IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 14, 2019, accepted January 27, 2019, date of publication February 19, 2019, date of current version March 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900289

A Mobility-Supported Routing Mechanism in Industrial IoT Networks

JIAN CAO[1,2], XINGWEI WANG[1], MIN HUANG[3], AND XINHAO ZHOU[4]
[1]College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
[2]School of Computer Science and Technology, Beihua University, Jilin 132001, China
[3]College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[4]Software College, Northeastern University, Shenyang 110169, China

Corresponding author: Xingwei Wang (wangxw@mail.neu.edu.cn)

This work is supported by the National Key R&D Program of China under Grant No. 2017YFB0801701; the Program for Liaoning Innovative Research Term in University under Grant No. LT2016007; the National Natural Science Foundation of China under Grant No. 61872073, 1572123, and 61702090; the Major International(Regional) Joint Research Project of NSFC under Grant No. 71620107003; and the MoE and ChinaMobile Joint Research Fund under Grant No. MCM20160201.

ABSTRACT Nowadays, the Industrial Internet of Things (IIoT) has emerged as a promising paradigm for the future network, which intends to connect all kinds of objects, including smart home equipment and intelligent drive devices. Despite the fact that IIoT routing has already attracted a lot of attention from both academia and industry, current researches on IIoT routing cannot effectively solve the mobility problem in a self-adaptive and self-organized manner, because the number of IIoT devices connected is extremely large, and mobility is a very important feature of these IIoT devices. In this paper, a novel routing mechanism based on probability calculation and segment routing is proposed to solve the mobility and scalability problem in IIoT routing. On the one hand, probability calculation can address the extremely unbalanced load condition caused by mobility by forwarding packets to different routings with a certain probability. On the other hand, segment routing enables scalability and flexibility for packet forwarding, and it can also be used to bypass the over-loaded links, thus to achieve load balance. Finally, the simulation results indicate that the proposed mechanism not only solves the mobility problem effectively but also achieves a better performance in many aspects.

INDEX TERMS Internet of Things, routing, mobility, segment, mechanism.

I. INTRODUCTION

The Industrial Internet of Things (IIoT) can be regarded as a network which consists of many objects (e.g., smart vehicles and home appliances) embedded with sensors [1], software [2], actuators [3], etc. Among these objects, network connectivity is established. As an emerging and novel resolution, IIoT has been considered as one of the most promising technologies for driving the development of industrial Internet [4]. Firstly, IIoT is responsible for managing all the connected objects which offer services, collect data and enable application, etc. In this way, it can provide real-time tracking behavior and comprehensive decision-making analysis [5]. Secondly, IIoT enables the Machine-to-Machine (M2M) communication manner between devices. Compared with traditional communication method, M2M is more transparent and can offer higher communication quality for the physical devices connected [6]. Thirdly, IIoT enables the automation of control and management, since physical devices are connected and controlled digitally and centrally with wireless infrastructure [7]. Therefore, IIoT has greatly pushed the development of industrial Internet by offering the mainly required factors such as collecting, transmission and computing.

Despite the benefits brought by IIoT, it also introduces a lot of challenges, among which the mobility is a key problem to be solved. According to incomplete statistic, the number of IIoT devices grows at the speed of 31% year-over-year [8]. The overall quantity of IIoT devices has reached 8.4 billion at the end of 2017 and it is about to exceed 50 billion by 2020 [9]. With such a tremendous number of IIoT devices, huge amount of data will be generated and transmitted in

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

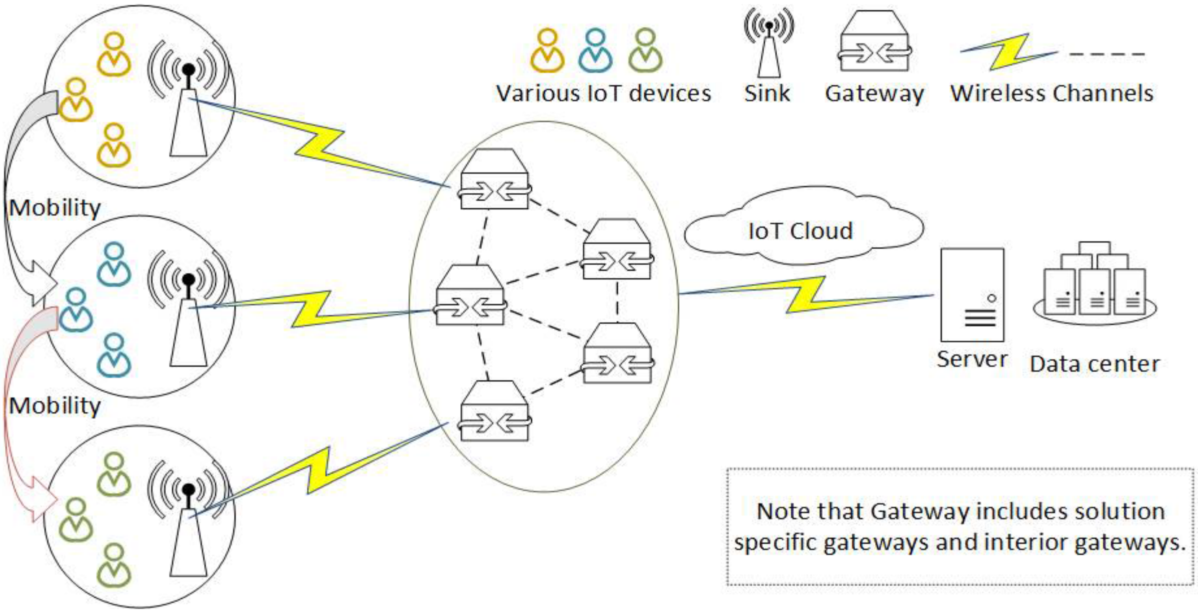


FIGURE 1. **The scenario with mobility enabled.**

the network. In this regard, congestion is easy to happen and may affect the network performance greatly [10]. What's worse is that these IIoT devices support mobility, which leads to unpredictable traffic patterns [11]. These two situations jointly make the routing in IIoT more complicated. Given this consideration, effective IIoT routing mechanisms should be proposed to solve the mobility problem.

Many work have already been proposed to solve IIoT routing. In particular, they can be classified into two categories. The first kind of work only intends to address the regular routing problem in IIoT, namely, delivering messages from IIoT devices to the destinations without considering the mobility features of IIoT devices, for example, [12]–[19]. These work lack enough scalability for general application in network. The second kind of work (for example, [20]–[24]) takes the mobility issue into consideration. Despite this, it usually focuses on managing instead of radically addressing the challenges faced, which makes them hard to be used in practice.

In this paper, the varying load conditions (which is typically caused by IIoT device mobility) is taken into consideration when addressing the IIoT routing problem, since the extreme imbalance network load may lead to worse situations. To fulfill such issue, the proposed mechanism is implemented from three aspects which are handover, forwarding and routing respectively. In particular, the handover module starts to work when there are devices moving from one sink to another. For the forwarding module, the forwarding probability is designed for packets, thus to average the suddenly increased traffic among different paths with different probabilities. Finally, the technology of segment routing [25] is applied to solve the routing problem in IIoT, due to its scalable and flexible characteristics.

The rest of this paper is organized as follows. Section II introduces the related work, while Section III presents the system framework and model. Section IV explains the proposed mechanism in detail. Section V presents the simulation results and the conclusion is summarized in Section VI.

II. RELATED WORK

Routing in IIoT has been studied for a long time and numerous works have been proposed from different aspects such as power consumption, delay, throughput, etc. Considering the main focus of this paper, the related work is discussed from two aspects, that is, IIoT routing without or with mobility considered.

A. IIoT ROUTING WITHOUT MOBILITY

To solve the routing problem in the scenario of Fig. 1, numerous categories of methods are proposed. The commonly used one leverages the naive routing to find the routing path from source to destination, which actually relies on the flooding strategy. For example, Sankaran and Sridhar [19] proposed an IIoT routing algorithm based on flooding and Markov chains, which intended to use protocol execution traces towards predicting power consumption. Actually, many routing strategies used in sensor networks can be applied in IIoT, due to the similarity between sensor network and IIoT. For example, RPL [26], the one used to enable bidirectional communication between sensors and access nodes, can also be applied to enable the bidirectional communication between IIoT devices and sink nodes. Many researches focus on integrating RPL with IPv6 and CoAP [27], such that RPL is widely recognized by many IIoT working groups (e.g., IETF IoT working group [28]).

