The Journal of Supercomputing
https://doi.org/10.1007/s11227-018-2514-3

Fog-based energy-efficient routing protocol for wireless sensor networks

Elham Mirzavand Borujeni[1] · Dadmehr Rahbari[1] · Mohsen Nickray[1]

Abstract

By exploiting the benefits of wireless sensor networks (WSNs), the Internet of Things (IoT) has caused many advances in the modern world. Since WSNs have limitations in energy usage, it is critical to save live nodes. Fog computing is a good solution to reduce the limitations of WSNs with its ability to meet the requirements of the IoT applications. Fog computing brings computing and storage resources closer to end users. P-SEP uses fog-based architecture to decrease energy consumption and increase network lifetime. To do so, in this paper, we introduce a new method based on P-SEP which uses FECR and FEAR algorithms in implementation. These algorithms improve the performance of fog-supported WSNs and prolong the lifetime of networks. The performance of the proposed approach is evaluated in comparison with P-SEP. The results of the simulation show that the average amount of energy usage in FECR protocol has been reduced by 9% and by 8% in FEAR. The number of live nodes saved in the network increased by 74% in FECR and 83% in FEAR in comparison with P-SEP protocol.

Keywords Wireless sensor network · Fog computing · Lifetime · Energy efficiency

1 Introduction

The Internet of Things (IoT) has brought a great evolution in the world of technology. Connecting many electronic devices, home and medical appliances, cameras, and

✉ Mohsen Nickray
m.nickray@qom.ac.ir

Elham Mirzavand Borujeni
e.mirzavand@stu.qom.ac.ir

Dadmehr Rahbari
d.rahbari@stu.qom.ac.ir

1 Department of Computer Engineering and Information Technology, University of Qom, Alghadir Ave., P.O. Box 3716146611, Qom, Iran

Published online: 04 August 2018

all types of sensors [1] to the Internet is the major goal of IoT. These devices are usually limited in computation power, battery, storage, and bandwidth. They outsource their computation by strong servers, mostly deployed in clouds. Cloud computing is considered a good solution to serve smart devices with elastic resources at low cost. However, it cannot support mobility, geo-distribution, location awareness, and latency for its end users [2]. Fog computing was proposed by Cisco to overcome limitations in cloud computing [3]. Fog will bring services closer to end users and on the edge of the network. Network devices equipped with additional storage and computational power can be used as fog servers in the network [4]. Wireless sensors, gateways, and routers are examples of fog servers keeping data and computation close to end users [3]. In fog computing architecture, fog nodes (FNs) function as middle-ware working between cloud and end users. They expand cloud and provide resources to the underlying sensors [5]. Some of the problems that can be solved by fog computing are listed below [6]:

– Latency: In WSNs, nodes are connected to the base station (BS). In many environments, low latency is an important factor. To do so, a large number of nodes located at the edge of the network (FNs) can be used. FNs are very close to the nodes in the network, so a slight communication delay through wireless links is possible.
– Delay jitter: In order to transfer large data with little delay, delay jitter, and reduction of packet loss, it is important to process data somewhere close to the network. As FNs can be located at the edge of the network, they are able to process large data and response more efficiently.
– Data security and integrity: More data are being transmitted in the network that is subject to attacks, even if it is encrypted. Fog computing can overcome this problem by providing the shortest possible distance.

Advantages associated with fog computing are as follows [2,4]:

1. Reduced network traffic: According to Cisco, the number of connected devices will be near 50 billion by 2020. These devices generate massive amount of data. Therefore, it can be helpful to provide computation ability closer to where devices are located. Through filtering, analyzing, and processing data, fog computing reduces the traffic being sent to the cloud.
2. Low latency requirement: Many applications like healthcare applications require real-time data processing. Since cloud data centers are geographically centralized, they often cannot provide real-time and low-latency communication for end users. Since FNs perform data processing very close to the end users, they are helpful solutions for real-time and latency-sensitive applications.
3. Scalability: Since the data generated by smart devices are too massive, sending all raw data to the cloud could make it a bottleneck. Fog servers with a wide and dense geographical distribution reduce the amount of data to be processed in the cloud and increase scalability in the network through providing the resources at the edge of the network and near the end user.

Healthcare, augmented reality, smart homes, smart grids, and smart vehicles are examples of applications that could benefit from fog computing [1,4,7]. With increasing technological facilities, people use higher number smart devices. The sensors used inside such devices allow them to know real-time information from their surroundings. With such definitions, IoT is associated with WSNs. The need to control the environment has led to the popularity of WSNs and made it possible to manage and monitor inaccessible environments [8]. With the popularity of WSNs rising every day, the technology is used in today's modern environments. WSNs can be used effectively in IoT, wherein all objects are connected. The capacity of today's WSNs can allow the execution of complex tasks such as data fusion [8]. Sensor nodes can be equipped with different components for different applications. An important disadvantage in WSNs is that they have limitations in power consumption, computation capability, and memory capacity [9]. Prolonging sensor network lifetime is also a major issue in WSNs protocols. There are many methods in order to reduce energy consumption and prolong the network lifetime in WSNs. One of them is hierarchical routing protocols. In this method, the network contains a set of clusters and every cluster has a cluster head (CH). In each cluster, CHs have the responsibility of sending and receiving data to and from the BS. In recent years, a combination of the notion of FNs with WSNs has improved energy efficiency in WSNs. Many protocols like [10–14] are presented to improve WSNs and prolong the lifetime. In these protocols, nodes send data to sink node and sink node is responsible to send data to the cloud. In these protocols, nodes send data to sink node and sink node is responsible to send data to the cloud. But in [15], the concept of fog computing is used in WSNs routing protocols. In this protocol, the CH selections are controlled in each round. This helps avoid selecting nodes with low energy as CH by applying the heterogeneity energy threshold. Optimization of the minimum distance between the CHs and FN, selection of some CHs to transmit data to FN, and decrease in the overhead on each FN could help prolong the lifetime of a network. In this paper, our fog-based energy-efficient routing protocol contains some FNs to cover some clusters and CHs sending/receiving data to and from FNs in the network. Our proposed network has more than one FN located on the edge of the network. After FNs receive data of CH, they filter and process received data and prepare data packets for sending to the cloud. They do this through a routing method. We represent two routing algorithms for this purpose. Therefore, optimized and efficiently implemented fog computing in the framework of WSN is a major concern in this study, in order to enhance potencies of smart devices and objects embedded within WSN. Our key contributions in this paper are as follows:

- Representing two algorithms, FECR and FEAR, for routing between FNs and cloud.
- Wide distribution of FNs as middle-ware between cloud and nodes at the edge of the network.
- Representing a new policy for selecting CHs, which nominate them according to their remaining energy.
- Sending data through CHs to the nearest FN to reduce their energy consumption and increase network lifetime.

The rest of this paper is organized as follows. In Sect. 2, related works are briefly reviewed. In Sect. 3, we introduced our fog-based routing algorithms in detail. In Sect. 4, the performance of our proposed algorithms is evaluated via simulation and a comparison between the results and the P-SEP protocol [15] is presented. Finally, in Sect. 5, the conclusion is discussed and suggestions are made for future work.

2 Related work

To date, many methods and algorithms have been proposed to optimize the performance of WSN with respect to lifetime, energy cost, latency, heterogeneity, scalability, support of different virtual networks, etc. However, they mainly employ basic concepts of LEACH and attempt to optimize it to some extent. Usually, the introduced protocols are evaluated through simulations while some authors have gone beyond simulations and examine their protocols outdoors. Here, some outstanding approaches as well as several relevant deficiencies and advantages are explained to provide a general picture of the development stream in order of their proposal dates. We divide these methods into hierarchical energy-aware protocols and ant colony optimization (ACO) routing protocol sections.

2.1 Hierarchical energy-aware protocols

LEACH is a post-traditional pioneering protocol for clustering of sensor networks [10,12], which includes randomized rotation of CH and a single CH in each cluster. The nodes send data packets to CHs, and the CHs fuse and forward them to the sink node. In comparison with the traditional algorithms, LEACH extends network lifetime, guarantees node communication with the CH that requires the lowest amount of transmit power, and aggregates data to reduce energy dissipation and latency in data transfer.

LEACH-Centralized (LEACH-C [12,16]) increases the energy efficiency to some extent by assuring production of a number of CHs during the setup phase. In this protocol, the BS is a central controller, which takes the responsibility of CH selection. Also, in 2008, LEACH-DCHS [17] was proposed to evaluate the probability of turning nodes to CHs by computing the residual energy level of each node. Another version of LEACH, namely PEGASIS [18], focuses on modifying the routing algorithm by permitting nodes to send or receive to and from close neighbors and become CH for transmitting data to the BS in turns, to achieve better energy-balancing and network lifetime. However, it inevitably makes some long chains among neighbor nodes while the CH rotation mechanism makes some nodes die earlier (specifically long distance transmissions cause nodes far from the sink nodes die early) and repeating resection of the chain heads at each round increases the communication overhead.

A protocol concerned with WSN heterogeneity is stable election protocol (SEP [13]). It extends the network's stability period. Its modified stable election protocol (M-SEP [11]) version classifies node types into advanced (higher initial energy) and normal (lower initial energy) nodes. The possibility of getting CH is higher for the advanced nodes. M-SEP prolongs network aliveness but it may freeze the system during some rounds. Its efficient modified version (EM-SEP [19]) can balance energy consumption and promote its efficiency. Due to the fact that advanced nodes are more likely to become CHs and from two equally probable sensors, the one with higher energy level will be selected, EM-SEP is limited to long-time maintenance of WSN aliveness and, unlike P-SEP [15], which prolongs the lifetime of fog-based sensor networks by balancing energy usage, it does not fluently decrease the energy of WSN.

Later, unequal cluster-based routing algorithm [20] was devised, in which sensor nodes were grouped with unequally sized components. Since the CH distance from the sink node increases, the cluster size will also increase. This algorithm concerns balanced energy cost and residual energies of sensor nodes. Furthermore, Karaboga et al. [21] introduced an ABC-based energy-efficient clustering mechanism (ICWAQ) in order to increase WSN lifetime through optimization of node clustering and defined CH routing gateways. ICWAQ maximizes network lifetime and employs a QoS mechanism by minimizing the delays between signals received from the clusters compared with LEACH, PEGASIS, and EEMSRA. In 2013, QoS-constrained jointly optimal congestion control, network coding, and adaptive distributed power control of multiple access interference wireless networks were focused to efficiently manage the available network resources when intra-session network coding is permitted [22].

