Received September 2, 2017, accepted September 23, 2017, date of publication September 29, 2017,
date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2758340

A Distributed TDMA Scheduling Algorithm Based on Exponential Backoff Rule and Energy-Topology Factor in Internet of Things

YIPING LI[1], XIAOTONG ZHANG[1], (Member, IEEE), TIE QIU[2], (Senior Member, IEEE), JUN ZENG[1], AND PENGFEI HU[1], (Student Member, IEEE)

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2]School of Software, Dalian University of Technology, Dalian 116024, China

Corresponding author: Xiaotong Zhang (zxt@ies.ustb.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2016YFB0700503 and Grant 2017YFB0702300.

ABSTRACT In the Internet of Things scene, the wireless sensor network (WSN) is widely used to monitor and perceive various context environments. The efficient utilization of time division multiple access (TDMA) slot resource has attracted more and more attention, especially for applications with high network performance requirements, for example, vehicular networks. The characteristics of the WSN, which have limited battery volume and variable topology structure, restrict the development of the centralized time slot allocation algorithm. Moreover, the traditional distributed time slot allocation algorithm is helpless to reduce the energy consumption, even if the variable topology is solved to some extent. In this paper, we propose the distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor, namely EB-ET-distributed randomized (DRAND) algorithm. We analyze the typical DRAND time slot assignment algorithm and the distributed TDMA slot scheduling algorithm based on energy-topology factor, which is proposed in our another work. By introducing the idea of Lamport's bakery algorithm, the priority control algorithm based on exponential backoff rules and energy-topology factor are presented to appropriately adjust the priority of time slot allocation and greatly reduce the probabilities of message collision and time slot allocation failure. Then, we introduce the implementation processes of the EB-ET-DRAND scheduling algorithm in various different states. The time slot structure and frame formats of algorithm are designed in detail. Finally, we implement a mesh network simulation system to evaluate the performance of proposed scheme. The experimental results indicate that the EB-ET-DRAND scheduling algorithm greatly improves the performance of time slot allocation and reduces the message complexity, time complexity, and energy consumption.

INDEX TERMS Internet of things (IoT), wireless sensor network (WSN), TDMA scheduling, exponential backoff, energy-topology factor, distributed time slot allocation.

I. INTRODUCTION

The development of Internet of Things (IoT), Cyber Physical System (CPS) and Wireless Sensor Network (WSN) greatly promoted the interconnection and intercommunication of all kinds of things [1], [2]. Physical space is connected into information space by various sensing technologies [3]. For example, vehicles are integrated with more and more sensors and communication technologies to derive and enable various vehicular network applications [4]. The importance of WSN promotes that some new technologies have been proposed to improve network connectivity and data delivery ratio, for example, social-oriented adaptive transmission mechanism [5], and the cooperation mechanism among network nodes based on the copy adjustable incentive scheme [6]. For IoT and WSN, context condition monitoring is the main application, so adaptability and real-time is the key to ensure the completion of the task [7], [8]. These requirements can be implemented by the rational allocation of network resources. The ideal resource allocation schemes can make the effective use of communication resources to improve network adaptability and prolong network lifetime [9]. Furthermore, in the latency-sensitive vehicular networks, the resource allocation scheme is great significance for the safe driving of vehicles [10]. Therefore, the resource allocation is one of the main research topics in the direction of IoT and WSN in recent years [11], [12]. The problem of time slot allocation is a Non-Deterministic Polynomial (NP) complete multi-objective optimization problem, which is the main challenge.

The energy supplies of the WSN nodes come from the batteries carried by the nodes, which are not suitable for charging or replacement for cost and ease of use. The energy of the network node is mainly consumed in the data communication module [13]. Therefore, how to improve the communication efficiency and reduce the working time of the communication module can effectively improve the life of network node [14], [15]. The traditional wireless communication protocol cannot be adopted, and the protocol needs to be redesigned. If node wants to access the channel, it needs be deployed through the Media Access Control (MAC) protocol [16]. The protocol not only bears the important tasks of the entire network resource allocation, but also determines the work mode of node data transmission and controls the communication module to start and stop. So the MAC protocol plays a decisive role in node energy consumption [17]. It is one of the important protocols to ensure the adaptability and real-time of network in the case of uncertain topology and energy constraint [18], [19].

In recent years, a large number of researches on MAC protocol and time slot allocation algorithm of WSN have made some important progress [20], [21]. But poor allocation efficiency, high energy consumption and other reasons make these research findings cannot be used into the practical applications. The lack of a complete theoretical system in this field restricts the development of WSN. The unpredictable network working environment, such as, wireless signal interference and nodes movement, makes the time slot resources allocation more complex [22], [23]. The centralized scheduling schemes can optimize the allocation algorithm for the actual requirements under the premise of the overall topology of the known network. However, they are not suitable for the network with uncertain topology over time. Therefore, it is very important to propose a distributed time slot allocation algorithm which is suitable for the dynamic change of network topology [24].

Some typical distributed time slot allocation algorithms, for example, DRAND algorithm [25] and Five-Phase Reservation Protocol (FPRP) [26], can reasonably allocate the time slot resources according to the requirements. So they can efficiently utilize the radio channel resources and reuse the time slots as much as possible. In our another work, we have proposed a distributed TDMA slot scheduling algorithm based on energy and topology factor (E-T-DRAND) by analyzing and summarizing the DRAND algorithm [27]. It presents the definition of energy-topology factor which is based on the influence of residual energy and topology on the time slot allocation. And it adopts energy-topology factor to arrange the priority of time slot scheduling. However, the message complexity, time complexity and energy consumption are too high. The exchange of too many data packets among neighbor nodes will lead to the non-convergence of the slot allocation time and the number of rounds when the continuous time slot request occurs. As a result, the amount of data packets switching and the running time of time slot allocation will increase.

In this paper, we analyze the DRAND algorithm and E-T-DRAND algorithm in detail. On this basis, by introducing the idea of Lamport's bakery algorithm, we propose the distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor, namely EB-ET-DRAND algorithm. And the priority control algorithm of the EB-ET-DRAND based on exponential backoff rule and energy-topology factor are presented to appropriately adjust the priority of time slot allocation. In addition, we introduce the implementation process of EB-ET-DRAND scheduling algorithm in various different states. The time slot structure and frame formats of algorithm are designed. Finally, we implement a large mesh network simulation system on the Objective Modular Network TestBed in C++ (OMNeT++) platform to evaluate the performance of proposed EB-ET-DRAND algorithm. The experimental results indicate that the EB-ET-DRAND scheduling algorithm greatly improves the performance of time slot allocation. It significantly reduces the message complexity, running time, number of rounds, and energy consumption.

In this paper, we focus on the research on distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor for WSN. The main contributions can be summarized as follows:

1) The distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor is proposed by introducing the idea of Lamport's bakery algorithm to improve the efficiency of time slot resources allocation and reduce the energy consumption.
2) The priority control algorithm based on exponential backoff rule and energy-topology factor are presented to appropriately adjust the priority of time slot allocation and greatly reduce the probabilities of message collision and time slot allocation failure.
3) The implementation process of proposed EB-ET-DRAND scheduling algorithm in various different states is introduced. And the time slot structure and the frame formats of algorithm are also designed in detail.
4) A large scale mesh network simulation system is implemented on the OMNeT++ platform to simulate and evaluate the performance of proposed scheme. Compared with some typical time slot allocation algorithms, the results indicate that the proposed scheme significantly reduces the message complexity, running time, number of rounds, and energy consumption.

The remainder of this paper is organized as follows. Section 2 reviews the related work on TDMA scheduling algorithm. Section 3 proposes the distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor. Section 4 presents the time slot structure and frame format design of EB-ET-DRAND scheduling algorithm. Section 5 presents the experiment and performance evaluation of the proposed algorithm. Section 6 concludes this paper.

II. RELATED WORK

The time slot resource allocation methods can have different classifications according to different focus. Based on whether to set up a centralized controller to assist the distribution scheme, it can be divided into the centralized and distributed resource allocation methods.

The centralized allocation methods mean that there is a central controller in the network, which is responsible for the calculation of the resource allocation scheme of the whole network. It needs to know the information of the whole network and distribute the allocation scheme to every node. The results generated by the centralized allocation of the entire network close to the optimal. But it is not suitable for the frequent changes of network topology. The frequent calculation and distribution of allocation scheme greatly increase additional cost.

