IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 April 2024, accepted 5 May 2024, date of publication 8 May 2024, date of current version 15 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3398348

RESEARCH ARTICLE

A Rapid Flooding Approach Based on Adaptive Delay and Low-Power Sleep for BLE Mesh Networks

LI WANG, (Member, IEEE), **JUNXIANG LI**, AND **MINGXIA LI**
School of Software, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

Corresponding author: Junxiang Li (lijunxiang@mail.nwpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61901388, in part by the Fundamental Research Funds for the Central Universities, and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515110406.

ABSTRACT Bluetooth Low Energy Mesh (BLE Mesh) is a promising protocol to interconnect massive transceivers in a low-power, high-flexibility and wide-coverage way. It applies managed flooding to realize efficient many-to-many communication, especially in the small packet scenario. However, the traditional approaches usually generate too much redundant packets, resulting in severe collision and even broadcast storm. Related works deal with this problem mainly from the aspects of transmission mode, probabilistic broadcast and node selection. These algorithms decrease the redundancy a lot but often come at a considerable delay. In this paper, we propose a rapid flooding approach based on adaptive delay and low-power sleep(FADLS) for BLE Mesh, which adjusts the maximum random delay and relay sleep time adaptively, based on a lightweight neighbor information acquisition mechanism. The simulation results show that the FADLS algorithm reduces the broadcast collision and transmission delay effectively. In addition, the adaptability enables the algorithm more suitable for changing scales of BLE Mesh networks.

INDEX TERMS BLE mesh, rapid flooding, adaptive delay, relay feature.

I. INTRODUCTION

In the age of Internet of Things (IoT), IoT devices experience an explosive growth [1]. Research and Markets [2] indicates that there will be more than 30 billion devices in 2035. Bluetooth Low Energy (BLE) [3] is of low power and high compatibility [4], thus has been widely used in IoT especially to interconnect sensors [5]. However, BLE is a short-range wireless technology mainly used in one-to-one and one-to-many communications, which limits the development towards wider area and more device application [6], [7]. As a promising protocol [8] proposed by Bluetooth Special Interest Group (SIG) in 2017, BLE Mesh is enhanced to achieve greater coverage and higher flexibility. In addition, unlike the routing technique used to realize many-to-many communication in WiFi [9], ZigBee [10] and Thread [11], managed flooding is used in BLE Mesh with the benifit of lower cost, lower overhead and lower latency.

Managed flooding is a simple and effective approach for multiple connections, since it shrinks the data scale through Time to Live(TTL) and limits the redundant relay through cache of received packets. Nevertheless, it is still critical to guarantee the delay caused by collisions on the limited broadcast channels, in particular when plenty nodes broadcast concurrently [12]. On the one hand, a straightforward solution is to introduce a random delay between receiving and transmitting of packets at the relays. On the other hand, an improper delay may even increase the total latency instead of reducing the broadcast collision [13]. Therefore, it is a challenge to design a rapid flooding algorithm which achieves a good tradeoff between broadcast collision and transmission delay.

The associate editor coordinating the review of this manuscript and approving it for publication was Nurul I. Sarkar.

Some studies change the transmission mode of BLE Mesh to reduce broadcast collisions. Raul Rondon et al. [14] proposed that nodes no longer transmit sequentially according to channels 37, 38, 39, and use the order of randomized broadcast channels to solve conflicts between multiple relays receiving packets simultaneously. This approach can avoid the contention of multiple nodes for the same channel on a certain scale and reduce broadcast collisions. Min-Yue Wang et al. [15] proposed an algorithm called ACE that uses all 37 Bluetooth data channels for data transmission. BLE Mesh node selects a data channel for continuous scanning and uses three broadcast channels to notify neighbouring nodes to establish a routing table. Neighbouring nodes use different data channels for data transmission, alleviate broadcast collisions issues. However, these algorithms need to note compatibility issues with standard BLE Mesh devices.

Another kind of algorithm uses probabilistic broadcasting to reduce redundant packets, and then reduces the probability of collision. GOSSIP1 [16] adopted a fixed probability approach. The node generates a random number before relaying. If the value is greater than the relay probability, the relay is performed. GOSSIP2 [17] algorithm proposed to dynamically adjust the relay probability according to the number of neighbors. This algorithm improves the adaptability to different environments, but may lead to network partitioning. APF [18] algorithm introduces two probability bounds P_{max} and P_{min}, and dynamically adjusts the relay probability according to the number of neighbors of the node. APF makes the relay probability more flexible to adapt to different network environments, especially in low-density network environments. It can improve the probability of packet transmission by adjusting the probability bounds, so as to ensure the connectivity of the network. However, the process of threshold adjustment depends on experience. At the same time, the algorithm does not give a specific scheme to obtain the information of neighbor nodes. Raqeebir Rab et al. [19]. proposed a new heuristic algorithm called dynamic probabilistic broadcast (DPB) algorithm. In this approach, the broadcast probability of each node is determined dynamically from a self-tailoring analysis model. The algorithm effectively solves the problem of setting probability thresholds by modelling network and transceiver parameters with broadcast probabilities.

Probabilistic broadcasting method can effectively reduce broadcast collision, but there are also problems of network connectivity and probability threshold setting based on experience without standardization. Therefore, some studies configure the relay features by selecting suitable nodes at the beginning of network establishment [20]. Using as few relay nodes as possible can reduce broadcast collisions and shorten the random relay delay while ensuring network connectivity. This kind of algorithms generally uses the Connected Dominating Set (CDS) [21] approach to select relay nodes, which can effectively achieve network coverage by minimising the number of relay nodes. According to the devices of algorithm execution, they can be divided into two categories: centralized and distributed. Genetic algorithm (GA) [22] is a centralized algorithm. The algorithm needs to obtain the topology information of the entire network, and the calculation complexity is high, and the calculation depends on a single node. Greedy Connect algorithm (GCA) [23] is a distributed relay node selection algorithm. Although the relay node set obtained by this algorithm is not the smallest, the algorithm requires less computing power of nodes and has higher reliability. However, after the node features are set, the role in the network is fixed. When a relay node for some reason (e.g., power failure, move out of the network), the network connectivity will be affected [23].