Besides, there are also many other kinds of routing schemes used in IIoT, for example, hierarchical routing [29] and query-based routing [30]. Specifically, for hierarchical routing, it is suitable for nodes in different groups and each group may need a head to decide the data forwarding. One specific example is PEECR [31] which bases on a predictive data transferring strategy to make the decision. Guo *et al.* [32] also proposed a hierarchical routing in heterogeneous IIoT networks, which leveraged IPv6 routing protocol to maximize the resource utilization. On the other hand, for the query-based routing, it is a complete shift from the previous two routings (i.e., naive and hierarchical routing), and aims at disseminating data among nodes such that they can retrieve data from the other nodes [30]. Typical examples of query-based routing can be found in [33] and [34].

Usually, different IIoT devices are designed for different purposes. Thus, different routing algorithms will be required. For example, in order to maximize the reliability in IIoT networks, [35] presented a reliable routing protocol to guarantee the reliability during the processes of data collection and control command delivery, while [36] proposed another IIoT routing algorithm for load balance in IIoT. Moreover, Kotagi *et al.* [37] not only considered load balance, but also tried to make the routing adaptive to IIoT networks.

Nevertheless, the above discussed references do not take the mobility of IIoT devices into consideration, while the arbitrary movement of IIoT devices would inevitably affect the network performance. Thus, the next review of the related research on IIoT is the routing problem with mobility considered.

B. IIoT ROUTING WITH MOBILITY

Many research work try to address the mobility issue of IIoT devices by establishing their identifiers and locators which could be used to supervise and recognize devices easily. Based on such idea, [38] tried to separate the design of identifier and locator, where the identifier was used to recognize the IIoT devices and the locator was used to determine the routing path for packets sent by these devices in their moving process. Such separation was also leveraged in [39]. However, apart from the separated definition, it also built a mapping relationship between the identifier and locator for the purpose of providing more accurate services. Such mapping relationship can be usually established by using anchor nodes [40] or global domain name servers [41].

In order to reduce overhead and live up to the human mobility pattern, [42] proposed a routing protocol which only delivered messages to the devices that were expected to forward them to the destinations. Meanwhile, [43] and [44] both took the mobility into consideration when designing their IIoT routing mechanisms. However, [43] focused on researching the end-to-end routing for mobility management, while [44] focused on performance analysis under different mobility models. Furthermore, [45] was based on wireless mesh networks to simulate the heterogenous terminals and to realize the IIoT services, while [46] intended to address the energy efficiency issue caused by IIoT mobility.

Despite this, the above mentioned research work mainly focus on managing all the mobile IIoT devices. Thus, they actually could not be applied to address the IIoT routing across different areas (or domains). Besides, these work are hard to serve the huge number of the mobile IIoT devices with small memory size and overhead. In this way, a mechanism is proposed in this work to address the IIoT routing with mobility across the access, transmission and core IIoT networks considered.

III. SYSTEM FRAMEWORK AND PROBLEM FORMULATION

In order to clearly explain this work, a simple IIoT scenario is firstly presented, in which the mobility is supported. Based on this scenario, the proposed system framework is next introduced, in which different mechanisms are developed to support IIoT routing with mobility.

A. SYSTEM FRAMEWORK

Considering the wide recognition of the three-layer IIoT architecture, it is also used in the scenario of this work as illustrated in Fig. 1. As shown, it consists of three main parts which are sensing layer, network layer and application layer from the left to the right of Fig. 1. Different from the traditional sensing layer of industrial Internet that contains sensors monitoring objects and their surrounding environments, this sensing layer of the proposed framework is composed of many IIoT devices including sensors, smart phones, smart vehicles, and even some 5G enabled objects (e.g., wearable devices). In particular, these IIoT devices in near vicinity using a common communication technology are grouped into one cluster which is connected to the corresponding solution specific gateway for data routing. Besides, due to the new mobility feature of many IIoT devices, they can easily move from one cluster to another. This naturally leads to the operation of leaving or joining cluster, which need to be handled carefully. The network layer is composed of many gateways which receive data from these IIoT devices and forward data to their destinations. With respect to the network layer, there are already many kinds of routing mechanisms for data transmission, for example, naive routing [28], hierarchical routing [29], query-based routing [30], etc. However, these existing mechanisms are not capable of handling the new features (e.g., mobility) enabled by IIoT. Thus, new mechanisms with new features should be developed for the IIoT routing. The application layer actually includes many servers or data centers which store the data received from the network layer. As artificial intelligence gets more and more popular, the huge amount of data stored will be used to perform model training and complex analysis for network optimization. Despite this, this part is out of the scope of this paper.

Based on the scenario of Fig. 1, the proposed framework is shown in Fig. 2. As depicted, it is composed of the following

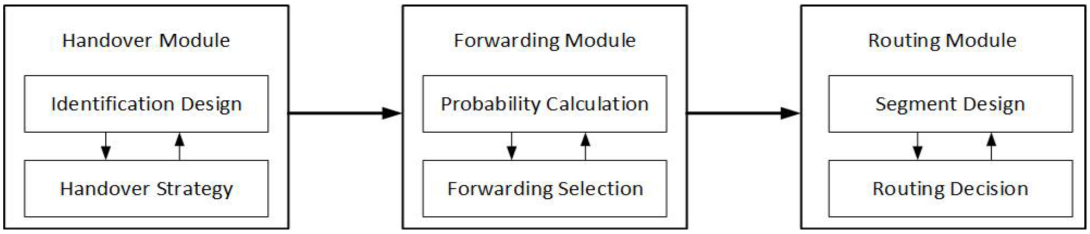


FIGURE 2. The proposed system framework.

three modules, that is, Handover Module (HM), Forwarding Module (FM) and Routing Module (RM). Among them, HM is responsible for designing the IIoT device identifications and managing the mobility feature of IIoT devices. FM is responsible for load balance calculation and data forwarding. In particular, the load balance calculation is used to handle the situation when the traffic from one cluster is huge. Thus, a lot of IIoT devices move to this area and join this cluster. RM is responsible for making routing decisions that mobility is taken into consideration.

B. PROBLEM FORMULATION

1) SENSING LAYER

Since various IIoT devices communicate with each other through wireless communication technologies, it is assumed that there are k kinds of wireless communication technologies and denote them by the set notation $T = \{t_1, t_2, \ldots, t_k\}$. Correspondingly, the transmission range of each technology is denoted by the set notation $R = \{r_1, r_2, \ldots, r_k\}$, that is, r_j is the transmission range of t_j, $\forall j \in [1, k]$.

Besides, the sinks are very important for IIoT devices to access the Internet. Therefore, it is assumed that there are m kinds of sinks which are denoted by the set notation $S = \{s_1, s_2, \ldots, s_m\}$. However, in order to formulate the mobility situation and avoid massive packet dropping situation in IIoT, it is assumed that each sink is capable of caching all the messages received.

Particularly, each sink has to support at least one kind of these wireless communication technologies before it can actually collect data and forward them to the Internet. To formulate the relationship between them, a binary variable is introduced as follows:

$$X_{t_j}^{s_i} = \begin{cases} 1, & s_i \text{ supports } t_j. \\ 0, & \text{otherwise.} \end{cases} \quad \forall i \in [1, m], j \in [1, k] \tag{1}$$

All the sinks are uniformly distributed in the local area. For all the IIoT devices in this area, they are denoted by the set notation $U = \{u_1, u_2, \ldots, u_p\}$. According to the location of these IIoT devices, they are connected to different sinks. Such relationship is formulated as follows:

$$Y_{u_j}^{s_i} = \begin{cases} 1, & u_j \text{ is connected to } s_i. \\ 0, & \text{otherwise.} \end{cases} \quad \forall i \in [1, m], j \in [1, p] \tag{2}$$

2) NETWORK LAYER

Assuming that there are n gateway nodes in the network layer, they are denoted by the set notation $G = \{g_1, g_2, \ldots, g_n\}$. Note that, there are both solution specific gateways and interior gateways in the network layer. However, for convenience, these gateways are regarded as the same category in this model and assume that all these gateways can support the technologies of sinks, that is, $T_g = T$ where T_g represents the set of technologies supported by any gateway $\in G$.

The distance between any sink s_i and any gateway g_j is denoted by $dis_{g_j}^{s_i}$. Usually, there should be at least one gateway for accessing Internet within the transmission range of s_i. In addition, given the gateway g_j, since it is assumed to support all kinds of technologies, the maximum transmission range of g_j is denoted by $maxD_{g_j}$ as follows:

$$maxD_{g_j} = \max\{r_1, r_2, \ldots, r_k\} \tag{3}$$

Within the distance of $maxD_{g_j}$, there should be at least another one gateway before g_j can actually forward data out.

3) APPLICATION LAYER

The application layer is composed of many servers and data centers. In particular, this layer is to supply services and store data for deep analysis. Each server or data center is connected to one gateway from the network layer. Therefore, it can be regarded as the edge node of the network layer.

Despite the importance of this layer, it is actually outside the scope of this paper, because the target in this sensing layer is to address the mobility issue occurred.