In the centralized allocation algorithm, graph coloring is used as a common method. However, there are many deficiencies in the case of multi-channel using this scheme, for example, interface constraints, bandwidth constraints, etc. Raniwala *et al.* [28] proposed the Load-Aware Channel Assignment (LACA) method which was a multi-channels heuristic allocation algorithm with multiple constraints. It increases the actual network throughput by restricting the number of channels, the requirements of network traffic and the number of interfaces. The LACA is based on the changes of network load so that it can be applied to multi-channel occasions. According to the estimated network load, the communication links are grouped according to the conflict relation, and then the communication resources are allocated to the nodes in the network. In the scheduling process, LACA algorithm constantly adjusts the channel resource scheduling program according to the network traffic. Because it is a centralized scheme, the algorithm was suitable for static networks.

For the distributed resource allocation methods, every node determines the occupancy of time slots and channels according to their own local information to generate allocation strategy in a distributed manner. The distributed allocation method is more suitable for dynamic networks, but the performance of the scheme may be lower than that of centralized allocation. The distributed resource allocation method can arrange the resources reasonably according to the demand, so the wireless channel resources can be used efficiently and the time slots can be multiplexed as much as possible. The dynamic TDMA protocol has a special control frame structure, so the control part and the data part do not interfere with each other. Some researchers have proposed many distributed time slot allocation schemes. The DRAND and DSA-AGGR algorithms are the typical distributed time slot allocation scheme.

Based on the centralized randomized time slot scheduling scheme (RAND) [29], Rhee *et al.* [25] proposed the DRAND scheme which was a distributed randomized time slot scheduling algorithm. It implements the RAND algorithm in a distributed way. In this algorithm, the time slot sequence is allocated by coordinating requests among network nodes. It can solve the conflicting node requests to assign the slot sequence of the TDMA. However, the time slot allocation with completely randomness decrease the efficiency, and also result in the high collision rate of the message and the low energy efficiency. The algorithm is suitable for the network with limited mobility which most nodes do not move.

Sato and Sakata [30] proposed a distributed TDMA slot scheduling algorithm with prioritized control based on Lamports bakery algorithm. It can constitute a localized network by measuring the distance to respective node. This distance-measurement-oriented scheme is shown as a possible replacement of DRAND scheduling algorithm. It improves the efficiency of TDMA slot allocation and reduces the time consumption on slot allocation, but increases the number of message required to allocate time slots.

Bryan *et al.* [31] proposed the Color Constraint Heuristic Algorithm (CCH) which is based on graph coloring theory and has a shorter duration than the DRAND algorithm. CCH calculate the weighted sum of the one-hop and two-hop neighbor nodes that have been allocated the time slot. And the weighted sum is used as the metric to determine the order in which the nodes are dyed, that is to say the node with high metric value are preferentially dyed, and the process is repeated. Based on this algorithm, the corresponding distributed algorithm (DSA-CCH) is developed. The algorithm uses the count backwards of above weighted sum to define the metric. When the preseted threshold is exceeded, the node begins to dye. Based on the DSA-CCH algorithm, the distributed Color Constraint Heuristic for data aggregation algorithm (DSA-AGGR) is proposed. It is optimized for the coloring aggregation tree to minimize the delay of aggregation value, and thus return to the base station for continuing the calculation. So the parent node will be allocated to the higher priority color than child node.

Zhu and Corson [26] proposed the FPRP which is a distributed heuristic TDMA slot allocation algorithm. It allocate time slot dynamically by the five-phase reservation process to realize non-conflicting time slot allocation. In this scheme, the real time is divided into many pairs of reservation and data transmission phases. A five-phase protocol is run in the each time slot of the data transmission phase to pick a winner of each slot. FPRP allows nodes to occupy multiple time slots simultaneously in a distributed wireless network. It supports the time slot space multiplexing, and the nodes outside two hops can reuse the same time slot.

Wang and Henning [32] proposed a deterministic distributed TDMA scheduling algorithm (DD-TDMA) which each network node allocates its own TDMA slot based on its neighborhood information. It presents that if the interval of transmission and reception slots are short, the energy consumption can be reduced by keeping the node in idle state, rather than in sleep state. However, the time needs to synchronize in this time slot allocation algorithm, and collision model is too idealistic, so it was difficult to apply in practice.

These existing distributed TDMA scheduling algorithms have improved the performance of time slot allocation to some extent from different perspectives. However, the problems, such as high time complexity, high message complexity and high energy consumption, still exist and need to be further in-depth studied and addressed.

III. THE DISTRIBUTED TDMA SCHEDULING ALGORITHM BASED ON EXPONENTIAL BACKOFF RULE AND ENERGY-TOPOLOGY FACTOR

In this section, we analyze the typical DRAND algorithm and the E-T-DRAND algorithm proposed in our another work firstly to indicate the enhancement scheme of performance improvement for time slot allocation. On the bases of these algorithms, we propose a distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor by introducing the idea of Lamport's bakery algorithm, namely EB-ET-DRAND algorithm. The exponential backoff and weight control rules of EB-ET-DRAND algorithm are presented. And we also present the implementation process of EB-ET-DRAND algorithm in detail.

A. THE ANALYSIS OF DRAND ALGORITHM AND E-T-DRAND ALGORITHM

In our another work, we have proposed a distributed TDMA slot scheduling algorithm based on energy and topology factor (E-T-DRAND) by analyzing and summarizing the DRAND algorithm [27]. This algorithm presents the definition of E-T factor which is based on the influence of residual energy and topology on the time slot allocation. And it adopts the E-T factor to arrange the priority of time slot scheduling to reduce the execution time and energy consumption of algorithm. Moreover, a large scale Mesh network system on the OMNeT++ platform has been simulated to evaluate the performance of E-T-DRAND and DRAND time slot allocation algorithms. Fig. 1 and Fig. 2 show respectively the average running time and number of rounds for successful allocating time slot in different size of neighbor nodes. Though the performance of E-DRAND algorithm is better than the DRAND algorithm and improves the efficiency of time slot allocation, there are still some defects need to be further improved.

The running time is proportional to the number of rounds for successful allocating time slot in different size of neighbor nodes, so their trends are the same in the pictures. According to the experimental results of DRAND and E-DRAND algorithms, the running time of time slot allocation is obviously increasing with the increase of the number of neighbor nodes. In addition, it can be observed that the running time of two algorithms has a significant increase within the range of 40-55. Through these analyses we can see that the exchange of too many data packets among neighbor nodes will lead to the non-convergence of the allocation time and the number of rounds when the continuous time slot requests occur. As a result, the amount of data packets switching and the time of time slot allocation will increase.

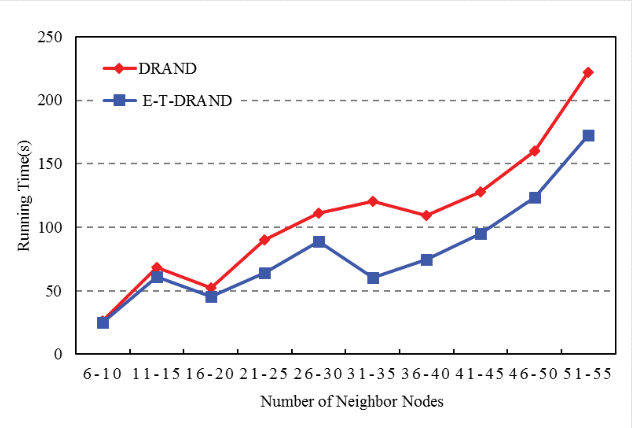


FIGURE 1. The average running time for successful allocating time slot in different size of neighbor nodes.

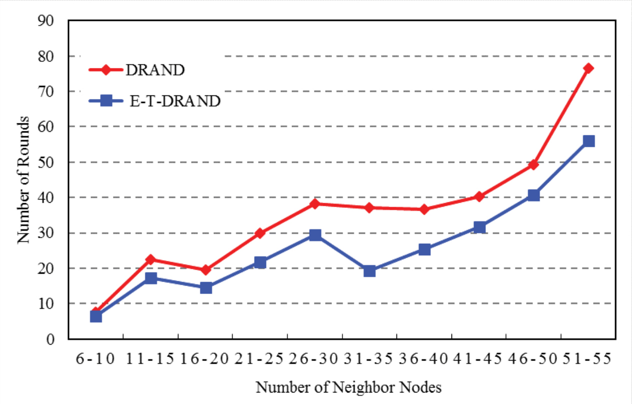


FIGURE 2. The average number of rounds for successful allocating time slot in different size of neighbor nodes.