Table 1 shows the existing algorithms in terms of compatibility with standard protocols, adaptability, and running devices.

TABLE 1. List of simulation experiment parameters.

Algorithm	Year	Compatibility	Self-adaptability	Distributed
ACE	2021	×	✓	✓
GOSSIP1	2002	✓	×	✓
GOSSIP2	2009	✓	×	✓
APF	2009	✓	×	✓
DPB	2016	✓	×	✓
GA	2018	✓	×	×
GCA	2020	×	×	✓
FADLS	2024	✓	✓	✓

As far as we know, the existing researches reduce the broadcast collision in BLE Mesh from different aspects [24], but often come at a considerable delay [25]. This paper proposes a rapid flooding approach based on adaptive delay and low-power sleep (FADLS) for BLE Mesh. The FADLS algorithm achieves rapid and low broadcast collision flooding through two core mechanisms: adaptive maximum random relay delay, relay low-power sleep mechanism. The main contributions are as follows.

1) Adaptive maximum random relay delay. According to the change of neighbor relay nodes, relay nodes calculate the optimal maximum random delay. This ensures that packets are rapidly transmitted with a low chance of broadcast collision.
2) Relay low-power sleep mechanism. During network operation, the mechanism can adjust the relay features based on received packets and neighbor information. This achieves network coverage with a minimum number of relay nodes, effectively reducing broadcast collisions. This mechanism also avoids the problems of single point of failure and manual configuration of node characteristics.
3) Lightweight neighbor information acquisition mechanism. A lightweight neighbor information acquisition mechanism was designed to solve the problem that the BLE Mesh packet payload is small and difficult to carry the neighbor node set.

The remainder of this paper is as follows. Section II mainly introduces the simulation model of BLE Mesh network. Section III presents the detailed design of the DAM algorithm. Section IV explains the configuration of the simulation and its results. Finally, conclusion is given in Section V.

II. SYSTEM MODEL

Traditional BLE devices can only communicate with connected devices. By introducing BLE Mesh, BLE devices can communicate with any device in the network. In addition to this, the BLE Mesh nodes can also be configured with different features, giving them special functions. The four main features are as follows:

1) Relay feature - the ability to receive and relay packets over the advertising bearer to enable larger networks.
2) Proxy feature - the ability to receive and relay packets between GATT and advertising bearers.
3) Low Power feature - the ability to operate within a mesh network at significantly reduced receiver duty cycles only in conjunction with a node supporting the Friend feature.
4) Friend feature - the ability to help a node supporting the Low Power feature to operate by storing messages destined for those nodes.

Relay nodes in BLE Mesh are essential for BLE to break short distance transmission. As shown in Figure 1, this is a simple BLE Mesh scenario. Through the relay node, our mobile phone does not need to establish a connection to control the fan outside its Radio-frequency (RF) range. All relay nodes in the network forward the control messages sent to the fan by the mobile phone.

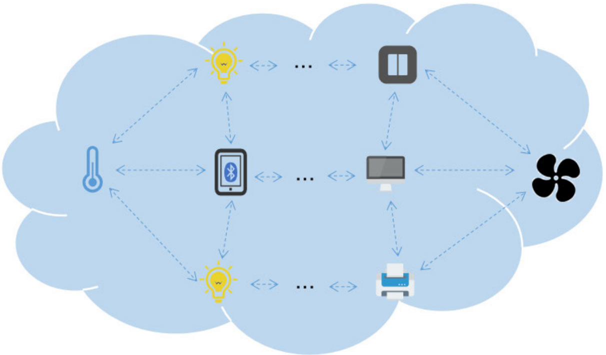


FIGURE 1. Example topology of a bluetooth mesh.

After understanding the network topology and node features of BLE Mesh, the next will introduce the specific transmission process of packets in BLE mesh. Through a network such as Figure 2, we analyze the the transmission delay and collision probability of BLE Mesh in the process of transmitting packets in detail.

In this network, the source node and the destination node are not within the RF range of each other, and the packets are relayed through two relay nodes. The propagation delay is not considered in the calculation of the transmission delay. Because the propagation speed of electromagnetic waves in the air is 3.0×10^8m/s, the propagation delay of packets is very small. The specific transmission process is shown in Figure 3.

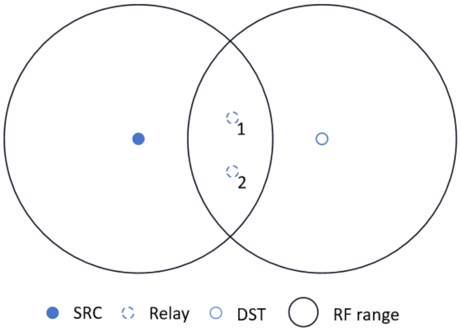


FIGURE 2. Scene model.

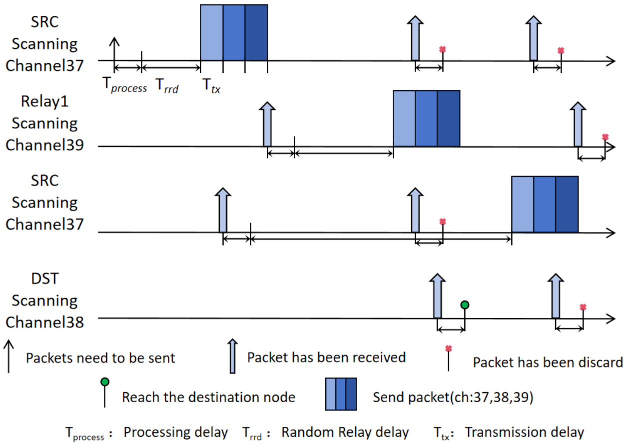


FIGURE 3. Example of communication flow between four devices in Bluetooth mesh.

The source node must send a packet to another node. The source node processes the packet first. Once the random delay has expired, the source node broadcasts packets sequentially on channels 37, 38, and 39. To ensure that no network packets are lost, the relay node's receiver duty cycle should be set to 100%. However, since nodes may be scanning different channels, the time at which they receive the packet may vary. For instance, as depicted in Figure 3, $Relay_1$ and $Relay_2$ receive packets on different channels. Once the packet is received, the relay node processes it and determines whether a relay is necessary. To prevent collisions between multiple relays, the protocol uses a random relay delay when relaying is necessary. Once this delay expires, the relay node will transmit the packet through the broadcast channel.

