Received March 5, 2022, accepted March 25, 2022, date of publication April 4, 2022, date of current version April 12, 2022.
Digital Object Identifier 10.1109/ACCESS.2022.3164693

A Mechanism for Load Balancing Routing and Virtualization Based on SDWSN for IoT Applications

ELHAM HAJIAN, MOHAMMAD REZA KHAYYAMBASHI, AND NASER MOVAHHEDINIA
Department of Computer Architecture, Faculty of Computer Engineering, University of Isfahan, Isfahan 81746-73441, Iran
Corresponding author: Mohammad Reza Khayyambashi (m.r.khayyambashi@comp.ui.ac.ir)

ABSTRACT Wireless sensor networks (WSN) are important communication components of an internet of things (IoT). With the development of IoT and the increasing number of connected devices, network structure management and maintenance face the serious challenge of energy consumption. By balancing the network load, the energy consumption can be improved effectively. In the conventional WSN architecture, the two prerequisites of the load-balancing mechanism, flexibility and adaptability, are difficult to achieve. software-defined networking (SDN) is a novel network architecture that can promote flexibility and adaptability using a centralized controller. In this paper, a novel SDN architecture aimed at reducing load distribution and prolonging lifetime is proposed, which consists of different components such as topology, BS and controller discovery, link, and virtual routing. Accordingly, a new mechanism is proposed for load-balancing routing through SDN and virtualization. Through direct monitoring of the link load information and the network running status, the employed OpenFlow protocol can determine load-balancing routing for every flow in different IoT applications. The flows in different resource applications can be directed to a base station (BS) via various routes. This implementation reduces the exchange of network status and other relevant information. Virtual routing aims to weigh forward nodes and select the best node for each IoT application. The simulation results show the distribution of load over the network in the proposed algorithm and are characterized by the balanced network energy consumption, but also it prolongs network lifetime in comparison to the LEACH, improved LEACH, and LEACH-C algorithms.

INDEX TERMS IoT, load balancing, routing, SDN, WSN.

I. INTRODUCTION

In recent years, IoT has been the subject of extensive research and industrialists [1], [2]. It connects millions of devices to the Internet via networking. The connection of various devices to the IoT would create a heterogeneous system. Located in an environment, IoT devices usually consist of sensor nodes, agents, RFIDs, *etc.* [3]–[5], which can observe, analyze, and decide on the information collected from the network. With the dispersion and development of the IoT network, the amount of generated data by the network increases. As a result, it is difficult to manage, control, and secure devices in a heterogeneous network [6]. Managing a heterogeneous and geographically distributed network structure mounts a technical challenge, especially in dynamic environments [7]. Given the nature of devices, it is impossible to make extensive changes to support this kind of network. SDN [8] is a novel technology to meet IoT requirements in terms of flexibility and management of heterogeneous devices [9], [10]. SDN manages these devices and generates large volumes of data by separating the data plane from the control plane, connecting them through the southbound interface of OpenFlow [11], and performing centralized control in a global network view. These devices are connected via a WSN, which is a well-known IoT connection participator [12], [6]. Given enormous data and energy constraints in WSNs, a solution should be proposed to balance the network energy consumption and prolong the network lifetime. Routing is an option for energy consumption reduction [13]. The network energy consumption can be balanced via correct routing. Several routing protocols prolong the network lifetime by forwarding data that usually employ hierarchical

The associate editor coordinating the review of this manuscript and approving it for publication was Sathish Kumar.

clustering algorithms. The topology-control structure can help balance limited energy consumption [14].

The Low-Energy Adaptive Clustering Hierarchy (LEACH) [15] protocol is a common algorithm for effective routing. By selecting a random Cluster Head (CH) in different rounds, the LEACH protocol distributes the energy consumption of the network. However, this protocol has three major drawbacks: 1) it fails to consider the energy factor in the nodes; 2) although CHs are connected to the base station (BS) through single-hop communications, the protocol does not consider CH-BS distance, and 3) since CH changes in every round and the cluster network are updated, the energy consumption is high. Irrespective of the nodes' conditions, it is not an optimal choice for CH and path. Considering the optimal number of clusters for the protocol improvement, LEACH centralized (LEACH-C) [16] provides a centralized controller protocol in the BS. A node with a low energy level is not authorized to become a CH. This protocol uses the sum of mean distances between nodes and a CH for clustering.

For the CH selection phase, LEAC-C uses node energy and position to save energy. The main drawback of LEACH-C is that it overlooks energy consumption in the data transfer mode. In this protocol, a CH communicates directly with the BS without considering energy, CH position, and the amount of data that should be transferred. Thus, this protocol is not optimal in terms of energy consumption. Based on LEACH, another algorithm known as improved LEACH was proposed [14], which selects the path with minimum energy cost from the source to the BS. This protocol considers node energy and position to select a CH, connects every node to a CH, and performs clustering based on distances between nodes and data transfer energy consumption. Considering their energy and distances from the BS, CHs might be unable to communicate directly with the BS in this protocol. Instead, they use neighboring clusters as relay nodes to communicate with the BS. All of these protocols have a local view in data forwarding. However, for appropriate routing, a global view of the network is needed, which could be achieved by SDN.

This paper draws on the advantages of SDN and virtualization to propose a novel routing protocol based on load balancing and virtualization. The SDN controller requires the global view of the network for optimal routing, which is provided by reducing the number of messages exchanged between nodes to discover the topology, identify neighbor nodes, and detect the BS. In most of the proposed SDN-related protocols, the controller, by sending a message to the nodes, determines how far they are from the controller. Each node then sends messages to adjacent nodes to detect its neighbor nodes. Based on these messages, the energy and distance from neighbor nodes are recorded in the node's neighbor table. Afterward, nodes send their neighboring tables to the controller to develop the network topology. These steps are taken through the exchange of messages in the network. This study proposes a method to reduce the number of messages, and consequently energy consumption and balance data where all these steps are incorporated in one algorithm. The controller does not need the entire neighboring tables for topology discovery because a large portion of data in those tables is redundant. In this method, the controller allocates weight to each node based on its parameters and connects links for routing. A coefficient is also allocated to each parameter based on specific applications using the analytical hierarchy process method (AHP-express) [17]. Nodes with lower weights are selected for routing based on IoT applications and objectives. Therefore, this paper aims to address the following objectives and contributions:

- Developing a novel centralized architecture for different IoT applications through virtualization and SDN to prolong the network lifetime;
- Proposing a novel algorithm for the discovery of BS and neighboring nodes by other nodes to decrease the number of messages sent and network energy consumption;
- Proposing a novel algorithm for topology discovery by the controller to decrease network energy consumption;
- Proposing a novel routing algorithm including network-related metrics and node weighting to distribute and balance the network load;

The rest of this paper is organized as follows. Section 2 reviews a few related works regarding SDN-based and WSN-based routing protocols with traffic load solutions. The system framework is presented in Section 3. The novel routing algorithm is discussed in Section 4. The forward metric is introduced in Section 5. Efficiency evaluation is performed in Section 6. Finally, conclusions are drawn in Section 7.

II. RELATED WORKS

The lifetime of a WSN depends on the energy capacity of its nodes. Several algorithms have been proposed for energy efficiency. Since the communication unit consumes more energy than the data collection and processing unit, the bulk of research has focused on routing and data transfer for energy consumption reduction [18]. Energy-efficient routing protocols are classified into four categories [19]: 1) network structure, 2) communication model, 3) topology-based, and 4) reliable routing.

Most routing algorithms are based on distribution, which consumes excessive energy due to the exchange of several messages. Therefore, centralized algorithms have low overhead and low failure probability on routes [20].

Load balancing is an energy-efficient method for routing. In many studies, load balancing has been utilized for energy efficiency. Load balancing pursues four goals: 1) resource utilization, 2) quality of service, 3) resilience, and 4) scalability [21]. Every algorithm achieves at least one of the above goals. In this section, some relevant papers are reviewed Table 1 presents a brief overview of these papers, whereas Fig. 1 describes the algorithms briefly in terms of different goals.

In [22], an intelligent routing scheme based on Deep Reinforcement Learning (DRL) is proposed for IoT-enabled

WSNs, which significantly reduces delay and increases network lifetime. The proposed algorithm divides the whole network into different unequal clusters depending on data load in the sensor node, which significantly prevents the immature death of the network. The experimental results demonstrate the efficiency of the proposed scheme in terms of the number of alive nodes, packet delivery, energy efficiency, and communication delay in the network.

A hierarchical SDN-based approach [23] is proposed to accelerate data management and balance load not only between IoT devices in a single domain but also between different network clusters. The proposed framework prevents the controller from transforming into a single point of failure 2 allowing the application of different kinds of management and load balancing strategies in a hierarchical and multistep scenario. According to experimental results, their algorithm reduced the average turnaround and waiting time and improved the processing performance. Furthermore, the proposed approach distributes tasks evenly throughout the network and improves the use of network resources.

A fuzzy topology discovery protocol (FTDP) has been proposed in [24] to tackle the energy-efficiency issue in a WSN setup by integrating SDN and WSN, which gave rise to a more robust system, software-defined wireless sensor network (SDWSN).

The purpose of the FTDP is to choose the best new hop of the decision parameters in the case of the remaining energy(RE), the node cost(NC), the number of neighboring nodes(NN), and the queue length(QL) in the SDWSN design.

In FTDP, the sink gets information about the nodes and decision parameters. At that point, the information is transmitted to the controller and the controller uses the fuzzy system and the information to decide on the next hop.

In [25], the authors focused on an SDN Based Load Balancing (SBLB) service, in which minimized response time and maximized resource utilization are considered for users on cloud servers. SBLB is constituted by an application module that runs along with an SDN controller and server pools to connect to the controller through SDN-enabled switches. The application module contains a dynamic load balancing module, a monitoring module, and a service classification module. All messages are handled in real-time and the host pool is maintained by the controller. Also, the algorithm decreases the average response and reply time.

An energy-efficient data collection technique via definite sensing control in two-level IoT-enabled software-defined heterogeneous WSN (2SD-HWSN) is formulated in [26] as an optimization problem, with transmission distance from smart sensors, residual energy of sensors, and load based on node density. The proposed algorithm is divided into two: set-up and transmission phases. In the set-up phase, the control server (CS) elects the best-suited control nodes (CNs) and sets up a schedule for coordinating data transmission. Further, normal nodes join appropriate CNs based on distance and residual energy. This way, CNs form clusters, and route sensed data during the transmission phase. Therefore, an alternative nature-inspired algorithm, that is, grey wolf optimization (GWO), is hybridized with particle swarm optimization (PSO) using a low-level co-evolutionary technique to improve its overall performance. This hybrid variant of GWO, known as HGWO-BC, offers balanced clustering (BC) via novel fitness function design.

In [27], a cluster-based flow control approach in hybrid SDNs is proposed to reduce the number of control messages. The algorithm is hybrid in the sense that it exploits distributed legacy routing and centralized SDN routing. In addition, it creates a trade-off between the granularity of flow control and communication overhead induced by the SDN controller. This approach partitions a network into clusters with a minimum number of border nodes. Instead of handling individual flows in each node, the SDN controller only manages incoming and outgoing traffic flows of clusters by border nodes, while the flows inside each cluster are controlled by a distributed legacy WSN routing algorithm.