These are not special cases. These problems have shown in the initial DRAND algorithm, which is derived from that the distribution of some nodes is very dense and the frequency of requests to retry is more than the surrounding scattered nodes. Therefore, if the frequency of requests to retry can be reduced, it will help improve the performance of the time slot allocation. So the enhancement scheme proposed in this section is intended to improve the efficiency of the requests to retry.

B. THE PROCESS AND STATE TRANSITION OF EB-ET-DRAND SCHEDULING SCHEME

Lamport's bakery algorithm is a mutex algorithm that prevents concurrent threads from entering critical area,

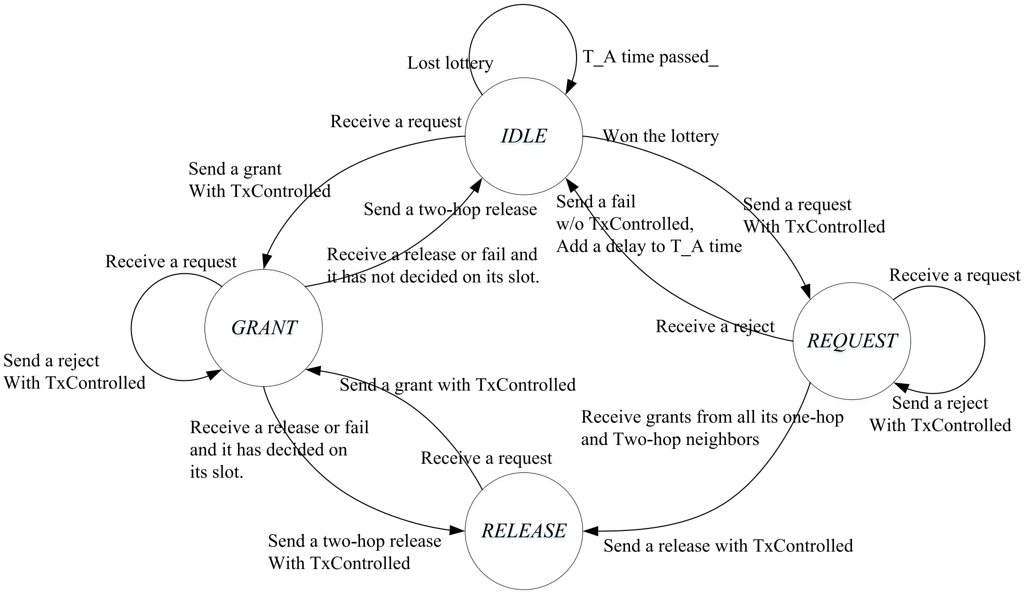


FIGURE 3. **The state diagram of the EB-ET-DRAND scheduling algorithm implementation.**

thereby eliminating the risk of data loss [33]. This algorithm idea can solve the high collision rate problem of messages during the process of time slot allocation.

This algorithm solves the problem caused by many asynchronous threads synchronization:

1) At any one time, there is only one thread at most in the critical area.
2) Each thread (unless it is stopped) must eventually be able to enter the critical area.
3) Any thread can be suspended beyond the critical area.

The basic idea of this algorithm is derived from the customer queuing principle in the bakery. Each customer is assigned a number firstly before entering the bakery, and then enters the bakery following the number from small to large order to buy bread. Here, the number assigned by baker is sorted from small to large, but two or more customers are likely to get the same number. So the number needs to be mutually-exclusive. If multiple customers catch the same number, the customer names are used to sort alphabetically, in which it is assumed that the names of all customers are not same. Lamport's bakery algorithm is designed for multi-threading, so the order in which threads are accessed can be controlled by priority.

Based on the idea of Lamport's bakery algorithm, we design the EB-ET-DRAND algorithm in this paper, which is a distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor. It uses different backoff time to distinguish the opportunity of the time slot request. This algorithm further improves and perfects the E-T-DRAND time slot allocation algorithm.

The EB-ET-DRAND algorithm is running round by round. The total time of each round is dynamically adjusted according to the network delay. There are four states for each node in the EB-ET-DRAND algorithm, which includes IDLE, REQUEST, GRANT, and RELEASE. According to the reference [25], Fig. 3 shows the state diagram of the EB-ET-DRAND scheduling algorithm implementation. Table 1 shows the state transition rules of EB-ET-DRAND scheduling algorithm. The starting conditions of each state transition occurrence and the next state are presented.

In EB-ET-DRAND scheduling scheme, the implementation of time slot allocation control is based on energy-topology factor and exponential backoff rules, rather than the random probability in DRAND algorithm. But most of the state transition procedures are similar with the DRAND algorithm. Compared with E-T-DRAND algorithm, the EB-ET-DRAND adds control rules to the backoff time and weight, which is shown in Table 1 and Fig. 3 with bold. In this scheme, for the nodes in one-hop range, only one node can send the time slot allocation request at the same time. In the process of state transition which includes IDLE, REQUEST, GRANT, and RELEASE, the time slot allocation policy based on energy-topology factor and exponential backoff rules is used to control the transmission of data packets. This scheme can effectively reduce the collision of messages.

C. THE IMPLEMENTATION PROCESS OF EB-ET-DRAND ALGORITHM

The state transition principles and processes of EB-ET-DRAND algorithm are described in detail in above section. Each node in the algorithm implementation contains four states, including IDLE, REQUEST, GRANT, and RELEASE. IDLE is the initial state of all nodes, which determines whether to apply the time slot by calculating its energy-topology factor. If it can apply a time slot, an unallocated time slot is selected through the state transition and negotiation

TABLE 1. The state transition table of EB-ET-DRAND scheduling algorithm.

No.	Current State	Event	Action / Condition	Next State
1	IDEL	Priority control algorithm is adopted (success), time slot request can be sent	Broadcast the time slot request to the one-hop neighbor nodes, start timer	REQUEST
2	IDEL	Time slot request messages of neighbor nodes are received	Send grant	GRANT
3	**IDEL**	**Priority control algorithm is adopted (failure), backoff time is calculated**	**Backoff, wait**	**IDEL**
4	REQUEST	Grant information of all one-hop neighbor nodes are received	Select own time slot resource, send the release information	RELEASE
5	REQUEST	No grant information for all one hop neighbor nodes are received in the timing	Resend the request to the nodes that have not sent the grant information	REQUEST
6	REQUEST	Request messages from other nodes are received	Send reject message	REQUEST
7	**REQUEST**	**Reject message is received, backoff time is calculated**	**Send failure message, backoff**	**IDLE**
8	RELEASE	Time slot request messages of neighbor nodes are received	Send grant	GRANT
9	GRANT	Request messages from other nodes are received	Send reject message	GRANT
10	GRANT	Released messages from other nodes or failure messages are received, and time slot is not allocated to itself	Broadcast this message within two hops	IDLE
11	GRANT	Released messages from other nodes or failure messages are received, and time slot has been allocated to itself	Broadcast this message within two hops	RELEASE

among neighbor nodes. If it is failure in these processes, exponential level backoff time will be produced. Algorithm 1 shows the EB-ET-DRAND time slot allocation algorithm of j node with pseudo-code, where $state_j$ represents the state information of j node.

D. THE EXPONENTIAL BACKOFF AND WEIGHT CONTROL RULES OF EB-ET-DRAND ALGORITHM

Compared to E-T-DRAND, EB-ET-DRAND algorithm has different realization form for achieving the acceptance of reject message. When a node receives a reject message, its backoff time is generated by an exponential backoff algorithm, which is timed to the next retransmission request allocation time. The algorithm is implemented in the following pseudocode, and the backoff time is changed according to the number of reject messages that have been received. The algorithm shows the same characteristics as the Lamport's bakery algorithm, and the opportunity of sending time slot request is separated by time.

Algorithm 2 shows the exponential backoff algorithm. The *recvRejCount*, which is the cumulative number of time slot allocation failure, is increased when the node receives the reject message. When the variable *recvRejCount* is less than k ($k = m = 8$ in later experiments), *T_Atime* will be that the *Ta* (transmission delay) multiplies by a random integer of no more than the *k-th* power of 2, which is evenly distributed. If *recvRejCount* is more than or equal to k, the random value will not exceed the *k-th* power of 2. The values of these parameters mainly consider the following aspects:

1) The maximum backoff time is not more than all the time slot allocation time;
2) The minimum back-off time cannot be less than 2;
3) The best parameter range verified by some experiments.

For the calculation of backoff time at line 6 and line 10 in algorithm 2, *Ta* is transmission delay, *pow* is exponent arithmetic, $recvRejCount < k?recvRejCount : k$ is that the maximum value of *recvRejCount* is k.