The packet relayed by $Relay_1$ is received by the source node. The source node then verifies the key information in the packet to confirm that it was sent by itself. If so, discard. Similarly, when $Relay_2$ receives a packet relayed by $Relay_1$, it checks if it has already received the packet and drops it if it has.

The destination node is within the RF range of $Relay_1$ and receives its packet. It determines that the packet is intended for itself and passes it to the upper layer of the protocol stack for processing.Once the random relay delay of $Relay_2$ has expired, all nodes within its RF range will have received the packet, and subsequently drop it.

Figure 3 shows that random relay delay can reduce the probability of broadcast collision, but it also increases the transmission delay of packets. We will mathematically model the packet transmission delay and collision probability, using the parameters explained in Table 2.

TABLE 2. **List of parameters used in the modeling of transmission delay and collision probability.**

Parameter	Explanation
T_{Delay}	End-to-end transmission delay
T_{rrd}	Random relay delay
$T_{process}$	Packet processing delay
T_{tx}	The sending delay of a single packet
h/hop	Hop number of the packet relay
x	1 indicates that the receiving node detects the packet on channel 37
	2 indicates that the receiving node detects the packet on channel 38
	3 indicates that the receiving node detects the packet on channel 39
RRD_{max}	Maximum random relay delay
E[T]	Expectation of minimum random relay delay
R	Number of relay nodes that participate in relaying
P_{col}	The probability of collision

The transmission delay is mainly composed of three parts. They are processing delay, random relay delay and sending delay. It can be expressed as:

$$T_{Delay} = T_{process} + T_{rdd} + x \times T_{tx} \tag{1}$$

Data packets often need to travel multiple hops to reach the destination node. Therefore, the transmission delay of data packet can be expressed as follows:

$$\begin{aligned} T_{\text{Delay}} &= T_{\text{process},1} + T_{\text{rrd},1} + x_1 \times T_{\text{tx}} + \ldots \\ &\quad + T_{\text{process},h} + T_{\text{rrd},h} + x_h \times T_{\text{tx}} \\ &= \text{hop} \times T_{\text{process}} + \sum_{h=1}^{hop} (T_{\text{rrd},h} + x_h \times T_{\text{tx}}) \end{aligned} \tag{2}$$

Assuming that there are R_h relay nodes in each hop, let the minimum transmission delay in each hop be:

$$\begin{aligned} &T_{\text{process}} + T_{\text{rrd},h,r} + x_{h,r} \times T_{\text{tx}} \\ &= T_{\text{process}} + \min\big(T_{\text{rrd},1} \\ &\quad + x_1 \times T_{\text{tx}}, \ldots, T_{\text{rrd},h} + x_h \times T_{\text{tx}}\big) \end{aligned} \tag{3}$$

Therefore, the transmission delay in BLE Mesh network can be expressed as follows:

$$T_{Delay} = hop \times T_{process} + \sum_{h=1}^{hop} (T_{rdd,h,r} + x_{h,r} \times T_{tx}) \tag{4}$$

During the relaying process of a hop, R nodes are involved in relaying. Each node generates a random relay delay independently and randomly within the range of $[0, RRD_{max}]$. These random values can be considered as independent and identically distributed (IID) random variables $(X_1, \ldots, X_R)$. According to D. Seita's calculation [26], if R IID variables take random values in [0,1], the minimum value can be expected to be:

$$E[T] = \frac{1}{R+1} \tag{5}$$

When R IID variables take random values in $[0, RRD_{max}]$, $E[T]$ can be expressed as:

$$E[T] = \frac{RRD_{max}}{R+1} \tag{6}$$

We analyze the packet collision probability through a simple model. The time for a node to send a packet on a channel is T_{tx}. The maximum random relay delay set by the node is RRD_{max}.

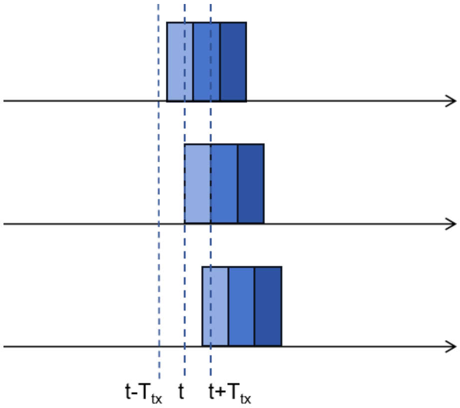


FIGURE 4. **Broadcast collision model.**

As shown in Figure 4, two relay nodes receive and prepare to relay the same packet simultaneously. One of the nodes transmits at time t. The probability of broadcast collision is equal to the probability that another node transmits between $[t - T_{tx}, t + T_{tx}]$. Then the collision probability (P_{col}) between two relay nodes can be expressed as follows:

$$P_{col} = \frac{2 \times T_{tx}}{RRD_{max}} \tag{7}$$

If there are R relay nodes that simultaneously receive and relay the same packet, then P_{col} can be expressed as:

$$P_{col} = 1 - (1 - \frac{2 \times T_{tx}}{RRD_{max}})^{R-1} \tag{8}$$

III. ALGORITHM DESIGN

A. OVERALL DESIGN OF RAPID FLOODING APPROACH BASED ON ADAPTIVE DELAY AND LOW-POWER SLEEP

This section presents the details of the FADLS algorithm. The FADLS algorithm has two core points: an adaptive maximum random relay delay and a relay low-power sleep mechanism.

The adaptive maximum random relay delay allows BLE Mesh to send packets quickly with a low collision probability. BLE Mesh can achieve network coverage with the least number of relay nodes by using a relay low-power sleep mechanism. When the relay feature of a node changes, it informs its neighbor nodes through a heartbeat message.

The neighbor nodes can then adjust the maximum random relay delay accordingly. Figure 5 shows the execution flow of the FADLS algorithm.