Several energy spaces have been considered in [20] based on the residual energy of nodes. In this method, the nodes communicate with each other in different spaces via wireless links, and the controller utilizes spaces with a higher energy rate for routing, if possible.

In [28], a Hybrid Fog-Cloud Offloading (HFCO) is introduced, where the tasks associated with complex applications are offloaded to the cloud servers to be executed and the results are sent back to the corresponding applications. It is a novel solution for sophisticated applications in IoT, where the IoT node is capable of offloading tasks to the best fog node or the cloud-based on the requirements of applications and conditions of the nearby fog nodes. In addition, fog nodes can offload tasks to each other or the cloud to balance the load and improve existing conditions, thereby allowing tasks to be executed more efficiently. The problem is formulated as a Markov Decision Process (MDP). Besides, a Q-learning-based algorithm is presented to solve the model and select the optimal offload policy. Numerical simulation results suggest that the proposed approach is superior to other methods in terms of lower delay, more executing tasks, and a balanced load.

A new energy-aware network topology obfuscation mechanism is proposed in [29], which maximizes the attack costs and is efficient and practical to be deployed. Network topology obfuscation is generally considered a promising proactive mechanism to mitigate traffic analysis attacks. The main challenge is to strike a balance among energy consumption, reliable routing, and security levels due to resource constraints in sensor nodes. first, a route obfuscation method is proposed by utilizing ranking-based route mutation, based on four different critical criteria: route overlapping, energy consumption, link costs, and node reliability. Then, a sink node obfuscation method is introduced by selecting several fake sink nodes that are indistinguishable from actual sink nodes, according to the k-anonymity model. As a result, the most suitable routes and sink nodes can be selected, and the highest obfuscation level can be reached without sacrificing energy efficiency.

In [30], SDN load balancing is carried out by determining the total number of demands and calculating the mean number of demands for every controller. The service-based link load balancing is then performed through link weight, which is determined by delay, loss rate, and bandwidth.

In [31] an optimized load balancing-based Admission Control Mechanism (Opt-ACM) is proposed for effective network flow management to decrease network congestion. To validate the efficiency of Opt-ACM, an optimization problem based on Mixed-Integer Linear Programming (MILP)-is formulated and tested using a well-known mathematical optimization solver called Gurobi.

A novel Efficient SDN-Based Wireless Sensor Network Architecture (ESD-WSN) has been proposed in [10] based on SDWSN and IoT by proxies. High-energy proxy nodes are selected to process control traffic and data aggregation. Proxies are determined in different rounds, and the proxy node assumes some of the controller's responsibilities.

In [32], a new routing algorithm has been proposed for IoT applications in wireless mesh networks through clustering. In this algorithm, the cluster head node and the relay node are employed to decrease the number of exchanged messages. The relay node can be used for inter-cluster communications.

In [33], the authors proposed reducing the flow table size and the number of its elements to decrease energy consumption. In this algorithm, flow entries with high probability remain in the table and other entries are deleted. If no entries in the flow table correspond during a specific interval, they are removed from the table by the Hidden Markov Model (HMM).

QoS-based routing protocol for Software-Defined Wireless Sensor Networks (QSDNWISE) is a novel architecture proposed in [34] for appropriate routing. In this architecture, a large number of clusters and cluster heads are utilized around the sink based on the high energy consumption of nodes. Separating data types, the algorithm performs independent routing for each data type.

Sink and controller localization and their optimal quantities were determined in [35] to reduce energy consumption. In this method, the number of source node hops from the sink and the controller declines, leading to lower energy consumption. Moreover, optimal quantities of sinks and controllers are employed to reduce their workload.

In [36], the number of flows is computed for load balancing using the number of links between controllers and between switches and controllers. A larger bandwidth with countless links can transfer a larger flow. Therefore, this parameter can be used to reduce the flow and perform load balancing.

To reduce the total load-energy cost, [37] exploits data traffic in WSN. For this purpose, the queue length of nodes and their residual energy are used.

A routing protocol is proposed in [38] based on virtualization and SDN. This algorithm employs Destination Oriented Directed Acyclic Graph (DODAG), which uses several routes for a specific source and a specific destination. In this routing protocol, specific weight is allocated to each node based on node congestion, link congestion, and energy to select its parent. Low-weight nodes are considered as parents.

The SDN Multiple Controller Load Balancing Strategy Based on Response Time (SSMCLBRT) protocol based on the response time is proposed in [39]. This protocol assigns a response time threshold for every controller. When the response time of a controller exceeds the threshold, the controller is said to be in overload conditions. The response time is defined as the difference between the packet-in transmitting and the packet-out receiving time by the controller. For this problem to be solved, the worst overloaded controller switch needs to migrate to controllers of lower overloads.

Dynamic load balancing is proposed in [40] for several controllers. By regulating the controller domain and improving the clustering effect, it assigns switches to the controllers of smaller domains. The Breadth-First Search (BFS) is used in different subdomains to assign switches to controllers in this algorithm. Also, considering the logical number of controllers, this algorithm performs the optimal switch-controller mapping to reduce communication costs and delays between switches and controllers.

III. SYSTEM FRAMEWORK

Fig. 2 shows the proposed SDN-based WSN model. In this model, the main idea is to separate the logic of control and data. The centralized controller aims to control nodes and the network, which is performed by the global network view. The proposed model consists of three planes, *i.e.* application plane, data plane, and control plane.

The application plane includes definitions, goals, services, and network functions, and the control plane covers management, information configuration, and the selection of forwarding nodes. Finally, the data plane is responsible for sensing and forwarding information. There are three major roles in this model including the models represented in [41], [42]: controller node, BS node, and SDN nodes. In these models, several BS nodes are used. The only difference is that, unlike previous models, each node can send information to several BS nodes in the proposed model. Using multiple BSs, the load is distributed and overloading is prevented in a BS. However, there is a need for new routing at any time to specify BS to send data from each node. Fig. 2 illustrates the part of the network model.

As the main controller of the network, the controller node is in charge of topology development and routing using the network states. It is also responsible for the management of the whole network and control functions. BS nodes are the network base stations, and all information is finally transmitted to one of these stations. The SDN node, resembling the OpenFlow switch in SDN, includes flow tables, and processes input flows based on the flow table. These nodes are responsible for matching and forwarding data streams in the WSN.

SDN is mainly characterized by separating the control plane from the data plane. With the global network view

TABLE 1. An overview of papers related to load balancing in SDWSN.

Year	Parameter	Feature	Load Balancing on Node or Controller	Controller	Routing	Clustering	Ref.
2021	Node congestion	Clustering based on network load	Node	×	✓	✓	[22]
2021	Number of activities	Distribution of tasks and the use of resources	Node	One controller	✓	×	[23]
2022	Energy, Queue length, and Number of neighboring nodes	Reducing the energy consumption with the fuzzy topology discovery protocol	Node	One controller	×	×	[24]
2021	Network congestion	Reduction of response time and resource utilization	Node	One controller	×	×	[25]
2022	Distance, Energy, and Load based on node density	Control nodes' selection, Load balancing, and Energy cost optimization	Node	One controller	✓	✓	[26]
2021	Number of flows	Reducing the number of controller messages	Node	One controller	✓	✓	[27]
2017	Energy	High-energy node routing	Node	One controller	✓	✓	[20]
2021	Number of activities	Offloading tasks to other nodes	Node	×	✓	×	[28]
2022	Route overlapping, Energy, Link costs, and node reliability	Energy-aware network topology obfuscation	Node	One controller	✓	×	[29]
2018	×	Dividing flow and decreasing traffic	Node	Multiple controllers	✓	×	[30]
2021	Network congestion	Flow management	Node	One controller	×	×	[31]
2018	Energy	Network load balancing through the proxy selection	Node	One controller	✓	×	[10]
2018	Node traffic	Using relay node and cluster head for routing	Node	×	✓	✓	[32]
2019	High-probability entries	Reducing flow table size	Node	One controller	×	×	[33]
2019	Separating traffic rates of time and reliability	Increasing the number of clusters around the sink	Node	One controller	✓	✓	[34]
2018	Number of hops–traffic	Controller and sink localization	Node and controller	Multiple controllers	✓	×	[35]
2020	Number of links and bandwidth	Passing flow reduction	Node	Multiple controllers	✓	✓	[36]
2020	Energy and node congestion	Balance routing	Node	×	✓	×	[37]

TABLE 1. *(Continued.)* **An overview of papers related to load balancing in SDWSN.**

2016	Link congestion and node congestion	DODAG and virtual routing	Node	One controller	✓	✓	[38]
2018	Response time	Switching migration to the controller with a lower load	Controller	Multiple controllers	×	✓	[39]
2019	Controller capacity and traffic emission delay	Switching to a controller with a lower load	Controller	Multiple controllers	×	✓	[40]
	Residual energy,Queue size, ,Bandwidth ,Delay	Routing different applications via different weights of nodes	Node	One controller	✓	✓	Proposed Algorithm

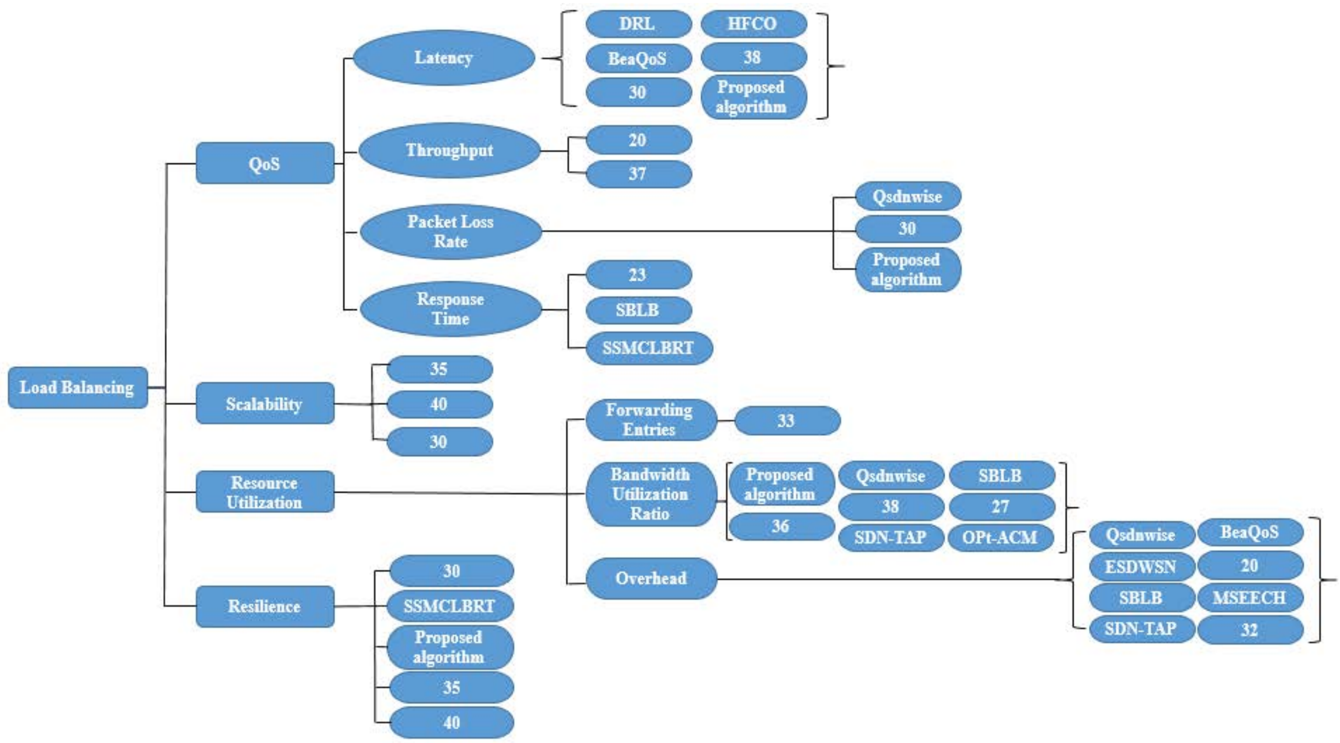


FIGURE 1. The tree diagram schema of studies reviewed by the authors.

and the network topology and information available, the controller tries to develop load balancing routing to forward packets.