4) OBJECTIVE

The mobility of IIoT devices would result in local network bottleneck which largely affects the throughput which is a very important indicator to evaluate the network performance. Therefore, the objective is to maximize the overall network throughput in this paper. Reviewing the proposed framework, it is composed of three different layers (or domains). However, due to the unpredictable load condition in network environment, many network nodes (e.g., sinks or gateways) may be overloaded, such that they become the bottleneck for maximizing the network throughput. In this research, such awkward situations are avoided by making a detour. Specifically, the obvious and large network bottlenecks may occur between sinks (i.e., sensing layer) and edge gateways (i.e., network layer) as well as between edge gateways and servers (i.e., application layer). Therefore, by formulating these potential bottlenecks, the objective is presented. Let's denote such two bottlenecks by $B_{s,g}$ and $B_{g,s}$ respectively.

In order to formulate $B_{s,g}$, it is assumed that the data sending rate of any IIoT device u_j at time instance t is denoted by $u_{j,t}^{rate}$. Then, the amount of traffic sent out by these IIoT devices (denoted by $Throughput(users)$) can be formulated as follows:

$$Throughput(users) = \sum_{s_i \in S} \sum_{u_j \in U} \sum_{t} Y_{u_j}^{s_i} \times u_{j,t}^{rate} \tag{4}$$

However, the practical situation should be taken into consideration. Specifically, these IIoT devices may generate data in an infinite manner, while on the contrary, the data processing capacities of sinks are limited. Thus, the maximum capacity of all these sinks is denoted by $Throughput(sinks)$ and it is formulated as follows:

$$Throughput(sinks) = \sum_{s_i \in S} \sum_{t_j \in T} \sum_{t} X_{t_j}^{s_i} \times maxC_{t_j}, \tag{5}$$

where $maxC_{t_j}$ indicates the maximum capacity of using technology t_j.

Based on equations (4) and (5), the first bottleneck can be formulated as follows:

$$B_{s,g} = \min\{Throughput(users), Throughput(sinks)\} \tag{6}$$

Similarly, to formulate $B_{g,s}$, the edge gateways are extracted from G. It is assumed that these edge gateways to the application layers are denoted by $\{eg_1, eg_2, \ldots, eg_q\}$ respectively, where $n > q$. Then, the other gateways are called intra-gateways and denoted by $\{ig_1, ig_2, \ldots, ig_{n-q}\}$. Assuming the amount of data transmitted from ig_i to eg_j using the technology t_k at the time instance t is denoted by d_{ig_i,eg_j,t_k}^{t}, then it follows that:

$$Throughput(ig) = \sum \sum d_{ig_i,eg_j,t_k}^{t} \times Z_{ig_i,t_k}^{t}, \tag{7}$$

where $Z_{ig_i,t_k}^{t} \in \{0, 1\}$ is a Boolean variable. In particular, $Z_{ig_i,t_k}^{t} = 1$ means that at the time instance t, only ig_i sends data using t_k in its transmission range, such that the transmission will not be interrupted.

Then, the maximum processing capacity of the edge gateways is denoted by $Throughput(eg)$ as follows:

$$Throughput(eg) = \sum_{s} \sum_{t} \max\{maxC_{t_1}, \ldots, maxC_{t_k}\} \times t \tag{8}$$

With equations (7) and (8), $B_{g,s}$ is presented as follows:

$$B_{g,s} = \min\{Throughput(ig), Throughput(eg)\} \tag{9}$$

So far, based on equations (6) and (9), two bottlenecks are obtained in network across different domains. According to the bucket theory, the overall objective is formulated as follows:

$$\textbf{Maximize} : \min\{B_{s,g}, B_{g,s}\} \tag{10}$$

However, in order to achieve the above objective, several constraints should be satisfied. Firstly, given any sink, it supports one technology based on practical situations.

$$\sum_{j \in [1,k]} X_{t_j}^{s_i} = 1, \quad \forall i \in [1, m] \tag{11}$$

Secondly, each sink has at least one gateway within its transmission range, and each gateway has at least one another gateway within its transmission range, such that the communication is enabled. Specifically, there is at least one $g_{j'} \in G$ making the following in-equations true.

$$\begin{aligned} \sum_{j \in [1,k]} X_{t_j}^{s_i} \times r_j &\geq dis_{g_{j'}}^{s_i} \\ maxD_{g_j} &\geq dist_{g_{j'}}^{g_j} \end{aligned} \tag{12}$$

Thirdly, each sink should be accessed by different IIoT devices.

$$\sum_{j \in [1,p]} Y_{u_j}^{s_i} \geq 0, \quad \forall i \in [1, m] \tag{13}$$

Finally, the maximum allowed traffic transmission should not be exceeded for both sinks and edge gateways. Given any sink s_i and technology t_j, it follows that:

$$\sum_{u_{j'} \in U} \sum_{t} Y_{u_{j'}}^{s_i} \times u_{j',t}^{rate} \leq \sum_{t_j \in T} X_{t_j}^{s_i} \times maxC_{t_j} \tag{14}$$

IV. THE PROPOSED MECHANISM

As explained in the system framework, the proposed mechanism consists of three modules which are HM, FM and RM respectively. Each module has specific responsibilities, and the inner-relationship among these three modules is shown in Fig.2. In particular, HM solves the mobility issue when IIoT devices move from one area to another, and makes sure that these devices can quickly connect to the new sink. FM forwards the data from IIoT devices and RM makes the routing decisions for traffic. The three parts are explained in the following subsections.

A. HANDOVER

The IIoT devices can easily move from one area to another. Thus, proper handover scheme should be well designed and it is presented from the perspectives of identification and strategy respectively.

The Multiple Protocol Label Switching (MPLS) is leveraged to design the identifications. Importantly, the IIoT devices and sinks are mainly considered, since they are closely related to mobility feature. For each IIoT device, it is given a unique number (or label in MPLS) denoted by *nodeID*. The same applies to each sink, where it is denoted by *sinkID*. The corresponding definition is as follows:

Definition 1: nodeID, sinkID are unique identifiers for recognizing each IIoT device and each sink in the same domain. In particular, they are also values that can be used as the label for using in forwarding and routing processes.

In this way, denoting the relationship between *nodeID* and *sinkID* by their joint identification (*JID*) and it can be built as follows:

$$JID = nodeID \times 10^{\lambda} + sinkID, \tag{15}$$

where λ exceeds the maximum length of *sinkID*. Thus, only storing one value can record the information of both IIoT device and sink for the identification during the forwarding and routing processes.

Besides, several signals are needed to identify the mobility characteristics of IIoT devices, which include detection, notification and updating. Assuming that the IIoT device is connected to one sink, when it is leaving the area of this sink, the corresponding signal will weaken, which can be detected by the sink. Thus, it will send a leaving detection message to the remote controller. After that, the controller sends notification message to the near sinks. When one near sink detects the arriving of this IIoT device, it returns an arriving detection message to the controller. Once the leaving-arriving match is captured by the controller, it then sends the updating messages to both sinks. For the pre-sink, it deletes the item related to this device. For the post-sink, assuming that the device moves from s_i to s_j, then the joint identification value should be changed as follows:

$$JID = JID - sinkID_{s_i} + sinkID_{s_j} \tag{16}$$

The handover of IIoT devices from one sink to another belongs to the horizontal handover and in this stage, only the intra-domain handover is considered in this work. In particular, all these sinks are assumed to be in the same autonomous system. For example, as shown in Fig. 3, when the IIoT device 1001 moves from one area of sink 601 to another sink of 602, the joint identification changes from 1001601 to 1001602, while the other situations all remain the same. Therefore, by using this kind of identification design, the goal can be achieved with minimum overhead.

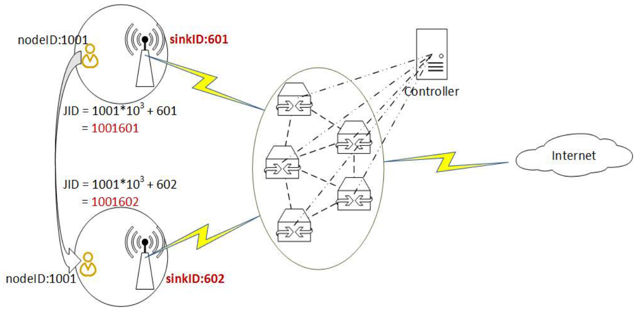


FIGURE 3. The handover example.

The MPLS protocol and segment routing scheme are both leveraged to design the handover strategy. In addition, by inserting the routing information into packet headers, it is easy to determine or change the routing paths by parsing and modifying the packet header. In this regard, no matter how the device moves, we can easily forward it to the destination according to its header, such that the routing scalability can be solved. Besides, due to such flexible design, this process can be finished within linear time. Therefore, no matter how much traffic arrive, they can be handled quickly, such that the traffic scalability can be solved.