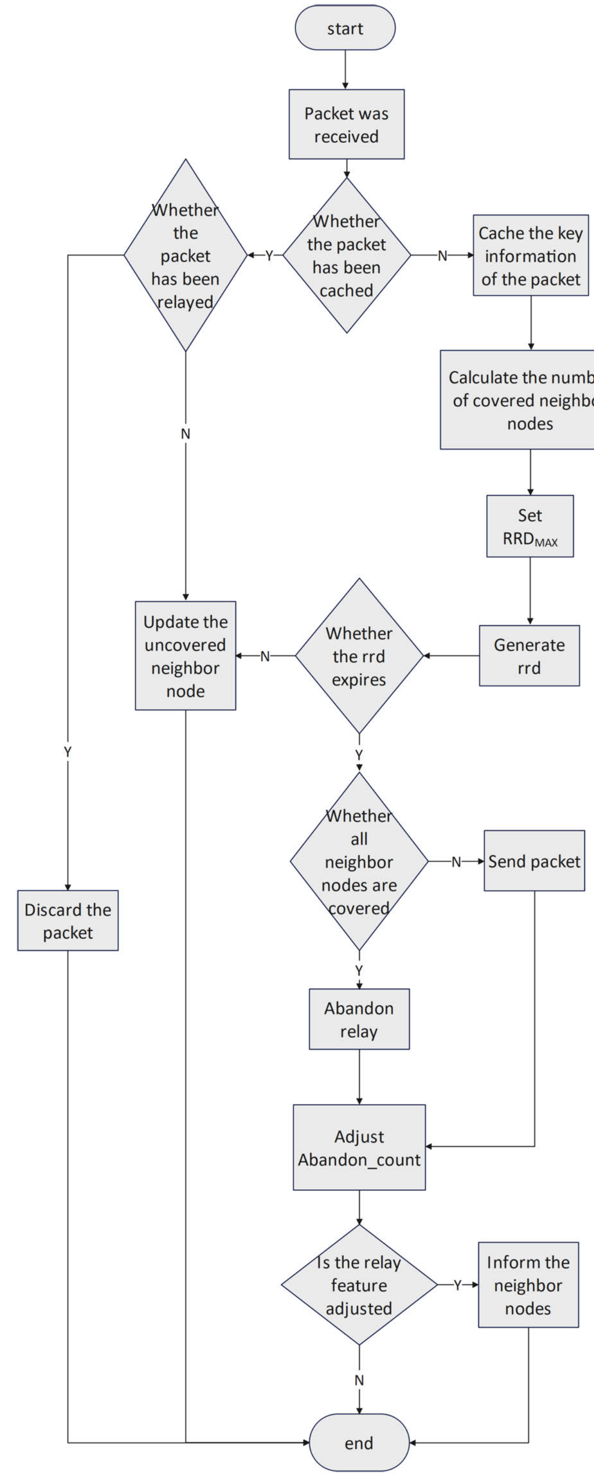


FIGURE 5. **Flowchart of the dynamic adaptive management flooding algorithm.**

Algorithm A: Dynamically Adjust Maximum Random Relay Delay

```
WHILE (A packet was received) DO
    IF (The packet has been cached) THEN
        IF (The packet has been relayed) THEN
            Discard the packet;
        ELSE
            Update the uncovered neighbor node;
        END IF
    ELSE
        Cache the key information of the packet;
        Calculate the number of covered neighbor nodes;
        Set RRD_MAX based on number of neighbours covereds;
    END IF
END WHILE
```

When a node receives a packet, it first determines whether the packet has been cached. If the packet has been cached and relayed, discard the packet. If the neighbor node is not relayed, the node update the uncovered neighbor node. If the packet has not been cached, the key information of the packet should be cached. Then, the node sets the RRD_{max} based on the number of covered neighbor nodes. When the RRD of a packet expires, the node decides whether to relay the packet based on the information of the neighbor node. The node will also adjust the relay feature based on the number of packets abandoned. If the relay feature changes, the node inform the neighbor node.

In the following, we will specifically introduce the design of two core points in the algorithm and lightweight neighbor information acquisition mechanism.

B. ADAPTIVE MAXIMUM RANDOM RELAY DELAY

Equation 6 shows that there is a positive correlation between the expectation of the minimum random relay delay and the maximum random relay delay. Equation 8, however, demonstrates a negative correlation between the collision probability and the maximum random relay delay. Therefore, nodes should adjust the maximum random relay delay adaptively based on neighbor information. To reduce the probability of packet collision in areas with a high node density, increase the maximum random relay delay. In areas with few nodes, reduce the maximum random forwarding relay to spread the message quickly.

The FADLS algorithm calculates the number of covered neighbor nodes when the relay node receives the packet for the first time. The maximum random relay delay is then dynamically adjusted based on this value. Below is the pseudocode for the algorithm:

C. RELAY LOW-POWER SLEEP MECHANISM

Equation 6 demonstrates a positive correlation between the expectation of minimum random relay delay and the number of relay nodes involved in the relay process. Equation 8 shows a negative correlation between the collision probability and the number of relay nodes participating in the relay.

Algorithm B: Dynamically Adjust the Relay Feature

```
IF (Random relaying delays expiration) THEN
    IF (All neighbor nodes have been covered) THEN
        Abandon relay;
        Abandon_count = Abandon_count + 1;
    ELSE
        Send packet;
        Abandon_count = Abandon_count - 1;
    END IF
    IF (Relay feature on) THEN
        IF (Abandon_count >= Threshold value) THEN
            Switch off relay feature;
            Use the heartbeat message to inform the neighbor nodes;
            Abandon_count = Threshold value;
        ELSE IF (Abandon_count <= 0) THEN
            Abandon_count = 0;
        END IF
    ELSE
        IF (Abandon_count <= 0) THEN
            Switch on relay feature;
            Use the heartbeat message to inform the neighbor nodes;
        ELSE IF (Abandon_count >= Threshold value) THEN
            Abandon_count = Threshold value;
        END IF
    END IF
END IF
```

Therefore, the node must adaptively adjust the relay feature based on the number of packets abandoning the relay. To achieve network coverage, the minimum number of relay nodes should be used. This technique can effectively reduce broadcast collisions. The adjustment of relay features is communicated to neighbouring nodes through heartbeat messages. This allows for a smaller maximum random relay delay to be set, which speeds up packet transmission.

