Chapter 7
Adaptive Forwarder Selection for Distributed Wireless Sensor Networks

Nor Azimah Khalid and Quan Bai

Abstract Wireless Sensor Network has emerged as a promising networking technique for various applications. Due to its specific characteristics, such as non-rechargeable, low-power multi-functional sensor nodes, limited sensing, computation and communication capabilities, it is challenging to build networking protocols for Wireless Sensor Networks. In this chapter, the focus is on addressing the routing issue with regards to energy efficiency and network lifetime. An adaptive and self-organized routing protocol for distributed and decentralized network, called Distributed Adaptive Forwarder Selection, is proposed. Multiple factors, involving cross layers were used for selecting the adequate forwarders for packets. The proposed approach is suitable for dynamic environments as there is no fixed topology or static role assignment for nodes in the WSN. In addition, the approach can allow sensor nodes to make flexible decisions based on their current capabilities and states. We have performed simulations of the proposed protocol and compared with two existing routing protocols in terms of node lifetime, average energy consumption and average residual energy. The results show that the proposed protocol performed better than some well known routing protocols such as LEACH and MOECS.

Keywords Distributed wireless sensor networks · Forwarder selection · Reinforcement learning

7.1 Introduction

In general, a Wireless Sensor Network (WSN) is a wireless network which consists of large numbers (hundreds to thousands) of irreplaceable and low-power multi-functional sensor nodes, operating in an unattended environment with limited sensing, computation and communication capabilities [1] used in a wide range of

N.A. Khalid (✉) · Q. Bai
Auckland University of Technology, Auckland, New Zealand
e-mail: nkhalid@aut.ac.nz

Q. Bai
e-mail: quan.bai@aut.ac.nz

Q. Bai et al. (eds.), *Multi-agent and Complex Systems*, Studies in Computational Intelligence 670, DOI 10.1007/978-981-10-2564-8_7

applications. Physical resources, such as memory, communication bandwidth and energy, can greatly limit the capability of sensor nodes and the performance of the whole WSN system [2]. A number of previous works have focused on these constraints for designing the communication and information processing elements for wireless sensor networks.

Communication process has been identified as highly resource consuming especially when the process is not well managed [3, 4]. There are two elements of communication for wireless sensor networks application: routing mechanism and media access control (MAC) protocol. In previous studies, several works for energy perspective issues have been briefly described on both elements. This involves reducing number of transmissions and distance of transmission via clustering or scheduling mechanisms at specific network layers.

For large scale networks, decentralized architectures are more appropriate as high transmission cost and delay might involve, especially if the central controller is located far away. Furthermore, in a centralized architecture, if the central node fails, then the entire network will collapse. On the other hand, decentralized control architecture are more reliable for large networks and can provide better collection of data and backup, in case of failure of the central node. However, decentralized approach is very challenging in terms of topology establishment and re-establishment especially in inaccessible applications such as battlefield and disaster management. These non replenish nodes need to self-organize themselves to sustain longer. This requires nodes in decentralized scheme to adapt accordingly with dynamic changes of environment (i.e., the network topology) and self-configure themselves without human intervention.

Shortest path algorithms based on either hop count or energy consumption are typically employed in routing protocols of ad hoc networks to achieve high energy efficiency [5]. However, relying on these parameters might cause hot spot scenario, i.e., sensor nodes that are frequently used might get depleted. In such case, a more reliable nodes should be selected. This leads to the need of having a more adaptive parameters to be considered during path selection. Furthermore, node preferences might be different, i.e., one node will choose a path which is nearer to it, which could be far to other nodes.

In this chapter, we focus on node selections for packet relaying (i.e., forwarder selection), and propose an adaptive forwarder selection approach for distributed WSNs. The proposed approach is called Distributed Adaptive Forwarder Selection (DAFS). In DAFS, suitable forwarders are selected in three phases, i.e., Eligibility Determination, Forwarder Selection and Receiver Acceptance. Multi-criterion parameters including energy, distance and buffer size, are considered in the approach. We claim that DAFS is an adaptive approach and suitable for dynamic environments, where nodes actions are determined based on their current capabilities and state of environment. In addition, it is suitable for largely distributed networks, where only decentralized approach is feasible.