B. FORWARDING

In this section, the main focus is to design the forwarding strategy. However, considering the unbalanced load situation that may be caused by mobility issue, a probability calculation is present firstly to average the load in network.

For each sink s_i, there may be many gateways within its capacity reach. One obvious situation is that if the traffic received by s_i can be forwarded to all the reachable gateways, load balance in network could be achieved. However, this will achieve extra increasing overhead, because forwarding one traffic to one more gateways would increase the energy consumption and the longer the propagation distance is, the more power consumption will be. In this way, a balance between them should be reached. In this section, it is required that the network imbalance should not be caused when minimizing the number of gateways connecting to each sink.

The amount of traffic received by s_i is denoted by Γ_{s_i} which will be used to evaluate the network load. To make it work, a threshold is firstly defined for s_i, denoted by ξ_{s_i} which is used to calculate the forwarding probability. In particular, ξ_{s_i} indicates the maximum amount of traffic that cannot be forwarded to one gateway under the normal condition. In order to choose the closet gateway at the first place, all these reachable gateways should be sorted according to their distances to the sink s_i. Assuming that these gateways are sorted and denoted by $\{g^i_{j_1}, g^i_{j_2}, \ldots, g^i_{j_\mu}\}$ respectively, the corresponding distances are formulated by the set of $\{dis^{s_i}_{g_{j_1}}, dis^{s_i}_{g_{j_2}}, \ldots, dis^{s_i}_{g_{j_\mu}}\}$ where $dis^{s_i}_{g_{j_1}} \leq dis^{s_i}_{g_{j_2}} \leq \cdots \leq dis^{s_i}_{g_{j_\mu}}$.

Based on the defined variables, the number of gateways (denoted by *NG*) that should be used for traffic forwarding can be calculated as follows:

$$NG = \lceil \frac{\Gamma_{s_i}}{\xi_{s_i}} \rceil, \tag{17}$$

where the traffic should be forwarded to the first *NG* gateways in $\{g^i_{j_1}, g^i_{j_2}, \ldots, g^i_{j_\mu}\}$.

The probabilities of forwarding traffic to different gateways can be different. However, for convenience, the traffic is forwarded to the first *NG* gateways with the same probability as follows:

$$\frac{1}{NG} \tag{18}$$

Nevertheless, it is noticed that the value of *NG* may exceed the number of all the reachable gateways (i.e., μ). In this case, equations (17) and (18) will not work correctly. In order to solve this kind of awkward situation, the value of the threshold ξ_{s_i} is increased by 5% repetitively until the following case is satisfied.

$$NG \leq \mu \tag{19}$$

Despite this, increasing the threshold is an emergency operation when the network load is extremely high. Once the network load decreases, the value of ξ_{s_i} should be recovered to its original one. Now, based on the forwarding probability calculated and the forwarding gateways selected, the received traffic of s_i can be forwarded out to the corresponding gateways, then following the forwarding information table to the destinations. The corresponding pseud-code is given in Algorithm 1 as follows:

Algorithm 1 Forwarding

```
Input: $s_i$, $\Gamma_{s_i}$, $\xi_{s_i}$, $G$
Output: Forward packets
System Initialization based on the concepts of MPLS and segment routing;
$Flag \leftarrow$ Check if the arriving packets of $s_i$ suffering from the mobility issue;
if $Flag$ is true then
    Change the $JID$ attached with the packets according to equation (16) and rebuilding the network connection between devices and sinks;
end
$GA \leftarrow$ Sort gateways according to the distances between them and the corresponding sink;
$NG \leftarrow \infty$;
while $|NG| > |GA|$ do
    $NG \leftarrow \lceil \frac{\Gamma_{s_i}}{\xi_{s_i}} \rceil$;
    if $|GA| \geq |NG|$ then
        Forward traffic to the first $NG$ gateways in $GA$;
        Break;
    end
    $\xi_{s_i} \leftarrow \xi_{s_i} + \xi_{s_i} * 5\%$;
end
return TURE;
```

C. ROUTING

In order to design routing strategy, a unique number or label is given to each gateway as *gatewayID* in the first place. Based on the uniquely defined *nodeID*, *sinkID* and *gatewayID*, the concept of segment routing is adopted to jointly take these labels into consideration and is used to form another important definition before they are actually used for routing, as follows:

Definition 2: segment indicates an instruction which can be executed by nodes on the arriving packets, for example, forwarding packets to a specific application before reaching destination based on the shortest path. In particular, by reparsing the JID calculated by equation (15), the information of the sink and the nodes that packets will traverse are both obtained. Thus, JID can be regarded as a segment. Many segments that work on the same packet can constitute a *segment list*, and the top one segment of this list is called the active segment that is used by the receiving router to process the packet.

In order to make the segment list work, several operations on this list should be set up as follows:

- PUSH: the operation that inserts a segment to the segment list. Initially, the segment is attached at the tail of this list to follow the first come first serve rule.
- POP: the top segment on the list is usually the active one. Once it is been processed, this operation would remove it and let the next segment become active.
- FORWARD: there are many hops before this packet reaches the next segment corresponding node. Under this condition, the active segment is still not finished, and the packet should be forwarded to the next hop before executing the next segment.

Based on the above definitions, an example is given to show how these operations actually work in Fig. 4. The packet header is encoded with a segment list which has four segments that are PUSHed at the initial stage. At first, node A (i.e., 101) is active. Before the packet reaches the other segment node (i.e., node c, 103), the FORWARD operation is used. When the packet arrives node c, node A is popped out and node C becomes active. The same procedure repeats until this packet reaches the destination node I.

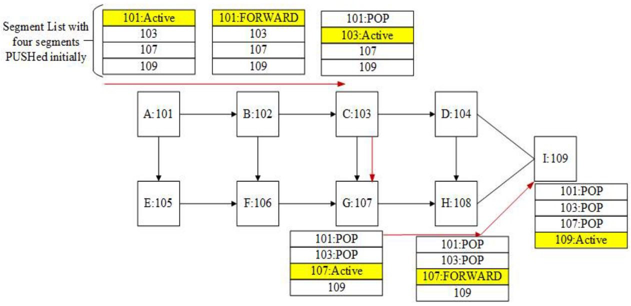


FIGURE 4. The example of segment routing.

According to the above illustration, it is aware that segments are usually used to steer traffic through specific network devices by encoding the segment list into the packet headers. However, considering the goal of this work (shown by equation (10)), this kind of design is leveraged to avoid the network bottlenecks and thus maximizing the network throughput. For example, as shown in Fig. 5, the line thickness indicates the link load. Assuming that the original segment list would be like this <Router1, Router2, Router3, Router4>, then it has to pass the heavy-loaded link between Router2 and Router3. PUSHing another segment between Router2 and Router3, that is, <Router1, Router2, Router5, Router3, Router4>, then the crowd link can be avoided, which naturally increases the throughput.

The shortest path first algorithm is adopted to calculate the default routing path and it is allowed that a midpoint-

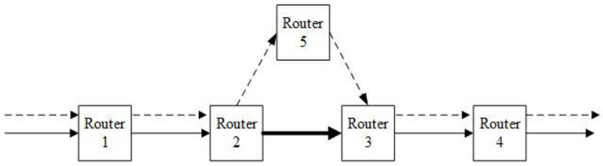


FIGURE 5. The example of using segment to avoid bottleneck.

segment can be inserted to the routing path to avoid network bottleneck. Therefore, the principle of the proposed algorithm is to override the path decisions made by the shortest path first algorithm based on network conditions. To implement it, a link state set is firstly defined and denoted by $\mathbb{L} = \{l_1, l_2, \ldots, l_{|\mathbb{L}|}\}$. In particular, the actual load on each link in $\mathbb{L}$ exceeds the 95% of its maximum capacity. It is aware that the controller is assumed to have a global network view, thus to make sure that the set of $\mathbb{L}$ contains all the heavy-loaded links as defined. In order to make the process clear, the routing paths consisting adjacent links of packets are firstly presented by using the default routing algorithm, that is, the shortest path algorithm. The target of this work is to partially override these routing paths. Therefore, it is assumed that the default routing path is already calculated. Now, taking the packets from the sink s_i, the intact routing path is denoted by $L^{s_i} = \{l_1^{s_i}, l_2^{s_i}, \ldots, l_{|L^{s_i}|}^{s_i}\}$ with the following constraint satisfied.

$$l_p^{s_i} \cap l_{p+1}^{s_i} \neq \emptyset, \tag{20}$$

where $p \in [1, |L^{s_i}|)$.

