IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 4, 2019, accepted December 18, 2019, date of publication January 3, 2020, date of current version January 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963723

A Smart Collaborative Routing Protocol for Delay Sensitive Applications in Industrial IoT

MINGQIANG ZHU[1], LIU CHANG[2], NAN WANG[3], AND ILSUN YOU[4], (Senior Member, IEEE)

[1]School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[2]Network Technology Research Institute, China Unicom, Beijing 100044, China
[3]Dawning Information Industry (Beijing) Company, Ltd., Beijing 100193, China
[4]Department of Information Security Engineering, Soonchunhyang University, Asan 31538, South Korea

Corresponding author: Ilsun You (ilsunu@gmail.com)

This work was supported by the Soonchunhyang University Research Fund and National Natural Science Foundation of China (No. 61872033).

ABSTRACT In the industrial Internet of things (IIoT), there is always a strong demand for real-time information transfer. Especially when deploying wireless/wired hybrid networks in smart factories, the requirement for low delay interaction is more prominent. Although tree routing protocols have been successfully executed in simple networks, more challenges in transmission speed can be observed in the manufacturing broadband communication system. Motivated by the progresses in deep learning, a smart collaborative routing protocol with low delay and high reliability is proposed to accommodate mixed link scenarios. First, we establish a one-hop delay model to investigate the potential affects of Media Access Control (MAC) layer parameters, which supports the subsequent design. Second, forwarding, maintenance, and efficiency strategies are created to construct the basic functionalities for our routing protocol. Relevant procedures and key approaches are highlighted as well. Third, two sub-protocols are generated and the corresponding implementation steps are described. The experimental results demonstrate that the end-to-end delay can be effectively cut down through comprehensive improvements. Even more sensor nodes and larger network scale are involved, our proposed protocol can still illustrate the advantages comparing with existing solutions within IIoT.

INDEX TERMS Industrial IoT, deep learning, routing protocol, tree topology, delay.

I. INTRODUCTION

According to statistics, the number of IoT devices continues to grow rapidly. It has reached 8.4 billion by 2017 and it is expected to reach 30 billion by 2020, which is far more than the current global population combined [1]. The IIoT is one of the new emerging concept with the development of sensor networks. Its goal is to realize the combination of people, machines and data, improve production efficiency [2]. The core content is that by taking advantage of high-precision intelligent sensor networks, achieve smart production and maintenance, improve the fabricating efficiency,reduce the manufacturing costs, extend the life of equipment and save non-renewable energy [3], [4]. In the traditional industrial scene, wired network is the main way of data communication. However, intelligent sensors based on wireless technology could enable data to be transmitted, published and Shared directly, which could cover various types of networks through wireless links [3]. Wireless communication method is able to provide high-bandwidth and flexible topology for the intelligent control equipment, mobile patrol robot and automated production line in factory [4]. More importantly, under some special circumstances, wireless network can effectively supplement the deficiency of wired network, further improve the coverage of industrial control network, and powerfully enhance its comprehensive communication performance. But there's no denying that with the development and application of high rate IoT devices, IIoT has been facing the explosion of data volume. This poses a serious challenge to the existing routing protocols in communication [5]. In the industrial scene, the routing technology required for networking is an extremely significant link in the supporting technology of IIoT, which has attracted more and more researchers' attention [6], [7].

At present, wireless local area network, cellular digital communication network and space satellite transmission

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

network have been widely utilized in civil field [2], [4]. However, facing the situations such as massive data, multi-hop links, node movement and poor link channels, all of them need to timely complete the complex tasks. For instance, adjustment and optimization of base stations, satellites and ground facilities for many times to ensure the communication quality. Wireless communication method has developed rapidly in recent year. The demand for fast, efficient and flexible networking is increasingly prominent [3], [8]. And the existing industrial environment is usually not conducive to wireless communication. Obviously, dust, smoke, extremely high or low temperatures, electrical noise levels, vibrations, locations underground or surrounded by thick concrete structures can all seriously destroy the reliability of wireless communications. Reliability and certainty are necessary for most industrial applications, so the reality is that IIoT often needs to employ the integration of wired and wireless through IP. Then, we realize the low-cost support for both wireless and wired protocols through the development of a single device [3], [4].

In recent years, wireless self-organizing network has become the focus of communication researchers due to its characteristics of non-centrality, self-organizing, multi-hop routing and dynamic topology. It has shown great application value in the fields of national defense, military, rail transit and emergency rescue. More realistically, as a local area network, wireless mesh network is more ideal for covering indoor industrial environments. With this kind of network, data can be transmitted directly to the upper system without passing through the base station. Moreover, data can be encrypted in various ways to ensure real-time, which secure interaction of logistics information in factory. In addition, logistics activities in factories has a very high standard of real-time data interaction, and delay is very easy to cause economic losses. In a decentralized wireless mesh network, any node can send and receive signals, and the transmission rate is fast. At the same time, if the nearest node fails or is disturbed, the data can jump to the alternate path to continue the transmission and maintain the stable operation of the network. The problem of signal delay is avoided to make the data exchange efficiently, which cannot meet the real-time demand of industrial production. Then, in order to solve this problem, the Internet task group specially established the mobile ad hoc network (MANET) engineering group, and proposed many routing algorithms that can adapt to dynamic topology. MANET is a kind of network where nodes move arbitrarily. Its topological structure is highly unstable. These nodes can form networks and transmit information as they move. Therefore, MANET needs to be capable of automatic creation, self-organization and management. Many research institutions and universities in the world have devoted themselves to the research of wireless Ad hoc network routing algorithms and obtained rich achievements [1]–[8].

Research on IIoT routing protocol based on wireless is a relatively complex and systematic engineering filed. Smart factory is a typical application of IIoT, which makes relevant routing protocols have to face complex mixed scenarios [2]. Especially when wired and wireless links are applied at the same time and a large number of fixed and mobile nodes coexist, massive industrial data often needs to be forwarded through intermediate nodes for multiple hops, and each hop will result at least some unavoidable delays [9], [10]. With the expansion of industrial production and the increasing number of sensor applications in the Internet of things, the scale of networking will also expand [4]. The problem is obvious – the more hops, the greater the cumulative total delay. At present, the solution to this problem is to add mesh nodes and use appropriate network protocols. Therefore, it is necessary to design a highly reliable and low delay routing protocol suitable for IIoT [11], [12]. Under that all kinds of nodes can simultaneously communicate to multiple other nodes through hybrid network, which could meet the requirements of multi-party communication and networking [13]. At the same time, the delay could be reduced as much as possible to ensure real-time information interaction and transmission [14]. This paper focuses on the optimization of the delay characteristics of the protocol. The purpose is to design a routing protocol which can meet the requirement of low delay and adapt to the complex mixed link scenario in smart factory environment. Then, it can effectively reduce the routing overhead brought by packet, improve the reliability and availability of industrial Internet. To sum up, the main work of our paper is as follows:

1. In order to study the applicable scenarios and workflow of the protocol, we establish a one-hop delay model for data transmission in IIoT.
2. The data packet forwarding and topology maintenance strategy are improved reliably, then, the wired link priority and data multicast scheme are also being introduced considering delay sensitivity.
3. The routing algorithm are designed to implement the protocol, and corresponding performance is validated via various experiments.