The mechanism's pseudocode is as follows:

D. LIGHTWEIGHT NEIGHBOR INFORMATION ACQUISITION MECHANISM

To accommodate the changing scale of the BLE Mesh network, this paper optimises the heartbeat message mechanism. The mechanism is designed to acquire neighbour information in a lightweight manner and includes the following two points:

1) To adjust the TTL value in the heartbeat message to two, the receiving node creates and maintains a list of two-hop neighbor information, as shown in Figure 6.
2) Adjust the frequency of sending heartbeat messages. In creating a BLE mesh, each node generates a list of neighbor information by periodically sending heartbeat messages. When the network structure stabilises, heartbeat messages are no longer sent periodically. They are only sent when the node's feature changes, to notify neighbouring nodes of the change.

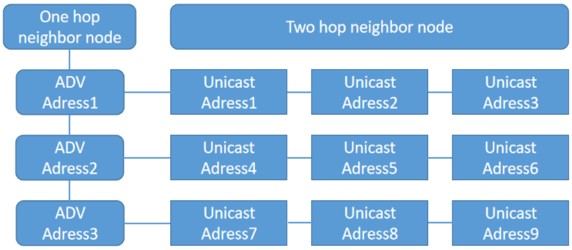


FIGURE 6. **Neighbor node information.**

Designing a lightweight neighbour information acquisition mechanism reduces the network load and provides the necessary information for FADLS. In this way, the algorithm adapts to the changing size of the BLE mesh network.

IV. SIMULATION RESULTS

A. SIMULATION EXPERIMENT DESIGN

In BLE Mesh networks, the number of nodes and topology have important effects on algorithm performance [27]. So we evaluated the FADLS algorithm in various BLE Mesh network scales.

Due to the complexity of developing a real test platform [28], simulation results have become a major benchmark for BLE Mesh [29]. To ensure the reliability of the simulation results, we conducted real experiments in the network scenario shown in Figure 2 to obtain device parameters. According to the Poisson distribution model, the packets sent from the source node to the destination node are set to reflect the random transmission of packets in the real network environment. The packet size is set according to the BLE Mesh standard protocol.

For networks with different number of nodes, we randomly generated 1000 connected network topologies based on the uniform distribution model. This satisfies the complexity of node placement in real network environments [30]. At the beginning of network operation, the relay feature of all nodes are kept on [31]. As the network runs, the FADLS algorithm achieves dynamic changes in the network configuration by adjusting the feature of the nodes.

We used Python to implement the simulation of BLE mesh and algorithm. Refer to Table 3 for detailed parameters.

TABLE 3. **List of simulation experiment parameters.**

Parameter	Value
Network Size	100m × 100m
Number of Nodes	100-1250
Network Topology	Uniform Random
Communication Range	15m
Physical layer rate	1Mbps
Packet Size	47Byte
Number of packets	100

As shown in Figure 7, this is an example network scenario with 120 nodes. The source and destination nodes are represented by red nodes 1 and 120, respectively. Node 80 has an RF range represented by the green circle, and nodes within its range are shown in green. Since the entire network is connected, the nodes will have neighbouring nodes within their RF range.

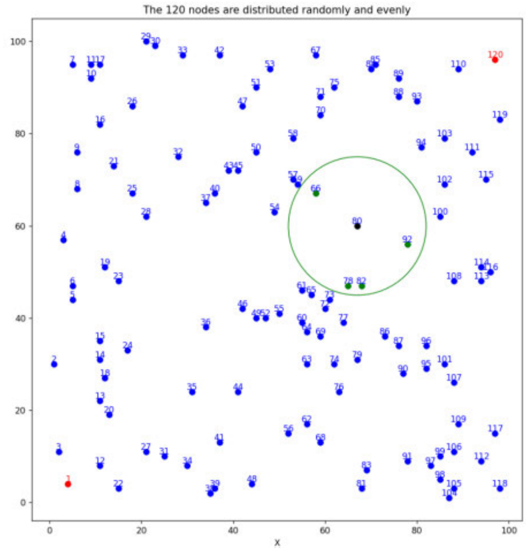


FIGURE 7. A network scenario with 120 nodes randomly and evenly.

We compared the FADLS algorithm with the Managed-flooding algorithm of the standard protocol and the more advanced DPB algorithm in several aspects.

B. ANALYSIS OF RESULTS

We counted the number of collisions generated during the delivery of 100 packets from the source node to the destination node. The following conclusions can be drawn from Figure 8:

1) The total number of packet collisions in FADLS algorithm is less than that in the other two algorithms.
2) FADLS algorithm increases the number of collisions more slowly.

FADLS algorithm reduces redundant relay nodes through relay sleep mechanism. According to the information of neighbor nodes, the maximum random relay delay was adjusted to reduce the collision probability. Its adaptability enables it to show better performance in changing scales of BLE Mesh networks.

We counted the transmission delay of a packet from the source node to the destination node. The following conclusions can be drawn from the Figure 9:

1) The transmission delay of FADLS algorithm is lower than the other two algorithms.
2) The transmission delay of FADLS algorithm can maintain low delay in changing scales of BLE Mesh networks.

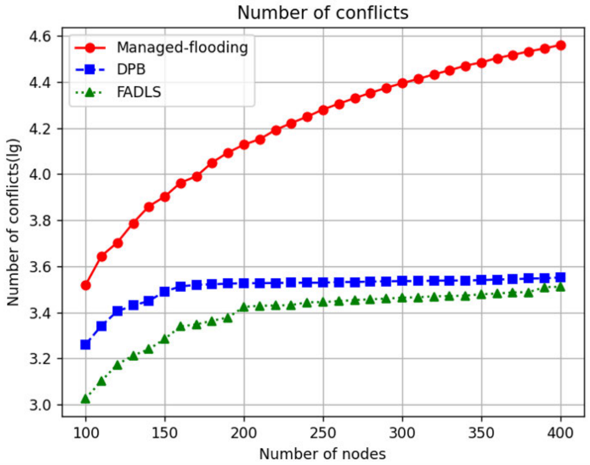


FIGURE 8. Relationship between the number of collisions and the number of nodes.