Equation (20) actually indicates that the end-node of $l_p^{s_i}$ is actually the begin-node of $l_{p+1}^{s_i}$. Based on this, a checking process is executed, that is, whether this intact routing path has suffered from any network bottleneck, by executing the following in-equation.

$$\mathbb{L} \cap L^{s_i} \begin{cases} \neq \emptyset, & \text{There exist over-loaded links in } L^{s_i}. \\ = \emptyset, & \text{otherwise.} \end{cases} \tag{21}$$

Once the over-loaded links in L^{s_i} have been detected, a midpoint node should be selected and the corresponding segment should be PUSHed into the original segment list to make a bypass. Assuming that the over-loaded link is $l_p^{s_i}$ which has start-node as $g_{p,start}$ and end-node as $g_{p,end}$. It is easy to determine the position of this link in the routing path through a search process. Thus, let's assume the corresponding segment list to be $< \ldots, g_{s,p}, g_{s,p+1}, \cdots >$ where $l_p^{s_i}$ locates between $g_{s,p}$ and $g_{s,p+1}$. In this way, this segment list is updated by PUSHing the start-node and end-node of $l_p^{s_i}$ as follows:

$$< \ldots, g_{s,p}, g_{p,start}, g_{p,end}, g_{s,p+1}, \cdots > \tag{22}$$

This operation has not influenced the essence of the original segment list which is supposed to go through the two nodes $g_{p,start}$ and $g_{p,end}$. However, the next move is to find a midpoint to bypass the over-loaded link between $g_{p,start}$ and $g_{p,end}$. Let's denote this midpoint by $g_{p,mid}$. Then the segment list is updated as follows:

$$< \ldots, g_{s,p}, g_{p,start}, g_{s,mid}, g_{p,end}, g_{s,p+1}, \cdots > \tag{23}$$

Therefore, the last step is to calculate $g_{s,mid}$ according to $g_{p,start}$ and $g_{p,end}$. Firstly, calculating all the candidate nodes for $g_{s,mid}$, and storing them in one set denoted by CN, as follows:

$$CN = G - \{\ldots, g_{s,p}, g_{p,start}, g_{p,end}, g_{s,p+1}, \ldots\} \tag{24}$$

For each candidate node of $g_{s,mid}$, the hops between it and $g_{p,start}$ or $g_{p,end}$ is calculated using the shortest path algorithm. Hence, the function $hops(*, *)$ is used to indicate the number of hops between any two nodes. Then, $\forall g_{s,mid}$, the extra number of hops caused by using it as a midpoint is calculated as follows:

$$hops(g_{p,start}, g_{s,mid}) + hops(g_{s,mid}, g_{p,end}) \tag{25}$$

Based on the value calculated by equation (25), all the candidate nodes are sorted in CN in an ascending order. However, to give a clear presentation, the sorted CN should be re-expressed in another form as follows:

$$CN = \{cn_1, cn_2, \ldots, cn_{|CN|}\}, \tag{26}$$

where $cn_1 \leq cn_2 \cdots \leq cn_{|CN|}$.

In order to avoid bottleneck, it is better to use as few hops as possible. Hence, each candidate in the order of $\{cn_1, cn_2, \ldots\}$ is checked by the following constraints:

$$\begin{aligned} &MaxT(g_{p,start}, cn_i, g_{p,end}) < MaxT(g_{p,start}, g_{p,end}) \\ &hops(g_{p,start}, cn_i) + hops(cn_i, g_{p,end}) < MaxHops \end{aligned} \tag{27}$$

where $MaxT\{*\}$ is a function that measures the maximum traffic throughput along the path traversing specific nodes, while $MaxHops$ indicates the maximum hops allowed to pass.

The first one candidate satisfying the above constraints will be selected as the midpoint. However, if no suitable solution is found, the original routing path remains the same, since spending more overhead to address the current situation may not be worthy and necessary. The entire routing mechanism can be obtained based on the above main parts. Now, the overall process is described in the form of pseudo-code in Algorithm 2 as follows:

Furthermore, in order to present a better clarification of the proposed mechanism, a flowchart is drawn to show their working process in Fig. 6.

D. ANALYSIS

The proposed mechanism is composed of three main parts. Therefore, its time complexity and space complexity can be analyzed from the perspective of these different parts. The following theorem is given.

Theorem 1: the time complexity and space complexity of the proposed mechanism are $O(N * L)$ and $O(N + L)$ respectively.

Algorithm 2 Routing

```
Input: $s_i$, $L^{s_i}$, $\mathbb{L}$
Output: Segment list
$SL$ ← Initialize the segment list to empty;
$L^{s_i}$ ← Calculate the routing path according to the default shortest path algorithm for packets arriving $s_i$;
$OL \leftarrow L^{s_i} \cap \mathbb{L}$;
$k \leftarrow 0$;
$m \leftarrow 0$;
if $OL \neq \emptyset$ then
    for $k \leq |OL|$ do
        $CN$ ← Calculate the candidate nodes to bypass the overloaded link $OL[k]$ according to equation (24);
        for $m \leq |CN|$ do
            $sumHop(CN[m]) \leftarrow hops(OL[k].start, CN[m]) + hops(CN[m], OL[k].end)$;
            $m++$;
        end
        Sort all the candidate nodes in $CN$ in an ascending order based on the number of hops calculated for each candidate node (i.e., $sumHop(CN[m])$);
        $m \leftarrow 0$;
        for $m \leq |CN|$ do
            if $sumHop(CN[m]) < MaxHops$ then
                $midPoint$ ← Set $CN[m]$ as the middle node to bypass this overload link $OL[k]$;
                Break;
            end
        end
        PUSH the segments $OL[k].start$, $midPoint$, $OL[k].end$ into $SL$ in order;
        $k++$;
    end
end
Insert the segment list $SL$ into the packet headers;
return segment list $SL$;
```

Proof: the proposed mechanism is composed of three parts which are handover, forwarding and routing respectively. For the first one, it only needs to parse and modify the packet header information. Therefore, given the number of packets, the corresponding handover operation can be executed in constant time complexity, denoted by $O(1)$. However, to fulfill such fast operation, extra spaces are required to store labels indicating nodes and links, which will be used to guide the forwarding and routing, and it requires $O(N + L)$ space complexity, since all the nodes and links should have a unique label in the system. For the second one, it is a iteration process until the constraint (19) is satisfied. Denoting the number of iterations by x, then the constrain can be re-expressed as $\lceil \frac{\Gamma}{\xi * 1.05^x} \rceil \approx N$. Solving this equation, we can obtain that $x \approx \log_{1.05}^{\frac{\Gamma}{\xi * N}}$. Therefore, the time complexity of the second part is $O(\log_{1.05}^{\frac{\Gamma}{\xi * N}})$. However, given the amount of traffic and the threshold, $O(\log_{1.05}^{\frac{\Gamma}{\xi * N}}) \leq O(N)$. Besides, to carry out the forwarding, extra spaces are required to store the segment list which needs the space complexity of $O(N)$, since the segment list can include at most N nodes. For the last one, the corresponding time complexity and space complexity are $O(N * L)$ and $O(N)$ respectively according to Algorithm 2.

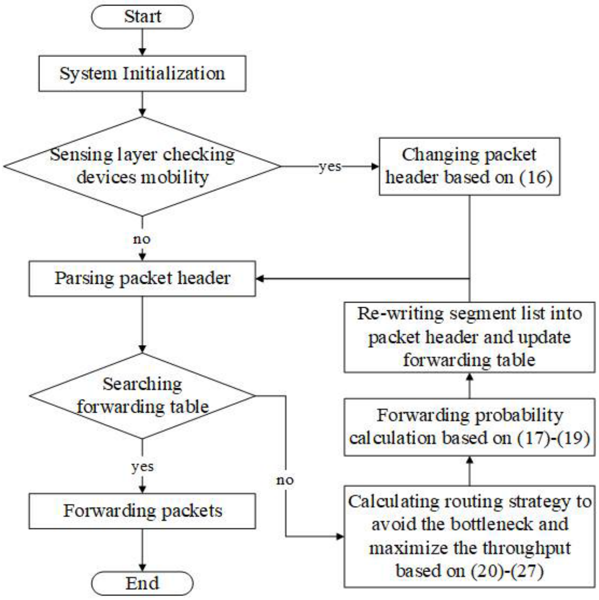


FIGURE 6. **A flowchart of the proposed mechanism.**

Since they formulate the proposed mechanism, the overall time complexity is actually $O(1) + O(N) + O(N * L) \approx O(N * L)$. Similarity, the overall space complexity is $O(N) + O(N) + O(N + L) = O(N + L + 2) \approx O(N + L)$.

V. PERFORMANCE EVALUATION

A. SETUP

In this simulation, the total IIoT environment is set to be a 40*40 m^2 square for the sensing layer. This square is divided into small areas in 4*4 m^2. Hence, there are $\frac{40*40}{4*4} = 100$ small areas. Each area has a sink to support different wireless communication technologies and to access the IIoT devices, and it is assumed that each sink can support 5 kinds of wireless communication technologies, that is, $k = 5$. In addition, the number of IIoT devices is set to be 1000, which indicates that there are 10 devices in each area in average. For the network layer, it is composed of many routers which follow the shape of a complete ternary tree with the height of 6, and each sink can access the edge routers within its transmission range.