The rest of the chapter is organized as follows. Some related work is introduced in Sect. 7.2. In Sect. 7.3, we describe the targeted problems and give some formal definitions. Section 7.4 introduces the interaction protocol applied in the proposed approach. Section 7.5 explains the forwarder selection processes in DAFS. The experimental results have been presented in Sect. 7.6. The chapter is finally concluded in Sect. 7.7.

7.2 Related Work

In this section, we briefly review related work on energy-efficient routing protocols and learning-based protocols. Most of the previous studies have shown that clustering approach can reduce the energy expenditure when merging all the information from the nodes into one cluster head which is responsible to process it and deliver it to the sink or base station. The limitation of sensor nodes usage for processing information has given a better energy consumption management which results to the sensor nodes lifetime to increase.

One of the well-known cluster-based protocols is Low Energy Adaptive Clustering Hierarchy (LEACH) [6]. In LEACH, data collection is perform periodically, which involves two phases, i.e., Cluster Head (CH) selection and cluster formation. The selection of CH in LEACH is based on closest distance. Each CH creates Time Division Multiple Access (TDMA) schedule for member nodes. CH also select code division multiple access (CDMA) to reduce inter-cluster interference. Members will collect information and use their allocated TDMA slots to transmit their collected data to CH. In [7], an energy efficient cluster formation algorithm (MOECS), was proposed based on a multi-criterion optimization technique. The selection of cluster heads is restricted to certain optimal value (i.e., optimal radius and distance from normal nodes).

Learning-based approach is commonly used in distributed systems [8]. Distributed Independent Reinforcement Learning (DIRL), is based on independent learning, i.e., each agent can autonomously and dynamically self-configure in order to maximize its own reward [9]. A reward-based dynamic approach based on two tier reinforcement learning scheme (micro learning and macro learning) were proposed in [10]. In their approach, an individual node was able to self-schedule its task using its local information and through learning [11].

Q-learning is a model-free RL technique, based on agents taking actions and receiving rewards from the environment in response to those actions [12]. In [13], Dimarogonas and Johansson proposed a combinatorial reverse auction that operates in two phases using RL and some economic models for energy optimization in sensor networks. The Q value was represented as an estimate cost of the route through neighbor comprised of hop count (account for energy efficiency) and minimum battery level among nodes. In [14], the potential of using energy aware metrics in RL based routing algorithms for WSN was studied, combining energy aware metrics with load balancing metrics. In [5], a machine-learning-based routing protocol for energy-efficient and lifetime-extended for UWSN, i.e., QELAR, was proposed. In QELAR, residual energy of each node and energy distribution among a group of nodes were used in it's lifetime-aware reward function, for calculating the Q-value (in selecting forwarder for packets). In [15], role-free clustering assignment was combined with learning dynamic network properties such as battery reserves. Less energy was consumed by using machine learning to enable nodes to independently decide whether or not to act as a cluster head on a per-packet basis in comparison to a traditional approach.

7.3 Problem Description and Definitions

In this chapter, we propose an adaptive, energy efficient and lifetime-aware forwarder selection approach, based on Q-learning technique. Using an action-value function (Q-value), which gives the expected reward of taking an action in a given state, the distributed learning agent is able to make a decision automatically. The proposed approach has the following features:

- Dynamic Network: In largely distributed network, link quality is not guaranteed. Link failure due to node's energy depletion, causes topology changes. Using Q-learning algorithm, selection of alternative link is possible as node selects next best forwarder based on current situations.
- Adaptive: We define our node role as forwarder, receiver and normal nodes. Node will decide on its role based on its present capabilities. It is adaptive to available resources i.e., a receiver can accept packets that they are capable to process, i.e., accept more when less busy. A node can be forwarder at a time but not at the other time if there is other more capable node to forward the packet etc.).
- General Framework: Q-learning behavior is determines by its reward function. We proposed a flexible and dynamic approach for the nodes to react based on its present capabilities.
- Load Balancing: Less energy is consumes when choosing a path based on shortest path. However, this may cause the link failure as choosing the same node to forward packets could drain its energy faster. We consider multiple parameters to allow alternative path selection.