The rest of this paper is organized as follows. Section II summarizes the related works. The one-hop delay model in IIoT was described in Section III. In this paper four main improvements to the low delay routing protocol(message forwarding strategy, route maintenance strategy, multicast and wired link priority strategies) are presented in Section IV, followed by proposed protocol implementation process in Section V. Then, in Section VI, we evaluate the performance of our protocol and compare it with existing main application protocols. Finally, the conclusion and future work are presented in Section VII.

II. RELATED WORKS

In a smart factory, if a wireless Ad hoc network is utilized for mixed wired and wireless networking, the main problem is that the network topology will constantly change with a large number of moving nodes. How to find the best route to the destination in shortest time is a very critical issue. Traditional internet-based routing protocols face many challenges. Firstly, it can't adapt to changing and constantly

moving scenes. Secondly, it needs more control messages to exchange routing information, which will produce a large overhead. In addition, the existing main routing protocols are generally lack of delay reduction schemes.

Researchers all over the world have been studying the routing protocols of wireless network for many years, peoples have made a number of achievements and obtained fruitful results. In 2018, Dhiviya *et al.* [15] proposed an energy-aware multicast Ad hoc on demand distance vector routing protocol The proposed scheme is based on multicast Ad hoc on demand distance vector protocol. It finds energy-efficient multicast routes from source node to a group of destination nodes. However, when its topology changes, its computational load and excessive consumption will increase dramatically, which cannot adapt to mixed scenarios well. Wang *et al.* [16] proposed an energy efficient cross-layer routing protocol. It is suitable for wireless sensors containing wake-up receivers. In order to save energy, the protocol employ the different transmission ranges of wake-up and main radios by skipping nodes during data transferring. If this method is used when the node density is high, the message cannot be forwarded directly and still needs to be sent to this point through sibling nodes, which cannot meet the requirement of reducing delay. In Wireless Sensor Networks, Carlier *et al.* [17] concluded by analysis that for low power and lossy networks, many internet protocols rely on the routing tree obtained based multicast. This conclusion points out one part of many wireless routing protocols which should be improved. Employing a mixed mode can optimized the whole average delay. A threshold determines from how many interested children onwards a broadcast should be used. Abo-Zahhad *et al.* [18] proposed rendezvous-based routing protocol. It creates a rendezvous region and constructs a tree in the middle of the network. This protocol contains two data transfer method, it brings the difficulty of management mechanism design. To improve the lifetime of wireless networks, Zhou *et al.* [19] proposed an collaborative distributed antenna routing protocol. It doesn't talk much about transmission delay characteristics. Y. Chen et al. proposed a new multicast routing protocol. It constructed multiple multicast trees and employed network coding [20]. This multicast solution brings a practical idea to reduce redundant packets. Reference [21]–[23] focused on the energy balance problems in wireless network routing protocol. However, when the packet needs to be transmitted through the root node with too many times, link congestion will occur in the nodes near the root node, which will lead to queuing and increase the delay, and even cause packet loss and other problems. In reference [24], [25] researchers studied the tree routing protocol. New packet sending and maintenance tree strategies are proposed respectively. However, their research focuses on the scenario of wireless mobile network, which is different from the actual needs of IIoT. Wu *et al.* [26] proposed a dynamic tree recombination strategy to reduce the delay caused by multicast. Its premise is that there are several independent trees in the topology and the network topology inside the kernel tree is relatively stable. Wang *et al.* [27] proposed a distributed priority tree-based routing protocol. It could be utilized in stereo space. A similar solution is also proposed by Ai *et al.* [28]. In mutually coordinated system, it can handle data transmission. However, this routing protocol is relatively complex, and it is very easy to cause network interruption once the links between nodes are unstable.

III. ONE-HOP DELAY MODELING

In many practical application scenarios of IIoT, wired and wireless hybrid links are often included, and fixed and mobile node forms also exist. Smart factories usually contain an overall control network with global management and monitoring capabilities. In our project, we take the network of a local production unit as an example, and the other parts of that can be extrapolated. At last, the final conclusion can be extended to the whole network. Its network functions include: the highest production control center and other secondary production control institutions to communicate with each other, the control institutions and various types of production equipment, inspection robots and grass-roots management node to communicate with each other. The system can give orders, all nodes can be quickly networking, and in the premise of low delay, to meet the needs of data transmission, voice and video service. As shown in Fig. 1, both wired and wireless links exist in the complete scene (in Fig. 1, the thick line is the wired link and the thin line is the wireless link). Each node has a corresponding level, and there are strict delay requirements for data communication between nodes. Currently, routing protocols of wired networks, which including rip and ospf, are only applicable to the large-scale fixed networks. And it is not to industrial production scenarios with certain mobility and complex environment. For a typica network with strict delay requirement, active routing protocol is more suitable. That is, when a node needs to send data, it can send it directly as long as the route to the destination node exists, so the delay is small.

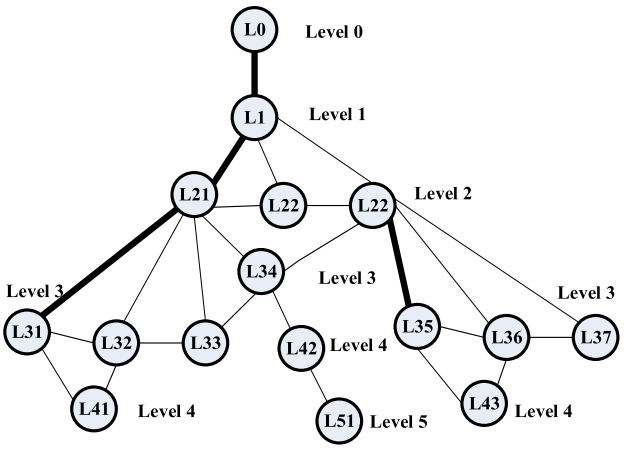

FIGURE 1. Frame format.

However, the nodes do not need to interact with all routing information. They only need to announce routing information in the process of joining the tree and maintain neighbors through message hello to achieve the purpose of maintaining the tree topology. Compared with other active routing protocols, this approach has a small overall overhead. In the case of wired link in this scenario, since its link stability is higher than that of wireless link, its bandwidth is larger, and its deployment is more fixed. The routing strategy can choose wired link in priority, so as to make full use of the advantage of large bandwidth. Furthermore, we introduce tree topology to make most nodes choose the parent node with wired link to complete the process of joining the tree, which can balance the load and reduce the delay of packet queuing, thus reducing the end-to-end delay. In this scenario, the parent node needs to distribute commands to the subordinate node. If multicast mode is adopted, the superior node only needs to send data to the subordinate node once, so the utilization rate of bandwidth resource could be effectively improved.