In addition, the backoff time is also required in the general state transition process, but this time is shorter than the *T_Atime*. This algorithm is intended to make the transmission of the request time slot allocation more discrete and reduce the probability of message collision by the backoff time, so that it can reduce the probability of request failure.

In EB-ET-DRAND algorithm, in addition to optimizing the exponential backoff rule, a weight control mechanism is also introduced. It divides a more pronounced priority with the residual energy of nodes. The priority control rules are as follows:

1) In the process of time slot allocation, the more neighbors and less residual energy a node has, the higher its priority. Because these nodes will generate more extra consumption once the time slot allocation fails. In the initial stage, the priority order of node is allocated by referring to the number of neighbor nodes firstly.
2) If the residual energy of a node is lower than that of all the one-hop neighbors around it, the node should have the highest priority. Because the extra consumption of these nodes will speed up the energy consumption once the time slot allocation fails. They will die once their energy is exhausted.
3) Within the two-hop range, if there are some nodes which have lower residual energy, the node should decrease its priority in the ranking.

Algorithm 1 The EB-ET-DRAND Time Slot Allocation Algorithm of Node *j*

```
1:  $state_j$: the four states of node $j$;
2:  The process that node j competitively requests time slot:
3:  if (Success in competition) then
4:      $state_j$ = REQUEST;
5:      Broadcast the request to the one-hop neighbor nodes;
6:  else
7:      $state_j$ = IDLE;
8:      Wait exponential back_off time;
9:  The node j receives a reject:
10: if (node $j$ receives a reject) then
11:     Broadcast the failure message to the one-hop neighbor nodes;
12:     $state_j$ = IDLE;
13:     Wait exponential back_off time;
14: The node $j$ receives the grant information from its all one-hop neighbor nodes (the time slot request is successful)
15: if (node $j$ receives the grant information from its all one-hop neighbor nodes) then
16:     Select a free time slot;
17:     $state_j$ = RELEASE;
18:     Send release message;
19: Node $m$ is a neighbor node of $j$, and receives the request of neighbor node
20: while (node $m$ receives a request) do
21:     if ($state_m$ = IDLE or $state_m$ = RELEASE) then
22:         $state_m$ = GRANT
23:         Send a grant message to node $j$;
24:     else
25:         Send a reject message to node $j$;
26: Node $m$ is a neighbor node of $j$, and receives the release of $j$
27: if (node $m$ is not assigned a time slot) then
28:     $state_m$ = IDLE;
29: else
30:     $state_m$ = RELEASE;
31:     Broadcast the release to the two-hop neighbor nodes;
```

Algorithm 2 The Exponential Backoff Algorithm

```
1:  recvRejCount: the cumulative number of time slot allocation failure;
2:  T_Atime: backoff time.
3:  Condition 1:
4:  if (receive the reject message of neighbor node) then
5:      $recvRejCount ++$;
6:      $T\_Btime = Ta * rand()\%pow(2, (recvRejCount < k?recvRejCount : k))$;
7:      Backoff $T\_Btime$;
8:  Condition 2:
9:  if (Request time slot failure) then
10:     $T\_Atime = Ta * rand()/RAND\_MAX * pow(2, (recvRejCount < m?recvRejCount : m))$;;
11:     Backoff $T\_Atime$;
```

Algorithm 3 The Priority Control Rules for Requesting Time Slot Allocation

```
Input: Node $A$; One-hop neighbor node $B$; Two-hop neighbor node $C$; Node set $V$.
Output: The next state transition of node.
1:  ticket_number[]: the priority control array of node, each node records the number of its neighbor nodes;
2:  if (has_unslotted_one-hop neibor_node B && has_smaller_residual_energy(unslotted_one-hop neibor_node B) then
3:      $ticket\_number[A] += 1$;
4:      $ticket\_number[B] += \alpha$;
5:  if (has_unslotted_two-hop neibor_node C && has_smaller_residual_energy(unslotted_two-hop neibor_node C) then
6:      $ticket\_number[A] += 1$;
7:      $ticket\_number[C] += \beta$;
8:  if (max($ticket\_number[]$) == $ticket\_number[A]$) then
9:      Send a requested time slot packet, and transit current state to REQUEST;
10: else
11:     random_backoff ($T\_Atime$);
```

Algorithm 3 shows the priority control rules for requesting time slot allocation. Algorithm 4 shows the priority adjustment rules after time slot allocation failure.

Where: the backoff time of line 11 in Algorithm 3 is calculated by the Condition 2 of Algorithm 2, and the backoff time of line 12 in Algorithm 4 is calculated by the Condition 1 of Algorithm 2.

In addition to the idea of weight control, these algorithms are based on the E-T-DRAND algorithm, so the preconditions and definitions of these algorithms are the same as E-T-DRAND algorithm. Since the weight control rule is applied, the value of the *ticket_number* for each node is calculated by α and β, which their values are mainly calculated by experiment and experience. In the final judgment stage, the priority of the time slot allocation is determined by the value of *ticket_number*. After joining this weight control rule, the node with the highest *ticket_number* can apply for a time slot firstly. The process that each node requests time slot by negotiation becomes more discrete. However, the introduction of backoff rule reduces the collision fundamentally.

Based on the priority control rules, the EB-ET-DRAND algorithm can ensure that only one node within its two-hop neighbors tries to apply the time slot allocation at the same time. This strategy greatly reduces the probabilities of time slot allocation failure and the messages collision. Due to the distributed control strategy, the priority control algorithms in each node are the same.

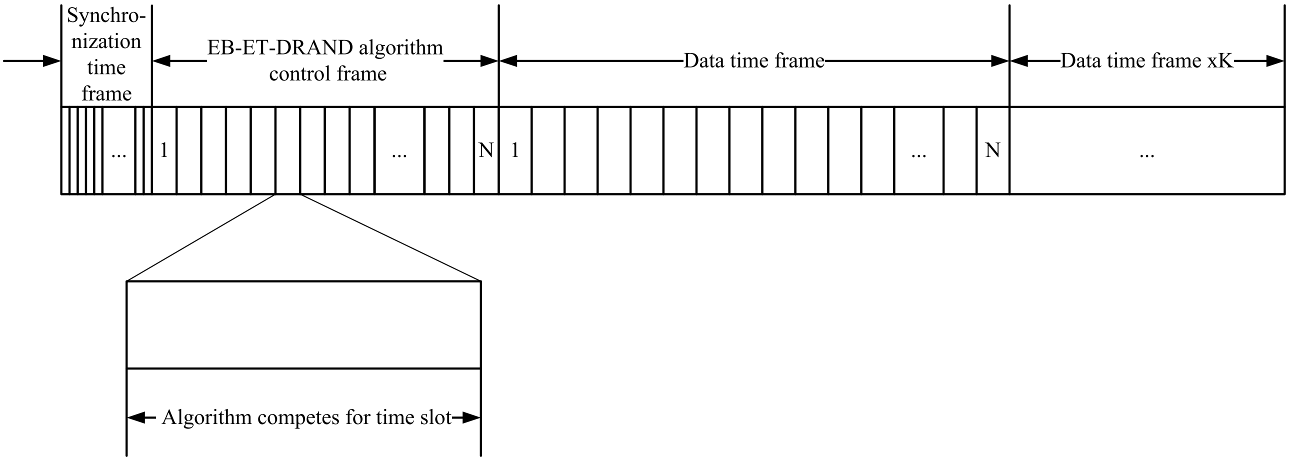


FIGURE 4. The time slot structure of EB-ET-DRAND algorithm in MAC layer.

Algorithm 4 The Priority Adjustment Rules After Time Slot Allocation Failure

```
Input: Node $A$; One-hop neighbor node $B$; Two-hop neighbor node $C$; Node set $V$.
Output: The next state transition of node.
ticket_number[]: the priority control array of node, each node records the number of its neighbor nodes;
Initialize: $ticket\_number[A] += 1$;
if (has_unslotted_one-hop neibor_node B && has_smaller_residual_energy(unslotted_one-hop neibor_node B) then
    $ticket\_number[A] += 1$;
    $ticket\_number[B] += \alpha$;
if (has_unslotted_two-hop neibor_node C && has_smaller_residual_energy(unslotted_two-hop neibor_node C) then
    $ticket\_number[A] += 1$;
    $ticket\_number[C] += \beta$;
if (max(ticket_number[]) == ticket_number[A]) then
    Send a requested time slot packet, and transit current state to REQUEST;
else
    random_backoff (T_Btime);
```

IV. THE TIME SLOT STRUCTURE AND FRAME FORMAT DESIGN OF EB-ET-DRAND ALGORITHM

In this section, we introduce the time slot structure and frame format design of proposed EB-ET-DRAND scheduling algorithm. The time slot structure of algorithm, which consists of the synchronization time frame, the algorithm control frame, and the data time frame, are presented. The frame formats of algorithm, which includes synchronization frame, time slot allocation request frame, notification frame of successful time slot allocation, and data frame, are also designed in detail.