7.3.1 *Definitions*

The network in our model is consider as a complex system comprising a number of adaptive sensor nodes, called agents.

Definition 1 A WSN is defines as connected undirected weighted graph $G = (V, E)$, where V is group in the network comprises of agents, i.e., $V = a_0, a_1, a_2, \ldots a_n$. $E = e_1, e_2, \ldots e_m$ is a set of edge in group. The edge, $e_k = (a_i, a_j)$ denotes the communication links between sensor a_i and sensor a_j (they are in each other's radio transmission range).

Definition 2 An agent a_i is defined as $a_i \in (R_i, Act_i, C_i, RWD_i)$. R_i is a_i's Role; Act_i is the action that a_i takes; RWD_i is the Reward that a_i gains (see Definitions 4); and $C_i = (EInit_i, ERes_i, ENeigh_i, Dist, BF_i, BFNeigh_i)$ is the capability of a_i, where $EInit_i$ is a_i's initial energy, $ERes_i$ is a_i's residual energy, $ENeigh_i$ is a_i's immediate neighbors' energy, Dist is distance between agents. BF_i is current buffer size of a_i, $BFNeigh_i$ is current buffer size of a_i's neighbors.

There are three types of Role, i.e., R in the proposed model, which are forwarder, receiver and normal node. In dynamic environment such as WSN, it is not practical to select a node as forwarder permanently, as it will cause the node to die faster. We propose a more distributed approach which allow flexibility in being forwarder, based on current capabilities. A forwarder can be a normal node at other time when its energy has degraded or it is currently processing many tasks.

Definition 3 An action set Act_i is defined as: $Act_i = (act_1, \dots act_i \dots act_n)$, where act_i is a possible action that a_i can perform. As a forwarder, an agent can take the following two actions:

- Forward packets received from one agent to another agent.
- Discard packets if no Forwarder is identified.

As a Receiver, there are three possible actions:

- Accept packets based on current capabilities.
- Reject packets if buffer is full (currently busy).

Definition 4 Reward function RWD, represents expected reward received by agents when transiting from one agent to another agent. The goal of our algorithm is to get the packet delivered from one agent to another agent, with maximum reward, i.e., minimum cost. The reward function is described in Eqs. 7.2–7.5.

7.4 System Framework and Interaction Protocol

There are three modules in the proposed approach, which are Eligibility Determination Module, Dynamic Forwarder Selection Module and Receiver Acceptance Module. Algorithm 1 shows the steps involved. When an agent has packet to transfer, they will compare among eligible neighbors, which one is the most capable (i.e., the one having the highest Q value). If no agent can accept the packet, after timeout, it will drop the packet. Among Eligible agents, once they receive packets, they will decides whether to accept or reject the packets. The amount of packets it will accept is depending on its current capabilities i.e., accept packets that they are able to process.

Algorithm 1: Forwarder Selection.

```
begin
    for each step of episode do
        Prior to any decision, individual agent will share its information ERes_i, Dist, BF_i
        and R_i with its neighbors;
        if current BF < maximum BF AND ERes > Minimum Energy
        then
            Decrease TIMEOUT ——(as it causes delay)
            Set Status as Eligible Forwarder
            Accept packet according to available BF
        ELSE
        if BF > maximum BF OR FULL OR ERes < Minimum Energy
        then
            Set Status as Not Eligible
            Choose Forwarder that give Q_max
```

7.4.1 Eligibility Determination

In the Eligibility Determination module, an agent decides whether to be a forwarder or not, based on its current capabilities (see Definition 2). The congestion or queue between receives or transmits will determine agent eligibility at the local level. A fully occupied buffer indicates agent is not capable to process any information at that particular time. The residual energy indicates eligibility at higher level, towards wider context i.e., network layer. Agent that decided to be Forwarder will inform it's neighbors about it's decision. In Dynamic Forwarder Selection, agent having more than one potential forwarder will select the best forwarder to forward packets based on Q value. Forwarder having the highest Q_{max} will be chosen. Q value is explained in Sect. 7.4.2. When forwarder receives packets from neighbors, it will process the received packets, according to its current capabilities.