To achieve this goal, as a key factor the kernel tree routing protocol should be studied and improved. The kernel tree routing protocol belongs to the active routing protocol, and the network can be deployed in the form of tree topology. At the same time, the nodes inform each other of the link status through the message hello. The node maintains the routing table in real time so that it does not need to reinitiate the routing process when it decides to send data, thus reducing the delay. At that point, in view of the characteristics of industrial scenarios, a low delay routing protocol based on tree topology is designed. Therefore, according to the actual requirements of communication networking in smart factory scenario mode, the highest production control center can be set as the root node of this tree, which is connected with other secondary control centers through wired lines. Each equipment, device and personnel, as nodes of the tree, network and communicate with each other through routing protocol.

The establishment of node one hop delay model is the key of protocol research. The MAC of wireless network studied in this paper is suitable for IEEE 802.11 protocol The process of back-off mechanism is represented by $b(t)$ and adopts discrete integer time scale. t and $t+1$ are respectively used to represent the starting time of two continuous time slots. The withdrawal timer is started at the beginning of the time slot. $s(t)$ represent the back-off order (0,1…, m). We have the following Settings.

$$W_i = 2^i W, \tag{1}$$

W is the minimum value of the competition window, $0 < i < m$, and m is the maximum back-off order. At the starting of each time slot, the node detects the channel. If the channel is idle, the back-off counter is reduced by 1. If it is busy, the counter remains unchanged. The node does not send data until the value of the counter is reduced to zero. Therefore, the actual service time can be modeled and analyzed by random process $\{s(t), b(t)\}$.

We model the random process $\{s(t), b(t)\}$ with a two-dimensional discrete Markov chain, and then analyze the actual service time, as shown in Fig. 2. The prerequisite for the establishment of this model is

a. There are no hidden nodes in the network.

b. n nodes within a hop range are all in the state of saturation, that is, every node has data waiting to be sent at any moment.

c. when node sends data, the probability of conflict is p, and it is considered fixed.

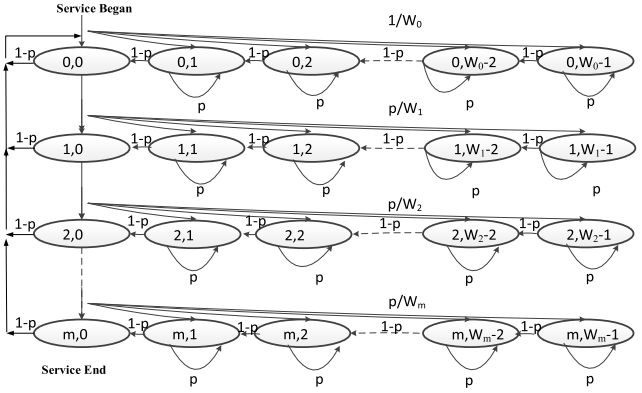

FIGURE 2. Markov model.

In the horizontal direction, a single step from right to left represents a decrease of 1 in the node back-off counter. In the vertical direction, the top-down one-step state transition indicates that the node has a collision. The one step transfer probability of this Markov chain is as follows:

1) When the node detects that the wireless channel is idle, the probability of its back-off counter being reduced by 1 is set as $P\{i, k|i, k+1\}$, it is computed by

$$P\{i, k|i, k+1\} = 1-p, \quad i \in (0, m), \ k \in (0, W_i-2). \tag{2}$$

2) When the node detects that the wireless channel is busy, the probability of freezing its retreat back-off counter is $P\{i, k|i, k\}$, it is computed by

$$P\{i, k|i, k\} = p, \quad i \in (0, m), \ k \in (0, W_i - 1). \tag{3}$$

3) When order i back-off ends and the back-off counter is reduced to 0, the data is successfully sent back to order 0 back-off. The probability that the value of the randomly selected back-off counter as k is $P\{0, k|i, 0\}$, it is computed by

$$P\{0, k|i, 0\} = \frac{1-p}{W_0}, \quad i \in (0, m-1), \ k \in (0, W_i-1). \tag{4}$$

4) If the data transmission fails, it will enter the $i+1$ order back-off. The probability that the value of the randomly selected back-off counter as k is $P\{i, k|i-1, 0\}$, it is computed by

$$P\{i, k|i-1, 0\} = \frac{p}{W_i}, \quad i \in (0, m), \ k \in (0, W_i-1). \tag{5}$$

5) When it reaches the maximum back-off order m, which represents the end of retreat. The counter value is reduced to zero, and then begins to send data, the node will reset the race window to the initial value. At this point for the next data, the randomly probability of selecting value of the retreat counter as k is $P\{0, k|m, 0\}$, it is computed by

$$P\{0, k|m, 0\} = \frac{1}{W_0}, \quad k \in (0, W_0 - 1). \tag{6}$$

Take $b_{i,k}$ as the steady-state distribution probability of Markov chain, and its calculation formula is

$$b_{i,k} = \lim_{n\to\infty} P\{s(t) = i, b(t) = k\}. \tag{7}$$

$b_{i,0}$, $b_{0,k}$, $b_{i,k}$ are respectively computed by Equ. (8)-(9).

$$b_{i,0} = b_{0,0} \times p_i, \tag{8}$$

$$b_{0,k} = \frac{b_{0,0}(W_0 - k)}{W_0(1-p)}, \tag{9}$$

$$b_{i,k} = \frac{b_{i,0}(W_i - k)}{W_i(1-p)}. \tag{10}$$

According to the steady-state distribution normalization condition, the following formula is established

$$\sum_{i=b}^{m} \sum_{k=0}^{W_i-1} b_{i,k} = 1. \tag{11}$$

Suppose that in a network with n nodes, each node is within the communication range of other nodes, and the packet arrival rate of any node I is λ_i. Saturation occurs when any node is sending packets. The probability of packet sent by node I in the free time slot is τ, while the probability of data transmission failure is p, and T_C is the average duration of a transmission conflict. The probability of any node in the system transmitting data in a random time slot can be expressed as

$$\begin{aligned} \tau &= \sum_{i=0}^{m} b_{i,0} \\ &= \sum_{i=0}^{m} p_i b_{0,0} \\ &= \frac{b_{0,0}\left(1 - p^{m+1}\right)}{1-p} \end{aligned} \tag{12}$$

The expression of data transmission failure probability is

$$p(n) = 1 - (1-\tau)^{n-1}. \tag{13}$$

By Equ. (8) and (9) can work out the numerical solution of p and τ.