Another important mission of LEACH optimization as (ALBA-R) is converge-cast in WSNs [23]. It is a cross-layer solution that combines awake/asleep schedules, MAC, routing, traffic load balancing, and back-to-back packet transmissions. Improvement in heterogeneous WSN energy is presented in EEMHR protocol [24]. In this approach, the nodes are divided into k levels of hierarchy and the total number of nodes are considered in order to compute the threshold of live CHs ratio in current round. Furthermore, [25] used topology control to present a multicast routing algorithm in order to save energy in the network. It focuses on reducing energy consumption and optimal cross-layer design by creating an end-to-end multicast routing used inside clusters.

TSRA [26] was introduced to enhance LEACH-based protocols through efficient determination of an optimal path. It uses the method of move and neighborhood search, which combines energy consumption and hop counts to make an efficient routing choice. It brings about a balanced transmission between nodes with low energy consumption and routing cost. Therefore, it increases the lifetime of WSN. In [27], authors attempted to improve QoS and energy efficiency. It makes ad hoc multiple paths in clusters. Cluster nodes involve a tolerable delay of data packets, and an adaptive routing protocol helps prolong WSN lifetime. In 2015, Kar and Misra [28] introduced the BRIDGE scheme to go over dynamic routing holes growing in numbers, especially in the case of using stationary WSNs, which causes temporarily misbehaving or trans-faulty nodes. The problem with this protocol is that its performance degrades as the hole area increases. Moreover, it cannot find and fix existing WSN deployment holes.

The scalability problem of traditional IoT architectures was addressed in mobile edge computing (edge IoT [29]), proposed to manage streams of data at the mobile edge. In this protocol, BSs are connected to FNs, which perform computing with local resources, and packet forwarding is facilitated through designing a software-defined networking (SDN) based on cellular core, on the top of the FNs. Furthermore, another IoT model, combining the benefits of SDN and fog computing, was presented in [30], to facilitate complex mechanisms implemented to control traffic and manage resources. In addition, this model enables analysis and management of some data at the network edge. Another issue in LEACH-based protocols is load balancing between multiple CHs at the same time that reduces the energy consumption of the inter-cluster routing. This issue is addressed via UCTCGT algorithm [9] for WSN, using unequal clustering and connected graph theory. Ref. [31] introduced multiobjective evolutionary algorithms to find the optimal lifetime and robustness of WSNs simultaneously. They emphasized that traffic distribution can optimize lifetime and obtain efficiency by finding the shortest paths and knitting edge disjoint paths to prune the search space. However, they neglected the influence of uncertainties caused by lack of information about link failure probabilities. In 2016, again an LEACH optimization protocol (H-LEACH [14]) was established to solve energy consideration problems during CH election. It applies some threshold condition in each round for the remaining and maximum energy of nodes, to select CHs and reduce the energy consumption of nodes during data transmission.

Optimizing gathered data and finding an appropriate route for mobile data collectors are presented in [32]. It divides the network into grid cells with the same size, provides a convenient construction of the spanning graph by using line sweep technique, and makes a complete graph according to the Warshall–Floyd algorithm. It presents a heuristic tour-planning algorithm on the basis of the complete graph. This method is successful in dispatching mobile data collectors and prolonging WSN lifetime. However, this work is limited to one mobile data collector and should be extended to many collectors.

Two protocols which focused on three-level heterogeneous WSNs are two-hop heterogeneity-aware centralized energy-efficient clustering (THCEEC) and advanced heterogeneity-aware CEEC (ACEEC) [33]. They were devised to enhance energy efficiency. They deal with the fluctuations in network deployments and adaptive transmission range of WSNs and provide longer network stability periods, promoted reliability of event reporting, reduced packet drop, avoided retransmission, and enabled economical energy consumption of the sensor nodes.

Alam and De [34] evaluated smoke WSNs outdoors by using three protocols of intrazone routing (IARP), interzone routing (IERP), and zone routing (ZRP). They stated that IERP protocol had great results in average energy consumption during sending and receiving data and also the number of packets dropping. In 2016, HFC [35], a hybrid fog and cloud interconnection framework for presenting a quick, effective, and automated management for the virtual network, was introduced. It uses an agent-based method, which permits different cloud services to interact with fog infrastructures. Scalability, flexibility, security, and L2 and L3 connectivity supports are the most important achievements of this protocol.

Finally, [36] attempted to solve the hot spot problem by adopting sink mobility through energy-efficient cluster-based dynamic routes adjustment (EECDRA), which tries to keep the cost of routes reconstruction down and maintain almost optimal routes to the latest location of the mobile sinks simultaneously. It achieves this goal by organizing the WSN into some equal clusters, choosing CHs within each cluster, and exploiting some communication-based rules to handle the process of routes reconstruction.

Table 1 represents a comparison of various protocols in WSNs. We have a short review on some protocols. Their important objective, benefits, disadvantages, and simulation environment are listed in this table.

Table 1 Comparison of various protocols in WSNs

Algorithm	Objectives	Environment	Features
LEACH [10]	Prolonging network lifetime	MATLAB	Reducing energy dissipation and latency
LEACH-C [12]	Developing and analyzing LEACH	Ns	Improving system lifetime, good for homogeneous WSNs
PEGASIS [18]	Improving Leach by greedy algorithm	Ns2	Distributing energy load among nodes, increasing lifetime, death of nodes due to long chains
SEP [13]	Prolonging network lifetime	MATLAB	Considering heterogeneity for nodes
P-SEP [15]	Prolonging lifetime of fog-based networks	MATLAB	Uniform nodes distribution, Prolonging the time interval of the system
ICWAQ [21]	Maximizing network lifetime	MATLAB	Minimizing the delays between received signals from the clusters
TSRA [26]	Improving LEACH by determining an optimal path	MATLAB	Decreasing energy consumption and routing costs
ALBA-R [23]	Optimizing LEACH	Ns2	Improving packet delivery ratio, decreasing end-to-end latency
SHE [27]	Energy efficiency and QoS	OMNet++	Prolonging network lifetime, not considering heterogeneous nodes
BRIDGE [28]	Efficient delivery of data to sink	MATLAB	Bypassing dynamic routing holes, cannot identify and bypass the available WSN deployment holes
UCCGRA [9]	Prolonging network lifetime	OMNET++	Balancing the energy consumption among nodes, Relieving the influence of energy-hole problem
H-LEACH [14]	Solving energy problems	MATLAB	Solving node energy issues, Which is a major disadvantage of the LEACH protocol
EECDRA [36]	Prolonging network lifetime	MATLAB	Using mobile sinks for reducing reconstruction of the route

2.2 ACO-based routing protocol

Routing by ACO helps find the best path from source to destination nodes in WSNs, classified by energy level, transmission distance, lifetime, quality of service, flat, hierarchical, pheromone, and heuristic routing protocols [37]. These methods were evaluated by location awareness, load balancing, computation complexity, and energy efficiency. The security, survival ability, and QoS awareness [37,38] are the critical issues for routing problem in WSNs. The optimization of energy consumption in WSNs by ACO is done in [39–41]. Also, the authors in [42] used ACO for WSNs routing. They optimized cost, adaptability, multipath transmission, energy consumption, and network lifetime parameters.

The authors in [38] proposed a QoS-aware routing protocol for heterogeneous wireless sensor networks by ACO (EAQHSeN). In this work, the entire traffic is classified into control traffic and data traffic as well as QoS constraints. They optimized packet delivery fraction, end-to-end delay, routing overhead, and minimum remaining energy compared to AODV and EEABR protocols. The researchers in [43] proposed an algorithm based on distance and direction of node communication and residual energy. They optimized the energy consumption and the lifetime of WSNs by improvement of pheromone technique in ACO and searching range in the network in comparison with LEACH protocol.

3 The proposed algorithm

In this section, we explain the proposed method in the fog-based WSN model. In WSNs, extending the lifetime of network and reducing energy consumption are two vital factors, and our method is introduced to achieve these goals.

3.1 The architecture of WSN fog-based model

In this section, we explain our fog-based WSNs. In Fig. 1, our fog-based model is shown, where the gray circles are advanced nodes (nodes with more energy than normal nodes), white circles are normal nodes, and CHs are determined. The existence of advanced nodes makes the network heterogeneous, which leads to the energy efficiency of the network. Because wireless sensor networks are widely used in IoT applications, they must meet IoT requirements. Since traditional networks have faced many challenges in serving IoT applications, fog computing was introduced as a practical solution. By processing data locally, these nodes avoid sending raw data to the cloud. Our network has some FNs with higher level of energy than other nodes (e.g., Wi-Fi). These nodes are wireless, they are not connected to the resource of energy, and they are fixed with predetermined locations in the network edge. We presume that the network contains normal and advanced nodes with two heterogeneous levels of energy. Based on the P-SEP protocol, each FN covers an area, called FN coverage (FNC) [15]. In each FNC:

– There are some normal nodes and advanced nodes.

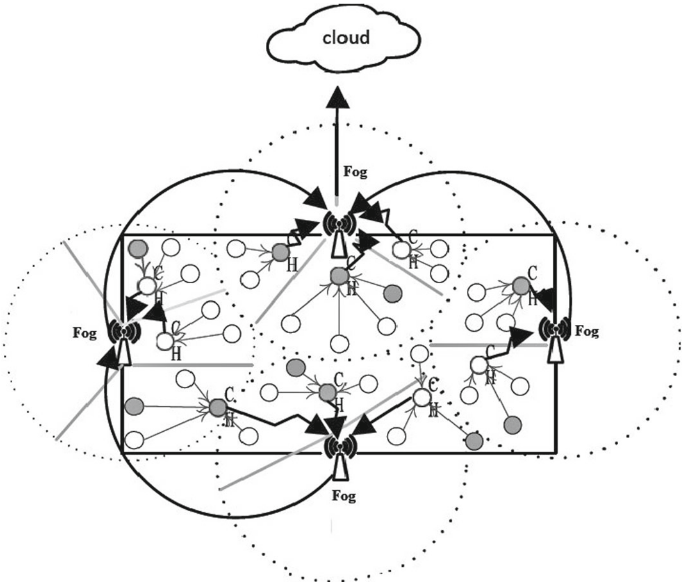

Fig. 1 Fog-based model and architecture of the relation system

- There are some CHs.
- More than one cluster can exist.