7.4.2 Forwarder Selection

To assist agents to select suitable forwarders, we use Q-learning approach where using this approach, agent tends to select forwarder that gives maximum Q-value. In the proposed approach, both successful and failure transmissions contribute to the calculation of the Q-values. Furthermore, the approach not only concerns on selecting the best forwarders but also allows forwarders to negotiate as they wish, namely, a two directional selection. The expected reward that can be received by taking an action at time t and the state at time t is denotes in Eq. 7.1:

$$Q(s_t, Act_t) = RWD_{total} + \gamma \Sigma P^{Act_t}_{s_t, s_{t+1}} max Q(s_{t+1}, a) \quad (7.1)$$

In Eq. 7.1, $Q(s_t, Act_t)$ is the expected reward that an agent can receive by taking an action a_t at the state s_t. RWD_{total} is the total reward gained by the agent, which can be calculated by using Eq. 7.5. γ ($\gamma \in [0, 1]$) is the discount factor, which determines how important the future rewards are. When γ is set to 0, the system only considers the current reward and it acts similarly to a greedy algorithm. When γ is set to 1, the system will strive for a long-term high reward. The typical value of γ is within [0.5, 0.99]. Each forwarding action may succeed or fail. $P^{Act_t}_{s_t,s_{t+1}}$ is the success rate of taking action Act_t when s_t choosing s_{t+1} as the next forwarder. On the other hand, the failure rate is, $1 - P^{Act_t}_{s_t,s_{t+1}}$. $max Q(s_{t+1}, a)$ in Eq. 7.1 denotes the optimal value when taking an action, a. In this paper, we only consider the current reward and for such case, the second part of Eq. 7.1 is omitted.

As explained in Sect. 7.4.1, agent's capabilities are evaluated when determining Eligibility as forwarder. It is also used as input in reward functions, which is then applied in Q value calculation. We defined two reward functions, as in [5], comprises $RWD_{success}$ as in Eq. 7.2 and RWD_{fail} as in Eq. 7.4. If the packet forwarding attempt from a_i to a_j is successful, the reward function is shown in Eq. 7.2.

$$RWD_{success} = -g - \alpha(c(a_i) + c(a_j)) \tag{7.2}$$

In Eq. 7.2, g is the constant cost when a_i tries to forward a packet. As forwarding packet consumes energy and bandwidth, the farther an immediate node is from destination node, the more negative reward it would receive. Thus, agent will use a shorter path to reduce this cost. The weight of g is set to be 1. $c(a_i)$ and $c(a_j)$ are cost functions of residual energy of a_i and a_j respectively, which can be calculated by using Eq. 7.3 and α is the weight and is set to be 0.5. By definition, $c(a_i)$ is in the range of [0, 1], to balance the parameters in Eq. 7.2.

$$c(a_i) = 1 - ERes_i/EInit_i, \tag{7.3}$$

where $ERes_i$ is the residual energy of a_i and $EIniti$ is a_i's initial energy (refer to Definition 2).

On the other hand, if the forwarding attempt from a_i to a_j fails, the reward function is defined as the equation below.

$$RWD_{fail} = -g - \beta c(a_i), \tag{7.4}$$

where β is weight for the cost function that can be tuned. The value of β can be set to 0.5.

Based on Eqs. 7.2 and 7.4, the total reward gained by a_i (i.e., RWD_{total}) can be calculated by using Eq. 7.5.

$$RWD_{total} = RWD_{success} + RWD_{fail} \tag{7.5}$$

RWD_{total} is used in Q value calculation (Eq. 7.1) above. The far an agent from other agent is, the more energy is consumes for transmission. Thus, it will choose forwarder that is nearer to it.

7.4.3 Receiver Acceptance

In the Receiver Acceptance module, upon receiving a packet, agent will check its current processing task. It will accept packet according to its current capabilities, i.e., if it is currently processing certain task but still have available buffer, it will accept an amount of packets based on its remaining buffer.

