j, E_{Max} represents the initial energy level. H_{cur} denotes the distance from a base station to the current node, H_{Min} denotes the minimum distance of all neighboring nodes of the current node, H_j denotes the distance from a base station to the neighboring node j. ERR_j represents the error ratio between the current node’s neighboring nodes. A_j denotes the existence of other agent in a neighboring node j. Equation (2) and (3) shows how to calculate ERR_j and A_j.

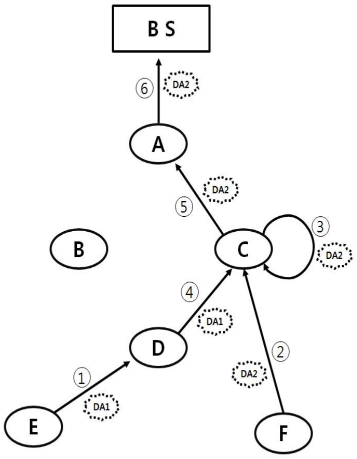
$$ERR_j = \frac{ER_j - ER_{Min}}{ER_{Max} - ER_{Min}} \tag{2}$$

$$A_j = \begin{cases} 1, & \text{if } Node\ j\ has\ DA, \\ 0, & otherwise \end{cases} \tag{3}$$

ER_{Max} and ER_{Min} represent the maximum and minimum number of errors of all neighboring nodes. ER_j is the number of error of neighboring node j.

The weight values(W_1, W_2, W_3) in (1) determine how agent perform the migration behavior. If W_1 is bigger than other values, each agent can migrate to the neighbor node which has smallest distance value. On the other hands, if W_3 is the biggest value, each agent migrates to neighbor node which has other DAs. It increases a chance of aggregation to reduce the number of communications. Each agent is able to avoid hazards using error rate information.

Simple migration example is shown in Fig. 2. The circled numbers represent the sequence of migration steps. ① Node E generates DA1 and DA1 determines whether migrates to. Node E choose node by comparing weighted sums. In weighted sums, node B is closer to a base station than node D but node B has a small amount of battery. DA1 migrates to node D. ② Node F generates DA2. DA2 calculates (1) to make a decision. DA2 migrates to node C. ③ In node C, DA2 waits for other agents to aggregate during one duty-cycle. ④ DA1 compares node B with C. DA1 decides to migrate to node C. DA 2 waited in node C aggregates DA1 and contains sensing data of DA1. ⑤-⑥ DA2 migrates to node A and a base station.



Node	Dist	Energy	Error
A	1	0.8	0.2
B	2	0.2	0.7
C	2	0.75	0.5
D	3	0.9	0.3
E	3	0.7	0.8
F	3	0.8	0.9

DA	P_w	Nw_{max}	W_1	W_2	W_3
DA1	0.25	3	0.7	-0.3	0.4
DA2	0.13	5	0.5	-0.2	0.3

Fig. 2 Aggregation and Migration Example of Proposed Framework

C. *Elite Agent Election*

Delivery agents consider three conflicting objectives: cost, latency, aggregation efficiency of their migration from source node to a base station. Equation (4)-(6) represent each objective.

$$Cost = \frac{The\ number\ of\ migration}{Distance\ of\ the\ source\ node} \tag{4}$$

$$Latency = \frac{Agent\ traveling\ delay}{Actual\ distance} \tag{5}$$

$$Aggregation\ Efficiency = \frac{The\ number\ of\ aggregation}{Total\ number\ of\ aggregation} \tag{6}$$

Equation (4) shows the overheads occurred to find efficient way for migration. (5) represents the time required for an agent to travel to a base station from a source node. Aggregation efficiency is measured by (6).

At the end of each round, a base station elects elite agents using (4)-(6). All agents arrived in current round are measured by these equations.

D. *Genetic Operation*

P_w, WR_{max} and weight values(W_1, W_2, W_3) are allocated randomly when system is initialized. We adopt a genetic algorithm to optimize these variables.

At the beginning of each round, a base station propagates elected elite agents to each node. Each SA selects one elite agent which has the most similar variables(genes). Gene similarity is measured with the Euclidean distance between the values of two genes. It executes half-to-half crossover process with the elite agent. When SA generates DA, it inherits its gene to DA and occurs mutation with a certain probability. Mutation position is selected randomly and mutation range is predefined by a base station. Fig. 3 shows a flowchart of our genetic operation.