In the network, each node has a sensing area responsible for collecting data of the environment. Nodes send their gathered data to CHs. Both normal and advanced nodes can be chosen as CH. These CHs are responsible for receiving, gathering, and fusing data packets from nodes. CHs send gathered data to FNs. Here, our FNs send data to the cloud. FNs collect data from CHs and then fuse the received data and transmit it to the nearest FN in their neighborhood. Beside FNC, an assumed network has a remote cloud for gathering data from the FNs.

3.2 Nodes distribution and their energy model

The network size is assumed to be M, and it is a rectangle. Also, r interprets the current round (cycle), r_{max} is the total round, and n is the total number of nodes (both normal and advanced nodes). In the network, while the locations of the normal nodes are given randomly, the locations of advanced nodes are determined in advance. The energy of advanced nodes is greater than all normal nodes in the network. The fraction of advanced nodes, whose energy is α times more than the energy of other normal nodes, is considered to be m. The total number of advanced nodes is $n \times m$ (the total number of normal nodes will be: $n - n \times m$). If the initial energy of normal nodes is E_f, then the energy of advanced nodes will be $E_f \times (1 + \alpha)$ [15]. The total energy of network will be $(n - n \times m) \times E_f + n \times m \times E_f \times (1 + \alpha) = n \times E_f \times (1 + \alpha \times m)$ which means the network has more energy. We consider our network with some FNs, which are wireless, use battery as a resource of power, have no energy limitation and predetermined locations. The communication of each FN is limited to its adjacent nodes in each round. FEAR and FECR algorithms are used in order to transmit information between FNs and send the combined data to the cloud. CHs send their gathered data to FNs, and the data move among FNs and become fused. The optimal probability of a node to become a CH is p_{opt} [13], which is different for normal and advanced nodes. The clusters are re-established in each round [10], and one epoch is equal to $\frac{1}{p_{opt}}$ rounds [13]. We consider p_{NA} as the weighted probability for selecting normal nodes and p_{AD} as the weighted probability for selecting the advanced nodes as CH in the rth round. Due to minimized energy consumption in each round within an epoch, the average number of CHs in each round in each epoch should be constant and equal to $n \times p$ [13] where p is the probability of the nodes to become CH in rth round, determined as follows:

$$p_{NA} = \frac{p}{1 + \alpha \times m} \times \frac{E_c}{E_{max}}, \tag{1}$$

$$p_{AD} = \frac{p \times (1 + \alpha)}{1 + \alpha \times m} \times \frac{E_c}{E_{max}}, \tag{2}$$

where E_c is the current energy of nodes and E_{max} is the maximum energy of nodes (equal to E_f in the first round). To calculate the probability of the nodes becoming CH, the fraction $\frac{E_c}{E_{max}}$ is used, which gives us the amount of remaining energy of a node in each round [14]. It can help choose nodes according to their residual energy. Each node has a radio circuit, which consumes a certain amount of energy to send an L-bit message over a distance d (where d is the distance from CH to FN, to obtain an admissible signal-to-noise ratio (SNR)), named radio energy dissipation. According to radio model in [15], the free-space channel model is exploited for the communication distance lower than d_0. The multipath model is used for communication distance larger than d_0. The transmission energy will be:

$$E_{TX}(l, d) = L \times E_{elec} + e_{amp} \times D, \tag{3}$$

where D is the coefficient of power loss. According to free-space model, D is equal to d^2 while in the multipath fading model D will be d^4, e_{amp} is the amplification factor (e_{fs} or e_{mp}), which depends on the model of transmitter amplifier, and d is the distance between the sender and receiver. Therefore, E_{TX} is calculated as:

$$E_{TX}(l, d) = \begin{cases} L \times E_{elec} + L \times e_{fs} \times d^2 & \text{if } d \le d_0, \\ L \times E_{elec} + L \times e_{mp} \times d^4 & \text{if } d > d_0, \end{cases} \tag{4}$$

where $E_{elec} = E_{TX} + E_{DA}$ [15]. E_{DA} is considered as the data aggregation energy for each node. The consumed energy by the receiver will be $E_{RX} = L \times E_{elec}$ [15]. The energy of CH after the rth round will be updated as $E_i(r) = E_i(r - 1) - E_{TX}$; E_i is the energy of ith node.

3.3 Calculating the threshold

$T_{NR}(r)$ is used to express the threshold of normal nodes, and $T_{AD}(r)$ is used to represent the threshold of advanced nodes. So we have [15]:

$$T_{NR}(r) = \begin{cases} \frac{p_{NR}(r)}{1-p_{NR}\times(r \bmod \frac{1}{p_{NR}(r)})}, & \text{if } NR \in G' \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The threshold applied to normal nodes is $T_{NR}(r)$, r is the current round, G' is the set of nodes that are not chosen as CHs in the last $\frac{1}{p_{NR}}$ rounds of the epoch. Similarly for the advanced nodes, we have [15]:

$$T_{AD}(r) = \begin{cases} \frac{p_{AD}(r)}{1-p_{AD}\times(r \bmod \frac{1}{p_{AD}(r)})}, & \text{if } AD \in G'' \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$T_{AD}(r)$ is the threshold applied to advanced nodes; G'' is the set of nodes that are not chosen as CHs in the last $\frac{1}{p_{AD}}$ rounds of the epoch. To send the data of nodes to CHs and CHs to FNs, we need to calculate the distance between them [15]. We assume d_{CH} as the distance between the nodes and CHs, and d_{FN} as the distance between CHs and FNs. In Algorithm 1, steps of the proposed method are shown. In order to select normal and advanced nodes as CH, threshold values of T_{NR} and T_{AD} are used. $d_{to}CHs$ is used for calculating the distance between nodes in the network, and CHs and $d_{to}FNs$ are used to calculate the distance between CHs and FNs. $E_c(n_i)$ is the current energy of ith node.

3.4 Fogs to cloud

Our network contains more than one FN allocated at the edge of the network. Each FN covers some CHs. CHs select the nearest adjacent FN for sending data. FNs receive data packets from CHs, and then fuse and send them to the cloud by a routing algorithm. The two algorithms FECR and FEAR are used for routing between FNs and sending data to the cloud. The goal of these algorithms is to prolong the network lifetime and reduce the energy consumption of all nodes in the network.

3.5 Pegasis-based routing of FNs

The communication of each FN is limited to its adjacent nodes in each round, and there is just one node that will be chosen to send the integrated data to the cloud. The PEGASIS [18] algorithm is used in order to transmit information between FNs and send the combined data to the cloud. The chain of nodes formed by PEGASIS algorithm enables the nodes to communicate with their neighbors within that chain. CHs send their gathered data to FNs, and these data move between FNs and become fused. Eventually, an FN will be chosen to transmit data to the cloud. Algorithm 2 shows the

Algorithm 1 Fog-based Energy-efficient Routing

```
1: Build the random sensor network as nodes.
2: Determine the node positions for fog and advanced nodes.
3: Determine parameters of Algorithm 2 (or Algorithm 3) by Table 3 (or Table 4).
4: Create the model based on the FN positions.
5: for r = 1 to r_max do
6:    CH_number ← 0
7:    Calculate p_NR, p_AD, E_max, E_c Eqs. (1) and (2).
8:    Calculate T_NR(r), T_AD(r) Eqs. (5) and (6).
9:    if Trand_NR ≤ T_NR & E_c(n_i) > 0 & (G'(i) > 0) then
10:       Selection of normal nodes by probability as CH.
11:       CH_number = CH_number ∪ {n_i}, where n_i here is normal node.
12:    end if
13:    if Trand_AD ≤ T_AD & E_c(n_i) > 0 & (G''(i) > 0) then
14:       Selection of advanced nodes by probability as CH.
15:       CH_number = CH_number ∪ {n_i}, where n_i here is advanced node.
16:    end if
17:    Calculate distance d_to CHs between nodes and CHs.
18:    Calculate distance d_to FNs between CHs and FNs.
19:    Send gathered data to FNs by minimum distance.
20:    Update the node's energy by using Eqs. (3) and (4).
21:    Send packets from FNs to cloud by Algorithm 2 (or Algorithm 3).
22: end for
```

Algorithm 2 Pegasis-based Routing of Fog Nodes (FECR)

```
Input: FNs.
Output: Best path of FNs to cloud.
1: Calculate distance of FNs to cloud as in Eq. (7).
2: N_1 = furthest FN from cloud as leader Eq. (8).
3: Add N_1 to chain.
4: Add the closest not visited neighbor of N_1 to chain Eq.(9).
5: Calculate D_chain(FNs) & E_chain(FN).
6: Select best chain by Min(D_chain).
7: Transfer packet from leader FN of chain to cloud.
8: Repeat steps 1 to 12 for all FNs.
```

steps of the application of PEGASIS, where D_{chain} (FNs) is the distance(chain), and E_{chain}(FN) is energy(chain) [18]. Figure 2 shows an example of the chain between FNs and their communications. The chain was formed by the farthest node from the cloud to make sure that nodes farther from the cloud have a close node in their neighborhood as in Eqs. (7) and (8). In Eq. (8), F is the set of FNs in the network. As shown in this figure, nodes connect to their nearest neighbors as in Eq. (9), and one node will be selected as the leader responsible for sending gathered data to the cloud. Each node receives data from adjacent nodes, fuses them with its own data, and transmits them to another node in the chain. Finally, the leader gathers the data and transmits them to the cloud. To select the leader, the distance of FNs to the cloud and their energy should be mentioned. Forming a chain among FNs and selecting one as the leader to transmit data to the cloud (instead of each FN sending its own data to the cloud) could help save more energy in the network. Also, CHs can transmit data to the closest FN in their neighborhood (instead of sending data to the cloud, which may be far from of

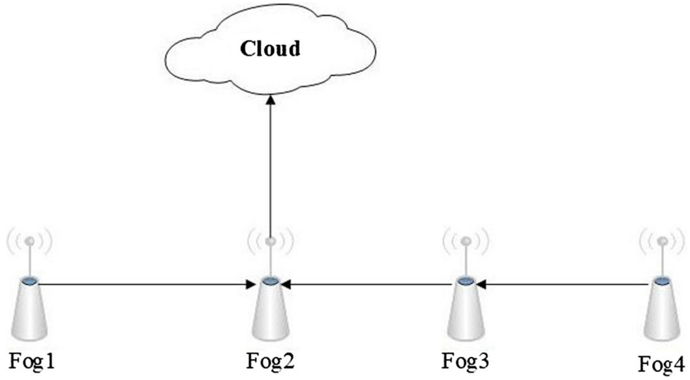

Fig. 2 FNs and their communication through a chain

them), leading to prolonged network lifetime.

$$d_{Cloud} = \sqrt{(x_{FN} - x_{Cloud})^2 + (y_{FN} - y_{Cloud})^2} \tag{7}$$

$$Node_{tail} = \max_{f_i \in F}(d_{Cloud}(f_i)), \qquad \forall f_i \in F \tag{8}$$

$$Node_{distance}(f_i) = \sqrt{(x_{f(i)} - x_{f(j)})^2 + (y_{f(i)} - y_{f(j)})^2}, \quad \forall f_j \in \{F - f(i)\} \tag{9}$$

3.6 ACO-based routing of FNs

Another algorithm used for routing between FNs and cloud is ACO. With a given list of nodes and their pairwise distances (d_{ij}),), the goal is to find the shortest possible path [44].