In order to obtain the transmission delay of MAC, assume that the transmission rate of wireless channel is R bit/s, the length of packets to be transmitted in the network all follows the general distribution, and the time slot length is T_r, and then the calculation formula of the average of channel idle period can be obtained as:

$$T_r \sum_{n=1}^{\infty} n(1 - P(n))^n = T_r / p(n). \tag{14}$$

Therefore, the average number of time slots that at least one packet arrives can be set to T_B, which can be expressed by the following Equ.(15)

$$T_B = T_r \sum_{n=1}^{\infty} n(1 - P(n))(P(n))^n = 1/1 - p(n). \tag{15}$$

In RTS \ CTS mechanism, collision interval T_C is expressed as

$$T_C = T_{RTS} + T_{DIFS}. \tag{16}$$

The successful transmission interval T_S can be expressed as

$$T_S = T_{RTS} + T_{DIFS} + T_{PHY} + T_{DIFS}. \tag{17}$$

The mean value of channel busy period B can be expressed as Equ. (18).

$$B = T_S T_B (1-p) + T_C T_{BP}. \tag{18}$$

The Equ. (18) says that B is equal to the number of time slots that have been sent successfully plus the one which have conflicted. Thus, the time slot number S of successful transmission can be expressed as

$$S = T_S T_B (1-p). \tag{19}$$

Based on the above, we can express the utilization rate of wireless channel as U, and its calculation formula is

$$U = \frac{S}{B+1} = \frac{T_S T_B (1-p)}{T_S T_B (1-p) + T_C T_{BP} + 1}. \tag{20}$$

Above, we analyze the influence of MAC layer protocol parameters and establish a one-hop delay model. The formulas express the relationships between one hop delay and the size of sending packet, network transmission rate and nodes number.

IV. A SMART COLLABRATIVE ROUTING PROTOCOL

The essence of proposed smart collaborative routing protocol is a fast and reliable packet forwarding rule based on tree structure. We employ three strategies to optimize the protocol, which involves forwarding, maintenance and efficiency. The most idea is to reduce unnecessary packets forwarding, maintain the tree topology of nodes actively and enhance network real-time communication capability.

A. FORWARDING STRATEGY

The principle of the kernel tree routing algorithm is that packets can only be forwarded to the parent node or child. When the destination node is in the communication range of a node, it is still necessary to forward the message to the parent node, thus increasing the forwarding hops of the packet, resulting in an increase in delay. As shown in Fig. 3, when node 6 sends data to node 5, according to the original protocol, the data packets shall be sent to node 5 through node 1, node 2 and node 3 successively.

In order to solve the problem of more packet forwarding times and maintain the tree-like topology structure of the network, we adopted the method of adding neighbor list.

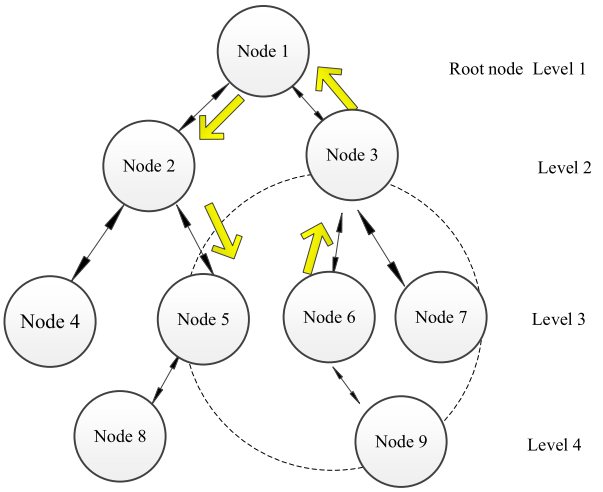

FIGURE 3. Strategy of forward packet.

In other words, when a packet is forwarded, it shall first query the neighbor list. If the destination node of the packet exists in the neighbor list of one hop, it shall be directly forwarded, so as to reduce the number of packet forwarding hops and time delay.

In the process of packet forwarding, the number of packet forwarding should be reduced as far as possible in order to prevent routing detour. Therefore, we propose a strategy of building neighbor node list. For those other nodes that can communicate directly with a node, we add them directly to the list of neighbor nodes of that node and maintain the list of neighbors by sending a hello message. When a packet is forwarded, first find out if the destination node is a member of the neighbor list. If yes, it can be submitted directly; If not, forward along the tree according to the routing table.

As shown in Fig.4, keeping the original tree topology unchanged, we allow the nodes within one hop range to communicate directly. At this point, if there is a packet that

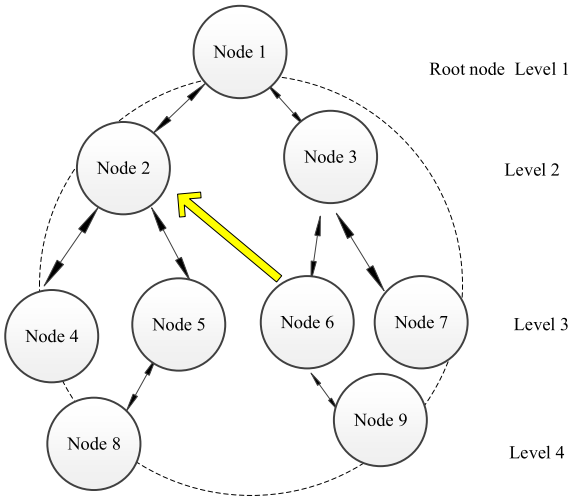

FIGURE 4. Improvement of forwarding strategy.

wants to be sent to destination node 2, 5, 8, 9, 7 and 3 through node 6, it could be directly submitted, thus reducing the forwarding hops of the route. After that, each node needs to maintain its own list of neighbors in real time. As a whole, the network topology is still a tree topology, and its parts become a central star topology. The introduction of a neighbor list gives alternative routes to packets. In the network at this time, the total forwarding times are reduced, which will reduce the forwarding times of packets. However, in order to maintain the neighbor list in real time, the routing overhead needs to be increased.

B. MAINTENANCE STRATEGY

According to the original kernel tree protocol algorithm, when a node finds itself detached from the kernel tree, it will inform its child nodes and all nodes under this branch to immediately detach from the kernel tree and restart the process of joining the tree. This approach will bring more redundancy to other nodes in route maintenance.

The improvement of the maintenance strategy is as follows:

1) when a node finds that its chain is broken with a child node, it acts as the parent and sets the routing entry of the child in the next hop of the routing table to temporarily unavailable state. At the same time, a notice is issued to its parent. In Fig. 3, when node 1 finds that it has broken its chain with child node 3, node 1 sets all routing items in its routing table whose next hop is node 3 to temporarily unavailable state. Since it is the root node, it is not required to report upward.

2) when a node finds that it has broken its chain with its parent node, it restarts the process of joining the tree as a child node, and does not inform all the nodes under its branch at this time.

3) after the node successfully joins the tree, the node under its branch will update the message through routing and send up to the new parent, and report up to the root node step by step.

As shown in Fig. 5, node 6 finds that the link with child node 3 is disconnected, and node 6 initiates the process of joining the tree again at this time. It picks node 5 as its parent and readds it to the tree. At the same time, it notifies the position of its child nodes 9, 10, and 11. Message of the routing update is received by node 5 from node 6 and adds nodes 9, 10 and 11 to its routing entries.

Through this improvement strategy, nodes 9, 10 and 11 do not need to initiate a request to rejoin the tree, but only node 6 needs to rejoin the tree and inform the nodes under its branch upward. In this way, the original tree-like topology of nodes 6 and 9, 10 and 11 is maintained, and the delay of rejoining the tree process of nodes 9, 10 and 11 is reduced, and the overall overhead of the protocol is greatly reduced.