FADLS algorithm adjusts the maximum random relay delay according to the information of neighbor nodes. When the number of nodes in the network increases, FADLS algorithm reduces redundant relay nodes through relay sleep mechanism. Keeping active relay nodes can further reduce the maximum random relay delay to transmit packets faster.

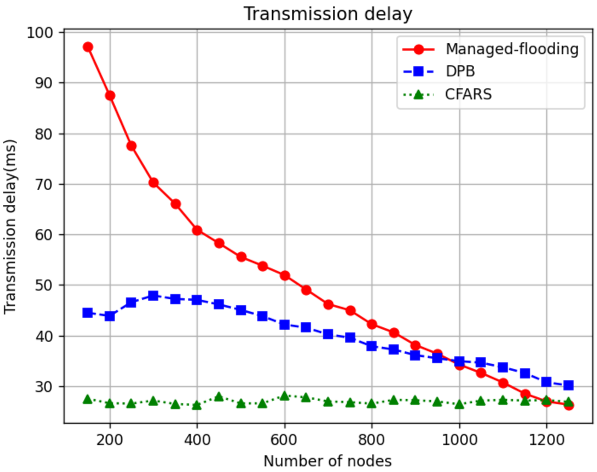


FIGURE 9. Relationship between transmission delay and the number of nodes in the scenario.

We counted the proportion of nodes involved in relaying in the process of sending packets from the source node to the destination node. From Figure 10 we drew the following conclusions:

1) FADLS algorithm can effectively reduce the redundant relay nodes.
2) The number of nodes participating in the relay in the management flooding algorithm gradually increases to 100%.
3) The management flooding algorithm based on neighbor information optimization can also limit the relay of redundant nodes to a certain extent.

The FADLS algorithm uses the relay sleep mechanism to retain only the critical nodes that keep the network connected. Covering an area of a certain size, the number of nodes used is roughly the same, so as the number of nodes increases, the proportion of nodes that maintain the relay feature will gradually decrease.

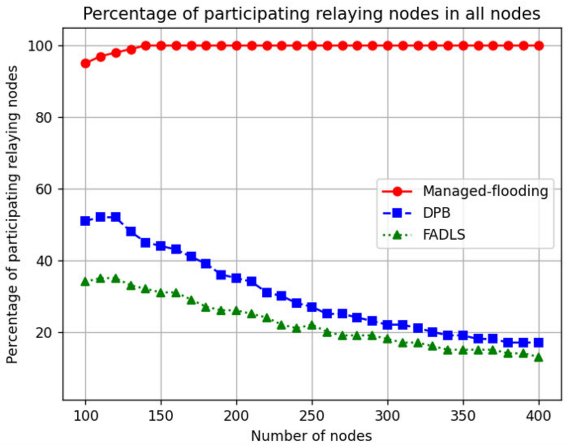


FIGURE 10. **Proportion of the number of nodes involved in the relay.**

The source node of the three algorithms sends 100 packets to the destination node. We calculate the packet processing delay, random relay delay, and sending delay for all nodes in the network. From Figure 11 we can learn:

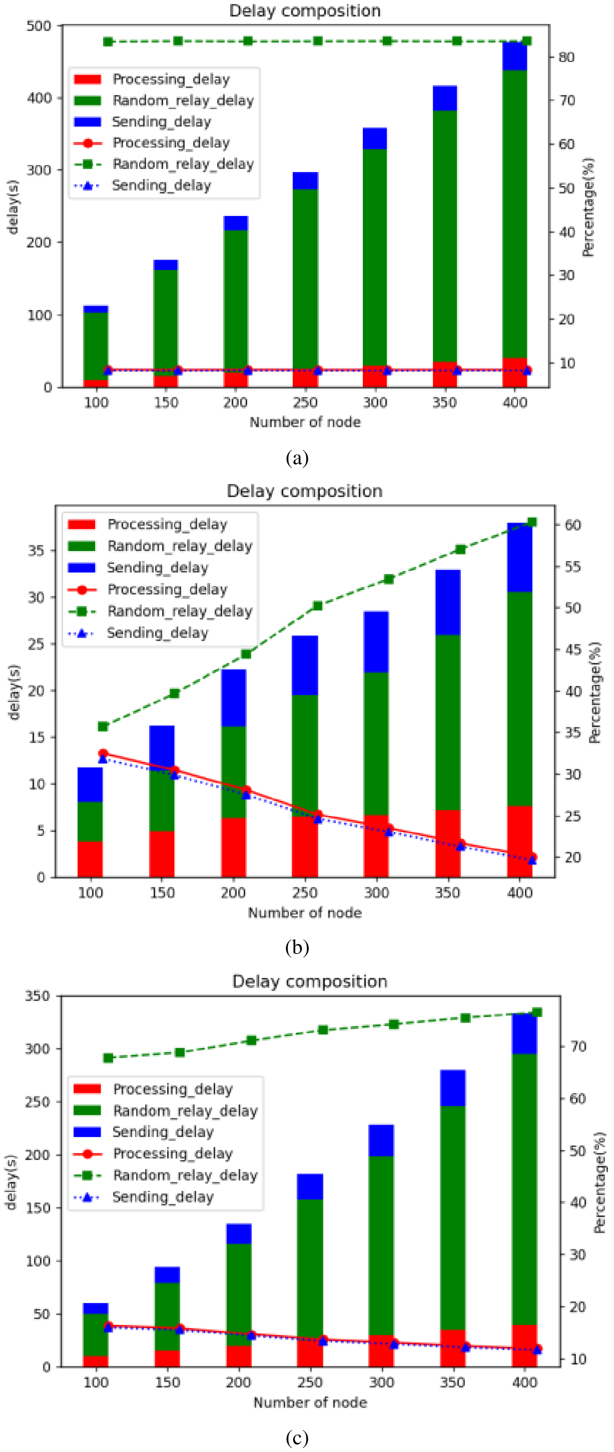
1) The time taken to transmit 100 packets using the managed flooding algorithm increases gradually. The proportion of processing delay, random relay delay and sending delay is unchanged.
2) The total delay of the FADLS algorithm is significantly lower than the other two algorithms. The proportion of random relay delay is gradually increasing.