Algorithm 3 ACO-based Routing of FNs (FEAR)

Input: FNs.
Output: Best path of FNs to cloud.

```
Calculate distance of FNs to cloud.
for i = 1 to LastAnt do
    Calculate d_ij
    Repeat until ant(i) has completed a tour.
    Select the FN j to be visited next.
    Calculate the energy of each FN.
end for
Replacing higher-performance solutions.
Update the amount of pheromones.
The best solution as optimal allocation is selected as the output.
Transfer packet from FNs to cloud.
```

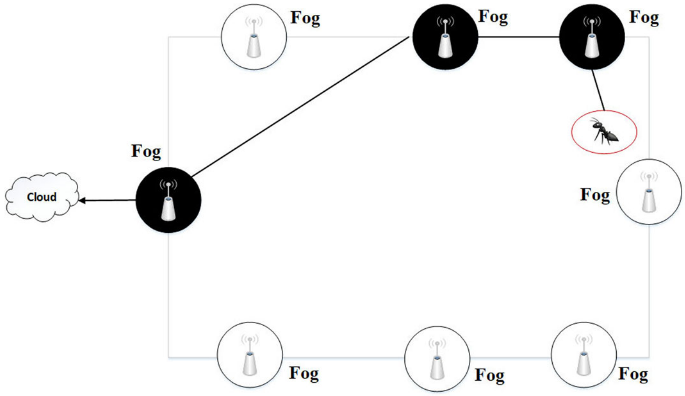

Fig. 3 FNs routing through ACO

ACO was first proposed by Dorigo Maniezzo [42]. The algorithm is based on the behavior of ants seeking a path between their colony and a source of food [45]. At first, the ants wander randomly. As shown in Fig. 3, when an ant finds a source of food, it walks back to the colony leaving pheromones to show the path has food. This helps other ants follow this path with a certain probability. As more ants find the path, it becomes stronger as Eq. (10) and is followed by ants with greater probability.

$$p_{xy}^{k} = \frac{\tau_{xy}^{\alpha}\eta_{xy}^{\beta}}{\sum_{k\in y}\tau_{xk}^{\alpha}\eta_{xk}^{\beta}} \tag{10}$$

In Eq. (10), p_{xy}^{k} is the probability of moving from x to y for ant_k, τ_{xy} is the amount of pheromone in transition from x to y, α is the control parameter for τ_{xy}, η_{xk} is the desirability of transition from x to y, β is the control parameter for η_{xy}, also τ_{xk} and η_{xk} are the attractiveness of other possible transition.

$$\tau_{xy} = (1-\rho)\tau_{xy} + \sum_{k} \Delta\tau_{xy}^{k} \tag{11}$$

In Eq. (11), ρ is the pheromone evaporation coefficient, $\Delta\tau_{xy}^{k}$ is the pheromone of ant_k with zero when ant_k does not use xy transition, and $\frac{T}{Cost_k}$ so that T is a constant value and $Cost_k$ is the tour cost of ant_k.

4 Experiment result

The performance of our proposed methods is presented in this section. We simulate FECR and FEAR and compare them by different parameters. Also, their results are compared with P-SEP [15].

Table 2 Parameters of FECR and FEAR protocols [15]

Parameter	Value
M	100^2, 500^2
n	100, 500
m	0.1, 0.2, 0.3
k	4000 (bit)
p	0.1
E_f	0.5 (Joule)
E_{DA}	5 (nJoule)/(bit)
E_{elec}	50 (nJoule)/(bit)
E_{fs}	10 (pJoule)/(bit/m^2)
E_{mp}	0.0013 (pJoule)/(bit/m^4)

Table 3 Parameters of FECR and FEAR protocols

Parameter	Value
n	5
E_f	25*0.5 (Joule)
ETX_{fog}	125 (nJoule)
ERX_{fog}	125 (nJoule)
EDA_{fog}	5 (nJoule)/(bit)
E_{elec}	125 (nJoule)/(bit)
Efs_{fog}	10 (pJoule)/(bit/m^2)
Emp_{fog}	0.0013 (pJoule)/(bit/m^4)

4.1 Simulation setup

The proposed algorithm has been evaluated in the MATLAB platform. In Table 2, network parameters used in our protocol are listed. The network contains normal nodes, advanced nodes, and some fixed-position FNs, $\{m = 0.1, m = 0.2, m = 0.3\}$ show the fraction of advanced nodes where $m = 0.1$ means the ratio of advanced to total nodes is 10%. The performance of algorithms FECR and FEAR in networks with 500 and 1000 nodes for different values of α and $m = 0.3, 0.2, 0.1$ was evaluated. In all simulations, the cloud is located outside the network and at a relatively far distance. To compare the performance of the algorithms with P-SEP, the number of live nodes and the average remaining energy of nodes were evaluated in network sizes of 500, 1000, 2000, and 5000 for different values of $m = 0.2, 0.3$ and $\alpha = 3$. Also, the average number of packets sent to the fog and CHs, the average energy consumption by nodes in the network, and the number of CHs for the FECR, FEAR, and P-SEP were evaluated.

Table 3 shows fog parameters in the proposed algorithm. The number of FNs is considered to be 5, and we assume each FN can cover 100 nodes. According to what was said above, the energy of FNs is much higher than that of other nodes in the network. The energy of FN is considered to be $25 \times E_f$. The sensing range of each

Table 4 Parameters of FEAR protocol

Parameter	Value
$nAnt$	40
Q	1
α	1
β	1
ρ	0.05

node (d_0) is 5.28 (m) [15] in our algorithm, and the position of the cloud is fixed while its location is far away.

In Table 4, the ACO parameters are listed. The number of ants is considered to be $nAnt$ while Q is a constant, α is pheromone exponential weight, $beta$ is heuristic exponential weight, and rho is evaporation rate.

4.2 Network lifetime and energy

To evaluate network lifetime and energy, we tested FECR and FEAR for a number of live nodes and the average energy remaining in nodes, under different values of m and α in different network sizes.

In Fig. 4a, c, the number of live nodes for the FECR and FEAR algorithms is shown. The network is evaluated at $\alpha = 3$ and various values of m (0.1, 0.2, 0.3) during 4500 rounds. As demonstrated, both algorithms have the largest numbers of live nodes at m=0.3. By increasing the percentage of advanced nodes ($m = 0.1$ means that a number of advanced nodes are 10% of total nodes) in fixed network size, the number of live nodes and energy of nodes increased, too.

As the number of advanced nodes in the network increases with the increase in m, the result is more distribution of these nodes in the network, which allows the clusters to be able to send in longer periods.

Figure 4b, d shows the amount of energy remaining on nodes in the network size $M = 500^2$, $\alpha = 0.3$, and various m at $M = 500^2$ through 4500 rounds. In the FEAR algorithm, with the increase in m, more energy is stored in the network. The FECR algorithm shows better results for the first 2500 rounds. The increase in the number of advanced nodes in the network has caused clusters to have more advanced nodes, and subsequently, these nodes become candidates for the CHs in higher rounds.

Figure 5a, c shows the number of live nodes in FECR and FEAR for different values of α during 4500 rounds at $M = 500^2$. In these evaluations, the value of m is fixed and equal to 0.3. The value of α indicates the amount of energy is higher in an advanced node than in a normal node. As α increases, each advanced node will have more energy. The greater the energy of advanced nodes, the higher the lifetime of these nodes and the network. Both FECR and FEAR algorithms for $\alpha = 3$ have more live nodes at the end of 4500 rounds. Figure 5b, d also shows the amount of energy remaining in the nodes after 4500 rounds. FEAR algorithm for $\alpha = 3$ retains more energy in the network.