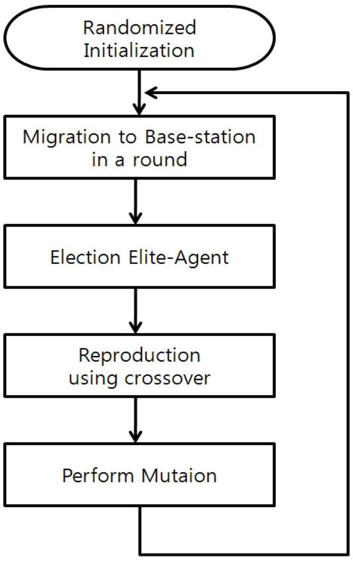

Randomized Initialization
Migration to Base-station in a round
Election Elite-Agent
Reproduction using crossover
Perform Mutaion

Fig. 3 Flowchart of Genetic Operation

IV. Simulation

A. Simulation Model

To simulate our proposed scheme, we have implemented our proposed framework in NS-2(Network Simulator ver. 2.31)[11]. LEACH[1] and MONSOON[6] is implemented to compare with our proposed framework. Table 1 shows the basic parameters configuration in simulation.

TABLE 1.
BASIC PARAMETERS CONFIGURATION

Network Scale and Node Environment	
Network Size	100m X 100m
Maximum Transmission range	10m
Total number of Sensor nodes	25-100
Topology Configuration mode	Randomized
MAC Protocol	IEEE 802.15.4
Initial Energy	1 J
Sensing data Specification	
Size of Sensing data	Default : 1 Kbytes
Duty-Cycle Duration	1 second
Round Duration	60 seconds
Genetic Operation Specification	
Mutation Probability	0.025
Range of Pw, WRmax, W1, W2, W3	0 ≤ Pw, WRmax, W1, W2, W3 ≤ 1

B. Simulation Results

Fig. 4 shows the number of activate nodes during 20 rounds. As the graph shows, LEACH consumes much energy in cluster head nodes. It reduces the lifetime of the wireless sensor networks. MONSOON and our proposed framework reduce a number of transmissions using aggregation method. In our proposed framework, each agent selects migration node with remaining energy. Equation (1) makes agent easy to achieve load balancing. Therefore successful delivery ratio of all agents is higher than other techniques. This result is represented in Fig. 5.

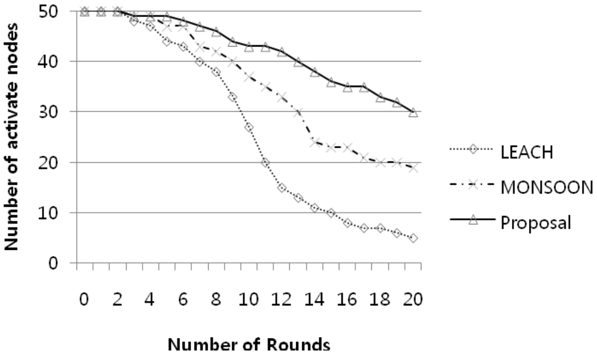


Fig. 4 Activate nodes with 50 nodes

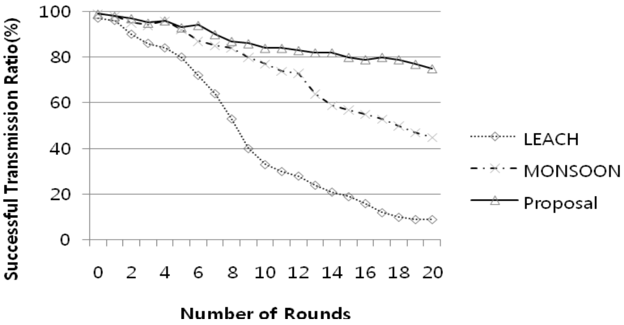


Fig. 5 Successful delivery ratio

The efficiency of genetic operations is shown in Fig. 6. At the beginning of a simulation, our proposed framework and MONSOON consume more power then LEACH because agents use random behavior policies. However, agents evolve their behavior policies as time goes by. Each agent achieves near optimal behavior policies using heuristic genetic operations.