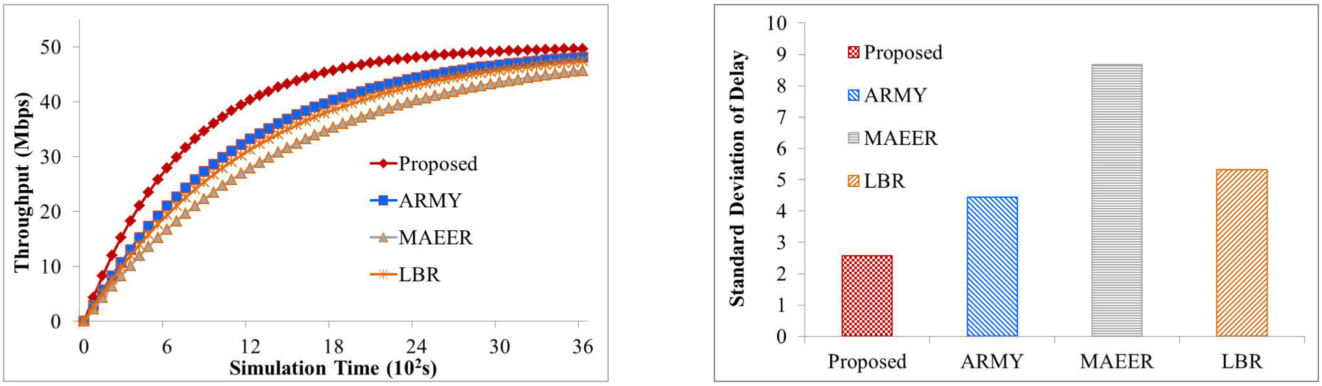
The overall simulation will sustain in one hour, during which each IIoT device is given a probability for generating [20,100] Kbytes data randomly. The occurring frequency of data generation is set to 10 seconds. In particular, in order to simulate the over-loaded situations in practical world, it is allowed that IIoT devices can move in different patterns. In this simulation, two walking patterns are tested, which are random and cascading [47] respectively. In addition, the node distribution is set to be uneven at the beginning. The proposed mechanism is compared with the other three approaches which are Load Balanced Routing (LBR) [36], Adaptive Routing using Multi-technologY (ARMY) [37] and Mobility Aware Energy Efficient Routing (MAEER) [46] respectively. We re-implement these benchmark approaches and test them in the setup environment for the following comparison.

B. RESULTS

The performance benefits of the proposed mechanism with respect to LBR, ARMY and MAEER are studied in this section. Since the objective of this work is to make the network load more balance, the throughput is selected as the first evaluation metric. In particular, it is studied and evaluated under the four mechanisms, and the load balance situation is analyzed based on the results achieved.

The corresponding results of throughput are shown in Fig. 7 and Fig. 8 in terms of the random and cascading walking models respectively. It is observed that one common phenomenon in both Figs. 7(a) and 8(a). Specifically, the throughput increases until it finally reaches to a stable condition. However, it is also noticed that the overall performance increase trend of Fig. 7(a) is better than that of Fig. 8(a). The main reasons are that the IIoT device walking pattern in Fig. 7(a) is random, such that the network load can be uniformly distributed. Meanwhile, the cascading walking pattern is used in Fig. 8, where all the IIoT devices follow the cascade distribution, such that the number of IIoT devices in different areas varies a lot and results in the unbalanced situation.

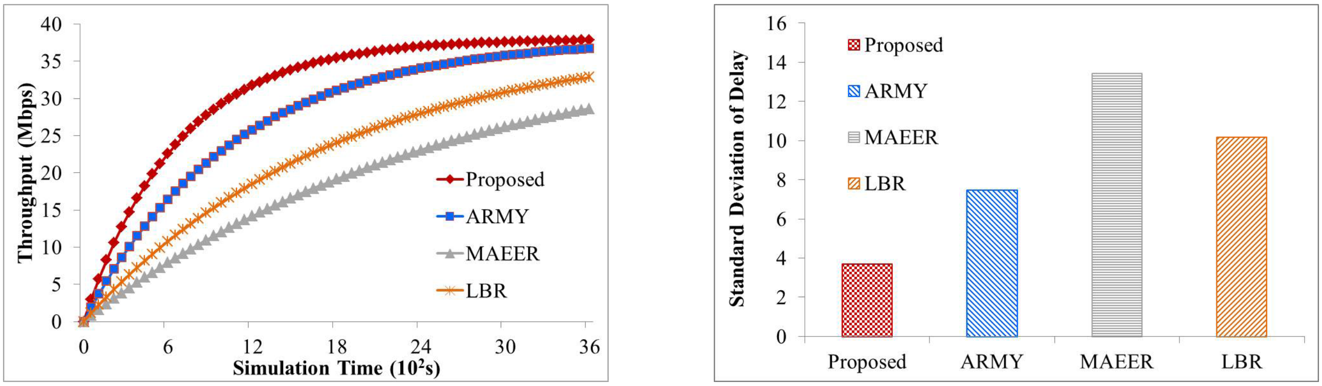
Let's look closer in Fig. 7(a) and Fig. 8(a), and it is discovered that the proposed mechanism has higher throughput than the other compared approaches, because the proposed mechanism can change the forwarding probabilities to different interfaces dynamically according to the current network load. In this way, the proposed mechanism can forward data to the destination using multiple paths. However, the data forwarding in MAEER generally follows a particular path, thus achieving lower throughput. For the approaches of LBR and ARMY, they both intend to achieve load balance in network. LBR forwards data through a changing path to the destination to achieve load balance. Despite this, LBR still bases on one path for data forwarding. Thus, the achieved throughput is lower than the proposed approach. ARMY uses



(a)

(b)

FIGURE 7. Throughput and the standard deviation using random walking model. (a) Throughput. (b) Standard deviation of throughput.



(a)

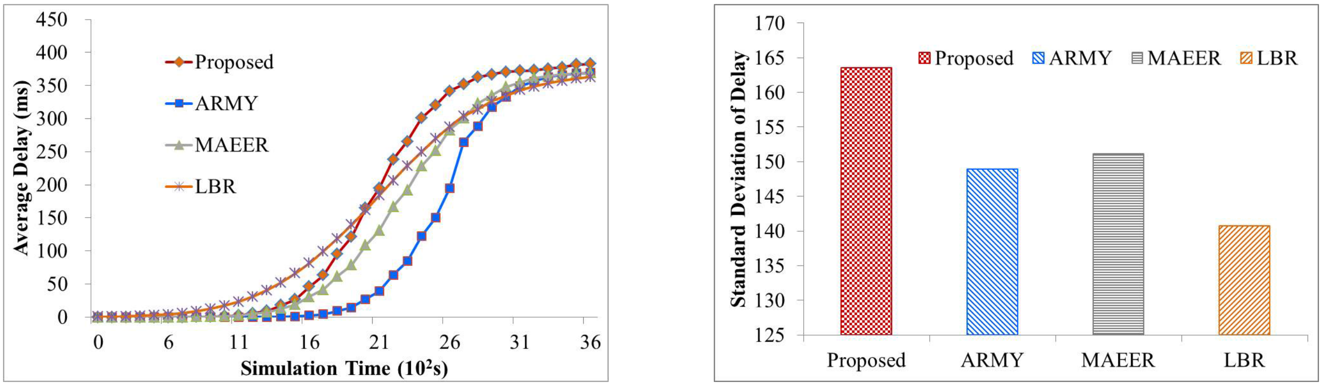
(b)

FIGURE 8. Throughput and the standard deviation using cascading walking model. (a) Throughput. (b) Standard deviation of throughput.

multiple paths for data forwarding based on particular probability. Comparatively, the proposed approach can change the forwarding probabilities according to the varying network load situation, which is more efficient and achieves higher throughput.