7.5 The Distributed Adaptive Forwarder Selection (DAFS)

Many energy efficient and lifetime-aware approaches proposed solutions either at Physical layer, MAC layer, Network layer, Transport or Application layer. Even though such solutions can improve network performances in terms of network lifetime, energy efficiency, power consumptions etc., both analytical studies and experimental works in WSN highlight the important interactions between different layers of the network stack [3]. In this research, we consider multi-variables parameters involving Network layer, MAC layer and as well distance between nodes. In this section, we will elaborate on those parameters, which are used in our reward functions.

7.5.1 Multi-variables Parameters

In this research, agent capabilities are determined by energy, buffer size and distance. For most applications, a wireless sensor node is not replenish. Therefore, there is strong dependence on battery lifetime. Similar to traditional network layer, data transmission is linked to data communication area, which relates to certain layer; the link layer or MAC layer, Network layer (routing protocols) and transport layer (transport protocol).

7.5.1.1 Communication Energy

The main task of sensor node is to detect events, perform local processing and transmit the data. Power consumption can be divided into sensing, communication and data processing. In decentralized network, nodes may need to know its neighbors' latest state. However, in such network, continuous updates will require a lot of energy. We minimize such energy consumption by allowing only effected nodes to update and updates will only be sent if there is changes (i.e., if its energy is depleted an reaching a threshold value or if there is topology change, such as a new node joining the network). Hence, our concern is on communication energy as sensor node expends the maximum during this phase (transmitting and receiving data). The energy model in [7] is adopted where the amount of energy consumed for transmission, i.e., E_{TX}, of an ℓ-bit message over a distance d is given by:

$$E_{TX} = \ell \times EElect + \ell \times \varepsilon_{fs} \times d^2, \tag{7.6}$$

where ℓ is the length of message (4000 bits), $EElect$ is the base energy required to run the transmitter or receiver circuitry (50 nJ/bits) and ε_{fs} is the energy consumed in an amplifier (10 pJ/bit/m^2). The energy expended in receiving an ℓ-bit message, i.e., E_{RX} is given by:

$$E_{RX} = \ell \times EElect \tag{7.7}$$

7.5.1.2 Local Congestion Control—MAC Layer Solutions

The second issue considered is concerning local congestion, by limiting the traffic that an agent can relay. An agent may participate in the communication if it can relay the packet which is based on its communication activity. For this reason, buffer size is considered as another important factor in the proposed model, i.e., when packets arrive, they have to be processed and transmitted. If packets arrive faster than the agent can process them, the agent puts them into the buffer until it can get around to transmit them. The maximum queuing delay is proportional to buffer size. The longer the line of packets waiting to be transmitted, the longer the average waiting time is. The queue of packets waiting to be sent also introduces a potential cause of packet loss. Since the agent has a finite amount of buffer memory to hold the queue, an agent which receives packets at too high rate may experience a full queue where the agent has to simply discard excess packets.

7.5.1.3 Distance

In some cases, agents may be located far away from each other or from the Sink. Direct communication or peer-to-peer communication between nodes, especially in large distributed area is impossible, as it causes higher transmission cost and deplete faster. Thus, we consider distance as another important parameter. For example, if there are two Forwarders that is within agent's proximity, where forwarder A having more energy and less buffer, the agent might choose forwarder B, which has less energy and buffer compared to forwarder A but is nearer to it, taking into account, the significant energy consumption for longer distance communication.

7.6 Simulation Results

In this section, we evaluate DAFS by comparing it with two cluster-based approaches, i.e., LEACH and MOECS. The simulations were conducted using C++ platform. Two metrics were used to measure the performance of different protocols: first node death time and average residual energy. The first metric needs to be maximized, while second metric needs to be minimized. First node death time is the time when the battery of the first sensor node is depleted. Each sensor node has the goal of maximizing its own packet delivery to destination (that is to avoid packet loss by sending only to forwarder that is the most capable). Table 7.1 provides the common simulation parameters, which is also used in our experiments. Network lifetime is the most important performance metric for WSNs. Using this metric, DAFS, LEACH and MOECS protocols were evaluated.

The nodes in each simulation are distributed in a 100×100 m^2 region, where the location of nodes are selected randomly and that no two points have the same location. The Sink is given a fixed location. All the nodes are homogeneous and have the same capability.