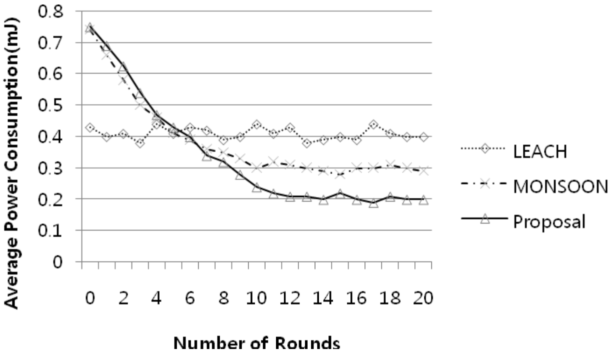


Fig. 6 Average power consumption

V. Conclusion and Future Works

Wireless sensor networks are widely applicable to various environments. Sensor nodes have a lot of limitations such as battery, memory, computational power, etc. Energy-efficient techniques are main research challenge in wireless sensor networks.

This paper proposes an energy efficient agent-based framework in wireless sensor networks. Agent operates automatically with their behavior policies as a gene. When a node has sensing data, a sensor node generates an agent. The agent contains the sensing data and carries the data toward the base station hop by hop. The base station elects elite agents that they have more appropriate behavior policies in current network situation by periods. The base station propagates elected elite agents to each node. Agents in each node inherit behavior policies of elite agents.

Simulation results show that our proposed framework increases the lifetime of each node. Each agent selects a next-hop node with neighbor information and behavior policies.

This process provides self-healing, self-configuration, self-optimization properties to sensor nodes.

In simulation, we don't measure the efficiency of aggregation and the self-healing results. We are going to measure and compare those in our future works. Additionally, parameters (reproduction probability, mutation probability, crossover point) of genetic algorithm have to be decided reasonably.

ACKNOWLEDGMENT

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund)(KRF-2007-313-D00533) and the Brain Korea 21 Project in 2008.

REFERENCES

[1] W.R.Heinzelman, A.Chandrakasan, and H.Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000.

[2] Min Chen, Taekyoung Kwon, Yong Yuan, and Victor C.M. Leung, "Mobile Agent Based Wireless Sensor Networks", In *Journal of Computers*, Vol. 1, NO. 1, pp.14-21, April 2006.

[3] Tracy Mullen, Viswanath Avasarala, and David L. Hall, "Custom-Driven Sensor Management", In *IEEE INTELLIGENT SYSTEMS*, Vol. 21, NO. 2, pp. 41-49, March/April 2006.

[4] Geoffrey Mainland, David C. Parkes, and Matt Welsh, "Decentralized, Adaptive Resource Allocation for Sensor Networks", In *Proceedings of the 2nd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005)*, May 2005.

[5] Pruet Boonma, and Junichi Suzuki, "BiSNET: A biologically-inspired middleware architecture for self-managing wireless sensor networks", In *Computer Networks*, Vol.51, NO.16, pp.4599-4616, 2007.

[6] Pruet Boonma, and Junichi Suzuki, "MONSOON: A Coevolutionary Multiobjective Adaptation Framework for Dynamic Wireless Sensor Networks", In *Proceedings of the 41st Hawaii International Conference on System Sciences*, 2008.

[7] Pruet Boonma, and Junichi Suzuki, "Exploring Self-star Properties in Cognitive Sensor Networking", In *Proc. of IEEE/SCS International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, Edinburgh, UK, June 2008.

[8] Ronan Mac Ruairi, and Mark T. Keane, "An Energy-Efficient, Multi-Agent Sensor Network for Detecting Diffuse Events", In *IJCAI*, pp 1390-1395, 2007.

[9] Chien-Liang Fok, Gruia-Catalin Roman, and Chenyang Lu, "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications", In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 653-662, 2005.

[10] Hongliang Ren, and Max Q.-H. Meng, "Biologically Inspired Approaches for Wireless Sensor Networks", In *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, June 25 - 28, 2006, Luoyang, China.

[11] The networks simulator-ns-2.
http://www.isi.edu/nsnam/.