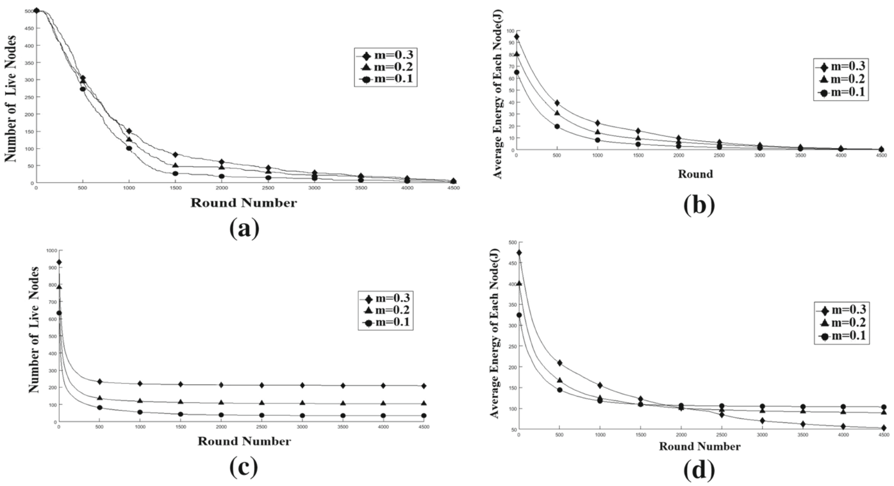

Fig. 4 Number of live nodes and average energy remaining in nodes for FECR and FEAR under different $m = \{0.3, 0.2, 0.1\}$ and $\alpha = 3$ through $r = 4500$. **a** FEAR lives: $(n, M) = (500, 500^2)$. **b** FEAR energy: $(n, M) = (500, 500^2)$. **c** FECR lives: $(n, M) = (500, 500^2)$. **d** FECR energy: $(n, M) = (500, 500^2)$

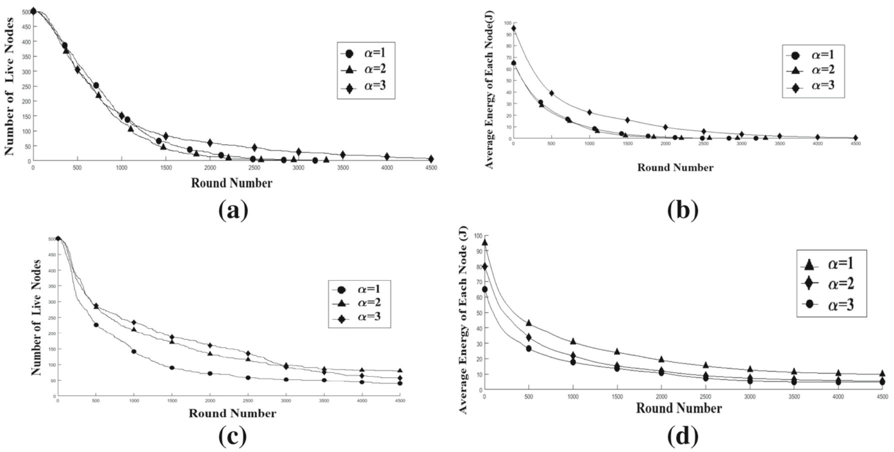

Fig. 5 Number of live nodes and average energy remaining in nodes for FECR and FEAR under different $\alpha = \{3, 2, 1\}$ and $m = 0.3$ through $r = 4500$. **a** FEAR lives: $(n, M) = (500, 500^2)$. **b** FEAR energy: $(n, M) = (500, 500^2)$. **c** FECR lives: $(n, M) = (500, 500^2)$. **d** FECR energy: $(n, M) = (500, 500^2)$

The number of live nodes as well as the amount of energy remaining in the nodes for a 1000×1000 network size is shown in Fig. 6a–d. These evaluations are obtained for different values of m and $\alpha = 3$ over 4500 rounds. In both algorithms, for $m = 0.3$ the network maintains a greater number of live nodes. Also, both algorithms save more energy in the network for $m = 0.3$.

Figure 7a, c shows the number of live nodes, and Fig. 7b, d shows the average remaining energy in the nodes in the network for different values of α with $m = 0.3$ over 4500 rounds. The number of nodes in the network is considered 1000. In FEAR

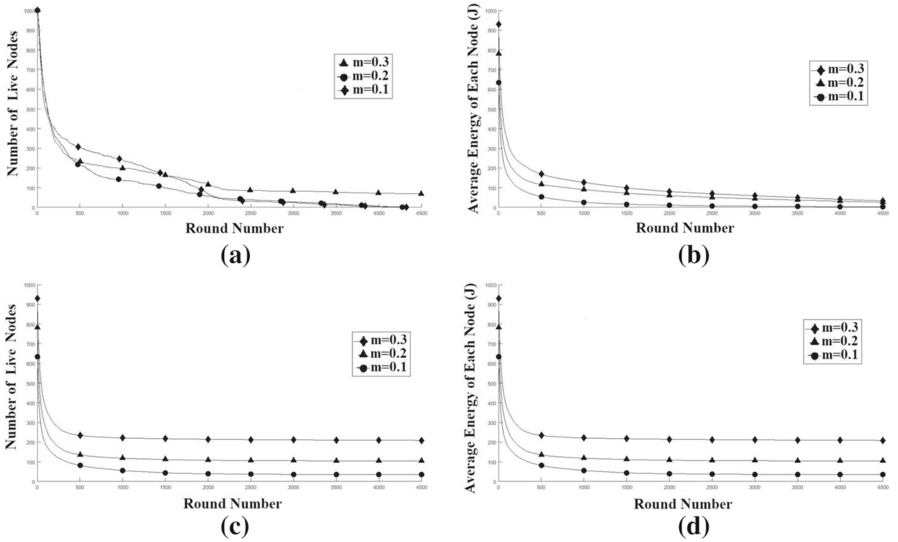

Fig. 6 Number of live nodes and average energy remaining in nodes for FECR and FEAR under different $m = \{0.3, 0.2, 0.1\}$ and $\alpha = 3$ through $r = 4500$. **a** FEAR lives: $(n, M) = (1000, 1000^2)$. **b** FEAR energy: $(n, M) = (1000, 1000^2)$. **c** FECR lives: $(n, M) = (1000, 1000^2)$. **d** FECR energy: $(n, M) = (1000, 1000^2)$

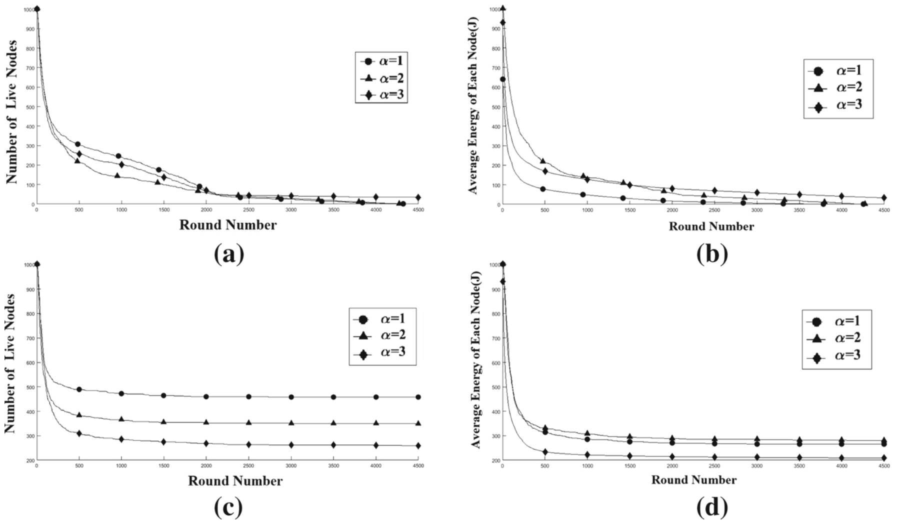

Fig. 7 Number of live nodes and average energy remaining in nodes for FECR and FEAR under different $\alpha = \{3, 2, 1\}$ and $m = 0.3$ through $r = 4500$. **a** FEAR lives: $(n, M) = (1000, 1000^2)$. **b** FEAR energy: $(n, M) = (1000, 1000^2)$. **c** FECR lives: $(n, M) = (1000, 1000^2)$. **d** FECR energy: $(n, M) = (1000, 1000^2)$

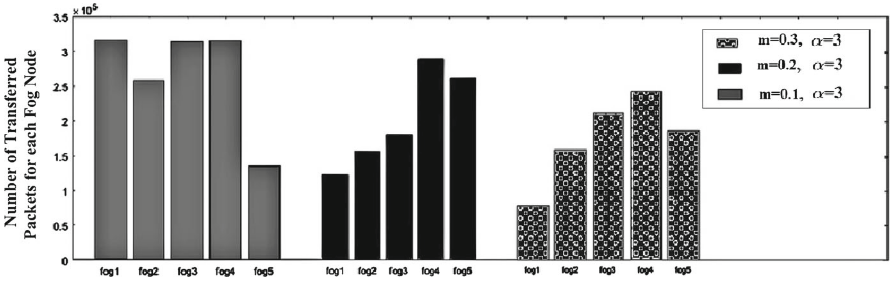

Fig. 8 Number of transferred packets to FNs in FECR under the network size = 500 at $\alpha = 3$ and various m and round number 4500

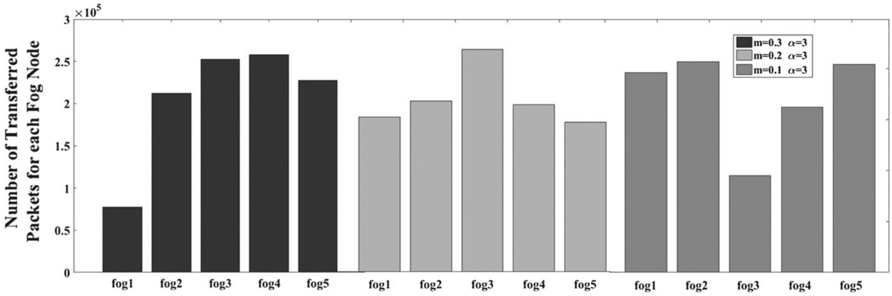

Fig. 9 Number of transferred packets to FNs in FEAR under the network size = 500 at $\alpha = 3$ and various m and round number 4500

algorithm for $\alpha = 3$, we have the largest number of live nodes in the network, while in the FECR algorithm, the maximum number of live nodes was gained for $\alpha = 1$. In addition, FEAR algorithm for $\alpha = 3$ is able to store more energy in the network, while the FECR algorithm for this α value has not been able to obtain a better result.

Based on results for live nodes and the amount of energy remaining in nodes in the networks with the size of 500 and 1000, it can be concluded that the values of m and m and α should be selected according to network size. This means the number of advanced nodes must be selected depending on the network size. Moreover, this number should be neither very small nor very big. On the one hand, choosing a small value of m means a smaller number of advanced nodes. The smaller number of advanced nodes leads to a higher probability of nominating normal nodes for CHs. To become CHs, normal nodes should consume much more energy. As a result, they will be depleted of energy sooner. As the time passes, we will have fewer CHs, which means a shorter lifetime of the network.