Figure 7.1a shows the results of the first node death (round number) for two different network sizes. The first node death for network size 200 nodes, occurs at 710 rounds in LEACH, at 920 rounds in MOECS and at 3940 rounds in DAFS. While for network size 500 nodes, the first death round occurs at 730 rounds in LEACH, at 980 rounds in MOECS and at 3472 rounds in DAFS. This might be due to communication involves during clustering phase in LEACH and MOECS. In addition, as more criteria are considered in DAFS, i.e., including nodes buffer size allows nodes to choose other alternative forwarder.

In DAFS, multiple parameters that influence energy consumption were included. These parameters include communication cost from sensor node to the forwarder, communication cost from forwarder to the Sink, and the forwarder's residual energy, which help sensor nodes achieve balanced energy dissipation in the system.

Table 7.1 Simulation parameters

Number of nodes	100–500
Deployment area	100×100 m
Data packet size	500 bytes
Control packet size	25 bytes
$EElect$	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 J/bit/m^4
Initial energy for sensor node	0.5 J
Network topology	Random

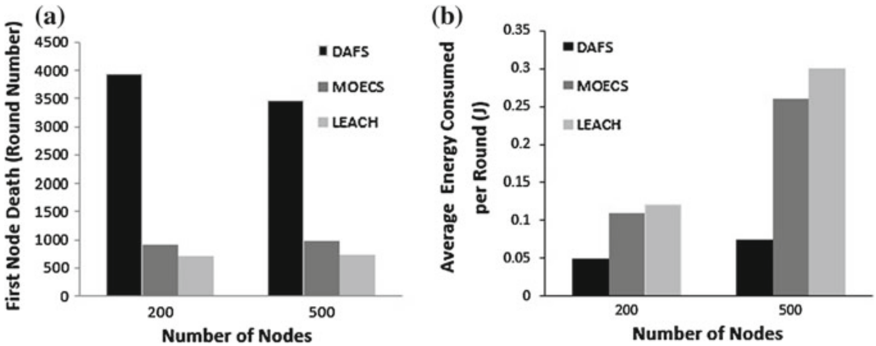

Fig. 7.1 **a** First node death in DAFS, LEACH and MOECS. **b** Average energy consumed per round in DAFS, LEACH and MOECS

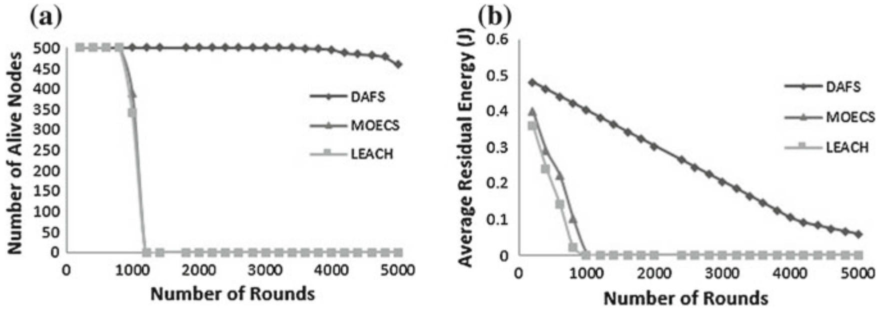

Fig. 7.2 **a** Number of alive nodes for 5000 rounds. **b** Average residual energy in DAFS, LEACH and MOECS

Figure 7.1b depicts the results for average energy consumed per round for two different network sizes using random topology which shows that our DAFS approach performs better than the other two. In addition to the balanced energy dissipation behaviors, such as distance, helps DAFS achieves minimum energy consumption compared to LEACH due to MOECS.

Figure 7.2a shows number of alive nodes in the network after 5000 rounds where nodes in DAFS survives much longer compared to the other two. Figure 7.2b illustrates results for the random topology where y-axis indicates the average residual energy and x-axis denotes the number of rounds. The residual energy of the system can also provides estimation of the network life. It can be observed that the mean residual energy of the system in the case of DAFS is higher than that of the other protocols. Hence, the network life under DAFS is enhanced compared to LEACH and MOECS. Unlike these cluster-based approaches (LEACH and MOECS), our approach did not involve cluster formation phases and is a distributed approach, as selection of forwarder is based on learning i.e., the Q_{max} value.