IV. THE PROPOSED LOAD BALANCING ROUTING ALGORITHM

This section introduces the proposed algorithm in which different routes are used for data routing with different QoSs. An objective function is employed to route data with different QoSs. For instance, in previous algorithms, an objective function might reduce energy, but the end-to-end delay could be increased by finding the correct route from a node to the BS. Therefore, this mounts a serious challenge. The proposed method identifies an objective function for data with different applications in routing.

To do so, the controller must collect network information, forward it to the controller, develop a topology structure, and finally perform routing. Each of these steps is performed by a specific module.

According to Fig. 3, the system architecture consists of three modules for nodes and three main modules for the controller.

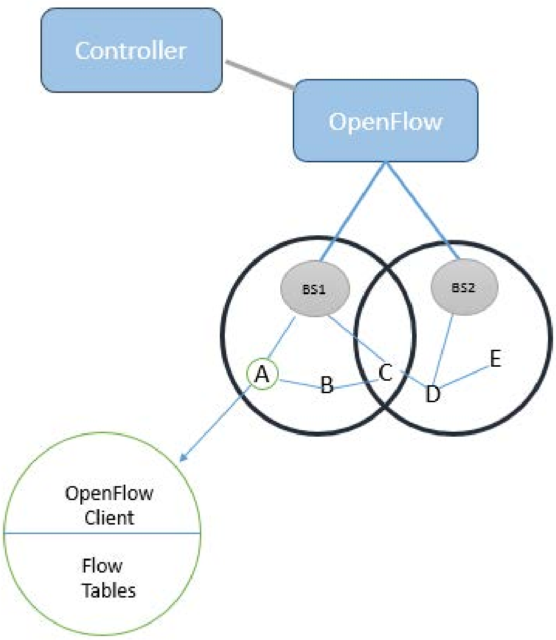


FIGURE 2. The part of the SDN-based WSN model.

The discovery module of BS, controller and neighbor nodes are in the data plane, whereas the discovery module of link, topology discovery, and virtual routing are in the controller. These modules are discussed in detail in the following subsections.

A. DISCOVERY MODULE OF BS, CONTROLLER, AND NEIGHBOR NODES

These three modules are integrated into a single module to be performed in one algorithm. This module is responsible for the discovery and maintenance of the neighbor nodes' tables, BS discovery, and controller discovery. In previous algorithms such as [20], [43], the discovery of neighbor nodes, controller, and BS is performed separately by sending separate messages. For this action, a message is sent from a controller with zero hops to discover the controller and the path. Upon reaching each neighboring node, the number of unit steps is added and the ID of the neighboring node along with the number of steps to the controller are stored in the routing table. Each node then sends this message to neighboring nodes. Thus, by broadcasting this message, all nodes will identify their path to the controller. Also, to discover neighbors, nodes re-send a message to their neighbors, requesting their energy and the number of steps to the controller.

When a reply is sent to this message by the neighbors, the neighbors' table is formed based on the neighbor ID and its information. Therefore, in the previous methods, it is necessary to send several messages in the network for each operation, which consumes more energy. In the proposed algorithm, however, all operations are carried out by transferring fewer messages and lowering energy consumption. For this purpose, the controller sends a topology discovery message to the BS. In this message, the number of hops to the BS is considered zero. The BS sends this message to its neighbor nodes (SDN nodes). The message contains the sender node's ID, its energy, and the number of hops that are sent to the specified BS. After receiving this message, the receiver node develops the neighbor table's entries including the sender node's ID, energy, and the number of hops to the BS. The number of hops is updated by adding one unit, which is sent along with the latter node's energy and its ID to the neighbors. When this message is broadcasted throughout the network, all nodes complete their neighbor tables with specific entries for each neighbor node. The output of this algorithm is a three-dimensional array INFO[n][n] [2]. n represents the number of nodes. The first and second dimensions are related to the proximity of nodes, which is set to 1 for adjacent nodes, and the third dimension is activated by the remaining energy and the number of hops. If non-adjacent node, there is no third dimension and the dimension is considered zero.

After the completion of these steps, every node identifies its neighbors, the number of their hops to the BS, and their energy levels. As a result, the neighbor nodes, BS, and controller are discovered. The proposed algorithm, by sending one message from the controller to nodes and the broadcasting message, discovers neighbor nodes, the controller, and BS. Thus, by reducing the number of messages sent, it decreases energy consumption.

B. LINK DISCOVERY MODULE

This module is responsible for discovering and maintaining the status of all physical links in the network.

In previous algorithms [43], nodes send the information of all neighbors to the controller, but in the proposed algorithm, when the previous step is finalized, nodes only send the information of neighbors with a lower ID number to the BS and the controller that it is in the third dimension INFO[n][n] [2] and includes their energy and number of hops. This can prevent the sending of duplicate data from different nodes. For instance, the link between nodes 1 and 2 is only sent once by node 2. The controller then recognizes the presence or absence of links between nodes. The links in the network are of two-way data type.

C. TOPOLOGY DISCOVERY MODULE

This module is responsible for developing and maintaining the network topology. When the previous steps are finalized, the controller develops topology by discovering the links between nodes based on the global network view. Then, the controller realizes the formation of nodes and BSs as clusters as well as the common node between clusters. Since there are no mobile nodes in the network, the topology remains constant until a node loses its energy and leaves the cycle.

D. VIRTUAL ROUTING MODULE

This module seeks to find an optimal route for every flow based on its application. By determining the adjacent

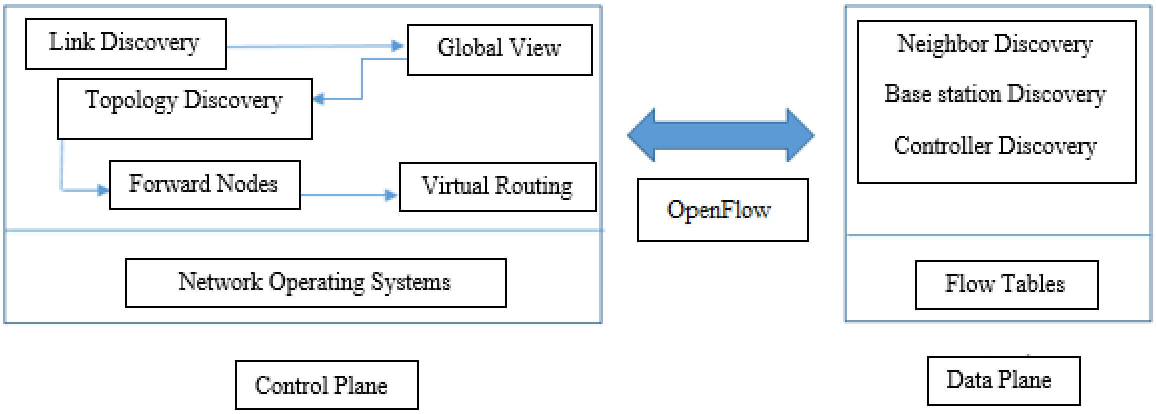


FIGURE 3. **The proposed system architecture.**

neighbors and finding the next hop for every network node based on a specific flow, this module facilitates the cooperation of SDN nodes and the network controllers.

Node routing is performed for a variety of purposes in applications, and the node input flows are directed to the BS with different QoSs through diverse routes. Every node has a flow table, which contains the next hop's node, specific operations including data forwarding to the next hop, and the number of hops to the BS for the flow with a specific QoS [8], [44].

The next hop' node is determined by the controller and sent to the nodes to be stored in the flow table. For routing, the controller needs network information such as nodes and links. This algorithm utilizes node queue length, distance from the BS, node energy, and link bandwidth. It requires the node and link conditions. Table 2 outlines the parameters required by the objective function.

1) PACKET QUEUE LENGTH

The packet queue length of every node in the next round is determined by a forward action in the current round. Eq. (1) [37] shows the process.

$$L_i^{t+1} = \min\left\{\left[L_i^t + N_{in,i}^t - N_{out,i}^t\right], Buffersize_i\right\} \tag{1}$$

L_i^t represents the queue length of the ith node at time t. $N_{in,i}^t$and$N_{out,i}^t$ are the number of input and output packets of the ith node at time t, respectively, and $Buffersize_i$ is the buffer size of the ith node. $N_{out,i}^t$ is presented in Eq. (2).

$$N_{out,i}^t = min\left\{L_i^t, \frac{e_i}{e_p}, \frac{Bwest_i}{B_p}\right\} \tag{2}$$

$\frac{e_i}{e_p}$ indicates the maximum number of packets sent by node i based on the energy of the node and $\frac{Bwest_i}{B_p}$ denotes this number based on bandwidth. $Bwest_i$ is the bandwidth of the node and B_p is the number of bits of packet p.

2) BANDWIDTH AVAILABLE FOR EVERY LINK

The available bandwidth of link (i,j) is calculated according to noise, interference, and probability of a successful transmission through Eq. (3).

$$Bwest_{i,j} = \left(1 - \left(IR_{i,j} + N_{i,j}\right)\right) * PT_{i,j} * BW_{i,j} \tag{3}$$

$IR_{i,j}, N_{i,j}, PT_{i,j}, BW_{i,j}$ are interference, noise, successful transmission probability and normal bandwidth of link (i,j), respectively.