Figures 8 and 9 report the number of transferred packets to FNs for network size $M = 500^2$ during 4500 rounds with $\alpha = 3$ and various values of m. Due to factors such as their distance from the FNs, the CHs select the appropriate FNs to send data. As m increases, clusters are able to send data in higher rounds. Therefore, the FNs that cover these clusters receive data from them for a longer period of time. In lower values of m however, as the population of advanced nodes is reduced, the clusters will have lower CHs candidates, so some clusters that have been sending data to the adjacent

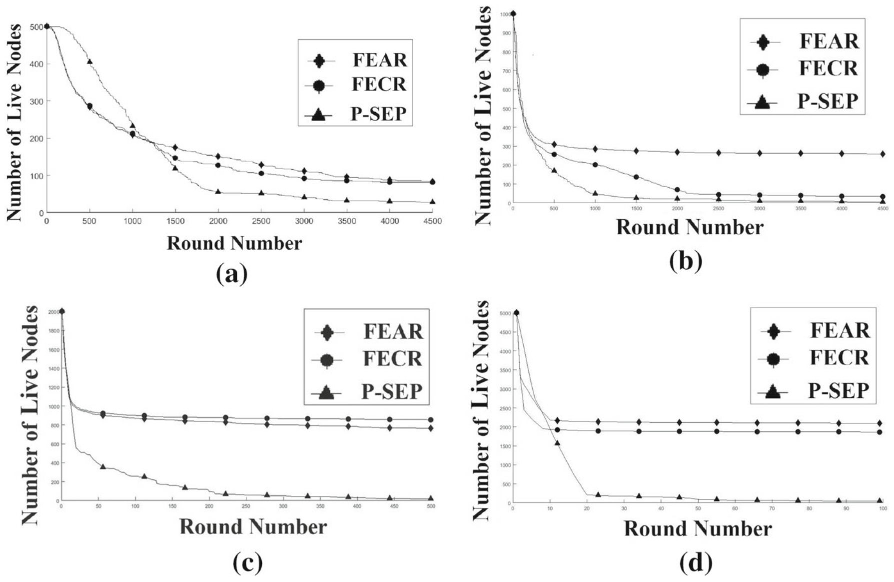

Fig. 10 Number of live nodes for FECR, FECR(ACO), P-SEP [15] for $m = 0.3$ and $\alpha = 3$. **a** Lives: $(n, M) = (500, 500^2)$. **b** Lives: $(n, M) = (1000, 1000^2)$. **c** Lives: $(n, M) = (2000, 2000^2)$. **d** Lives: $(n, M) = (5000, 5000^2)$

nodes in $m = 0.3$ are unable to send the data in higher rounds. Thus, packets received by FNs will vary with different α values. The wide distribution of FNs at the network edge allows CHs to send data to the nearest neighboring FN. These data are processed or stored in FNs. Responses are also sent at shorter distances and reach CHs in less time.

4.3 Comparison of protocol performance

In this paper, the proposed algorithm was simulated in networks under the size of 500, 100, 2000, and 5000 for different values of $m = \{0.3, 0.2\}$; then, it was compared with the P-SEP protocol [15]. In the simulation with $M = 2000 \times 2000$ round numbers considered to be 500 and for $M = 5000 \times 5000$, the total round is 100.

Figure 10a–d reports the number of live nodes for FECR, FEAR, and P-SEP at $\alpha = 3$ and $m = 0.3$ under the network size = 500, 100, 2000, 5000. The number of live nodes in all algorithms is decreased by an increase in the round number, but FECR and FEAR can save more live nodes in the network after the last round. It means FECR and FEAR have been able to keep more live nodes in the network by dividing the data transmission load between FNs. This is very important for IoT applications to sense the environment for a longer time. CHs send their data to the nearest FN, which is able to process and store data locally. As clear from the results, the proposed algorithms for 5000 nodes in the network have been able to achieve a better result than P-SEP and can sense data of the environment for longer time.

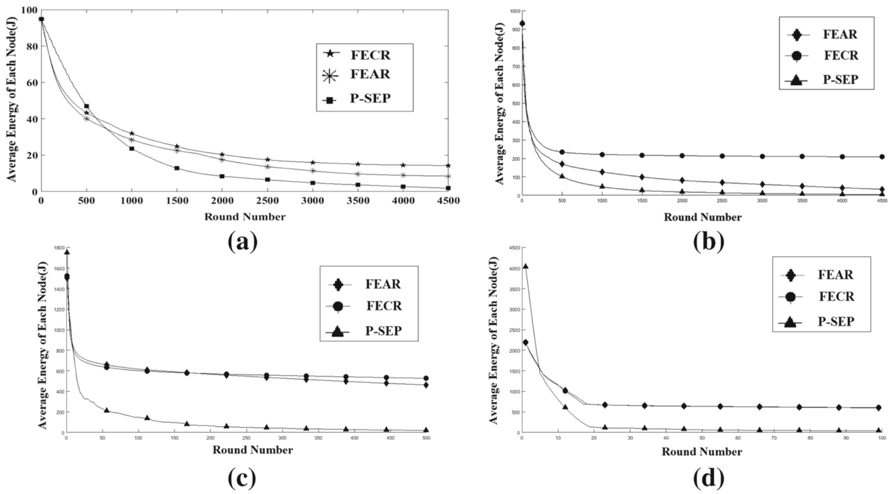

Fig. 11 Remaining energy in FECR, FECR(ACO), P-SEP [15] for $m = 0.3$ and $\alpha = 3$. **a** Energy: $(n, M) = (500, 00^2)$. **b** Energy: $(n, M) = (1000, 1000^2)$. **c** Energy: $(n, M) = (2000, 2000^2)$. **d** Energy: $(n, M) = (5000, 5000^2)$

In Fig. 11a–d, the average energy of nodes with $\alpha = 3$ and $m = 0.3$ for network sizes $M = 500^2, 1000^2, 2000^2, 5000^2$ is presented. As the round number increases, the energy of nodes decreases. FECR and FEAR led to better results in keeping more energy in the network, because of FNs which gather data from CHs with each CH able to send data to the closest FN in its neighborhood. Proposed algorithms generally have a lower energy consumption in comparison with P-SEP. Therefore, in FECR and FEAR, nodes lose their energy at lower rates and it can save more energy in the network.

In Figs. 12 and 13, the number of remaining live nodes, the average remaining energy of nodes for networks with the sizes of 500, 1000, 2000, and 5000, and the values of $m = 0.2$ and $\alpha = 3$ were evaluated. In these evaluations, the two proposed algorithms show better results compared to algorithm P-SEP. Therefore, it can be concluded that the proposed method has better results for different network sizes and is scalable. In total, with the increase of the rounds in all three methods, the number of nodes and the remaining energy in the nodes are reduced by sending data, but the two proposed methods have been able to achieve better results in comparison with P-SEP. In these two methods, the selection of CHs based on the amount of energy remaining in them as well as data transmission from CHs to the nearest adjacent node has resulted in an increase of network lifetime and reduced energy consumption compared to P-SEP.

In Fig. 14, the number of CHs is shown in three algorithms FECR, FEAR, and P-SEP. This estimate is obtained for networks with 500 and 1000 nodes, $\alpha = 3$ and $m = 0.2$ for 4500 rounds. As shown in this figure, with the network size increasing, the number of CHs also increased. In the two proposed algorithms, the number of CHs was higher than in the P-SEP method. The number of CHs for the FEAR method in higher rounds was more than that of the P-SEP method. In Fig. 15, the number of CHs for networks with the size of 500 and 1000 for $\alpha = 3$ and $m = 0.3$ over 4500 rounds

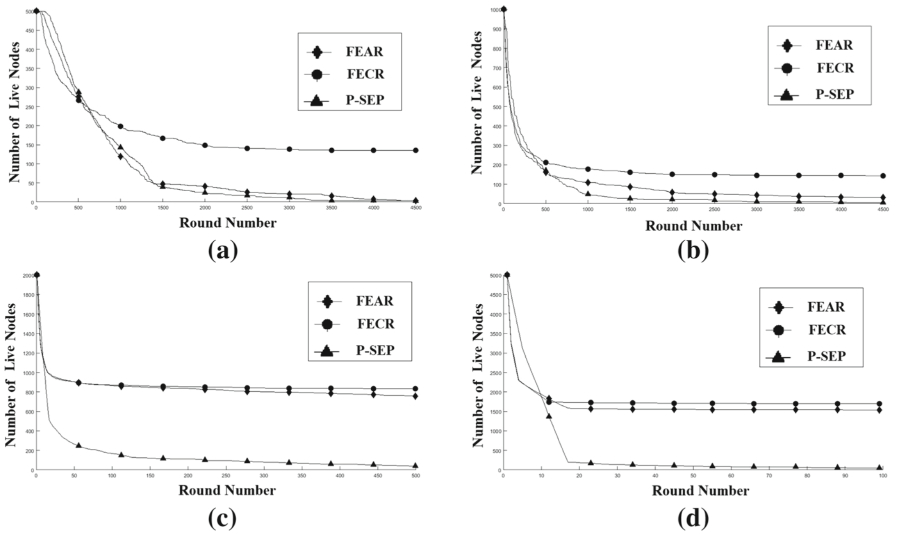

Fig. 12 Number of live nodes for FECR, FECR(ACO), P-SEP [15] For $m = 0.2$ and $\alpha = 3$. **a** Lives: $(n, M) = (500, 500^2)$. **b** Lives: $(n, M) = (1000, 1000^2)$. **c** Lives: $(n, M) = (2000, 2000^2)$. **d** Lives: $(n, M) = (5000, 5000^2)$

