Sensors and Materials, Vol. 28, No. 6 (2016) 739–743
MYU Tokyo

S & M 1231

Combining a Multi-Hop Distributed Energy-Efficient Cluster Algorithm with Fuzzy Logic Control for Wireless Sensor Networks

Young-Long Chen,* Li-Hsien Chang, and Tuo-Jing Huang

Department of Computer Science and Information Engineering,
National Taichung University of Science and Technology No. 129, Sec. 3, Sanmin Rd., Taichung 404, Taiwan

(Received April 1, 2015; accepted April 22, 2016)

Keywords: WSN, cluster, energy-efficient, coverage problem, fuzzy logic control

In this paper, we propose a multi-hop distributed energy-efficient cluster fuzzy logic control (MHDEECFLC) architecture to solve the coverage problem of wireless networks, prolong network lifetime, and achieve a balance regarding the energy consumption of nodes. We use fuzzy logic control (FLC) to choose fitter nodes that work as cluster heads (CHs). MHDEECFLC architecture enables the choice of fitter nodes that work as CHs to select the phase of the CHs to achieve better energy efficiency and to prolong a wireless sensor network (WSN) lifetime. According to the simulation results compared with previous proposed methods, MHDEECFLC architecture has a longer lifetime and greater coverage efficiency. MHDEECFLC architecture can improve both energy consumption between nodes and the lifetime of a network.

1. Introduction

The wireless sensor network (WSN) technology involves combining sensors. The sensor nodes (SNs) can then communicate with the wireless network.(1) The WSNs are used in military, medical, environmental, and agricultural applications, among others.(2,3) An important issue is the design of energy-efficient algorithms to prolong the lifecycle of a WSN.(4) The coverage problem(5) is also an important issue in a WSN.

The low-energy adaptive clustering hierarchy (LEACH)(6) is divided into the setting phase and the steady-state phase. In the setting phase, we assume that there are n SNs in the WSN.

Fuzzy logic control (FLC) is widely used in many fields. It is especially used for reducing resource consumption in a WSN.(7) FLC is combined with a fuzzification interface, a rule base, an inference engine, and a defuzzification interface. The fuzzification interface maps the input values onto suitable linguistic values. The rule base represents the thinking rules within the entire system, usually according to experts' experience or by building a training sample. The inference engine makes a proper output determination when the input parameter triggers many rules simultaneously. The defuzzification interface calculates the inference results.(8)

This paper proposes a multi-hop distributed energy-efficient clustering with FLC (MHDEECFLC) architecture. The MHDEECFLC architecture chooses the cluster head (CH) using FLC; therefore, energy consumption between nodes can be reduced and the lifetime of the WSN prolonged.

---

*Corresponding author: e-mail: ylchen66@nutc.edu.tw

ISSN 0914-4935 © MYU K.K.

2. MHDEECFLC Architecture

In this section, we describe the architecture of MHDEECFLC. The MHDEECFLC architecture is divided into choosing the CH phase, building the cluster phase, and the steady state phase, and recombining the cluster phase.

2.1 Choosing the CH phase

In choosing the CH phase, the base station (BS) sends all nodes an initial message. Each node calculates the number of neighboring nodes ND_i; the equation is

$$ND_i = count(\{i|dist(i, N_\mathrm{a}) \le R_\mathrm{c}, i \ne N_\mathrm{a}\}), \tag{1}$$

where i represents the ith node in WSN, N_a represents the normal node a, $dist(i, N_\mathrm{a})$ is computed as the distance from i to N_a, and R_c represents the nodes' broadcast range. After calculating the ND_i, each node waits for some delay time T_i and broadcasts a density message. The equation for T_i is

$$T_i = \beta e^{\frac{1}{ND_i}}, \tag{2}$$

where β is a constant to ensure $0 < T_i < T_\mathrm{max}$. Next, every node calculates the distance from the BS to its location DBN_i. Next, DBN_i, ND_i and the remaining energy of node $E_{\mathrm{re}(i)}$ are used to calculate the probability that a CH will be selected ($PCCH_i$). In this phase we use FLC to choose fitter nodes that serve as CHs. The FLC is divided into a fuzzification interface, a rule base, an inference engine, and a defuzzification interface. As Fig. 1 shows, every node calculates its $PCCH_i$ using FLC through fuzzification, inference, and defuzzification. After calculating the $PCCH_i$, every node sends a competitive message in each broadcast range and competes for $PCCH_i$ with neighboring nodes. If a node discovers that its $PCCH_i$ is the highest in the competitive message, the node declares that it has become a CH. Otherwise, the node gives up on becoming a CH and waits for a state message.