Bandwidth information can be used as a routing metric for the dynamic selection of forwarding path(s), which can offer better service to a flow. These nodes estimate the bandwidth according to the channel utilization over a period by considering the combination of link capacity and link quality.

a: LINK CAPACITY

Link capacity is the maximum throughput or data rate of the link. Each node monitors channel utilization by measuring the throughput of transmitting a packet. Thus, our MAC layer measurement of available bandwidth accounts for the effects of both contention and physical errors caused by fading and interference. It should be noted that the available bandwidth is measured only by the successful MAC layer transmissions. Therefore, $PT_{i,j}*BW_{i,j}$ is the bandwidth affected by successful transmission probability.

b: LINK QUALITY

Measuring the link quality at every node helps balance the load between nodes. The link quality can be measured in terms of Interference Rate (IR) and noise rate. Transmissions on a wireless link may interfere with those on another provided that they both use the same radio link and are within the interference range of each other. This may also be affected by noise. Therefore, transmission interference and noise are critical factors for determining the achievable data rate. The interference and noise depend on factors such as network topology, traffic on neighboring links, etc. It has been shown that interference and noise can seriously affect the capacity of wireless networks in a multi-hop setting. A routing metric needs to detect the potential interference experienced by the links to find paths that suffer less interference and noise. As such, the overall network capacity could be improved with low transmission time. Therefore, $\left(1-\left(IR_{i,j}+N_{i,j}\right)\right)*PT_{i,j}*BW_{i,j}$ is the available bandwidth after considering noise and interference.

c: NOISE AND INTERFERENCE FOR EVERY

The degrees of the first and last nodes in every link are considered to calculate RT and noise rate. According to the Barabasi-Albert network, every link is a part of the real network, which reflects the RT and noise rate [45]. A higher link node degree will increase the noise and interference. Eq. (4) calculates the link noise and interference. K_1 and K_2 are the first and last node degrees of the link (i,j), respectively. The measure of noise and interference is directly related to the degree of link nodes and can be considered as the rate of both.

$$IR_{i,j}+N_{i,j}\sim\beta(K_1+K_2) \tag{4}$$

d: RESIDUAL ENERGY

The residual energy is a key metric for routing decisions. In routing, the residual energy of a neighbor node is used to select the forward node. Nodes with higher energy levels are preferred and selected as forwarding nodes.

In the LEACH energy model [15], [16], the l-bit data energy consumption model suggests that if the distance between a transmitter and a receiver (d) exceeds d_0, the multipath mode with a path loss coefficient of 4 will be used. Otherwise, the free-space mode with a path loss coefficient of 2 will be employed. Eq. (5) indicates this model for information transmission, whereas Eq. (6) shows the energy required for l bit transmission in both free-space and multipath modes separately.

$$E_{Tx}\left(l,d\right)=E_{Tx-elect}\left(l\right)+E_{Tx-amp}\left(l,d\right) \tag{5}$$

$$E_{Tx}\left(l,d\right)=\begin{cases} lE_{elect}+l\varepsilon_{fs}d^2 & d<d_0 \\ lE_{elect}+l\varepsilon_{mp}d^4 d\geq & d_0 \end{cases} \tag{6}$$

$E_{Tx-elect}$ represents the energy required to activate electronic circuits in the transmission state. $E_{Tx-amp}(l,d)$ is the energy required to activate the amplifier in that state, and E_{elect} is the energy required to activate electronic circuits.

On the receiver's side, the energy consumed for the l bit of data is obtained from Eq. (7).

$$E_{Rx}\left(l\right)=E_{Rx-elect}\left(l\right)=lE_{elect} \tag{7}$$

$E_{Rx-elect}$ expresses the energy required to activate electronic circuits in the reception state.

e: DELAY

In this algorithm, the distance parameter is regarded as a delay. The number of neighbor node hops to the BS is used to select the forward node. A node with the minimum distance is the best option for forwarding.

V. FORWARD METRIC

A key challenge is to design an objective function for different IoT applications. The proposed algorithm is recommended for an IoT program in a smart home with several applications such as lightening, face recognition, e-health, and home sensors. It needs an extensive objective function to justify all features. Instead of four objective functions, weighted parameters are proposed for all types of flow based on QoS application and each neighbor node's requirements for forwarding purposes.

The main parameters of face recognition, e-health, lightening, and home sensor applications are bandwidth, delay, energy, and queue, respectively. As described in the previous section, the main components of the objective function are delay, energy, queue, and bandwidth. The objective function is the jth node weight for routing as the next node of the ith node. A node with the minimum weight is selected as the forward node. The objective function is calculated from Eq. (8).

$$W_j=\frac{\alpha_1*L_j+\alpha_2*D_j}{\alpha_3*Erem_j}-\alpha_4*Bwest_{ij} \tag{8}$$

where α_i is the relative coefficient for ith parameter indicating the importance of the parameter for the routing decision. For an application with queue length significance, α_1 is selected to be greater than other coefficients, while for an application with the delay significance, α_2 is considered to be greater than other coefficients. The rest of the coefficients are defined similarly.

- D_j is the distance between the jth neighbor node and the BS. The jth node is a neighbor of the ith node. In this case, distance is defined as the number of hops. A node with the minimum distance from the BS is selected as the best forward node. Therefore, node weight is directly related to distance. $(W_j\alpha D_j)$
- $Erem_j$ is the residual energy of the jth node. A node with maximum residual energy takes priority as the forward node. Therefore, weight is indirectly related to the residual energy $(W_j\alpha\frac{1}{Erem_j})$.
- L_j is the queue length of the jth node. A node with a minimum queue length is preferred as the forward node. Hence, weight is directly related to the queue length $(W_j\alpha L_j)$.
- $Bwest_{ij}$ is the bandwidth estimation based on the link efficiency. It includes the rates of loss, noise, and interference. A node with higher bandwidth should be selected. The bandwidth is then subtracted from the total weight to increase the probability of selecting a node as the forward node. A higher bandwidth increases the selection chance of a node.

The importance of a parameter is considered in the IoT application to obtain coefficients. Accordingly, the values of these coefficients are calculated from the AHP-express algorithm, which is based on pairwise comparisons. In other words, the parameters are considered to be in the same application. The AHP-express is a multi-criteria decision-making method that uses a hierarchical or network structure to reveal

TABLE 2. The necessary parameters in the proposed algorithm.

Parameter	Definition
L_i^t	queue length of the *i*th node at time t
$N_{in,i}^t$	Number of input packets of the *i*th node at time t
$N_{out,i}^t$	Number of output packets of the *i*th node at time t
$Buffersize_i$	Buffer size of the *i*th node
e_i	The energy of the *i*th node
e_p	Energy consumption for transmission of a packet (p)
$Bwest_i$	The bandwidth required by the *i*th node
B_p	Number of packet bits
$Bwest_{i,j}$	The bandwidth of the link between *i* node and *j* node
$IR_{i,j}$	Interference for the link (i,j)
$N_{i,j}$	Noise for the link (i,j)
$PT_{i,j}$	Probability of successful transmission for the link (i,j)
$Bw_{i,j}$	Normal bandwidth for the link (i,j)
K_1	The first node's degree of the link
K_2	The last node's degree of the link
$E_{TX}(l,d)$	The energy consumed to send *l* bit over distance *d*
$E_{TX-elect}(l)$	The required energy required to activate electronic circuits in the transmission state for *l* bit
$E_{TX-amp}(l,d)$	The required energy to activate the amplifier in the transmission state for *l* bit in distance d
E_{elect}	The required energy to activate electronic circuits
ε_{fs}	Emission coefficient of the amplifier energy in the free-space state
ε_{mp}	Emission coefficient of the amplifier energy in the multi-path state
d	Distance
$E_{RX}(l)$	The energy consumption for the reception of *l* bit
$E_{RX-elect}(l)$	The required energy to activate electronic circuits in the reception state for *l* bit
D_j	Distance between the *j*th node and the BS
L_j	Queue length of the *j*th node
$Erem_j$	The residual energy of the *j*th node
β	Relative coefficient of interference and noise
α_1‘α_2‘α_3‘α_4	Coefficients of different parameters

a problem and determine priorities based on decision-maker's judgments along a process. As regards user acceptability, the score of AHP-express is higher than other multi-criteria decision-making methods; therefore, it has been adopted in this paper to find coefficients.

Table 3 shows the coefficients obtained for every application using AHP-express.

A. ROUTE UPDATING

If the node energy falls below the threshold along the flow path or the node congestion exceeds the threshold, that node will directly communicate with the controller and request a new route to balance the load. Based on the parameters, the controller selects the alternative node and informs the node of changes in its forward node. In this way, the route is updated. In this process, the node energy lasts longer and the network lifetime is extended.

VI. EFFICIENCY EVALUATION

A. SETTING PARAMETERS

This section evaluates the efficiency of the proposed method through computer simulation. For this purpose, a simple network with 100 nodes, 1 controller, and 5 base stations was designed. The nodes were developed randomly in the network. The proposed method was implemented in four applications defined in the previous section. The data traffic of every application is allocated to each node based on its application. Finally, the proposed algorithm is compared to the LEACH, LEACH-C, and improved LEACH algorithms in four applications. The simulation was performed in NS 2.35 on Ubuntu 10 running on VMware 13. A schematic view of the network is shown in Fig. 4.

Depending on its cluster, every node is connected to a base station to transmit its data and information to that BS. A node might be placed in the cluster corresponding to two base stations. In the proposed method, the controller calculates every node's weight as the forward node in each application and then selects the lowest weight according to Eq. (8). The forward node is then introduced to the node. Every application uses its specific coefficients, as outlined in Table 3, to calculate the weight.

Table 4 presents an overview of simulation information and parameters.

B. EFFICIENCY ANALYSIS

The proposed algorithm is compared to LEACH, LEACH-C, and improved LEACH algorithms for efficiency analysis. Like the proposed algorithm, the above three algorithms, based on clustering, are useful for distributed routing. These algorithms also employ network parameters for routing. These items justify choosing top three algorithms for comparing.

The improved LEACH is the extended version of the LEACH algorithm. Like the other two algorithms, the improved LEACH aims to balance the network load.

TABLE 3. Coefficients required for every application in the objective function.

Lightening	α1=0.1	α2=0.1	α3=0.75	α4=0.1
Face recognition	α1=0.18	α2=0.18	α3=0.077	α4=0.55
E-health	α1=0.19	α2=0.59	α3=0.08	α4=0.11
Sensor home	α1=0.64	α2=0.12	α3=0.09	α4=0.12

By doing so, the algorithm improves efficient data transfer and prolongs the network lifetime. These algorithms are compared to the proposed algorithms in terms of residual energy, the number of dead nodes, received data, and queue length. The following comparison charts are presented for e-health, face recognition, lightening, and home sensor applications.

1) E-HEALTH

Fig. 5 indicates the results of simulation for the e-health application Fig. 5a shows the mean residual energy for every algorithm. Node energy in LEACH and LEACH-C algorithms is consumed faster because some factors such as energy and nod distance are disregarded in determining to route until all nodes lose their energy. In these algorithms, the balance of consuming energy is not considered. As regards, network factors, both the proposed algorithm and improved-LEACH have higher residual energy. In comparison to the improved LEACH, the proposed algorithm consumes energy and distributes the network load for routing in a balanced manner. The improved LEACH and the proposed algorithm have nearly equal amounts of residual energy. At the beginning of the simulation, the residual energy in the improved LEACH algorithm is higher level than the proposed algorithm. As the network grows stable, the residual energy of the proposed algorithm rises.