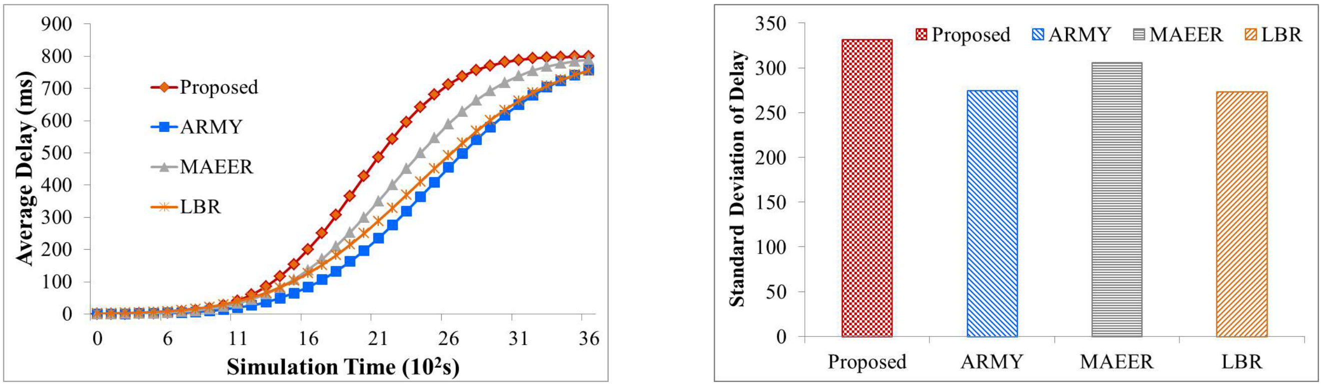
In order to present a comprehensive evaluation, the values of standard deviation of throughput of all the gateway nodes achieved by using different approaches are also calculated. Generally, the lower the standard deviation value, the more stable the network will be. The corresponding results are shown in Figs. 7(a) and 8(a) respectively. Since MAEER and LBR only use one path for data forwarding, their probabilities of causing more congestion traffic become large. Despite the fact that LBR can change the routing path, it still relies on one path to forward data, which may not be enough when the amount of traffic becomes large. Moreover, ARMY cannot adapt to the extremely changing network environment, because it does not change the forwarding probability to one path even though it suffers from a network trouble. Instead, the proposed approach not only uses multiple paths for data forwarding, but also changes the forwarding probability according to the changing network states. Therefore, the proposed approach can achieve a more stable condition, as indicated by the results shown in Figs. 7(b) and 8(b), that is, the standard deviation of throughput achieved by the proposed approach is smaller than that of the other benchmarks. Such results can prove the correctness of the proposed mechanism in a certain extent.

Apart from throughput, the delay achieved by using different approaches is also evaluated. The corresponding results are shown in Fig. 9 and Fig. 10 respectively. Unlike the results of throughput, the proposed approach does not show more and better performance than the other benchmarks in both Figs. 9(a) and 10(a). In contrast, the average delay achieved by the proposed approach is even higher than that of the others. For example, at the simulation time between [2100,3000]s in Fig. 9(a) and at the simulation time between [1100,3000]s in Fig. 10(a), the average delay achieved by the proposed approach is apparently higher than that of the other benchmarks. On one hand, the proposed mechanism uses multiple paths for data forwarding. Some of them are longer, because a detour is usually used for load balance purposes. On the other hand, the fixed paths used by the other benchmarks could result in a very stable average network delay, even though some packets may be dropped. Nevertheless, the average delay achieved by the proposed approach is



(a) (b)

FIGURE 9. Average delay and the standard deviation using random walking model. (a) Average delay. (b) Standard deviation of average delay.



(a) (b)

FIGURE 10. Average delay and the standard deviation using cascading walking model. (a) Average delay. (b) Standard deviation of average delay.

within 1 second which is under acceptance. Apart from this, the corresponding standard deviation of the achieved delay is also calculated. The results are shown in Figs. 9(b) and 10(b) respectively. Different from the result of high delay, the proposed approach has the lowest value of standard deviation, while MAEER has the largest value of standard deviation. It means that the delay achieved by the proposed approach is more stable than that of the other benchmarks.

To make up for the deficiency on average delay, the average packet loss probabilities achieved by different approaches are also calculated and presented. The detailed results of packet loss probability are shown in Fig. 11 and Fig. 12 in terms of the random and cascading walking models respectively. Apparently, the packet loss probability of the proposed approach is the lowest, while that achieved by ARMY is the second lowest, and that of MAEER is the largest. Such phenomenon can be observed in both figures. The proposed approach splits the traffic among different paths and it can change the forwarding probability according to the varying network conditions, thus to reduce the network congestion in a certain extent. ARMY also bases on such principle to avoid network congestion. However, the proposed approach can adjust the forwarding probabilities to different paths according to the real-time network condition. Thus, the proposed approach can achieve the better performance in packet loss probability. As for MAEER and LBR, they use one single path to transfer data to destinations. MAEER uses one fixed path for data forwarding. Meanwhile, LBR can change the current path to another one if network congestion happens on the current path. Despite this, it still relies on one single path for data forwarding, which could easily lead to network congestion and eventually cause packet loss.

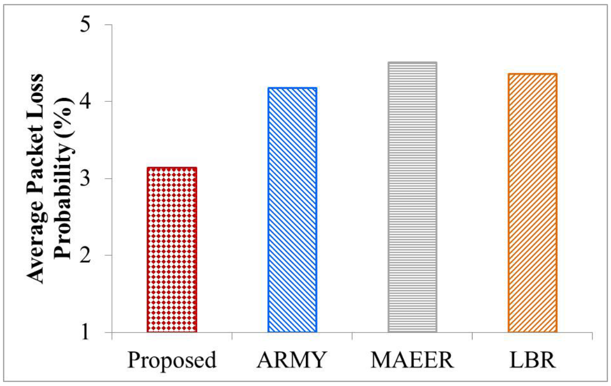


FIGURE 11. **Average packet loss probability using random walking model.**

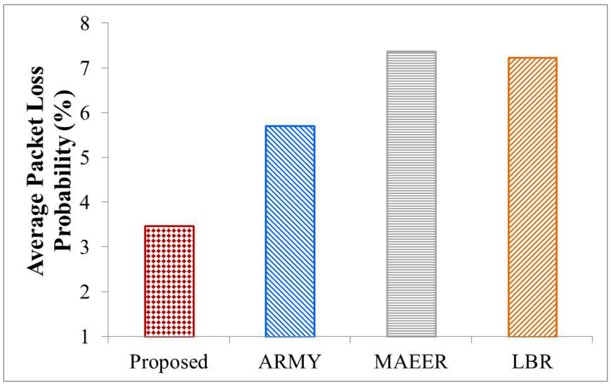


FIGURE 12. **Average packet loss probability using cascading walking model.**

Based on the statistic results, the proposed approach increases network delay by up to 45.1%, while decreasing packet loss probability by up to 107.75%, which is far larger than the previous value. In this regard, the proposed approach still has benefits.

VI. CONCLUSION

The mobility is a very important feature for current fashion IIoT devices. Due to such case, the routing problem in IIoT becomes more and more difficult. In order to address the IIoT routing with mobility considered, this work proposes a multi-step mechanism which involves device handover, data forwarding and routing. Each step is responsible for the corresponding duty, and the three steps are executed in order. In particular, this work not only designs the calculation of forwarding probability to balance the network load, but also introduces the segment routing to achieve a scalable and flexible routing. Based on such mechanism, good performance is achieved with respects to the average packet loss probability, throughput, delay, etc. The future work intends to optimize the IIoT routing towards other aspects such as energy and reliability.

REFERENCES

[1] D.-G. Zhang, H.-L. Niu, S. Liu, and X.-C. Ming, "Novel positioning service computing method for WSN," *Wireless Pers. Commun.*, vol. 92, no. 4, pp. 1747–1769, 2017.

[2] D.-G. Zhang, X. Wang, and X.-D. Song, "New medical image fusion approach with coding based on SCD in wireless sensor network," *J. Elect. Eng. Technol.*, vol. 10, no. 6, pp. 2384–2392, Nov. 2015.

[3] D.-G. Zhang, X. Wang, X. Song, and D. Zhao, "A novel approach to mapped correlation of ID for RFID anti-collision," *IEEE Trans. Services Comput.*, vol. 7, no. 4, pp. 741–748, Oct. 2014.

[4] D.-G. Zhang, K. Zheng, T. Zhang, and X. Wang, "A novel multicast routing method with minimum transmission for WSN of cloud computing service," *Soft Comput.*, vol. 19, no. 7, pp. 1817–1827, Jul. 2015.

[5] Z. Ma, D.-G. Zhang, S. Liu, J. Song, and Y. Hou, "A novel compressive sensing method based on SVD sparse random measurement matrix in wireless sensor network," *Eng. Comput.*, vol. 33, no. 8, pp. 2448–2462, Nov. 2016.

[6] D.-G. Zhang, S. Liu, T. Zhang, and Z. Liang, "Novel unequal clustering routing protocol considering energy balancing based on network partition & distance for mobile education," *J. Netw. Comput. Appl.*, vol. 88, no. 15, pp. 1–9, Jun. 2017.

[7] D.-G. Zhang, S. Zhou, and Y.-M. Tang, "A low duty cycle efficient MAC protocol based on self-adaption and predictive strategy," *Mobile Netw. Appl.*, vol. 23, no. 4, pp. 828–839, May 2017.