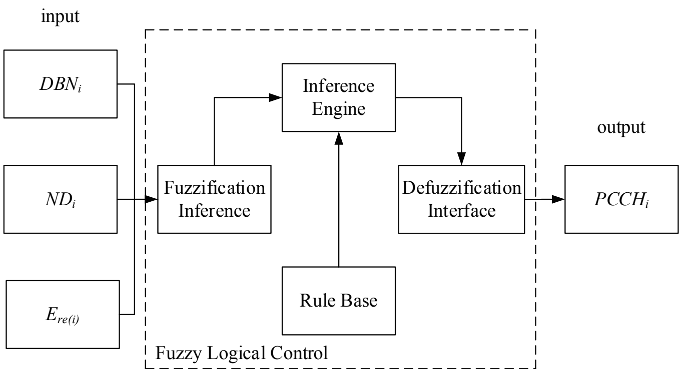

Fig. 1. Using FLC select CH selection probability.

2.2 Building cluster phase

In the building cluster phase, the CH sends a state message to neighboring nodes in its broadcast range. A non-cluster node receives the CH's state message and replies with a joint message to the CH. The CH notes and updates its sub-node table. This non-cluster node becomes a 1-hop node. The 1-hop node sends a state message in its broadcast range, and the non-cluster node replies with a joined message to the 1-hop node from which it received the state message. The 1-hop node updates its sub-node table, and this non-cluster node then becomes a 2-hop node. This step is repeated constantly until all nodes become cluster members in the sensing field. The remote area nodes are allowed to expand their broadcast range to find a cluster.

2.3 Steady-state phase

In the steady-state phase, the SNs sense data. After sensing data, the n-hop node transfers the data to the $(n - 1)$-hop node. The medium hop k-hop node in a cluster needs to receive data from a $(k + 1)$-hop node and fuse the data to itself. The node transfers the fused data to CHs or to its parent node. The CH_f must receive data from the 1-hop node and fuse all of the received data. Finally, the CH transfers the fused data to the BS. To avoid causing rapid node death, we set a threshold value $E_{\mathrm{th(CH_f)}}$, the equation for which is

$$E_{\mathrm{th(CH_f)}} = E_{\mathrm{CH_f}} \times N_{\mathrm{frame}}, \tag{3}$$

where N_{frame} is the number of frames. Equation (3) enables the calculation of the energy consumption of CH_f in the current round. After the CHs transfer the data, they compare the retained energy $E_{\mathrm{re(CH_f)}}$ with the energy threshold value of the CH. When $E_{\mathrm{re(CH_f)}} > E_{\mathrm{th(CH_f)}}$, the CHs continue serving CHs in the next round; otherwise, the CHs broadcast a dismissal message to cluster members.

2.4 Recombining cluster phase

In the recombining cluster phase, after the CH communicates with the BS and data transfer is complete, that is defined as a round. After the rounds end, all nodes return to a non-cluster state and repeat the selection phase, the building cluster phase, and the steady-state phase.

3. Simulations Results

In this section, we discuss the performance of our MHDEECFLC architecture compared to LEACH, and distributed energy-efficient clustering with improved coverage (DEECIC). These methods are implemented in MATLAB. We assumed that the coordinate of BS was (50, 175). The sensing field was set at 100 × 100 m^2. The initial energy of each node was set at 0.25 J. The sensing range R_s was set at 10 m. The broadcast range R_c was set at 20 m. The distance threshold value d_0 was set at 75 m. The upper boundary of the waiting time T_{max} was set at 500 ms. For the simulation, we set the number of nodes N at 100 so that every node was randomly deployed. The data packet size was set at 4000 bits for each transmission time. The value of β was set at 0.18. β

is the constant used to calculate T_i. We set E_{elec} =50 nJ/bit, E_{DA} = 5 nJ/bit, $\varepsilon_{\mathrm{fs}}$ = 10 pJ/bit/m^2, and $\varepsilon_{\mathrm{mp}}$ = 0.0013 pJ/bit/m^4.

In Fig. 2, we show a comparison of the number of live nodes in every round. The first node death in LEACH occurred in round 258; the first node death in DEECIC, occurred in round 125; and the first node death in MHDEECFLC architecture, occurred in round 245. To keep sufficient coverage, we always selected higher node density areas to serve as CHs, which caused the nodes in higher density positions to consume more energy.