Fig. 5b shows the number of data received for a specific application in the network. Accordingly, there were more successful transmissions in the proposed algorithm than in the other three algorithms because it had large volumes of received data. Based on the network conditions, this algorithm selects the best route for transmission and therefore makes fewer unsuccessful transmissions. The network congestion is also lower in this algorithm than in other algorithms. Fig. 5c depicts the number of dead nodes in each algorithm. In the LEACH algorithm, the loss of node energy is faster than other algorithms. In the proposed algorithm, many packets are initially sent and received, so its number of dead nodes is greater than that of the improved LEACH. However, in comparison to the improved LEACH, the number falls when the network becomes stable. To gain further insights into load balancing in the proposed algorithm, the Mean Square Error (MSR) chart of residual energy is displayed in Fig. 5d. When the number of dead nodes in the proposed algorithm is lower than the improved LEACH, its MSR exceeds that of the improved LEACH. This shows that

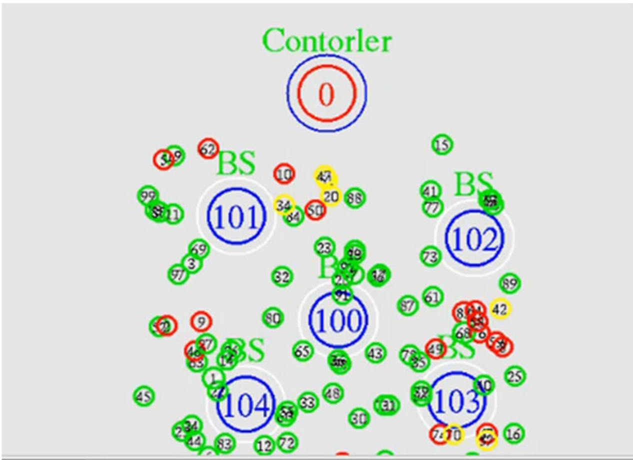


FIGURE 4. Schematic view of the simulation.

TABLE 4. Values of necessary parameters for simulation.

Parameters	Values
Number of Sensors	100
Number of BS	5
SDN Controller	1
Packet Size	4000
Initial Energy	0.5 joule
E_e	50 nJ/bit
εfs	5pj/(bit.m^2)
εmp	1.3fj/(bit.m^2)
Network area	100*100
Simulation Time	450
Queue Type	Drop Tail (FIFO)
Queue Size	200 Packets
Antenna Type	Omni antenna

the proposed algorithm consumes energy in a more balanced manner in the network. It is because the best routes are selected for data transmission, which reduces the uneven energy consumption. The large MSR of the residual energy in the proposed method suggests that more nodes are utilized and active in the routing process. Thus, the network load is distributed over more nodes and the load is routed by the network in a more balanced way. Over time, MSR residual energy also decreases due to energy loss by the nodes.

Fig. 5e shows control overhead. This figure is considered packets that are sent from the MAC layer and are received by the network layer and are declared as routing overhead. As shown in the figure, the proposed algorithm operates better than other algorithms because according to Fig. 5b has more received data and fewer lost packets. If there are more resending packets, control overhead will be more.

Fig. 5f illustrates the delay of received successful packets from source to destination. Accordingly, the delay of the proposed algorithm is lower than any other algorithm. Because, based on the condition of the network and nodes, a route is selected that has the lower delay.

Fig. 5g shows the average path cost from source to destination. This parameter is considered based on hop count. Initially, because of transferring many control packets in the network, the hop count ascends with a steep slope and almost stands still afterward. Then it decreases because of using near nodes with low delay, with transferring data packets.

2) FACE RECOGNITION

This section presents the simulation results for face recognition. Fig. 6 illustrates the corresponding comparison charts.

Fig. 6a shows the residual energy of nodes. In comparison to the previous applications, the residual energy is higher in this application than in the improved LEACH.

Fig. 6b draws a comparison between all algorithms in terms of the number of received data for face recognition. In this application, the proposed algorithm delivers more data to the destination over time and has a lower rate of packet loss than other algorithms.

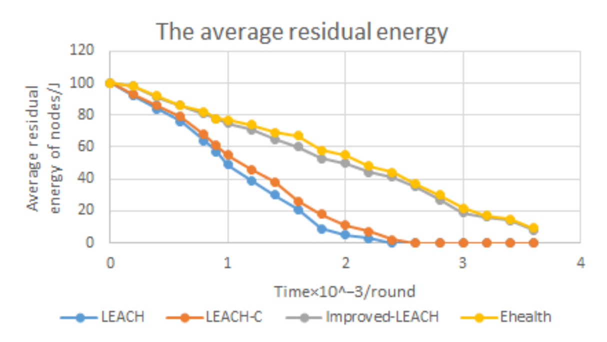
Fig. 6c shows the number of dead nodes. Compared to the previous applications, the proposed algorithm requires more time than the improved LEACH to decrease the number of dead nodes. When the network becomes stable, the number of dead nodes rises in the proposed algorithm for this application.

Fig. 6d draws a comparison between the MSR of algorithms. Given the energy consumption balance of the network, the proposed algorithm has a larger MSR in the residual energy. Compared to previous applications, its MSR is improved with a better slope due to its higher residual energy. The proposed method activates a larger number of nodes by sending data in different paths and avoiding load imposition on specific nodes by distributing the load over the network, Hence, the MSR of residual energy is higher in this method. Over time, many nodes participate in routing, and the energy of many nodes is low. So their residual energy MSR is reduced.

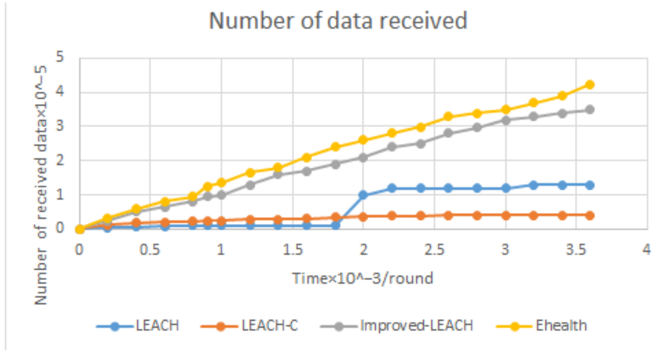
Fig. 6e shows control overhead in face recognition applications. The proposed algorithm operates better than other algorithms because this algorithm considers the condition of the network and nodes and selects the best nodes for routing that has fewer lost packets. So it has less control overhead. Initially, because of transmitting control packets in the proposed method, this parameter is almost equal to other algorithms.

Fig. 6f shows the delay of successful packets from source to destination in this application. The delay of the proposed algorithm is lower than others. Initially, because of the same conditions of all nodes, route with nodes is selected that have a lower delay. During the time, because these nodes have less energy and more queue size, according to forwarding metric nodes with higher delay will select.

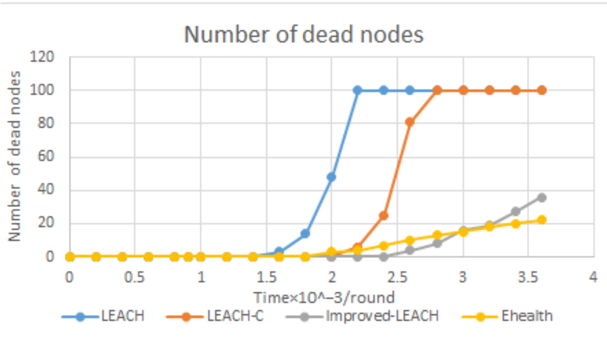
Fig. 6g presents the hop count for received successful packets as the average path cost. Initially, the number of hops is also largely due to the exchange of many packets of



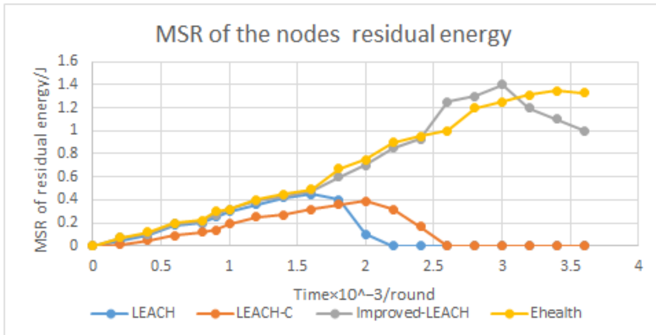
5a: the average residual energy



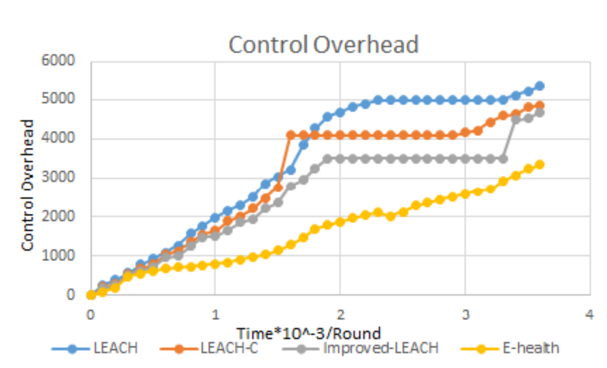
5b:number of received data



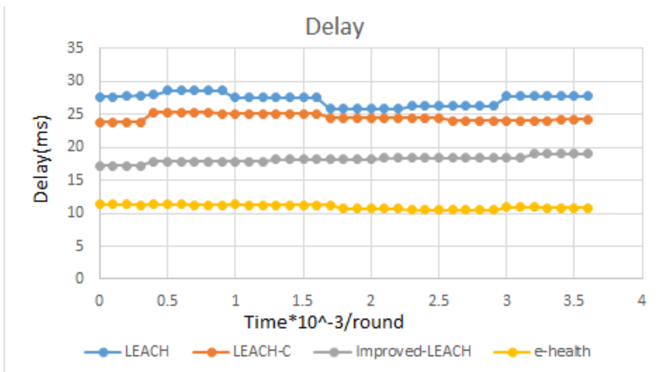
5c: Number of dead nodes



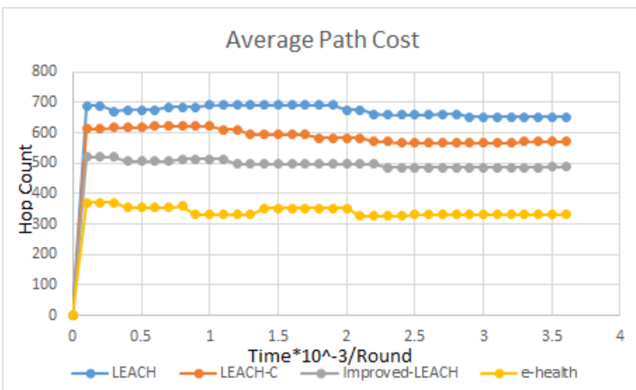
5d: MSR of the nodes residual energy



5e: Control Overhead



5f: Delay



5g: Average Path Cost

FIGURE 5. **Comparison charts of e-health application.**

data and control. Then, that decreases with transferring data packets only.

3) LIGHTENING

This section depicts charts associated with lightening applications. Fig. 7a shows the residual energy.