[8] D.-G. Zhang, Y.-N. Zhu, C.-P. Zhao, and W.-B. Dai, "A new constructing approach for a weighted topology of wireless sensor networks based on local-world theory for the Internet of Things (IOT)," *Comput. Math. Appl.*, vol. 64, no. 5, pp. 1044–1055, Sep. 2012.

[9] Y.-P. Liang and D.-G. Zhang, "A kind of novel method of service-aware computing for uncertain mobile applications," *Math. Comput. Model.*, vol. 57, nos. 3–4, pp. 344–356, 2013.

[10] D.-G. Zhang, C.-P. Zhao, Y.-P. Liang, and Z.-P. Liu, "A new medium access control protocol based on perceived data reliability and spatial correlation in wireless sensor network," *Comput. Elect. Eng.*, vol. 38, no. 3, pp. 694–702, May 2012.

[11] D.-G. Zhang, X. Song, X. Wang, K. Li, W. Li, and Z. Ma, "New agent-based proactive migration method and system for big data environment (BDE)," *Eng. Comput.*, vol. 32, no. 8, pp. 2443–2466, Nov. 2015.

[12] D.-G. Zhang, X.-D. Song, X. Wang, and Y.-Y. Ma, "Extended AODV routing method based on distributed minimum transmission (DMT) for WSN," *AEU-Int. J. Electron. Commun.*, vol. 69, no. 1, pp. 371–381, Jan. 2015.

[13] D.-G. Zhang and X.-D. Zhang, "Design and implementation of embedded un-interruptible power supply system (EUPSS) for Web-based mobile application," *Enterprise Inf. Syst.*, vol. 6, no. 4, pp. 473–489, Oct. 2011.

[14] D.-G. Zhang, "A new approach and system for attentive mobile learning based on seamless migration," *Appl. Intell.*, vol. 36, no. 1, pp. 75–89, Jan. 2012.

[15] D.-G. Zhang, K. Zheng, D.-X. Zhao, X.-D. Song, and X. Wang, "Novel quick start (QS) method for optimization of TCP," *Wireless Netw.*, vol. 22, no. 1, pp. 211–222, Jan. 2016.

[16] D. Zhang, X. Kang, and J. Wang, "A novel image de-noising method based on spherical coordinates system," *EURASIP J. Adv. Signal Process.*, vol. 2012, p. 110, Dec. 2012.

[17] D.-G. Zhang, W.-B. Li, S. Liu, and X.-D. Zhang, "Novel fusion computing method for bio-medical image of WSN based on spherical coordinate," *J. Vibroeng.*, vol. 18, no. 1, pp. 522–538, Feb. 2016.

[18] Z. Ma, D.-G. Zhang, J. Chen, and Y.-X. Hou, "Shadow detection of moving objects based on multisource information in Internet of Things," *J. Exp. Theor. Artif. Intell.*, vol. 29, no. 3, pp. 649–661, Jul. 2016.

[19] S. Sankaran and R. Sridhar, "Modeling and analysis of routing in IoT networks," in *Proc. Int. Conf. Comput. Netw. Commun.*, Trivandrum, India, Dec. 2015, pp. 649–655.

[20] D.-G. Zhang, S. Zhou, J. Chen, and S. Liu, "New mixed adaptive detection algorithm for moving target with big data," *J. Vibroeng.*, vol. 18, no. 7, pp. 4705–4719, Nov. 2016.

[21] T. Zhang *et al.*, "Novel optimized link state routing protocol based on quantum genetic strategy for mobile learning," *J. Netw. Comput. Appl.*, vol. 122, pp. 37–49, Nov. 2018.

[22] C. Chen and Y. Cui, "New method of energy efficient subcarrier allocation based on evolutionary game theory," *Mobile Netw. Appl.*, pp. 1–14, Sep. 2018.

[23] S. Liu and X. Liu, "Novel dynamic source routing protocol (DSR) based on genetic algorithm-bacterial foraging optimization (GA-BFO)," *Int. J. Commun. Syst.*, vol. 31, p. e3824, Dec. 2018.

[24] J. Chen, C. Li, W. Liang, D.-G. Zhang, and G. Mao, "Capacity of cooperative vehicular networks with infrastructure support: Multiuser case," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1546–1560, Feb. 2018.

[25] A. Giorgetti, A. Sgambelluri, F. Paolucci, F. Cugini, and P. Castoldi, "Segment routing for effective recovery and multi-domain traffic engineering," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 2, pp. 223–232, Feb. 2017.

[26] T. Winter, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document IETF RFC 6550, Mar. 2012.

[27] Z. Shelby, K. Hartke, C. Bormann, *The Constrained Application Protocol (CoAP)*, document IETF RFC 7252, Jun. 2014.

[28] Accessed: Dec. 12, 2018. [Online]. Available: https://www.ietf.org/topics/iot/

[29] N. Nake and P. Chatur, "A review on hierarchical routing methods in sink based wireless sensor networks," in *Proc. 2nd Int. Conf. Adv. Electr., Electron., Inf., Commun. Bio-Inform. (AEEICB)*, Feb. 2016, pp. 259–262.

[30] N. Al-Hoqani, S.-H. Yang, D. P. Fiadzeawu, and R. J. Mcquillan, "In-network on-demand query-based sensing system for wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.

[31] D.-G. Zhang, H.-L. Niu, and S. Liu, "Novel PEECR-based clustering routing approach," *Soft Comput.*, vol. 21, no. 24, pp. 7313–7323, Dec. 2017.

[32] J. Guo, P. Orlik, and K. Ishibashi, "Resource aware hierarchical routing in heterogeneous wireless IoT networks," in *Proc. 8th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, 2016, pp. 599–604.

[33] D. Zhang, G. Li, X. Ming, Z.-H. Pan, and K. Zheng, "An energy-balanced routing method based on forward-aware factor for wireless sensor network," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 766–773, Feb. 2014.

[34] D.-G. Zhang, X. Wang, X.-D. Song, T. Zhang, and Y.-N. Zhu, "A new clustering routing method based on PECE for WSN," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, p. 162, Jun. 2015.

[35] J. Guo, P. Orlik, J. Zhang, and K. Ishibashi, "Reliable routing in large scale wireless sensor networks," in *Proc. 6th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2014, pp. 99–104.

[36] H.-Y. Kim, "An effective load balancing scheme maximizes the lifetime in wireless sensor networks," in *Proc. 5th Int. Conf. IT Converg. Secur. (ICITCS)*, Aug. 2015, pp. 24–27.

[37] V. J. Kotagi, F. Singh, and C. S. R. Murthy, "Adaptive load balanced routing in heterogeneous IoT networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 589–594.

[38] Z. Zhu, R. Wakikawa, and L. Zhang, *A Survey of Mobility Support in the Internet*, document IETF RFC 6301, Jul. 2011, pp. 1–30.

[39] A. Sharma, X. Tie, A. Venkataramani, D. Westbrook, and A. Yadav, "A global name service for a highly mobile Internet," *ACM SIGCOMM*, Chicago, IL, USA, Aug. 2014, pp. 1–13.

[40] D. Zhang, T. Zhang, Y.-Y. Cui, X. Liu, and G. H. Ge, "New multi-hop clustering algorithm for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[41] D.-G. Zhang, T. Zhang, Y. Dong, X.-D. Zhang, and J. Zhang, "A kind of effective data aggregating method based on compressive sensing for wireless sensor network," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, p. 159, Dec. 2018.

[42] J. Lee, S.-K. Kim, J. Jo, J.-H. Yoon, and S.-B. Yang, "Short paper: A human mobility pattern-based routing protocol for delay tolerant networks," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 1–2.

[43] M. Hata, M. Soylu, S. Izumi, T. Abe, and T. Suganuma, "Design of SDN based end-to-end routing over multiple domains for mobility management," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, Tokyo, Japan, Nov. 2017, pp. 1–4.

[44] J. Rangaraj and M. Anitha, "Performance analysis of proactive and reactive protocol under different mobility models for MANET," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Feb. 2017, pp. 637–644.

[45] J. Kim and J. Lee, "Cluster-based mobility supporting WMN for IoT networks," in *Proc. IEEE Int. Conf. Green Comput. Commun.*, Besancon, France, Nov. 2012, pp. 700–704.

[46] S. S. Chaudhari, S. Maurya, and V. K. Jain, "MAEER: Mobility aware energy efficient routing protocol for Internet of Things," in *Proc. Conf. Inf. Commun. Technol. (CICT)*, Gwalior, India, Nov. 2017, pp. 1–6.

[47] X.-P. Han, X.-W. Wang, X.-Y. Yan, and B.-H. Wang, "Cascading walks model for human mobility patterns," *PLoS ONE*, vol. 10, no. 4, p. e0124800, 2015.

Authors' photographs and biographies not available at the time of publication.

• • •