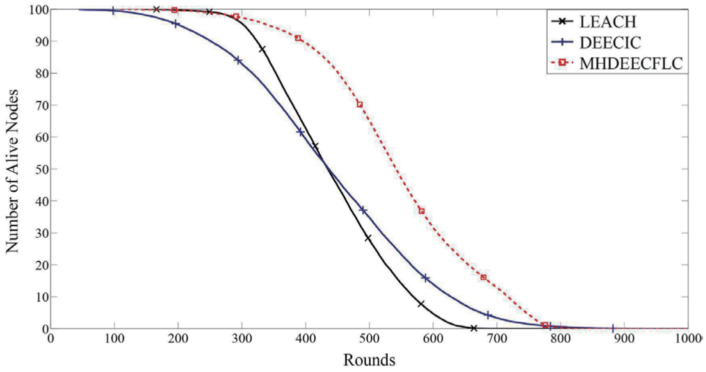

Fig. 2. (Color online) Number of live nodes in every round.

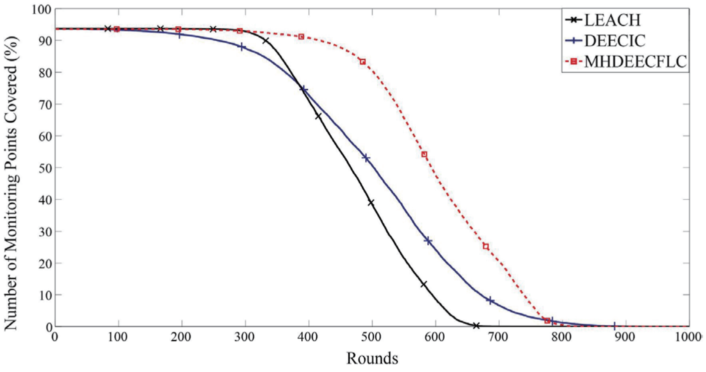

Fig. 3. (Color online) Sensor node coverage rate in every round.

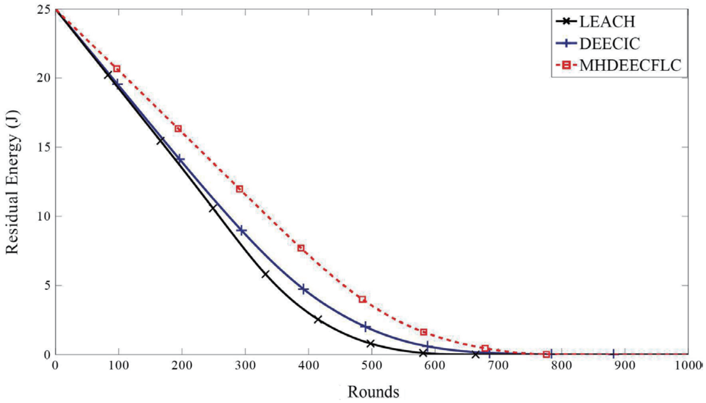

Fig. 4. (Color online) Total residual energy in every round.

In Fig. 3, we show a comparison of the coverage of SNs in every round: LEACH maintained a 31.2% coverage rate to round 520; DEECIC, a 45.1% coverage rate to round 520; and MHDEECFLC architecture, a 75.3% coverage rate to round 520.

Figure 4 shows that LEACH kept 72.4% of the total remaining energy to round 120; DEECIC, retained 73.36% to round 120; and MHDEECFLC architecture, 78.4% to round 120.

4. Conclusions

In terms of prolonging WSN lifetime, the proposed architecture was compared with LEACH. It demonstrated better performance from 1.55 to 76.7%, with DEECIC, from 31.2 to 264.8%. In coverage efficiency, in round 520, compared with LEACH, the proposed architecture achieved better performance from 138.7 to 168.9%; compared with DEECIC, from 58.3 to 86%.

Acknowledgements

This work was supported in part by the Ministry of Science and Technology (MOST) of the Republic of China under grant Nos. MOST103-2221-E-025-010 and 104-2221-E-025-001.

References

1 B. Wei, H. Y. Hu, and W. Fu: Proc. IEEE ICCEE, eds. (IEEE, Taipei, 2008) pp. 398–401.
2 J. Ko, T. Gao, R. Rothman, and A. Terzis: IEEE Eng. Med. Biol. Mag. **29** (2010) 103.
3 Y. L. Chen and X. Z. Shi: IJETI **2** (2012) 216.
4 Y. L. Chen and H. P. Lai: Comput. Math. Appl. **64** (2012) 688.
5 Manju, S. Chand, and B. Kumar: IJETI **6** (2016) 16.
6 W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan: IEEE T. Wireless Commun. **1** (2002) 660.
7 J. Kim, S. Park, Y. Han, and T. Chung: Proc. Int. Congr. Anti-Cancer Treat., eds. (ICACT, Phoenix Park, 2008) pp. 654–659.
8 Y. L. Chen, Y. S. Lin, J. W. Wang, and J. H. Wen: Comput. Commun. **31** (2008) 1901.