As the number of network nodes increases, the number of nodes involved in relaying increases, and hence the total delay increases. FADLS algorithm reduces the number of relay nodes participating in the relay through the relay sleep mechanism according to the information of neighbor nodes, so the total transmission delay is less than the other two algorithms. The maximum random relay delay is adjusted with the number of nodes in the network. So it is sensitive to the number of nodes. When the number of nodes increases, the maximum random relay delay increases in order to avoid collisions, so the proportion of random relay delay increases.

We counted the number of hops used by each packet to reach the destination node. Figure 12 shows the the experimental results.

1) The number of hops used by all three algorithms is gradually decreasing.
2) The FADLS algorithm has a faster descent rate.

As the number of nodes increases, the probability that a relay node exists at the edge of the RF range of the sending node increases. These nodes relay packets to reduce the number of hops. In FADLS algorithm, the maximum random relay delay of the relay node which is farther from the sender node is smaller. In this way, the path with the least number of hops can be found faster, while the path with more hops will be blocked.



(a)

(b)

(c)

FIGURE 11. **The delay and delay ratio of the Managed-flooding algorithm(a), the FADLS algorithm(b) and the DPB algorithm(c).**

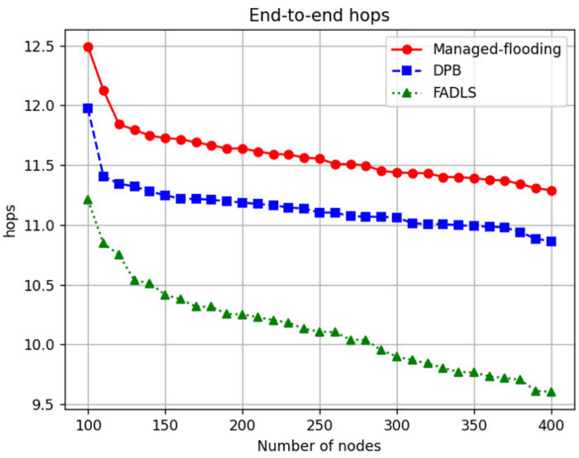


FIGURE 12. The number of hops used during packet transmission.

V. CONCLUSION

The FADLS algorithm proposed in this paper reduces broadcast collision and propagation delay in BLE Mesh networks. The algorithm is adaptable to changes in BLE Mesh network scale. The neighbor information acquisition mechanism designed in this paper is more lightweight and flexible than the heartbeat message in the protocol. In summary, the FADLS algorithm enables BLE Mesh to achieve a larger network size and greater flexibility.

In this paper, we focus on reducing broadcast collision and propagation delay. However, there is a lack of research on the energy consumption of BLE Mesh [32], [33]. BLE Mesh data transmission is based on broadcast and scanning, which requires nodes to continuously scan to prevent packet loss. This approach is not conducive to the long-term operation of battery-powered nodes.

In the future work, We will enhance the long-term stability of the network by improving the relay sleep mechanism in future work. Our design will promote lower energy consumption by exploring the interaction between sleeping and active nodes.

REFERENCES

[1] M. S. Sharbaf, "IoT driving new business model, and IoT security, privacy, and awareness challenges," in *Proc. IEEE 8th World Forum Internet Things (WF-IoT)*, Oct. 2022, pp. 1–4.

[2] M. Al Rawajbeh, S. AlZu'bi, and A. Alkhatib, "IoT-paradigm: Evolution challenges and proposed solutions," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Sep. 2023, pp. 1–5.

[3] *Bluetooth Core Specification Version 5.1*, Bluetooth Special Interest Group, Kirkland, WA, USA, 2019.

[4] K. Sornalatha and V. R. Kavitha, "IoT based smart museum using Bluetooth low energy," in *Proc. 3rd Int. Conf. Adv. Electr., Electron., Inf., Commun. Bio-Inform. (AEEICB)*, Feb. 2017, pp. 520–523.

[5] L. Leonardi, G. Patti, and L. Lo Bello, "Multi-hop real-time communications over Bluetooth low energy industrial wireless mesh networks," *IEEE Access*, vol. 6, pp. 26505–26519, 2018.

[6] P. Pierleoni, A. Gentili, M. Mercuri, A. Belli, R. Garello, and L. Palma, "Performance improvement on reception confirmation messages in Bluetooth mesh networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2056–2070, Feb. 2022.

[7] D. Pérez-Díaz-De-Cerio, Á. Hernández-Solana, M. García-Lozano, A. V. Bardají, and J.-L. Valenzuela, "Speeding up Bluetooth mesh," *IEEE Access*, vol. 9, pp. 93267–93284, 2021.

[8] *Bluetooth Mesh Profile Specification*, Bluetooth Special Interest Group, Kirkland, WA, USA, 2017.

[9] Y. Ge, A. Taha, S. A. Shah, K. Dashtipour, S. Zhu, J. Cooper, Q. H. Abbasi, and M. A. Imran, "Contactless WiFi sensing and monitoring for future healthcare–emerging trends, challenges, and opportunities," *IEEE Rev. Biomed. Eng.*, vol. 16, pp. 171–191, 2023.

[10] C. A. G. Silva, E. L. Santos, A. C. K. Ferrari, and H. T. S. Filho, "A study of the mesh topology in a ZigBee network for home automation applications," *IEEE Latin Amer. Trans.*, vol. 15, no. 5, pp. 935–942, May 2017.

[11] W. Rzepecki and P. Ryba, "IoTSP: Thread mesh vs other widely used wireless protocols—Comparison and use cases study," in *Proc. 7th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2019, pp. 291–295.

[12] A. Hernandez-Solana, D. Perez-Diaz-De-Cerio, M. Garcia-Lozano, A. V. Bardaji, and J.-L. Valenzuela, "Bluetooth mesh analysis, issues, and challenges," *IEEE Access*, vol. 8, pp. 53784–53800, 2020.

[13] M. Baert, J. Rossey, A. Shahid, and J. Hoebeke, "The Bluetooth mesh standard: An overview and experimental evaluation," *Sensors*, vol. 18, no. 8, p. 2409, 2018.