A. THE TIME SLOT STRUCTURE DESIGN OF PROPOSED EB-ET-DRAND SCHEDULING ALGORITHM

In the centralized time slot allocation scheme, a unified network resource allocation node is responsible for allocating the entire network, that is, each time slot of the TDMA frame structure is assigned to its users. In this time slot, whether the node is in the sending or receiving state is also determined by the network resource allocation node. However, in the distributed time slot allocation scheme, the time slot resources are allocated by the coordination among the nodes. That is to say the time slot allocation of TDMA phase is implemented by the participation of all nodes.

In MAC layer, the time slot structure of proposed EB-ET-DRAND algorithm consists of the synchronization time frame, the algorithm control frame, and the data time frame. Fig. 4 shows the time slot structure of EB-ET-DRAND algorithm in MAC layer. The functions of each frame are as follows:

1) The synchronization time frame is responsible for executing synchronization algorithm to implement network synchronization of all nodes in the system. Even if the distributed time slot allocation algorithm, it is also a TDMA mode. So the nodes need to synchronize with each other and keep the clock consistent to avoid message collisions when the data is sent and received [34].
2) The EB-ET-DRAND algorithm control frame is responsible for the implementation phase of time slot allocation algorithm. The proposed EB-ET-DRAND algorithm is executed in this frame. The time slot resources are allocated through competition and negotiation among neighbor nodes.
3) The data time frame is responsible for data communication among nodes.

The EB-ET-DRAND algorithm control frame and the data time frame are composed of N time slots, where N is the maximum number of nodes allowed in the network. In order

TABLE 2. The synchronization frame format.

Frame head	Type	ID of originating node	Local time	CRC checking
2 Bytes	1 Byte	1 Byte	2 Bytes	2 Bytes

to reduce the proportion of time slot resource consumption caused by the synchronization time frame and algorithm control frame, the number of data time frames can be appropriate to increase under the condition that the clock deviation is not obvious. In each data time frame, the number of time slots assigned to nodes is the same. The notification area of neighbor node has M time slots, where M is the maximum number of neighbor nodes. Therefore, the synchronization time frame is composed of M time slots. The synchronization time frame, EB-ET-DRAND algorithm control frame, and multiple data time frames form a frame cycle. And they continue to cycle operation in the time domain.

B. THE FRAME FORMAT DESIGN OF PROPOSED EB-ET-DRAND SCHEDULING ALGORITHM

The frame of EB-ET-DRAND time slot allocation algorithm consists of four formats, including synchronization frame, time slot allocation request frame, notification frame of successful time slot allocation, and data frame. We assume that the upper limit of network nodes is 32 and the upper limit of neighbor nodes for each node is 8. So the EB-ET-DRAND algorithm control frame and the data time frame are composed of 32 time slots, and the synchronization time frame is composed of 8 time slots.

1) SYNCHRONIZATION FRAME

The synchronization frame format is shown in Table 2.

The frame type of corresponding to the different type value is shown in Table 3:

TABLE 3. The synchronization frame format.

Type value of frame type	Corresponding frame type
0x01	Synchronization frame
0x02	Time slot allocation request frame
0x03	Notification frame of successful time slot allocation
0x04	Data frame

In the synchronization frame format, the local time is represented by 2 Bytes. The first Byte represents the time slot number, and the first Byte represents the micro-time slot number. Each time slot is composed of 1000 micro-time slots, which is enough to represent with 1 Byte.

2) TIME SLOT ALLOCATION REQUEST FRAME

The time slot allocation request frame format is shown in Table 4. In this case, the IDs of all neighbor nodes need to be sent. The maximum number of neighbor nodes is 8, and 0 is complemented if it is insufficient.

3) NOTIFICATION FRAME OF SUCCESSFUL TIME SLOT ALLOCATION

Table 5 shows the release frame format sent after successfully allocating time slot. The time slot number of originating node is used to broadcast the slot number in which the originating node state changes. It is used for the update of neighbor nodes.

4) DATA FRAME

Table 6 is shown the data frame format. The data of the data frame is collected from the various sensors information of nodes. Its maximum length is 200 Bytes.

V. EXPERIMENT AND PERFORMANCE EVALUATION

In this section, we implement a Mesh network simulation system on the OMNeT++ platform to evaluate the performance of proposed EB-ET-DRAND time slot allocation algorithm. Firstly, we take a sample and small-scale topology structure for instance to analyze the feasibility of algorithm. Then we implement a large Mesh network simulation system to evaluate the performance of proposed algorithm from multiple aspects.

A. A TIME SLOT ALLOCATION EXAMPLE ANALYSIS BASED ON EB-ET-DRAND ALGORITHM

In order to verify the feasibility of algorithm, a small-scale verification experiment is implemented on the OMNeT++ platform by reference to the [35]. Fig. 5 shows the topology and time slot allocation example of nodes. There are six nodes in the network. Because of the distributed system, all the nodes run the same program, and the network does not have the central node. The numbers of (X, Y) in parentheses represent the coordinates of the nodes in the plane coordinate system.

Except the B-C-D-E node group, A-B and D-F perform an independent time slot allocation algorithm. And the time slot allocation graph also shows that each node is assigned into a different time slot within the two hops range, which is consistent with the expectation of proposed algorithm. In the process of time slot allocation for DRAND algorithm, the nodes start time slot allocation at random time. However, in the EB-ET-DRAND algorithm, the time slot allocation start time of nodes is decided by the priority of the surrounding nodesąŕ energy and topology. In the case of the topology shown in Fig. 5, each group executes the program of time slot allocation request in parallel, eventually forming three groups: D-F, A-B and B-C-D-E. It can be observed that the maximum number of time slots is optimized when the nodes within any two-hop range are not allocated within the same time slot.

TABLE 4. The time slot allocation request frame format.

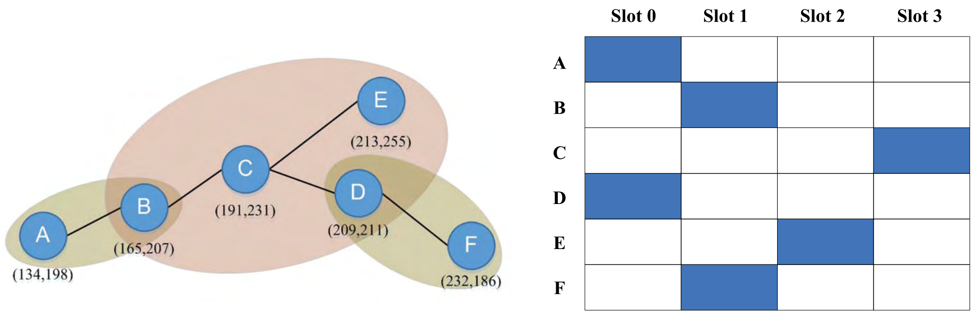
Frame head	Type	Length	ID of originating node	Flag of time slot request	Residual energy information	Node states information	ID of neighbor nodes	CRC checking
2 Bytes	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	8 Bytes	2 Bytes

TABLE 5. The notification frame format format of successful time slot allocation.

Frame head	Type	ID of originating node	Time slot number of originating node	Residual energy information	ID of neighbor nodes	CRC checking
2 Bytes	1 Byte	1 Byte	1 Byte	1 Byte	8 Bytes	2 Bytes

TABLE 6. The data frame format.

Frame head	Type	Length	ID of originating node	ID of destination node	Data	CRC checking
2 Bytes	1 Byte	1 Byte	1 Byte	1 Byte	200 Bytes	2 Bytes



	Slot 0	Slot 1	Slot 2	Slot 3
A	■			
B		■		
C				■
D	■			
E			■	
F		■		

FIGURE 5. The topology and time slot allocation example.

TABLE 7. The number of rounds required for each node to successfully obtain the final time slot for the DRAND and EB-ET-DRAND algorithms.