Fig. 7b shows the number of data received in this application. Accordingly, the number of data received the proposed algorithm is larger than in other algorithms; therefore, the proposed algorithm has a lower rate of packet loss.

Fig. 7c shows the number of dead nodes in the lightening application. Similar to the other two applications, the node energy was lost sooner in the LEACH algorithm than in other algorithms; however, in the proposed algorithm, the energy was lost over a longer period.

Fig. 7d presents the MSR of the residual energy. Like the other applications, when there are fewer dead nodes in the proposed algorithm than in the improved LEACH algorithm, the MSR of the proposed algorithm will be larger than that of the improved LEACH, and vice versa. This confirms the simulation validity.

Fig. 7e illustrates control overhead in lightening application. The figure is considered packets that send from the MAC layer and receive to the network layer and are declared as routing overhead. The more resend packages, the more control overhead. In the proposed algorithm exists fewer lost packets because it considers the condition of the network and selects the best nodes for routing. Initially, because of transmitting control packets in the proposed method, this parameter is almost equal to other algorithms. During the time, this parameter is much lower than other algorithms but will rise again because of sending control packets for new routing.

Fig. 7f depicts the delay of packets in this application. Like the top applications, a route is selected that has the lowest delay.

Fig. 7g shows the average path cost in the lightening application. After completing the transferring of control packets, this parameter is almost constant for the proposed algorithm and is lower than other algorithms.

4) SENSOR HOME

The following charts display the sensor home application. Fig. 8a shows the residual energy of nodes. As can be seen, the levels of residual energy in the proposed algorithm are higher than in other algorithms.

Fig. 8b shows the number of received data in the sensor home application. This application is significantly different from the improved LEACH. Accurate data reception is also more important in this application than in others. Therefore, its rate of data reception will be higher than other applications.

Fig. 8c illustrates the number of dead nodes in the sensor home application. Given the larger number of sent and received packets in the proposed method, initially more nodes lose their energy compared to the improved LEACH. However, as time elapses and the network is stabilized, this rate takes a downturn until it falls behind that of the improved LEACH.

Fig. 8d depicts the MSR of residual energy in the sensor home application. Based on the number of dead nodes, the MSR of residual energy increased when there were fewer dead nodes in the proposed algorithm than in the improved LEACH.

Fig. 8e shows control overhead in sensor home application. The proposed algorithm operates better than other algorithms in this parameter.

Fig. 8f shows the delay in sensor home application. Accordingly, the delay of the proposed algorithm is slower than other algorithms. During the time, this parameter increases because the nodes with low delay lose their energy and have long queues. According to the forward metric, these nodes aren't good for routing, and it will have to use more delayed nodes.

Fig. 8g illustrates the average path cost in this application. This parameter in the proposed algorithm is lower than other algorithms.

C. AVERAGE ANALYSIS

Below, the average evaluation of the parameters of the previous section is examined.

Fig. 9 compares the residual energy charts of the proposed algorithms to the proposed algorithm in terms of four applications. Residual energy is higher in the proposed algorithm than in others because it selects the best route based on several parameters for the selection of the forwarding node. In general, the level of residual energy is higher in the proposed algorithm than in other algorithms and this difference increases over time. It means the proposed algorithm would produce the best results in the long term.

Fig. 10 draws a comparison between algorithms in terms of the number of dead nodes. In short, there are fewer dead nodes at a specific time in the proposed algorithm than in other algorithms due to the balanced energy consumption and load distribution. Initially, there are more dead nodes in the proposed algorithm due to calculations and information exchange, which would decrease over time until the best results are achieved.

Fig. 11 compares the proposed algorithm with four applications to other algorithms in terms of the packet loss rate. Accordingly, the packet loss rate was lower in the proposed algorithm than in others because reduced congestion through load balance and distribution curtailed the queue length of nodes. As a result, the number of lost packets decreased. In the proposed algorithm, the routing was performed properly and the load was balanced across the network. As the figure shows, in the proposed algorithm and improved LEACH, some nodes faced congestion within [3.2, 3.6], which was characterized by a high packet loss rate, and the number of dead nodes surged. The proposed algorithm outperformed other algorithms even in the state of congestion.

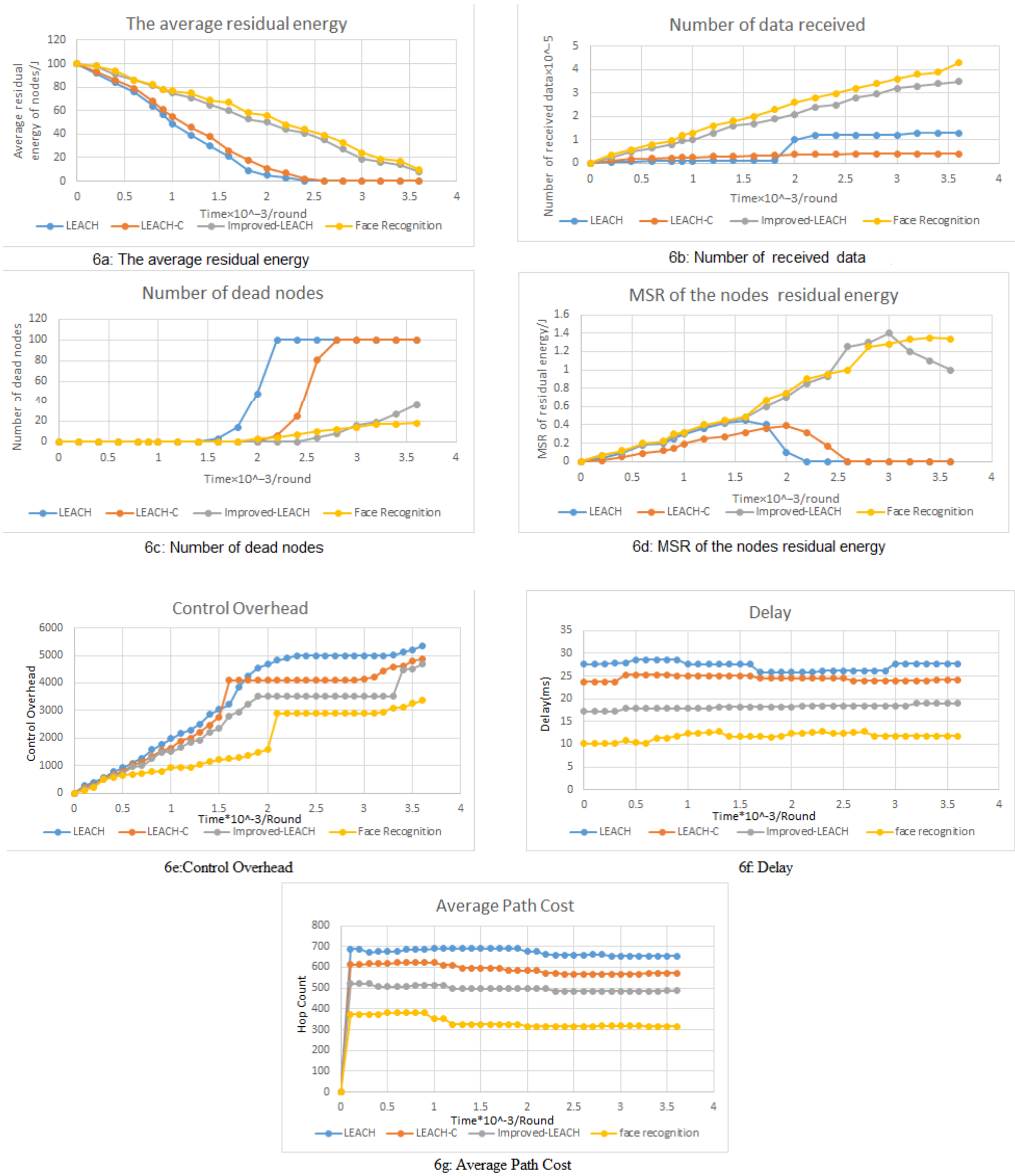


FIGURE 6. **Comparison charts for face recognition.**

Fig. 12 compares the proposed algorithm with four applications to other algorithms in terms of the control overhead. As shown in Fig. 11, since the number of lost packets in the proposed algorithm is less than other algorithms, the overhead control to resend the lost packets is less. This means that the proposed algorithm requires