[14] R. Rondón, A. Mahmood, S. Grimaldi, and M. Gidlund, "Understanding the performance of Bluetooth mesh: Reliability, delay, and scalability analysis," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2089–2101, Mar. 2020.

[15] M. Wang, Y. Li, J. Lv, Y. Gao, C. Qiao, B. Liu, and W. Dong, "ACE: A routing algorithm based on autonomous channel scheduling for Bluetooth mesh network," *Electronics*, vol. 11, no. 1, p. 113, Dec. 2021.

[16] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Societies*, Jun. 2002, pp. 1707–1716.

[17] A. Mohammed, M. Ould-Khaoua, L. M. Mackenzie, and J.-D. Abdulai, "Dynamic probabilistic counter-based broadcasting in mobile ad hoc networks," in *Proc. 2nd Int. Conf. Adapt. Sci. Technol. (ICAST)*, Sep. 2009, pp. 120–127.

[18] A. M. Hanashi, I. Awan, and M. Woodward, "Performance evaluation with different mobility models for dynamic probabilistic flooding in MANETs," *Mobile Inf. Syst.*, vol. 5, no. 1, pp. 65–80, 2009.

[19] R. Rab, A. Rahman, and F. T. Zohra, "Analytical modeling of self-pruning and an improved probabilistic broadcast for wireless multihop networks," *Ad Hoc Netw.*, vol. 52, pp. 106–116, Dec. 2016.

[20] M. Reno, R. Rondón, L. L. Bello, G. Patti, A. Mahmood, A. Lombardo, and M. Gidlund, "Relay node selection in Bluetooth mesh networks," in *Proc. IEEE 20th Medit. Electrotechnical Conf. (MELECON)*, Jun. 2020, pp. 175–180.

[21] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 1, pp. 14–25, Jan. 2002.

[22] E. A. J. Hansen, M. H. Nielsen, D. E. Serup, R. J. Williams, T. K. Madsen, and R. Abildgren, "On relay selection approaches in Bluetooth mesh networks," in *Proc. 10th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Nov. 2018, pp. 1–5.

[23] A. Bęben, A. Bąk, and M. Sosnowski, "Efficient relay node management method for BLE MESH networks," *Int. J. Electron. Telecommun.*, pp. 29–35, Sep. 2019.

[24] I. Natgunanathan, N. Fernando, S. W. Loke, and C. Weerasuriya, "Bluetooth low energy mesh: Applications, considerations and current state-of-the-art," *Sensors*, vol. 23, no. 4, p. 1826, Feb. 2023.

[25] A. Hawbani, X. Wang, A. Abudukelimu, H. Kuhlani, Y. Al-Sharabi, A. Qarariyah, and A. Ghannami, "Zone probabilistic routing for wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 728–741, Mar. 2019.

[26] D. Seita. (2016). *The Expectation of the Minimum of IID Uniform Random Variables*. Accessed: Jan. 13, 2024. [Online]. Available: https://danieltakeshi.github.io/2016/09/25/the-expectation-of-the-minimum-of-iid-uniform-random-variables/

[27] J. Silvestre-Blanes, J. C. G. Ortiz, and V. M. Sempere-Payá, "Dynamic autonomous set-up of relays in Bluetooth mesh," in *Proc. IEEE 28th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2023, pp. 1–7.

[28] B. Lin and H. Su, "Experimental evaluation and performance improvement of Bluetooth mesh network with broadcast storm," *IEEE Access*, vol. 11, pp. 137810–137820, 2023.

[29] D. G. Reina, S. L. Toral, P. Johnson, and F. Barrero, "A survey on probabilistic broadcast schemes for wireless ad hoc networks," *Ad Hoc Netw.*, vol. 25, pp. 263–292, Feb. 2015.

[30] J.-Y. Jung, D.-Y. Seo, and J.-R. Lee, "Counter-based broadcast scheme considering reachability, network density, and energy efficiency for wireless sensor networks," *Sensors*, vol. 18, no. 2, p. 120, Jan. 2018.

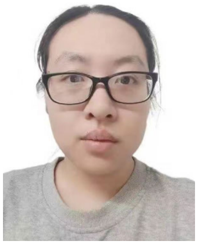
[31] M. Zhuravinskyi, D. Maletskyi, I. Kravets, O. Kapshii, and O. Farenyuk, "Decentralized provisioning algorithm for BLE mesh network," in *Proc. IEEE 18th Int. Conf. Comput. Sci. Inf. Technol. (CSIT)*, Oct. 2023, pp. 1–4.

[32] D. Villa, C.-K. Lin, A. Kuenzi, and M. Lang, "Bluetooth low energy mesh network for power-limited, robust and reliable IoT services," 2022, *arXiv:2208.04050*.

[33] S. M. Darroudi, R. Caldera-Sànchez, and C. Gomez, "Bluetooth mesh energy consumption: A model," *Sensors*, vol. 19, no. 5, p. 1238, 2019.

LI WANG (Member, IEEE) received the B.E. degree in communications engineering from Shandong University, Jinan, China, in 2009, and the M.S. and Ph.D. degrees in communications engineering from Xidian University, Xi'an, China, in 2012 and 2018, respectively. She was a Visiting Student with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada. She is currently an Assistant Professor with the School of software, Northwestern Polytechnical University, Xi'an. Her research interests include the IoT networks, resource allocation, and interference management.

JUNXIANG LI received the B.E. degree in software engineering from Qingdao University, Qingdao, China, in 2021. He is currently pursuing the master's degree with Northwestern Polytechnical University. He continues to deepen his knowledge and expertise in the field of computer science and engineering with Northwestern Polytechnical University. With a passion for exploring the intersection of technology and innovation. His research interests include the IoT networks, communication protocols, and routing algorithms. His academic pursuits are driven by a desire to contribute to the advancement of the technology landscape and address the challenges presented by the evolving digital world.



MINGXIA LI received the B.E. degree in software engineering from Northwestern Polytechnical University, Xi'an, China, in 2021, where she is currently pursuing the master's degree with the School of Software. Her research interests include the IoT networks, interference management, and integrated sensing and communication (ISAC).

• • •