Node	DRAND algorithm						EB-ET-DRAND algorithm					
	First	Second	Third	fourth	fifth	Average number of rounds	First	Second	Third	fourth	fifth	Average number of rounds
A	2	3	2	2	2	2.2	2	2	1	2	1	1.6
B	3	4	4	3	3	3.4	2	3	3	2	3	2.6
C	4	6	4	5	4	4.6	4	4	3	4	4	3.8
D	3	2	3	3	4	3	2	3	2	2	2	2.2
E	2	2	3	3	3	2.6	2	1	2	2	2	1.8
F	2	3	2	2	2	2.2	2	2	1	1	2	1.6

Table 7 shows the average number of rounds required for each node to successfully obtain the final time slot for the DRAND and EB-ET-DRAND algorithms. Each algorithm performs a total of five times, and the sixth column is the average number of rounds. As is shown from this simple example, the EB-ET-DRAND algorithm with priority control and exponential backoff rules can effectively reduce the number of rounds required for the time slot request. It reduces the collision and improves the efficiency of algorithm implementation. Further large-scale experiments and evaluations will be presented in the following sections.

B. EXPERIMENTAL SETUP

In our previous another work, we have implemented a large Mesh network simulation system on the network simulator OMNeT++ platform to evaluate the performance of E-T-DRAND time slot allocation scheme [27]. In this paper, we still adopt this simulation system and apply the proposed EB-ET-DRAND time slot allocation algorithm in it.

In this simulated network, there are 250 nodes which are randomly distributed in a 300x300m plane. The random distribution of nodes reflects the irregularity of topology, which is more scientific for the performance evaluation of the algorithm. Fig. 6 shows the topology structure of OMNet++ simulation system. The dotted line represents a connection link between the two nodes. Each node has a broadcast communication range of 40m and a link capacity of 2 Mbps.

In our simulation experiment, we compare the proposed EB-ET-DRAND algorithm with DRAND and E-T-DRAND algorithms from the message complexity, time complexity and energy consumption. And the distributed Color Constraint Heuristic for data aggregation (DSA-AGGR)

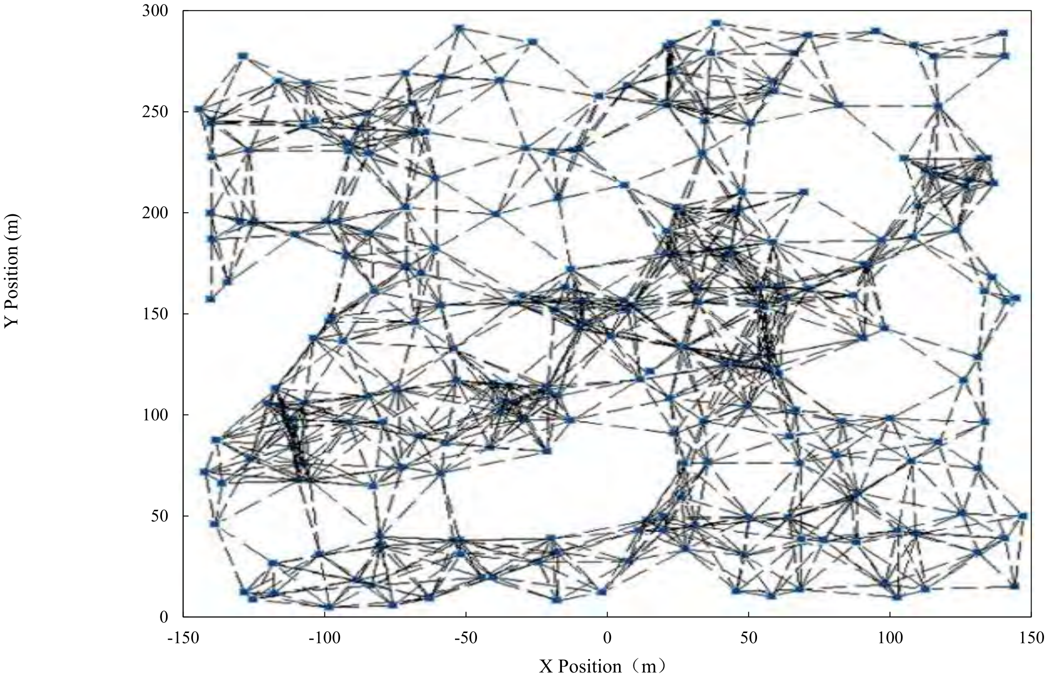


FIGURE 6. **The topology structure of OMNet++ simulation system.**

algorithm proposed by Bryan *et al.* [31], which is based on distributed slot assignment with the Color Constraint Heuristic (CCH), is also used as comparison. So the basic simulation parameter settings and conditions are configured according to the reference [25]. The specific simulation parameters are shown in Table 8. In this experiment, we perform 10 repeated experiments and calculate the average value as experimental results.

TABLE 8. **Simulation parameters.**

Parameter	Value
Link capacity	2Mbps
Broadcasting range	40m
Initial energy	1900mAh
Energy consumption for receiving	15mA
Energy consumption for transmitting	20mA
Energy consumption for idling	5mA
Radio operating time for transmitting or receiving	300us
Number of nodes(random distribution on 300m300m square)	250
Range of neighbor nodes	[5,55]
Average transfer delay	1s

C. EXPERIMENTAL RESULTS AND PERFORMANCE

We evaluate our proposed EB-ET-DRAND algorithm by comparing with DRAND, E-T-DRAND and DSA-AGGR algorithms from multiple aspects, including message complexity, time complexity and number of rounds, energy consumption. The detailed performance analyses are as follows:

1) MESSAGE COMPLEXITY

The message complexity, also known as the maximum number of messages, is the maximum number of messages required for all the nodes in a given topology to perform the time slot allocation algorithm. Fig. 7 shows the average number of messages per node for allocating time slot in different size of neighbor nodes. Generally, as the number of neighbor nodes increases, the number of messages sent by these algorithms increases. At the initial stage, the performances of all these algorithms are similar. With the increasing of the neighbor nodes number, the average number of transmission messages for the DRAND algorithm increases rapidly. For the DSA-AGGR algorithm based on colored aggregation tree, the average number of transmission messages is small because only a small amount of information are consider and aggregated. By taking the number of neighbor

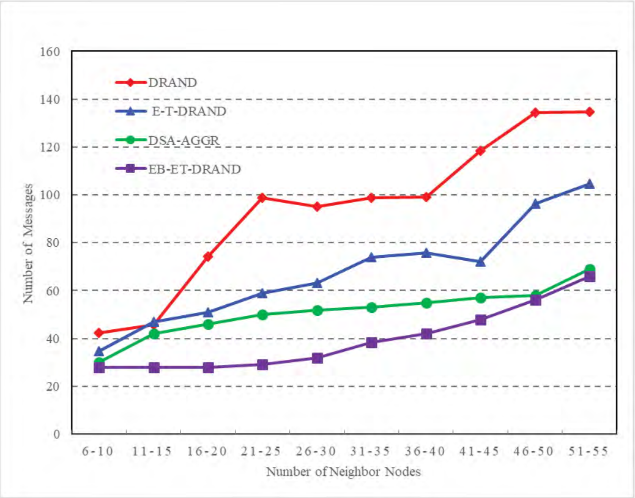


FIGURE 7. **The average number of messages per node for allocating time slot in different size of neighbor nodes.**

nodes and energy information into consideration, the average number of transmission messages for the E-T-DRAND algorithm increases slowly, and the average is 28.1% less than DRAND. Because of further optimizing the exponential backoff methods, the average number of transmission messages for the EB-ET-DRAND algorithm is 41.1% less than DRAND, which reduce the number of messages more significantly. The results indicate that our proposed algorithm can reduce the message complexity greatly.

2) TIME COMPLEXITY AND NUMBER OF ROUNDS

The time complexity, also known as the longest run time, is the maximum run time required for all the nodes in a given topology to perform the time slot allocation algorithm. Fig. 8 and Fig. 9 show the average running time and average number of rounds for successful allocating a time slot in different size of neighbor nodes. For the time slot allocation algorithm, the running time is proportional to the number of turns, so their trends are similar. As is shown in the figures, the EB-ET-DRAND algorithm achieves the best performance in all these time slot allocation algorithms. With the increase of the number of neighbor nodes, it is necessary to acquire time slots by through multiple rounds. So the running time and number of rounds for these algorithms are increasing along with the increase of the number of neighbor nodes. After the range of 36-40 in the x-axis, the running time and number of rounds for DRAND, E-T-DRAND and DSA-AGGR algorithms increase significantly and show no convergence. That is because the continuous time slot request causes a sharp increase of the collision when the neighbor nodes are too many. Further, the increase of time slot allocation failure results in the increases of running time and number of rounds. For the EB-ET-DRAND algorithm, the increases of running time and number of rounds are convergent. The reason is that the scheduling algorithm efficiently allocates the time slot resources according to exponential backoff rule and the priority control algorithms, and greatly reduces the probabilities of time slot allocation failure and the messages collision. The running time is reduced by 41.2% compared with that of DRAND algorithm, and the maximum can be reduced by 67.3%. The results show that our proposed algorithm is effective for reducing the running time and improving the efficiency of time slot allocation.