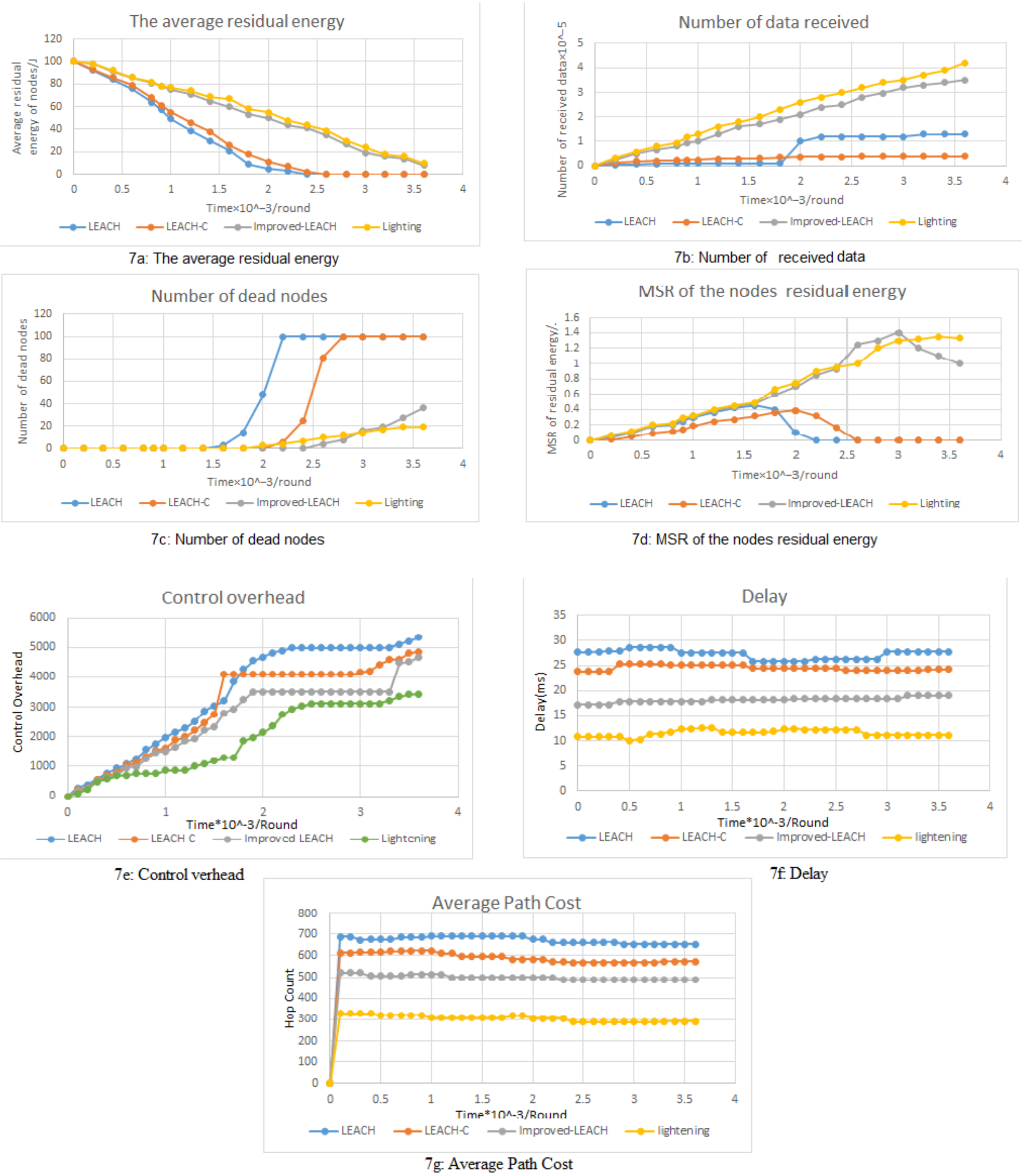
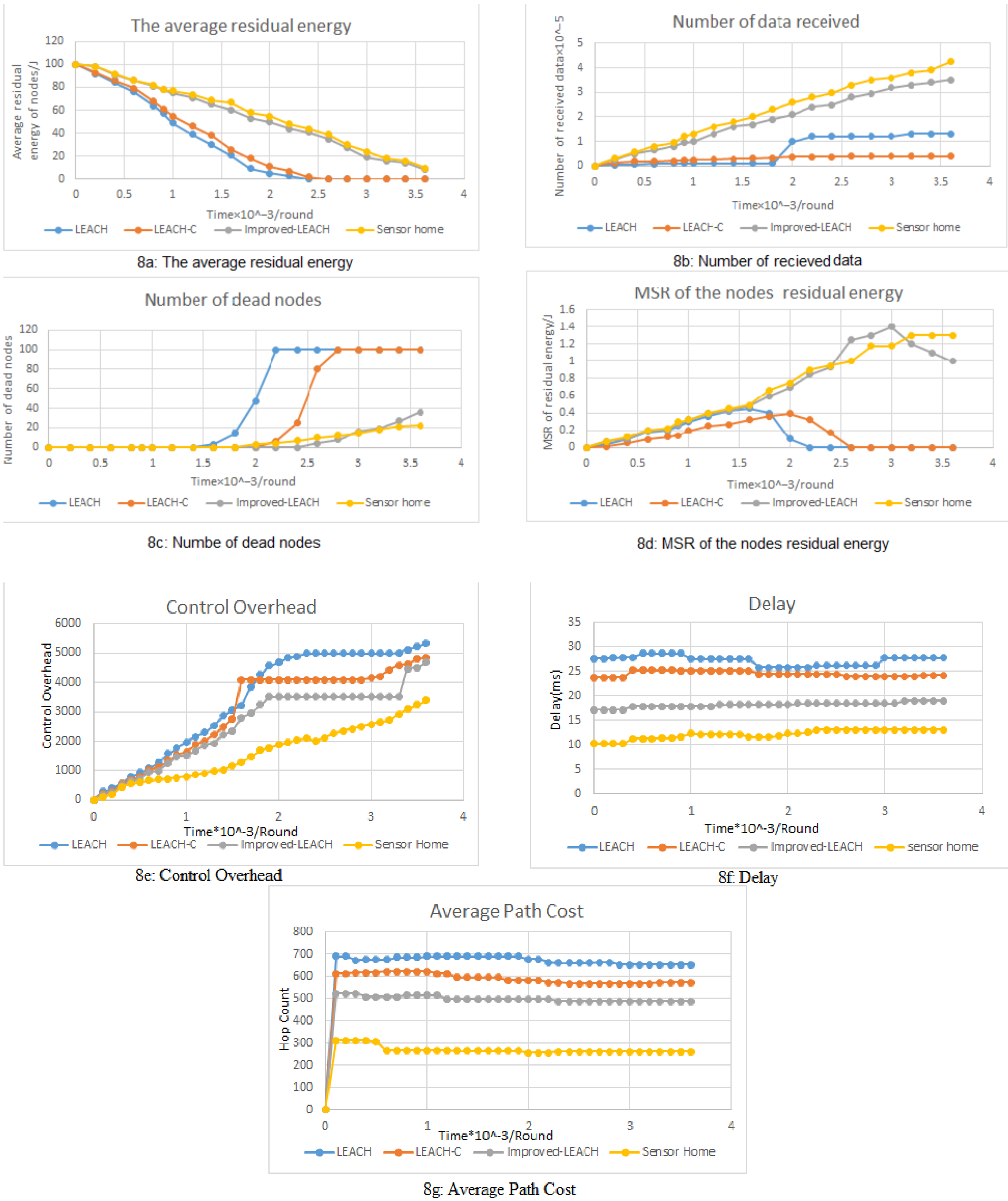


FIGURE 7. **Comparison charts of lightening application.**

fewer packets resending and more packets to reach their destination.

Fig. 13 shows the mean queue length for different applications. It suggests the longest and shortest mean queue lengths are associated with face recognition and lightening applications, respectively. The mean queue lengths of different applications were nearly balanced in all nodes.



8a: The average residual energy

8b: Number of recieved data

8c: Numbe of dead nodes

8d: MSR of the nodes residual energy

8e: Control Overhead

8f: Delay

8g: Average Path Cost

FIGURE 8. **Comparison chats of sensor home application.**

Fig. 14 shows the mean queue lengths of all algorithms. As can be seen, the queue length of the proposed algorithm was shorter than other algorithms due to data distribution across the network. As a result, fewer queues were created in

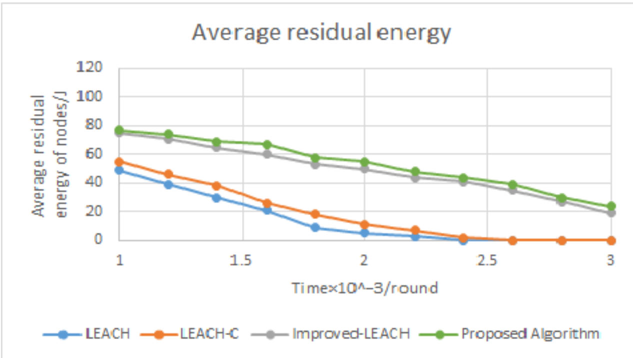


FIGURE 9. Comparison of all algorithms in the residual energy of nodes.

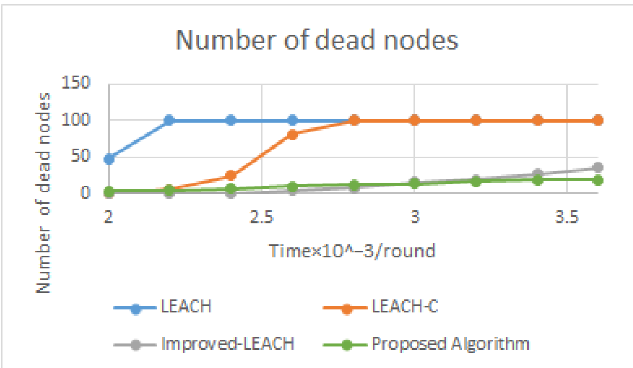


FIGURE 10. Comparison of all algorithms in terms of the number of dead nodes.

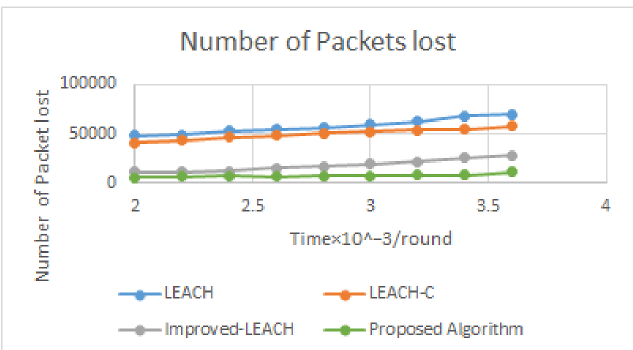


FIGURE 11. Comparison of all algorithms in terms of the number of lost packets.

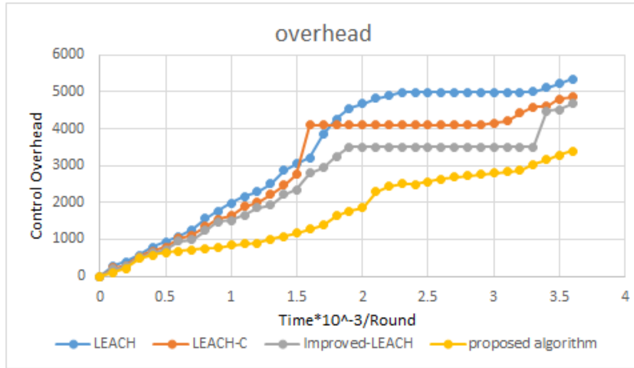


FIGURE 12. Comparison of all algorithms in terms of the control overhead.

FIGURE 13. Mean queue lengths of different applications.

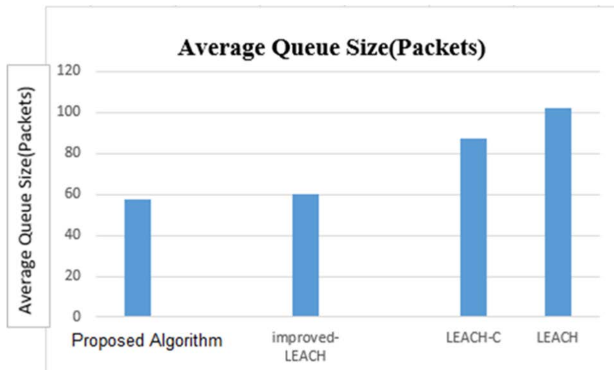


FIGURE 14. Mean queue lengths of all algorithms.

nodes, and data loss decreased. Routing over various nodes decreases queue length in nodes.

The last six figures yield the following results: given the importance of energy, the number of active nodes, and the number of packets received and control overhead in the networks facing energy constraints such as WSN and considering load balancing and residual energy augmentation, it can be concluded that discussing load balancing only through the analysis of residual energy among nodes is impossible. It is because the residual energy of the nodes is high but the load would not be balanced. That is, while some nodes might not have any energy at all, others have high levels of energy. However, analyzing the number of dead nodes contradicts the abovementioned conclusion drawn from the proposed algorithm there were fewer dead nodes compared to other algorithms. Therefore, many of the nodes were active and had high levels of energy. This finding is confirmed by the analysis of the MSR chart of residual energy. Moreover, activating most of the nodes and decreasing their load through the queue length reduced or balanced network congestion. Lost packets in the proposed algorithm were fewer than other algorithms. Hence, it balanced load across the network by increasing the number of active nodes and the residual energy of every node and at the same time reducing network congestion and queue length. Despite load distribution, the number of received packets kept rising, and resending packages and control overhead is decreasing.