C. EFFICIENCY STRATEGY

Multicast is not required in the original kernel routing protocol. In the whole scenario of smart factory, video and audio data forwarding may be considered in addition to the

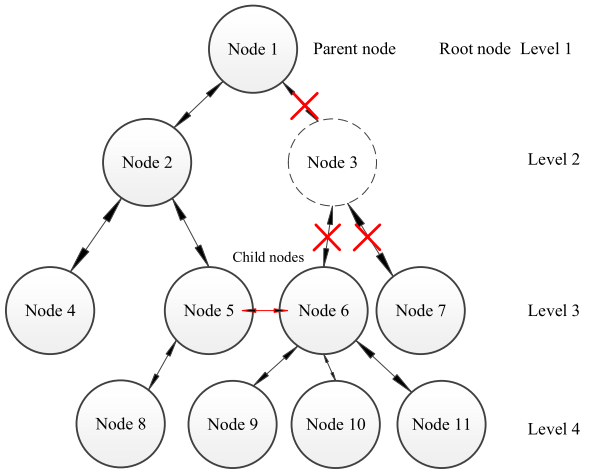

FIGURE 5. Maintenance of tree topology.

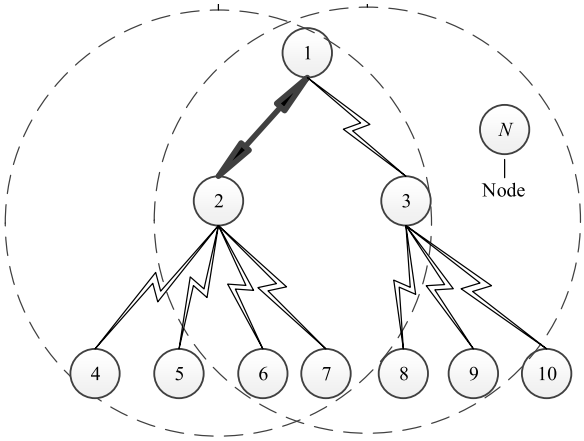

FIGURE 6. Strategy of wired link priority.

transmission requirements of various industrial detection data. Using multicast strategy can save packet cost, reduce packet forwarding times, and reduce time delay. The packet transfer process is shown in Fig. 6. From it, we can see that node 2 multicasts node 4, 5, 6, 7, 8, 9. Node 3 copies the packet and forwards it. Nodes 5 and 6 then copy the packet and forward them.

The multicast method based on tree topology has the characteristics of high efficiency. Multicast sources can distribute groups to each recipient of a multicast group with a minimum number of copies. And based on the tree topology, the routing decision of the node becomes simple, only need to forward up or down.

Take node 5 sending data to other nodes in the network as an example. According to the previous unicast strategy, the total number of hops which has be needed to send packets is $H_5 = 2+2+2+1+1+1+3+2+2+4+4 = 25$ hops. In the process of multicast storage and forwarding, the total number of hops which sending the packet is 12, and also the packet is copied 6 times. Therefore, the multicast method can reduce the forwarding times of packet, improve the utilization of link and reduce the time delay.

At the same time, under the condition that wired links and wireless links coexist, The advantages of stable wired links and large bandwidth could be made full use of, and the nodes with wired links are selected as the parent nodes in priority

As shown in Fig. 7, nodes 6 and 7 are both within the communication range of nodes 2 and 3. According to the original kernel tree routing protocol, node 6 and 7 select their father's selection strategy: when nodes 2 and 3 are of the same rank, select the node that receives the reply to join the tree first. At this point, if nodes 6 and 7 are hung under node 3, the wired link is not fully utilized, and the links between nodes 1 and 3 may be congested, so the effect of load balancing is not achieved.

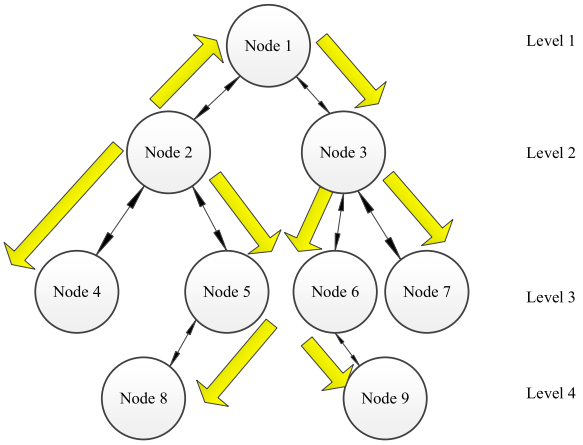

FIGURE 7. Multicast transmission strategy.

Therefore, we distinguish the wired link from the wireless link. When the node with wired link replies to join the protocol, it will carry the label that there is wired link. The node receives the join reply message with wired link label node, which is selected first as the parent node to be joined into the tree. As shown in the figure, nodes 6 and 7 select node 2 as the parent node to join the tree. By this method, the wired link can be effectively used to reduce the time delay.

D. ROUTING ALGORITHM PROCEDURES

1) GENERATION

(1) The root is specified and started to produce a kernel tree with one only root node, and its level is set to 1.

(2) For the nodes that can communicate directly with the root, the nodes with priority links are firstly selected to join the kernel tree by joining tree protocol, and their level is set to 2.

(3) If other nodes do not join the network, they will send messages to join the tree periodically. By joining tree protocol, the nodes with priority links are selected first, and the nodes with low node level are selected as the parent nodes. When it is joined to the kernel tree, it sets its level to the level of the parent node + "1".

(4) Repeat step 3 for other nodes that have not joined the network until all the nodes have joined the kernel tree topology. The node level that has joined the kernel tree will be 0.

(5) After the above steps, each node maintains two tables: one for the routing table and the other for the list of neighbors.

2) MAINTENANCE

(1) When a node joins the kernel tree successfully, it sends a message hello periodically, maintains links with its parent, child and neighbor nodes, and declares its existence.

(2) A message hello is received by each node from its neighbors and updates the timer in the neighbors list.

(3) If the node cannot receive the message hello within a certain time segment, the link is interrupted.

(4) If the parent timer timeout occurs, the node and the ones under the tree are detached from the kernel tree. Based on the improved maintenance policy, the node will quickly rejoin the protocol and apply to join the tree route. And it does not need to disconnect from the child node. When it joined to the tree successfully, the information of the child node is reported up to the root.

(5) If the timer of the child is timed out, all nodes under this node tree are separated from the kernel tree.At this time, the protocol is used to send up the update message and delete these nodes.

(6) When a message about joining the tree is received by one node from its new neighbor, it adds the node to the tree according to the previous joining tree protocol.

(7) If a message hello received by one node from a new neighbor, add the node to the neighbor list according to the neighbor maintenance policy.

3) ROUTING

Based on the newly added neighbors list, change the routing policy to:

(1) after receiving the message, the node will first query the list of neighbors. If the destination is the neighbor node directly connected to itself in the tree (including the parent node, child node and neighbor node), the node will be directly delivered through the network.