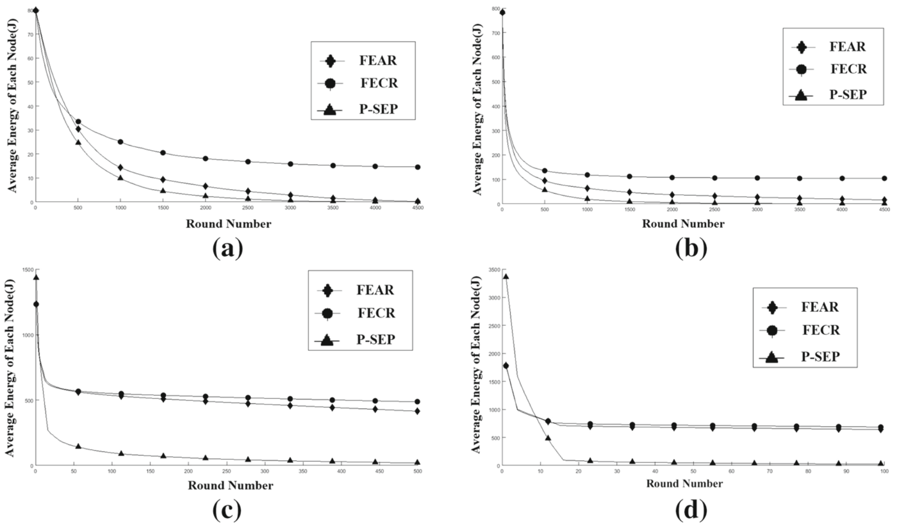

Fig. 13 Remaining energy in FECR, FECR(ACO), P-SEP [15] for $m = 0.2$ and $\alpha = 3$. **a** Energy: $(n, M) = (50{,}000^2)$. **b** Energy: $(n, M) = (1000, 1000^2)$. **c** Energy: $(n, M) = (2000, 2000^2)$. **d** Energy: $(n, M) = (5000, 5000^2)$

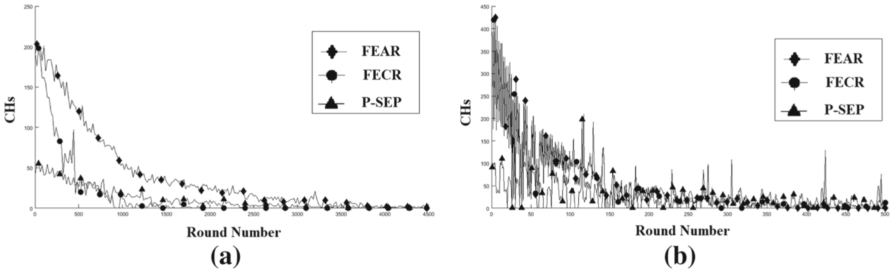

Fig. 14 Number of CHs in FECR, FEAR, P-SEP [15] for $m = 0.2$ and $\alpha = 3$. **a** CHs: $(n, M) = (500, 500^2)$. **b** CHs: $(n, M) = (1000, 1000^2)$

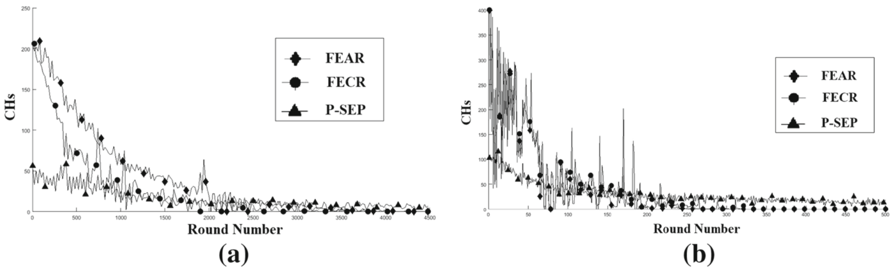

Fig. 15 Number of CHs in FECR, FEAR, P-SEP [15] for $m = 0.3$ and $\alpha = 3$. **a** CHs: $(n, M) = (500, 500^2)$. **b** CHs: $(n, M) = (1000, 1000^2)$

has been achieved. With increasing CHs, they are well distributed. This means that by increasing the value of m in the network, the proposed method has been able to send data to the CHs in the higher rounds.

Figure 16 and 17 show the average energy remaining and the average energy consumption of nodes under the network sizes 100×100 and 500×500 at $\alpha = 3$ and $m = 0.3$ for proposed algorithms and P-SEP. The nodes in the network have more energy after 4500 rounds compared to P-SEP. With the less energy consumption in FECR and FEAR, we can conclude that FECR and FEAR are more energy-efficient algorithms with longer stability period and they can prolong the lifetime of the network more than P-SEP. Less energy consumption has made the proposed network more sustainable and more energy efficient. This suggests that the proposed method also works best in small-sized networks. The average amount of energy usage in FECR and FEAR algorithms is reduced by 9 and 8%, respectively, while the number of live network nodes is increased by 74 and 83%, respectively, as compared to P-SEP.

Figure 18 shows the average number of packets transmitted to CHs and FNs under the network size $M = 500^2$, $\alpha = 0.3$, and $m = 0.3$ in the proposed algorithms and P-SEP protocols. The round number is considered 80% of total round numbers (4500), and as a result, all algorithms have enough live nodes for transferring data in the network during these rounds. According to Fig. 18, the average number of transferred packets to CHs in the proposed algorithms is lower than P-SEP, which means CHs in our method are less involved in the process of fusion and computations are done in FNs locally.

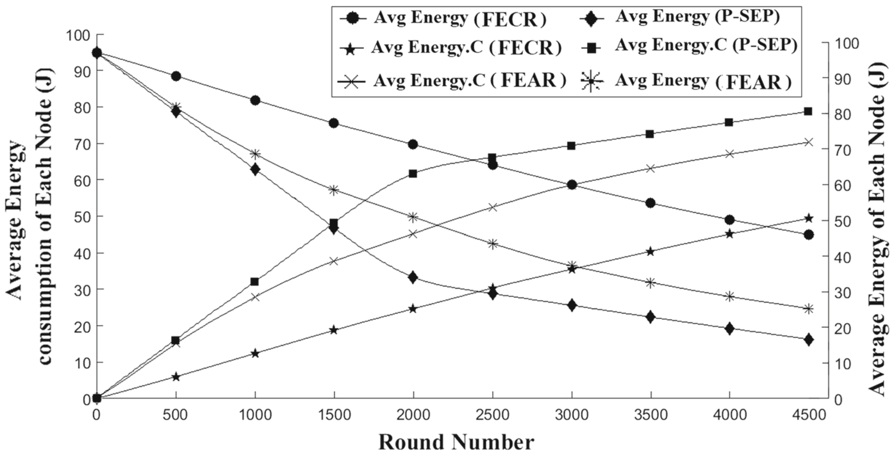

Fig. 16 Comparison between average energy consumption and average energy of nodes in P-SEP, FEAR, and FECR under the Network Size = 100 with $\alpha = 3$, $m = 0.3$ at Round Number 4500

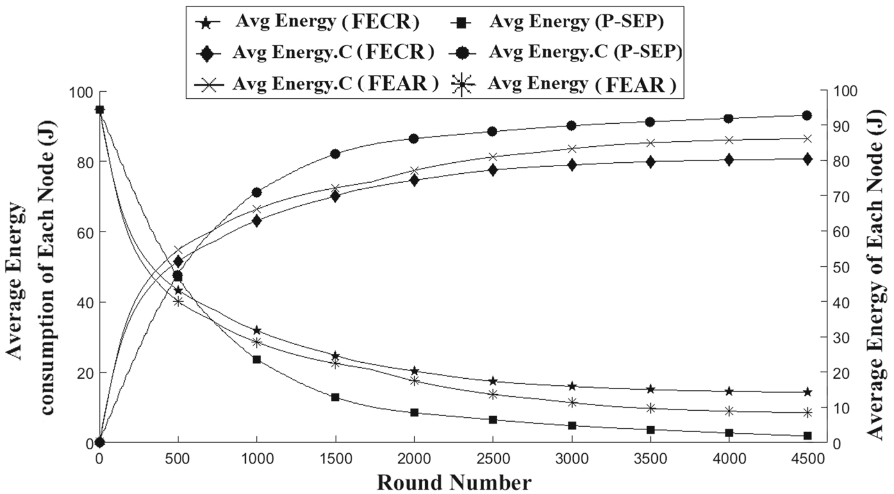

Fig. 17 Comparison between average energy consumption and average energy of nodes in P-SEP and FECR under the network size = 500 with $\alpha = 3$, $m = 0.3$ at Round Number 4500

Table 5 reports the first node dies (FND), some node die where 20% of nodes die (SND), half node die (HND), and the number of dead nodes at the last round (DN). The numeric results show that FND, 20%ND, and HND for P-SEP are better than FECR and FEAR, but at the end, the number of dead nodes in P-SEP is more than FECR and FEAR. It means that the proposed algorithms attempt to save more energy in the network by adapting themselves to prolong lifetime in the network. FECR and FEAR select appropriate CHs and FNs to transfer data. The proposed algorithms have more live nodes at the end of 4500 rounds in comparison with P-SEP.

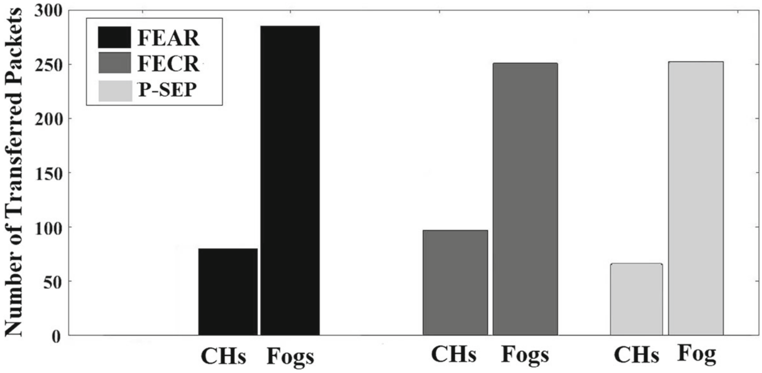

Fig. 18 Comparison between number of transferred packets to FNs in P-SEP, FEAR and FECR under the network size = 500 at $\alpha = 3$ and $m = 0.3$ through 4500 Round

Table 5 Comparison of FECR, FEAR, and P-SEP under network size 500 with r = 4500, $\alpha = 3$; FND = first node die, SND = 20% node die, HND = half node die; DN = number of dead nodes in the last round (r = 4500)

Method	FND	SND	HND	DN
FECR ($\alpha = 3$, $m = 0.3$)	43	194	732	395
FECR ($\alpha = 3$, $m = 0.2$)	33	145	548	369
FECR ($\alpha = 3$, $m = 0.1$)	21	169	486	343
FEAR ($\alpha = 3$, $m = 0.3$)	19	203	775	420
FEAR ($\alpha = 3$, $m = 0.2$)	23	102	319	423
FEAR ($\alpha = 3$, $m = 0.1$)	19	115	318	455
P-SEP ($\alpha = 3$, $m = 0.3$)	192	499	981	473
P-SEP ($\alpha = 3$, $m = 0.2$)	101	304	616	499
P-SEP ($\alpha = 3$, $m = 0.1$)	21	185	382	499