VII. CONCLUSION

The proposed algorithm presented a novel routing protocol for network load balancing. Using SDN and clustering, this algorithm provided a balanced routing for decreasing the number of sent messages. In the setup phase, the algorithm decreased energy consumption and ensured load balance by discovering neighbors and controllers in an operation and transmitting the information of neighbor nodes through controllers for topology discovery. In the routing phase, the algorithm guaranteed data transmission in an energy-efficient route by selecting the best node of the next-hop based on network parameters. These solutions distributed and balanced network load decreased network energy consumption, and prolonged network lifetime. For verification, the simulated algorithm was compared to the LEACH, LEACH-C, and improved LEACH algorithms with the simulation results lending credit to findings.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, pp. 2785–2805, Oct. 2010.

[2] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, 2012.

[3] S. Rani, R. Talwar, J. Malhotra, S. H. Ahmed, M. Sarkar, and H. Song, "A novel scheme for an energy efficient Internet of Things based on wireless sensor networks," *Sensors*, vol. 15, no. 11, pp. 28603–28626, Nov. 2015.

[4] O. Flauzac, C. Gonzalez, and F. Nolot, "Developing a distributed software defined networking testbed for IoT," presented at the 7th Int. Conf. Ambient Syst., Netw. Technol. (ANT), 2016.

[5] T. M. C. Nguyen, D. B. Hoang, and T. D. Dang, "Toward a programmable software-defined IoT architecture for sensor service provision on demand," presented at the 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC), Melbourne, VIC, Australia 2017.

[6] T. M. C. Nguyen, D. B. Hoang, and Z. Chaczko, "Can SDN technology be transported to software-defined WSN/IoT?" presented at the IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData), Chengdu, China, 2016.

[7] N. Bouchemal, S. Kallel, and N. Bouchemal, "A survey: WSN heterogeneous architecture platform for IoT," presented at the Int. Conf. Mach. Learn. Netw., Paris, France, 2018.

[8] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[9] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-defined WSN management system for IoT applications," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2074–2081, Sep. 2018.

[10] Z. Zhang, Z. Zhang, R. Wang, Z. Jia, H. Lei, and X. Cai, "ESD-WSN: An efficient SDN-based wireless sensor network architecture for iot applications," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.* Helsinki, Finland: Springer, 2017, pp. 735–745.

[11] N. McKeown, T. Anderson, H. Balakrishnan, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[12] H. C. Silva, A. E. H. Pereira, Y. K. Solano, B. T. D. Oliveira, and C. B. Margi, "WARM: WSN application development and resource management," presented at the Simpósio Brasileiro de Telecomunicacõe, Santarém, Portugal, 2016.

[13] S. Tomovic and I. Radusinovic, "Extending the lifetime of wireless sensor network with partial SDN deployment," *Telfor J.*, vol. 8, no. 1, pp. 8–13, 2016.

[14] Y. Zhang, P. Li, and L. Mao, "Research on improved low-energy adaptive clustering hierarchy protocol in wireless sensor networks," *J. Shanghai Jiaotong Univ. (Sci.)*, vol. 23, no. 5, pp. 613–619, Oct. 2018.

[15] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," presented at the 33rd Hawaii Int. Conf. Syst. Sci., Maui, HI, USA, USA, 2000.

[16] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[17] J. E. Leal, "AHP-express: A simplified version of the analytical hierarchy process method," *MethodsX*, vol. 7, 2020, Art. no. 100748.

[18] V. Srivastava, S. Tripathi, K. Singh, and L. H. Son, "Energy efficient optimized rate based congestion control routing in wireless sensor network," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 3, pp. 1325–1338, Mar. 2020.

[19] C. Xinying, W. Muqing, and L. Wenxing, "Energy efficient algorithm for SDWSNs based on DACR and energy hierarchy," presented at the 3rd IEEE Int. Conf. Comput. Commun., Chengdu, China 2017.

[20] L. Wenxing, W. Muqing, and W. Yuewei, "Design of multi-energy-space-based energy-efficient algorithm in novel software-defined wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 7, Jul. 2017, Art. no. 155014771771811.

[21] F. Chahlaoui and H. Dahmouni, "A taxonomy of load balancing mechanisms in centralized and distributed SDN architectures," *SN Comput. Sci.*, vol. 1, no. 5, p. 268, Sep. 2020.

[22] G. Kaur, P. Chanak, and M. Bhattacharya, "Energy-efficient intelligent routing scheme for IoT-enabled WSNs," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11440–11449, Jul. 2021.

[23] Z. Eghbali and M. ZolfyLighvan, "A hierarchical approach for accelerating IoT data management process based on SDN principles," *J. Netw. Comput. Appl.*, vol. 181, no. 4, 2021, Art. no. 103027.

[24] J. Kipongo, E. Esenegho, and T. G. Swart, "Efficient topology discovery protocol using IT-SDN for software-defined wireless sensor network," *Bull. Electr. Eng. Informat.*, vol. 11, no. 1, pp. 256–269, Feb. 2022.

[25] S. Kumar, K. Cengiz, S. Vimal, and A. Suresh, "Energy efficient resource migration based load balance mechanism for high traffic applications IoT," *Wireless Pers. Commun.*, p. 5111, Feb. 2022.

[26] P. Mishra, N. Kumar, and W. W. Godfrey, "An evolutionary computing-based energy-efficient solution for IoT-enabled software-defined sensor network architecture," *Int. J. Commun. Syst.*, Feb. 2021.

[27] Q. Liu, L. Cheng, R. Alves, T. Ozcelebi, F. Kuipers, G. Xu, J. Lukkien, and S. Chen, "Cluster-based flow control in hybrid software-defined wireless sensor networks," *Comput. Netw.*, vol. 187, Mar. 2021, Art. no. 107788.

[28] S. Aljanabi and A. Chalechale, "Improving IoT services using a hybrid fog-cloud offloading," *IEEE Access*, vol. 9, pp. 13775–13788, 2021.

[29] M. Bin-Yahya and X. Shen, "Secure and energy-efficient network topology obfuscation for software-defined WSNs," *IEEE Internet Things J.*, early access, Jan. 20, 2022, doi: 10.1109/JIOT.2022.3144873.

[30] F. Shang, L. Mao, and W. Gong, "Service-aware adaptive link load balancing mechanism for software-defined networking," *Future Gener. Comput. Syst.*, vol. 81, pp. 452–464, Apr. 2018.

[31] R. Kumar, U. Venkanna, and V. Tiwari, "Opt-ACM: An optimized load balancing based admission control mechanism for software defined hybrid wireless based IoT (SDHW-IoT) network," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107888.

[32] J. Li, B. N. Silva, M. Diyan, Z. Cao, and K. Han, "A clustering based routing algorithm in IoT aware wireless mesh networks," *Sustain. Cities Soc.*, vol. 40, pp. 657–666, Jul. 2018.

[33] C. Wang and H. Y. Youn, "Entry aggregation and early match using hidden Markov model of flow table in SDN," *Sensors*, vol. 19, no. 10, p. 2341, May 2019.

[34] X. Tan, H. Zhao, G. Han, W. Zhang, and T. Zhu, "QSDN-WISE: A new QoS-based routing protocol for software-defined wireless sensor networks," *IEEE Access*, vol. 7, pp. 61070–61082, 2019.

[35] H. R. Faragardi, M. Vahabi, H. Fotouhi, T. Nolte, and T. Fahringer, "An efficient placement of sinks and SDN controller nodes for optimizing the design cost of industrial IoT systems," *Softw., Pract. Exper.*, vol. 48, pp. 1893–1919, Jun. 2018.

[36] K. Choumas, D. Giatsios, P. Flegkas, and T. Korakis, "SDN controller placement and switch assignment for low power IoT," *Electronics*, vol. 9, no. 2, p. 325, Feb. 2020.

[37] Y. Chai and X.-J. Zeng, "Load balancing routing for wireless mesh network with energy harvesting," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 926–930, Apr. 2020.

[38] M. Alishahi, M. H. Y. Moghaddam, and H. R. Pourreza, "Multi-class routing protocol using virtualization and SDN-enabled architecture for smart grid," *Peer–Peer Netw. Appl.*, vol. 11, no. 3, pp. 380–396, May 2018.

[39] J. Cui, Q. Lu, H. Zhong, M. Tian, and L. Liu, "A load-balancing mechanism for distributed SDN control plane using response time," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1197–1206, Dec. 2018.

[40] G. Li, X. Wang, and Z. Zhang, "SDN-based load balancing scheme for multi-controller deployment," *IEEE Access*, vol. 7, pp. 39612–39622, 2019.

[41] K. M. Modieginyane, B. B. Letswamotse, R. Malekian, and A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportunities for efficient network management: A survey," *Comput. Elect. Eng.*, vol. 66, pp. 274–287, Feb. 2018.

[42] B. B. Letswamotse, R. Malekian, C.-Y. Chen, and K. M. Modieginyane, "Software defined wireless sensor networks (SDWSN): A review on efficient resources, applications and technologies," *J. Internet Technol.*, vol. 19, no. 5, pp. 1303–1313, 2018.

[43] J. Wang, Y. Miao, P. Zhou, M. Hossain, and S. M. M. Rahman, "A software defined network routing in wireless multihop network," *J. Netw. Comput. Appl.*, vol. 85, pp. 76–83, May 2016.

[44] Y. E. Oktian, S. Lee, H. Lee, and J. Lam, "Distributed SDN controller system: A survey on design choice," *Comput. Netw.*, vol. 121, pp. 100–111, Jul. 2017.

[45] M. Coscia and F. M. H. Neffke, "Network backboning with noisy data," presented at IEEE Int. Conf. Data Eng. (ICDE), San Diego, CA, USA, 2017.

ELHAM HAJIAN is currently pursuing the Ph.D. degree with the Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran. Her research interests include the area of computer networks, including wireless sensor networks, software defined networks, performance modeling, and evaluation.

MOHAMMAD REZA KHAYYAMBASHI was born in Isfahan, Iran, in 1961. He received the B.Sc. degree in computer hardware engineering from Tehran University, Tehran, Iran, in 1987, the M.Sc. degree in computer architecture from the Sharif University of Technology (SUT), Tehran, in 1990, and the Ph.D. degree in computer engineering and distributed systems from the University of Newcastle upon Tyne, Newcastle upon Tyne, U.K., in 2006. He is currently working as an Associate Professor with the Department of Computer Architecture, Faculty of Computing Engineering, University of Isfahan, Isfahan. His research interests include distributed systems, computer networking, software define network (SDN), mobile computing, the Internet of Things (IoT), and edge/fog/cloud computing. He has successfully supervised graduated Ph.D. and M.Sc. students and also supervising Ph.D. students in his research area.

NASER MOVAHHEDINIA received the B.Sc. degree from Tehran University, Tehran, Iran, in 1987, the M.Sc. degree from the Isfahan University of Technology, Isfahan, Iran, in 1990, and the Ph.D. degree in electrical and computer engineering from Carleton University, Ottawa, Canada, in 1997. He is currently a Full Professor with the Faculty of Computer Engineering, University of Isfahan. His research interests include wireless networks, internet technology, and intelligent systems.