(2) When the destination node is a node under the tree, it is forwarded to the corresponding child node directly according to the routing table. This is forwarded step by step down to the destination node.

(3) When the destination node is no longer a node in the tree, the packet is forwarded up to the parent.

(4) When the destination address of packet is multiple, multicast strategy is adopted. After receiving the packet, we first check to see if there are any nodes under our tree in the destination node. If so, copy the data contents of the packet and reassign the destination node of the packet. At the same time, take the node IP address under the tree as the destination node of the packet and make it forward downward. For the remaining destination nodes, repopulate the packet header and forward it up.

In order to better illustrate the improved algorithm, we present the improved packet multicast transmission process, as shown in figure 6.

Based on the topological relationship in Fig. 8, we can list the neighbor relationship between each node, the source-destination relationship, and the hops to each node. As shown in the Tab 1, we take nodes 1 and 2 as examples, and others can be followed.

In Fig. 8, node 1 forwards packets to a destination node of 6, 7, 9. Node 2 sends packets to a destination node of 4, 6, 7, 9. Node 3 copies and forwards the received packets to a destination node of 6, 7, 9.The packet received by node 6 is also copied and forwarded, and its destination node is 9. Node 7 receives the packet.

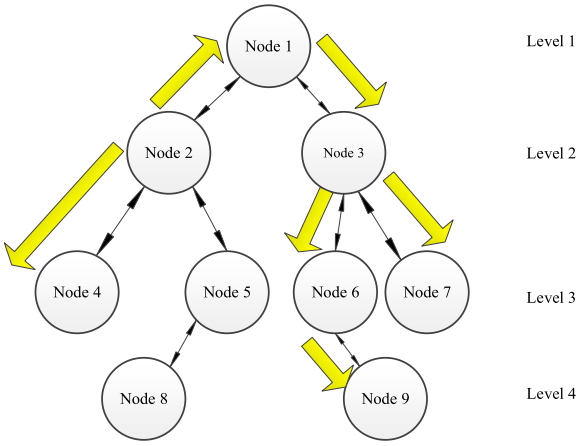

FIGURE 8. Improved packet multicast transmission process.

TABLE 1. Routing and neighbor relationship list.

Node (*N*)	Routing table		Neighbor list	
	Destination node	Node of the next hop	Node	Level
1	2	2	2	2
	3	3		
	4	2		
	5	2		
	6	3	3	2
	7	3		
	8	2		
	9	3		
2	4	2	1	1
	5	5	3	2
	8	5	4	3
	other	1	5	3
……	……	……	……	……

When the distance from one node to the destination is the last hop, the node could be delivered directly without forwarding through the tree structure. At the last hop, the topology changes to a star, and the network becomes a star network centered on the node itself. The addition of the neighbor list adds a new forwarding path to the packet, which optimizes the routing performance and reduces the forwarding times of the packet, thus reducing the delay.

V. IMPLEMENTATION

With the improvement strategies described in section III, The execution mechanism of the tree routing protocol has been redefined. In this section, the improved protocol is divided into two sub-protocols(generate tree protocol & maintain tree protocol), and we described them separately.

A. GENERATE TREE PROTOCOL

The nodes in the mesh can be automatically configured for networking. When the initialization starts, the node actively initiates the process of joining the kernel tree, which is implemented by joining the tree protocol. The node join tree protocol consists of three messages: Join-request, Join-reply and Join-report.

1) JOIN-REQUEST

It is a message sent to a node in the tree by the one making the request. By broadcasting, it could be sent to the one-hop node in the communication range. Its frame structure is shown in Table 2.

TABLE 2. Frame format of join-request.

Type	Level	Reserved 1	Reserved 2
Destination IP			
Source IP			

2) JOIN-REPLY

It is a response message that the nodes give to the requesting one unicast in the kernel tree. The frame structure of it is shown in Table 3.

TABLE 3. Frame format of join-reply.

Type	Level	Reserved	Link
Destination IP			
Source IP			

3) JOIN-REPORT

It is a unicast message that the requesting node sends to the other ones to select the appropriate parent node in the kernel tree. Its frame structure is the same as Join-request. Same as Table 3.

4) IMPROVED FLOW

The network devices in mesh need to be able to configure and organize the network automatically. After the initialization of a node, the node actively initiates the action of joining the kernel tree, and the process of forming a tree topology is called the process of joining the tree of the node. The protocol process of joining tree to kernel tree is as follows:

(1) The nodes broadcast Join-Request messages that can be received by all nodes in the wireless coverage. After other nodes receive join request messages, they first determine whether they have joined the kernel tree.

(2) If not, then other nodes cannot join the kernel tree by themselves, and the request message is ignored. If the node has joined the kernel tree, the node will then reply by unicasting the Join-Reply message.

(3) After receiving the Join-Reply, the requesting node adds the node that sent the message to its candidate list. When join request timer is timed out, the nodes that join the kernel tree with this request will join the kernel tree through the join strategy set at first (the nodes with wired links are selected first as the parent nodes, and the nodes with small node level are selected next).

(4) Conduct level identification. Level = candidate parent level +1.If a node join to the kernel tree at its first time, the message report is added to the parent node in unicast mode. If not, that is, the node has a children, then report message is added to the parent node in unicast mode, and the addresses of children and descendants are added to the send-IP of Update message, and Update is sent.

(5) The node that receives the message determines whether it is the destination node of the message. If so, add the applicant to its relevant form, send routing update message upward and forward to the root node. If not, discard the message.

B. MAINTAIN TREE PROTOCOL

Mesh has the ability of self-regulation. When the node finds fault or the link is disconnected, the network can repair itself. For the routing protocol of this project, we require the nodes to be able to self-maintain the whole tree topology. Node maintenance tree protocol includes three messages: Hello, Update and Kick, respectively.

The message hello is utilized to maintain the tree-like topology that has been formed in the kernel tree and to establish and maintain neighbor relationships. Through this message, the link state of the neighbor node can be determined. Its frame format is shown in Tab 4. Link - types of node links, including wired link and wireless link. Message update is used to Update routing information between nodes. When a node finds a topology change through message hello, the Update message needs to inform the nodes that need to be changed. This message could be passed from the sending node to the root, so that the root node could know the topology change of the network and accelerate the network topology convergence. Its frame format is shown in Tab 5.

TABLE 4. Frame format of Hello.

Type	Level	Reserved	Link
Destination IP			
Source IP			
None			

When the mesh is shut down or the link is unstable, it is easy to cause the wireless link to be disconnected. In order to deal with that, the nodes need to take timely countermeasures

TABLE 5. Frame format of update.

Type	Level	Reserved	Code
Destination IP			
Source IP			
Send IP（variable）			

and modify the routing table, which is called tree maintenance.

1) INITIAL MESSAGE

The node sends the message hello through broadcast, which can determine whether the relationship between superiors and subordinates has changed and confirm whether the neighbor node exists. The node defines the corresponding active timer in the neighbor list, and the protocol sets the length of the active timer to be 3 times that of the message hello sending timer. Within node initialization, after sending message join-report, the child node will update the type option in the neighbor list and set the active timer at the same time. After receiving message hello, the parent node will update the active timer of the child node in the neighbor list.