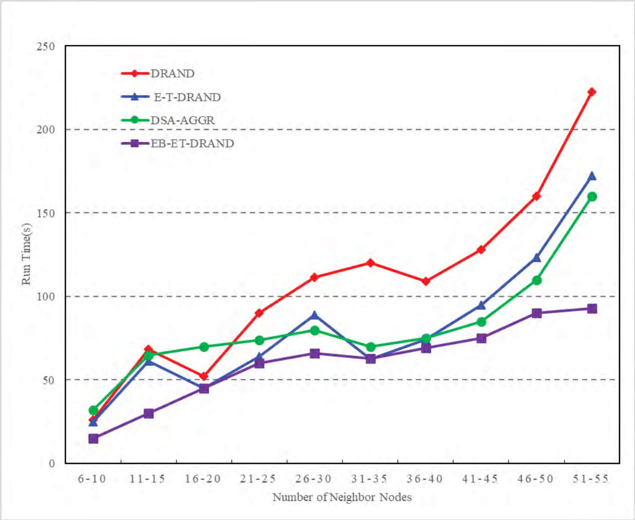


FIGURE 8. The average running time for successful allocating a time slot in different size of neighbor nodes.

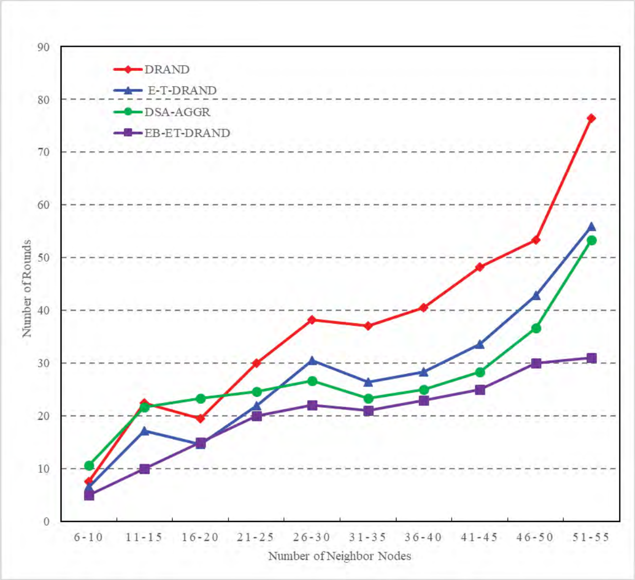


FIGURE 9. The average number of rounds for successful allocating a time slot in different size of neighbor nodes.

3) Energy consumption: By referring the energy model in OMNeT++ platform and the energy consumption analysis of reference [29], we can give the total energy consumption of EB-ET-DRAND algorithm by:

$$E_{total} = E_i + E_s + E_t + E_r. \tag{1}$$

Where E_i is the energy consumption in IDLE state, E_s is the energy consumption in SLEEP state but not discussed in this paper, that is, $E_s = 0$, E_t is the energy consumption in transmitting packets, E_r is the energy consumption in receiving packets.

According to the simulation results, we conduct energy consumption analysis of EB-ET-DRAND energy model. Fig. 10 shows the average energy consumption per node for EB-ET-DRAND algorithm in different size of neighbor nodes. Because the consumption of E_i and E_r is relatively large, the duration time of IDLE state has a great impact on the total system energy consumption. Therefore, the energy efficiency of time slot allocation can be improved notably by reducing duration time of IDLE state.

Fig. 11 shows the average energy consumption per node in different size of neighbor nodes. The E-T-DRAND algorithm reduces the energy consumption of the whole network by reducing the number of time slot allocation request. While EB-ET-DRAND algorithm reduces the energy consumption by not only reducing the number of time slot allocation request but also reducing the duration time of IDLE state. Therefore, EB-ET-DRAND achieves the best performance in

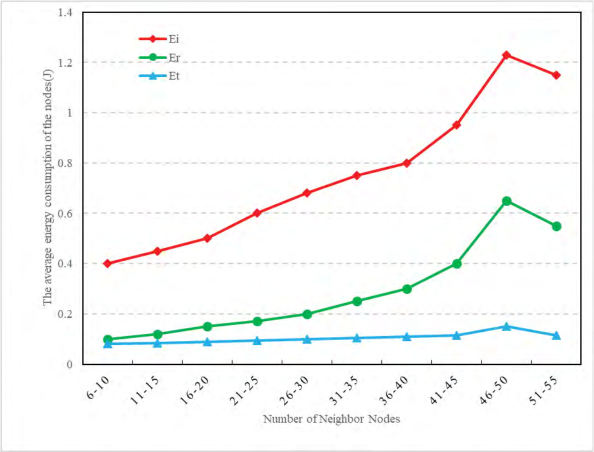


FIGURE 10. The average energy consumption per node for EB-ET-DRAND algorithm in different size of neighbor nodes.

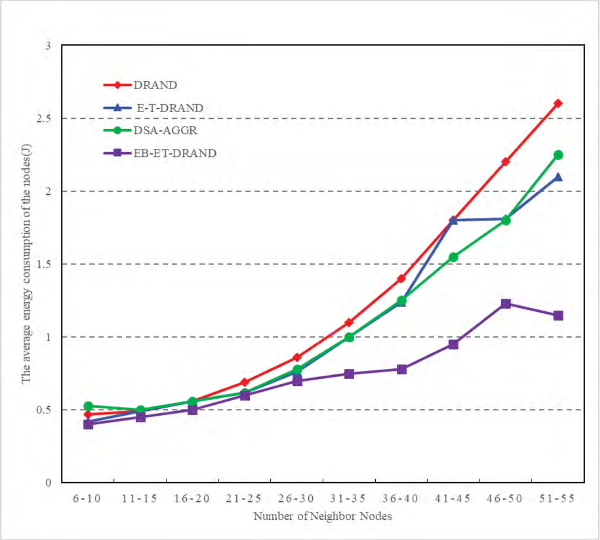


FIGURE 11. The average energy consumption per node in different size of neighbor nodes.

total energy consumption for all these time slot allocation algorithms. The energy consumption is reduced by 38.3% compared with that of DRAND algorithm, and the maximum can be reduced by 55.7%. The results indicate that the proposed EB-ET-DRAND algorithm greatly improves energy efficiency.

From the above simulation results and performance analysis, it is observed that the EB-ET-DRAND scheduling algorithm has greatly improved the performance of time slot allocation. It significantly reduces the message complexity, running time, number of rounds, and energy consumption. This will improve network efficiency and prolong network lifetime in the industrial application fields of IoT.

VI. CONCLUSION

In WSN applications, the efficient utilization of TDMA slot resource has attracted more and more attention. Especially, for some latency-sensitive applications (e.g., vehicular networks), the resource allocation scheme is great significance for the efficient and safe working of network. In generally, the WSN has limited battery volume and variable topology structure. These characteristics restrict the development of the centralized time slot allocation algorithm. Though the traditional distributed time slot allocation algorithms solve the variable topology to some extent, they are helpless to reduce the energy consumption. In this paper, by analyzing the typical DRAND algorithm and the E-T-DRAND algorithm, we have proposed the distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor, namely EB-ET-DRAND algorithm. By introducing the idea of Lamport's bakery algorithm, the exponential backoff and weight control rules have been presented to completely solve the collision problem and greatly reduce the probability of time slot allocation failure. The time slot structure and frame formats of algorithm have been also designed. By simulating the performance of proposed scheme on the network simulator OMNeT++ platform and comparing with some typical distributed time slot allocation algorithms, the experimental results indicate that the EB-ET-DRAND scheduling algorithm greatly improves the performance of time slot allocation and reduces the message complexity, time complexity and energy consumption. Research on the distributed TDMA scheduling algorithm has important academic and practical value, so it still needs further in-depth study in our future work.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] S. Li, L. Da Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[3] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing based face identification and resolution scheme in Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1910–1920, Aug. 2017.