5 Conclusion

FECR and FEAR are proposed as fog-based algorithms for WSNs. These algorithms represent a method to find optimal CHs in the network depending on the energy remaining in nodes. FNs send and receive data between cloud and WSNs. FECR and FEAR are presented as two routing algorithms for sending data to the cloud. FECR algorithm uses the PEGASIS algorithm to form the chain among the nodes and send data to the cloud. In FEAR algorithm, routing is performed based on ACO. According to the experimental results, the proposed algorithms save much higher energy in comparison with P-SEP protocol. In these proposed algorithms, CHs send gathered data from other nodes of their cluster to the nearest adjacent FN. Then, FNs fuse the data collected from CHs and send them to the cloud by one of these algorithms. By adding FNs in the network, it is not necessary to send all data to the cloud for processing. Only data that cannot be processed in FNs are sent, so the volume of computation in the CHs and also the cloud will be less. Adding FNs will allow processing and storage to be done locally, with less delay in end-user requests. As a result, the proposed architecture for IoT and latency-sensitive applications is more efficient. Also, the presence of different levels of energy in the proposed network causes the network to be heterogeneous and helps maintain energy in the network. As shown in this paper, our proposed method improves the energy efficiency and prolongs the lifetime of the network due to the higher energy of FNs. FNs have less energy limitation, and as a result, the network can send more data at higher number of rounds. The position of FNs is assumed fixed on the network edge. Moreover, they are wireless and use batteries as their energy resources. Fog computing reduces the latency, network traffic, and energy consumption. But since the FNs are composed of micro-data centers and each node has a balanced processing load, optimal resource allocation for them is one of the challenges of this method. In addition, data in the network are vulnerable and can be attacked by attackers. There are many solutions to maintain data security. Security topics and the overhead of implementing them in the network require further research. Moreover, a discussion of data fusion in fog-based WSNs and optimal resource allocation can be investigated by researchers in the future.

References

1. Dastjerdi AV, Buyya R (2016) Fog computing: helping the internet of things realize its potential. Computer 49(8):112–116
2. Mahmud R, Kotagiri R, Buyya R (2018) Fog computing: a taxonomy, survey and future directions. In: Internet of everything. Springer, pp 103–130
3. Ivanov S, Balasubramaniam S, Botvich D, Akan OB (2016) Gravity gradient routing for information delivery in fog wireless sensor networks. Ad Hoc Netw 46:61–74
4. Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: principles, architectures, and applications. In: Internet of Things. Elsevier, pp 61–75
5. Aazam M, St-Hilaire M, Lung C-H, Lambadaris I, Huh E-N (2018) Iot resource estimation challenges and modeling in fog. In: Fog Computing in the Internet of Things. Springer, pp 17–31
6. Firdhous M, Ghazali O, Hassan S (2014) Fog computing: will it be the future of cloud computing. In: The 3rd International Conference on Informatics and Applications (ICIA2014), pp 8–15
7. Yi S, Hao Z, Qin Z, Li Q (2015) Fog computing: platform and applications. In 3rd IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), pp 73–78
8. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (iot): a vision, architectural elements, and future directions. Fut Gener Comput Syst 29(7):1645–1660
9. Xia H, Zhang R-H, Yu J, Pan Z-K (2016) Energy-efficient routing algorithm based on unequal clustering and connected graph in wireless sensor networks. Int J Wirel Inf Netw 23(2):141–150
10. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd IEEE Annual Hawaii International Conference on System Sciences, p 10
11. Singh D, Panda CK (2015) Performance analysis of modified stable election protocol in heterogeneous WSN. In: IEEE International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), pp 1–5
12. Heinzelman WB, Chandrakasan AP, Balakrishnan H (2002) An application-specific protocol architecture for wireless microsensor networks. IEEE Trans Wirel Commun 1(4):660–670
13. Smaragdakis G, Matta I, Bestavros A (2004) Sep: a stable election protocol for clustered heterogeneous wireless sensor networks. Technical report, Boston University Computer Science Department
14. Razaque A, Mudigulam S, Gavini K, Amsaad F, Abdulgader M, Krishna GS (2016) H-leach: hybrid-low energy adaptive clustering hierarchy for wireless sensor networks. In: IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp 1–4
15. Naranjo PGV, Shojafar M, Mostafaei H, Pooranian Z, Baccarelli E (2017) P-sep: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks. J Supercomput 73(2):733–755

16. Heinzelman WB (2000) Application-specific protocol architectures for wireless networks. PhD thesis, Massachusetts Institute of Technology
17. Liu Y, Gao J, Jia Y, Zhu L (2008) A cluster maintenance algorithm based on leach-dchs protocol. In: IEEE International Conference on Networking, Architecture, and Storage, NAS'08. pp 165–166
18. Lindsey S, Raghavendra CS (2002) Pegasis: power-efficient gathering in sensor information systems. IEEE Aerosp Conf Proc 3:3–3
19. Malluh AA, Elleithy KM, Qawaqneh Z, Mstafa RJ, Alanazi A (2014) Em-sep: an efficient modified stable election protocol. In: Zone 1 Conference of the American Society for Engineering Education (ASEE Zone 1), 2014. IEEE, pp 1–7
20. Wang J, Yang X, Ma T, Wu M, Kim J-U (2012) An energy-efficient competitive clustering algorithm for wireless sensor networks using mobile sink. Int J Grid Distrib Comput 5(4):79–92
21. Karaboga D, Okdem S, Ozturk C (2012) Cluster based wireless sensor network routing using artificial bee colony algorithm. Wirel Netw 18(7):847–860
22. Baccarelli E, Cordeschi N, Polli V (2013) Optimal self-adaptive qos resource management in interference-affected multicast wireless networks. IEEE/ACM Trans Netw (TON) 21(6):1750–1759
23. Petrioli C, Nati M, Casari P, Zorzi M, Basagni S (2014) Alba-r: Load-balancing geographic routing around connectivity holes in wireless sensor networks. IEEE Trans Parall Distrib Syst 25(3):529–539
24. Tanwar S, Kumar N, Niu J-W (2014) Eemhr: Energy-efficient multilevel heterogeneous routing protocol for wireless sensor networks. Int J Commun Syst 27(9):1289–1318
25. Jiang D, Xu Z, Li W, Chen Z (2017) Topology control-based collaborative multicast routing algorithm with minimum energy consumption. Int J Commun Syst 30(1):e2905
26. Orojloo H, Haghighat AT (2016) A tabu search based routing algorithm for wireless sensor networks. Wirel Netw 22(5):1711–1724
27. Chen D-R (2016) An energy-efficient qos routing for wireless sensor networks using self-stabilizing algorithm. Ad Hoc Netw 37:240–255
28. Kar P, Misra S (2017) Detouring dynamic routing holes in stationary wireless sensor networks in the presence of temporarily misbehaving nodes. Int J Commun Syst 30(4):e3009
29. Sun X, Ansari N (2016) Edgeiot: mobile edge computing for the internet of things. IEEE Commun Mag 54(12):22–29
30. Tomovic S, Yoshigoe K, Maljevic I, Radusinovic I (2017) Software-defined fog network architecture for IOT. Wirel Pers Commun 92(1):181–196
31. Rahat AA, Everson RM, Fieldsend JE (2016) Evolutionary multi-path routing for network lifetime and robustness in wireless sensor networks. Ad Hoc Netw. 52:130–145
32. Xie G, Ota K, Dong M, Pan F, Liu A (2017) Energy-efficient routing for mobile data collectors in wireless sensor networks with obstacles. Peer-to-Peer Netw Appl 10(3):472–483
33. Aslam M, Munir EU, Rafique MM, Hu X (2016) Adaptive energy-efficient clustering path planning routing protocols for heterogeneous wireless sensor networks. Sust Comput: Inf Syst 12:57–71
34. Alam S, De D (2017) Cloud smoke sensing using iarp, ierp and zrp routing protocols for wireless senor network. CSI Trans ICT 5(1):119–124
35. Moreno-Vozmediano R, Montero RS, Huedo E, Llorente IM (2017) Cross-site virtual network in cloud and fog computing. IEEE Cloud Comput 4(2):46–53
36. Wang J, Cao J, Ji S, Park JH (2017) Energy-efficient cluster-based dynamic routes adjustment approach for wireless sensor networks with mobile sinks. J Supercomput 73(7):3277–3290
37. Liu X (2017) Routing protocols based on ant colony optimization in wireless sensor networks: a survey. IEEE Access
38. Malik SK, Dave M, Dhurandher SK, Woungang I, Barolli L (2017) An ant-based qos-aware routing protocol for heterogeneous wireless sensor networks. Soft Comput 21(21):6225–6236
39. Sharma S, Kushwah RS (2017) ACO based wireless sensor network routing for energy saving. In: IEEE International Conference on Inventive Communication and Computational Technologies (ICICCT), pp 150–154
40. Kannan M, Chinnappan S, Krishnamoorthy C (2017) Ant star fuzzy routing for industrial wireless sensor network. In: Third IEEE International Conference on Sensing, Signal Processing and Security (ICSSS), pp 444–446
41. Chen H, Lv Z, Tang R, Tao Y (2017) Clustering energy-efficient transmission protocol for wireless sensor networks based on ant colony path optimization. In: IEEE International Conference on Computer, Information and Telecommunication Systems (CITS), pp 15–19

42. Enxing Z, Ranran L (2017) Routing technology in wireless sensor network based on ant colony optimization algorithm. Wirel Pers Commun 95(3):1911–1925
43. Sun Y, Dong W, Chen Y (2017) An improved routing algorithm based on ant colony optimization in wireless sensor networks. IEEE Commun Lett 21(6):1317–1320
44. Yang J, Shi X, Marchese M, Liang Y (2008) An ant colony optimization method for generalized TSP problem. Progr Nat Sci 18(11):1417–1422
45. Dorigo M, Birattari M (2011) Ant colony optimization. In: Encyclopedia of machine learning. Springer, pp 36–39