2) MESSAGE TIMEOUT

When the active timer of a node in the node neighbor list timeouts, that is, it does not receive the hello sent by the sending node for three times, then it enters the process of message hello timeout. A node deletes these nodes from its neighbor list, and a neighbor node deletes this node from its neighbor list in the same way.

3) ROUTING UPDATE

Routing update messages are used when nodes discover changes in topology or links. When a node rejoins the tree or breaks its chain with other nodes, the node will send the relevant messages to the root step by step through the routing update protocol. The routing update process is as follows:

a: CHILD NODES JOIN TO THE KERNEL TREE FOR THE FIRST TIME

As a new child node joins the kernel tree through another one, it will add this new one to its routing table. Meanwhile, it will send update to its parent node, setting the code value in the message to 10 and the send IP in the message to the IP of the new child. The node receiving that message will write send IP to the location of destination IP in its routing table entry, then source IP to the location of next hop IP in its routing table entry. At the same time, in the message the send IP remains unchanged until it is delivered to the root.

b: THE NODE IS DISCONNECTED FROM THE KERNEL TREE DUE TO COMMUNICATION WITH ITS PARENT

When a node finds the active timer timeout of its parent node in the neighbor list, it indicates that the node has left the kernel tree. At this point, the node needs to re-initiate the join kernel tree process. At this point, when the node joins the kernel tree again, the node sends an update message to the parent. The code value of the message is set to 10, and the send IP in the message is set to the IP addresses of all the children and descendants in the routing table of this node at this time. The node receiving this message first queries whether send IP exists at the destination node in the routing entry. If so, the next hop node is updated. If not, add the send IP in the message to the destination IP address in the routing entry, set the source IP in the message to the next hop IP address in the entry, and forward the update message step by step. The send IP in the message remains unchanged until it is delivered to the root node.

c: THE NODE IS FORCED TO CHANGE THE ROUTING ENTRY DUE TO AN INTERRUPTION OF COMMUNICATION WITH THE CHILD NODE

When this node finds the active timer timeout of the neighbor list neutron node, it indicates that the child node has left the kernel tree. At this point, the node sends an update message to the parent. The code value in the message is set to 11, and the send IP in the message is set to the child and the descendant node under the child node. It receiving this message sets the destination node in the routing entry to temp for the routing entry status of those nodes, and starts inactive timer, waiting for update. Meanwhile, the message is forwarded up to the root node step by step.

d: NODES ACTIVELY LEAVE THE KERNEL TREE STRUCTURE

When nodes in the network find that they need to move in a large range or are about to fail, the nodes will actively initiate messages leaving the kernel tree. This notifies other nodes to leave the kernel tree, allowing them to route updates faster without having to wait for neighbor timers to time out. So the node sends a message Kick as a broadcast message, which is the same thing as leaving the kernel tree.

4) ROOT NODE RESELECTION

when a node of rank 2 finds that it is disconnected from the root node, there are two possibilities: one is for the node of rank 2 to leave the network, and the other is for the root node to leave the network. When the root node leaves the network, the network needs to select a new root node so that other nodes can rejoin the tree and update routing information in time.

When the original root node is destroyed, in order to minimize the change of the original topology, the root node selection strategy is changed to: we choose the node with the largest number of child nodes and descendant nodes and rank 2 as the new root node. Since it is a broadcast message, the node first determines whether it has received the same broadcast message, and if so, it discards it. If not, start the timer root-report. If the node level is 2, determine whether there is a root node in the neighbor list. If there is, the root node is not damaged. Therefore, the code format of message

root-reply is set to 11, and a reply message is sent to the sending node. If there is no root node, then the root node is destroyed. At this point, we need to select a new root node. If the count value of the message is less than the one of the node itself, the message root-reply is returned. The code format of message root-reply is set to 10, indicating that the node has more children and descendants and is more suitable for the root node.

VI. SIMULATION RESEARCH

When the kernel tree is established, the end-to-end delay of the packet mainly depends on the forwarding times (i.e., the hops) of the packet. By calculating the hops from each node in the kernel tree to all nodes in the tree, the time delay before and after improvement can be compared. At this point, if the hops of this node to all nodes in the tree are set as i, the hops of each node in the tree to other nodes can be calculated as.

Employing the original kernel tree routing protocol and the forward strategy along the tree, as shown in Fig. 9. The hop number is calculated as

$$\begin{aligned}\sum_{i=1}^{13} H_i &= 32+43+43+25+28+30+35\\ &\quad +39+37+46+46+48+48\\ &= 500 \end{aligned} \tag{21}$$

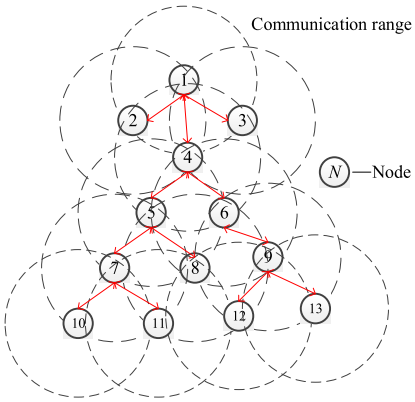

FIGURE 9. The tree topology.

When the improved routing and forwarding strategy is adopted, the change of hop count is

$$\begin{aligned}\sum_{i=1}^{13} H_i &= 32+42+42+23+25+25+30\\ &\quad +25+31+41+35+35+42\\ &= 428 \end{aligned} \tag{22}$$

From the above results, the total number of hops after adopting the improved strategy is 72 hops, which is less than that before the improvement. Considering that the majority of the delay is caused by queuing at the node, the less the number of hops, the less the delay. Similarly, if there are many nodes around a node, that is, when the nodes are densely distributed, the adoption of the new forwarding strategy has obvious advantages. In Fig. 4, there are 6 nodes in the communication range of node 8, and the sum of its hops to other nodes is 25, which is relatively small in total. Therefore, the improvement strategy is more suitable for scenarios with high node density

We employ the following three routing protocols and the proposed improved protocol for comparative analysis. Destination sequenced distance vector routing (DSDV) mainly needs periodic switching routing table between nodes to maintain routing information. Ad hoc on-demand distance vector (AODV) Routing mainly establishes routing information when packets need to be sent. Kernel tree routing protocol (KTRP) needs to establish the topology of tree structure to solve the problem of routing loop. Child nodes need to send routing tables to the parent node to maintain routing information, and each node also needs to maintain its own routing table. Its routing overhead is to periodically send a message to a neighbor node claiming its existence. We propose a improved kernel tree routing protocol (IKTRP). Its main feature is that when sending a group, it first queries the list of neighbors. If there is a destination node in the list of neighbors within a hop range, it will submit it directly. If not, it will follow the routing table directly.

A. PARAMETERS OF THE ONE-HOP DELAY MODEL

One-hop delay refers to the time required for data to be sent from one node to another in the industrial Internet of things, where the receiving node is within the communication coverage of the sending node. It includes sending delay, propagation delay and queuing delay. All parameters are shown in Table 6.