[4] O. Kaiwartya *et al.*, "Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.

[5] Z. Ning, F. Xia, X. Hu, Z. Chen, and M. S. Obaidat, "Social-oriented adaptive transmission in opportunistic Internet of smartphones," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 810–820, Apr. 2017.

[6] Z. Ning, L. Liu, F. Xia, B. Jedari, I. Lee, and W. Zhang, "CAIS: A copy adjustable incentive scheme in community-based socially aware networking," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3406–3419, Apr. 2017.

[7] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 553–576, 1st Quart., 2016.

[8] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 1, pp. 86–99, Jan. 2012.

[9] C. Li, H. Zhang, B. Hao, and J. Li, "A survey on routing protocols for large-scale wireless sensor networks," *Sensors*, vol. 11, no. 4, pp. 3498–3526, 2011.

[10] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular social networks: Enabling smart mobility," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 16–55, May 2017.

[11] X. Ren, W. Liang, and W. Xu, "Data collection maximization in renewable sensor networks via time-slot scheduling," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 1870–1883, Jul. 2015.

[12] T. Qiu, R. Qiao, and D. Wu, "EABS: An event-aware backpressure scheduling scheme for emergency Internet of Things," *IEEE Trans. Mobile Comput.*, to be published. [Online]. Available: http://dx.doi.org/10.1109/TMC.2017.2702670

[13] Z. Ning, Q. Song, Y. Yu, Y. Lv, X. Wang, and X. Kong, "Energy-aware cooperative and distributed channel estimation schemes for wireless sensor networks," *Int. J. Commun. Syst.*, vol. 30, no. 5, p. e3074, 2017.

[14] J. Park, S. Lee, and S. Yoo, "Time slot assignment for convergecast in wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 83, pp. 70–82, Sep. 2015.

[15] Q. Chen, Y. Hu, Z. Chen, J. N. Davies, and P. Excell, "Smart guaranteed time-slot allocation algorithm for industrial wireless sensor networks emergency message transmission," *IET Wireless Sensor Syst.*, vol. 5, no. 2, pp. 76–86, 2015.

[16] J. Bhar, "A mac protocol implementation for wireless sensor network," *J. Comput. Netw. Commun.*, vol. 2015, Mar. 2015, Art. no. 697153.

[17] F. Iannello, O. Simeone, and U. Spagnolini, "Medium access control protocols for wireless sensor networks with energy harvesting," *IEEE Trans. Commun.*, vol. 60, no. 5, pp. 1381–1389, May 2012.

[18] X. Fafoutis, A. Di Mauro, C. Orfanidis, and N. Dragoni, "Energy-efficient medium access control for energy harvesting communications," *IEEE Trans. Consum. Electron.*, vol. 61, no. 4, pp. 402–410, Nov. 2015.

[19] I. Amdouni, R. Soua, E. Livolant, and P. Minet, "Delay optimized time slot assignment for data gathering applications in wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Unusual Confined Areas*, 2012, pp. 1–6.

[20] T. Qiu, Y. Lv, F. Xia, N. Chen, J. Wan, and A. Tolba, "ERGID: An efficient routing protocol for emergency response Internet of Things," *J. Netw. Comput. Appl.*, vol. 72, pp. 104–112, Sep. 2016.

[21] S. Gajjar, N. Choksi, M. Sarkar, and K. Dasgupta, "LEFT: A latency and energy efficient flexible TDMA protocol for wireless sensor networks," *Int. J. Comput. Netw. Inf. Secur.*, vol. 7, no. 2, pp. 1–14, 2015.

[22] V. Gabale, B. Raman, P. Dutta, and S. Kalyanraman, "A classification framework for scheduling algorithms in wireless mesh networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 199–222, 1st Quart., 2013.

[23] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Netw.*, vol. 16, no. 4, pp. 985–997, 2010.

[24] A. Bhatia and R. C. Hansdah, "A distributed TDMA slot scheduling algorithm for spatially correlated contention in WSNs," *Mobile Inf. Syst.*, vol. 2015, 2015, Art. no. 234143.

[25] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 10, pp. 1384–1396, Oct. 2009.

[26] C. Zhu and M. S. Corson, "A five-phase reservation protocol (FPRP) for mobile ad hoc networks," *Wireless Netw.*, vol. 7, no. 4, pp. 371–384, 2001.

[27] Y. Li, X. Zhang, J. Zeng, Y. Wan, and F. Ma, "A distributed TDMA scheduling algorithm based on energy-topology factor in Internet of Things," *IEEE Access*, vol. 5, pp. 10757–10768, 2017.

[28] A. Raniwala, K. Gopalan, and T.-C. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, 2003.

[29] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Netw.*, vol. 5, no. 2, pp. 81–94, Mar. 1999.

[30] K. Sato and S. Sakata, "A distance-measurement-oriented distributed TDMA scheduling algorithm for sensor networks," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. Workshops*, 2011, pp. 1–3.

[31] K. L. Bryan, T. Ren, L. DiPippo, T. Henry, and V. Fay-Wolfe, "Towards optimal TDMA frame size in wireless sensor networks," Univ. Rhode Island, South Kingstown, RI, USA, Tech. Rep., 2007.

[32] Y. Wang and I. Henning, "A deterministic distributed TDMA scheduling algorithm for wireless sensor networks," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.*, 2007, pp. 2759–2762.

[33] L. Lamport, "A new solution of Dijkstra's concurrent programming problem," *Commun. ACM*, vol. 17, no. 8, pp. 453–455, 1974.

[34] T. Qiu, X. Liu, M. Han, M. Li, and Y. Zhang, "SRTS: A self-recoverable time synchronization for sensor networks of healthcare IoT," *Comput. Netw.*, to be published. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2017.05.011

[35] K. Sato and S. Sakata, "A power-efficient distributed TDMA scheduling algorithm with distance-measurement for wireless sensor networks," *Wireless Pers. Commun.*, vol. 75, no. 2, pp. 1511–1528, 2014.

YIPING LI received the B.E. degree from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China, in 2015, where she is currently pursuing the M.S. degree. Her research interests include Internet of Things, wireless sensor network, and resource scheduling.

XIAOTONG ZHANG (M'11) received the M.S. and Ph.D. degrees from the University of Science and Technology Beijing, in 1997 and 2000, respectively. He was a Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing. His industry experience includes affiliation with Beijing BM Electronics High-Technology Co., Ltd., from 2002 to 2003, where he was involved in digital video broadcasting communication systems and IC design. His industrial cooperation experience includes BLX IC Design Company Ltd., North Communications Corporation of PetroChina, and Huawei Technologies Co., Ltd. His research includes work in quality of wireless channels and networks, wireless sensor networks, networks management, cross-layer design and resource allocation of broadband and wireless network, signal processing of communication, and computer architecture.

TIE QIU (M'12–SM'16) received the B.Sc. degree from the Inner Mongolia University of Technology in 2003 the M.Sc. and Ph.D. degree from the Dalian University of Technology, China, in 2005 and 2012, respectively. He is currently an Associate Professor with the School of Software, Dalian University of Technology. He was a Visiting Professor in electrical and computer engineering with Iowa State University, USA from 2014 to 2015. He has authored/co-authored eight books and over 60 scientific papers in international journals and conference proceedings, such as ToN, TMC, TII, IEEE Communications, *Computer Networks*. He has contributed to the development of for copyrighted software systems and invented 15 patents. He is a Senior Member of China Computer Federation and the ACM. He serves as the general chair, PC chair, workshop chair, publicity chair, publication chair, or TPC member of a number of conferences. He serves as an Associate Editor of the IEEE ACCESS Journal, the *Computers & Electrical Engineering* (Elsevier journal) and the *Human-Centric Computing and Information Sciences* (Springer Journal), an Editorial Board Member of the *Ad Hoc Networks* (Elsevier journal) and the *International Journal on AdHoc Networking Systems*, a Guest Editor of the *Future Generation Computer Systems* (Elsevier journal).

JUN ZENG received the B.E. and M.S. degrees from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China, in 2013 and 2016, respectively. She is currently a Software Engineer in Sina Weibo, China. Her research interests include Internet of Things, wireless sensor network, and resource scheduling.

PENGFEI HU (S'16) received the B.E. degree from the School of Computer Science, Zhengzhou University of Aeronautics, China, in 2012. He is currently pursuing the Ph.D. degree from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He focuses on the objects modeling in cyber-physical space convergence and Internet of Things. His research interests include Internet of Things, identification and resolution of physical objects, and cyber-physical modeling.

• • •