7.7 Conclusion and Future Work

As a resource constraint node, the use of sensor node in large scale network has some challenges in terms of energy efficiency and decentralized approach. These challenges can be overcome by ensuring energy is not use unnecessarily in transmission (multiple redundant packet, frequent use of same nodes etc.) Thus, the selection of relay node, i.e., forwarder, is crucial.

Decentralized architectures are more appropriate in many WSN applications. However, without the present of central controller, node needs to make its own decision based on limited information. In this paper, we consider multi-criteria parameters in forwarder selection and assists nodes decision by using a distributed learning-based approach. Our solution is adaptive as it is based on agent's current capabilities, that are changing dynamically when it gets depleted etc. With this technique, it is possible to consider multiple individual metrics for forwarder selection which is critical for well balanced energy dissipation of the system.

Simulation results demonstrate that DAFS achieves significant energy savings and enhances network lifetime when compared to LEACH and MOECS protocols. Multiple parameters involved in forwarder selection process for DAFS help to dissipate energy at a much more balanced rate as compared to other protocols and also it shows that the ability of DAFS to scale both from the network deployment area and node density which makes it a viable energy efficient schemes for WSNs.

References

1. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. Ad hoc Netw. **3**(3), 325–349 (2005)
2. Karl, H., Willig, A.: Protocols and Architectures for Wireless Sensor Networks. Wiley (2007)
3. Akyildiz, I., Vuran, M..C.: Wireless Sensor Networks. Wiley, New York (2010)
4. Anastasi, G., Conti, M., Di Francesco, M., Passarella, Andrea: Energy conservation in wireless sensor networks: a survey. Ad hoc Netw. **7**(3), 537–568 (2009)
5. Tiansi, H., Fei, Y.: Qelar: a machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks. IEEE Trans. Mob. Comput. **9**(6), 796–809 (2010)
6. Bsoul, M., Al-Khasawneh, A., Abdallah, A.E., Abdallah, E.E., Obeidat, I.: An energy-efficient threshold-based clustering protocol for wireless sensor networks. Wirel. Pers. Commun. 1–14 (2013)
7. Aslam, N., Phillips, W., Robertson, W., Sivakumar, S.: A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks. Inf. Fus. **12**(3), 202–212 (2011)
8. Badica, C., Scafes, M., Ilie, S., Badica, A., Muscar, A.: Dynamic negotiations in multi-agent systems. In: ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, p. 8 (2011)
9. Shah, K., Kumar, M.: Distributed independent reinforcement learning (dirl) approach to resource management in wireless sensor networks. In: IEEE International Conference on Mobile Adhoc and Sensor Systems, 2007. MASS 2007, pp. 1–9. IEEE (2007)
10. Shah, K., Di Francesco, M., Kumar, M.: Distributed resource management in wireless sensor networks using reinforcement learning. Wirel. Netw. 1–20 (2012)

11. Shah, K., Di Francesco, M., Anastasi, G., Kumar, M.: A framework for resource-aware data accumulation in sparse wireless sensor networks. Comput. Commun. **34**(17), 2094–2103 (2011)
12. Barto, A.G.: Reinforcement learning: An introduction. MIT Press (1998)
13. Dimarogonas, D.V., Johansson, K.H.: Event-triggered control for multi-agent systems. In: Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009, pp. 7131–7136. IEEE (2009)
14. Devillé, M., Le Borgne, Y.A., Nowé, A., De Causmaecker, P., Maervoet, J., Messelis, T., Verbeeck, K., Vermeulen, T.: Reinforcement learning for energy efficient routing in wireless sensor networks. In: Proceedings of the 23rd Benelux Conference on Artificial Intelligence, pp. 89–96 (2011)
15. Forster, A., Murphy, A.L.: Clique: role-free clustering with q-learning for wireless sensor networks. In: 29th IEEE International Conference on Distributed Computing Systems, 2009. ICDCS'09, pp. 441–449. IEEE (2009)