TABLE 6. The simulation parameters.

Parameter	The numerical
Length of the MAC's head	272bits
Length of the physical layer's envelope	128bits
ACK	112bits+PHY
RTS	160bits+PHY
CTS	112bits+PHY
SIFS	10μs
DIFS	50μs
The RTS threshold	200bytes
Minimum competition window W	31
Length of the slot	20μs
Maximum Retransmission times	7
Range of scenarios	1000m*1000m
Packet sending rate	1/s
Maximum degree of Connection of nodes	5
The simulation time	200s

B. QUALITATIVE EVALUATION OF PROTOCOL MECHANISM

The routing protocol used by the IIoT should be adapt to the dynamic network topology. It provide stable and efficient routing algorithm, so that nodes can minimize the end-to-end

TABLE 7. Comparison of routing protocols.

	AODV	DSDV	IKTRP
Route discovery	On demand	Initiative	Initiativ
Routing collaboration between nodes	No interactive routing tables	Neighbor broadcast routing table	Route update messages are sent to the parent node
Avoid routing loops	Yes	Yes	Yes
Whether periodic updates	No	Yes	Yes
Anti-destroying ability	Routing local fixes	Broadcast message updates via routing	Rejoin tree
One-way link	Not support	Not support	Not support
Multicast	Not support	Not support	Support

delay of data and reduce the routing overhead. This time, we adopt qualitative evaluation to reflect the characteristics of routing protocol, and use quantitative data to verify our views.

1) ROUTE DISCOVERY

The discovery process of routing is mainly divided into two ways, active and on-demand. Active routing is the creation process of actively initiated routing, which requires periodic maintenance of topology, with low delay, and is better used when the node does not need to move too much. On-demand routing discovery is mainly based on the communication requirements. In this way, the node does not need to maintain all the routing entries of the topology, and the energy consumption can be reduced accordingly.

2) ROUTE COLLABORATION

Since the mesh we employed has no center, distributed routing management is generally adopted. The node acts as a router and maintains the routing by interacting with routing information.

3) ROUTING LOOPS AVOIDING

Due to the limited bandwidth resources of the network, the dynamic change of the topology and the instability of the link state, higher requirements are put forward for the process of route discovery and maintenance. The node needs to be protected from routing loops that cause packets to be forwarded aimlessly across the network until they are discarded.

4) ROUTING OVERHEAD

Reduce routing packets as much as possible, use routing information efficiently, and reduce bandwidth overhead.

5) ENERGY CONSUMPTION

The energy of the node is limited by the battery capacity. Sending route maintenance messages periodically will cause the node to consume too much energy. And limited by the CPU processing power of the node, the routing algorithm needs to be simple and stable.

6) PROTOCOL SECURITY

Nodes are usually composed of portable mobile devices that lack the necessary physical protection. Mobile nodes transmit information through wireless links, so they are vulnerable to eavesdropping, counterfeiting and other security problems. Therefore, routing protocol is still required to have security mechanism.

C. QUANTITATIVE EVALUATION OF ROUTING ALGORITHMS

The performance of the routing algorithm is mainly evaluated through quantitative testing, which mainly includes end-to-end data packet delay, routing overhead, packet delivery rate and other indicators [19].

1) END-TO-END AVERAGE DELAY TIME (T_{ADT})

It reflects the time characteristics of routing protocol when forwarding data packets. It mainly includes: routing discovery time, packet sending delay based on MAC layer collision avoidance strategy, packet arrival time and so on. End-to-end delay is affected by network bandwidth, routing discovery time, routing protocol parameters and other factors.

$$T_{\mathrm{ADT}} = \frac{1}{M}\sum_{i=0}^{M}\left(rt_i - st_i\right), \tag{23}$$

The total number of successful transmission packets is M, rt_i is the time received by the ith group, st_i is the time sent by the ith group.

2) ROUTING OVERHEAD (R_{o})

It represents the ratio of the routing control packets to the packets which actually received in the network. The smaller the proportion of routing overhead, the less the protocol overhead, and therefore the higher the efficiency of routing generation and maintenance.

$$R_{\mathrm{o}} = \sum_{i=0}^{N} rdb_i / \sum_{i=0}^{N} r_o b_i, \tag{24}$$

The total number of nodes is N, rdb_i is the number of routing packets sent by node i, and $r_o b_i$ is the number of packets actually received by node i.

3) PACKET DELIVERY RATIO (R_{PD})

It mainly reflects the reliability of routing protocol. It is equal to the ratio of the number of packets successfully received by the node to the one which successfully sent by the node. On the other hand, it shows the loss rate of the group.

$$R_{\mathrm{PD}} = \sum_{i=0}^{N} R_{num,i} / \sum_{i=0}^{N} S_{num,i}, \tag{25}$$

$R_{num,i}$ is the number of packets successfully received by node i, and $S_{num,i}$ is the number of packets successfully sent by node i.

4) NUMBER OF DIFFERENT NODES

According to Fig. 10, DSDV has the shortest end-to-end delay, followed by IKTRP protocol. All of them are active routing protocols that require real-time maintenance of routing information. When starting to send packets, IKTRP does not need routing discovery and other processes, but only needs to inquire about the routing and forward or directly deliver. At the same time, with the nodes number increasing, the delay decreases relatively, and it has the minimum delay in the range of 30 to 40 segments, because the routing protocol has modified the strategy of packet forwarding. If there is a destination node within a hop neighborhood, it is delivered directly. As the nodes number increases, IKTRP shows its advantages. When the node density increases, the number increases in a certain range, and the hops that required by the packet decrease.

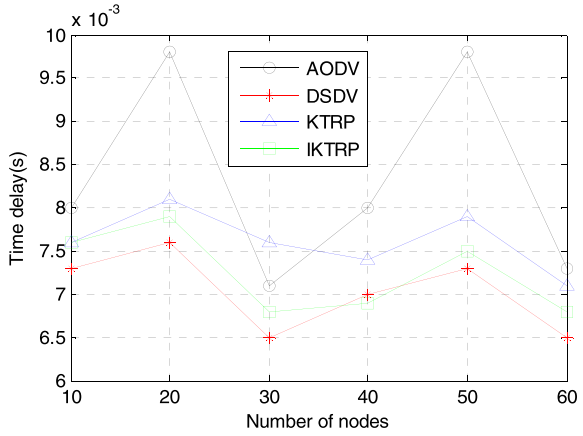

FIGURE 10. Delay with the number of nodes.

In Fig.11 the delivery rate difference between IKTRP and KTRP is small, but the delivery rate performance is not ideal compared with AODV. IKTRP needs to constantly update its routing information. when a node breaks its chain, which most likely resulting routing failure, the child node needs to initiate the request to join the group. If there is no new route, it is easy to cause packet loss. And as the nodes number increases, the delivery rate begins to decrease. Because when the number is large, the routing overhead is high, and there will be collisions or queues with normal packets. Also, packet loss occurs when the cache is saturated.

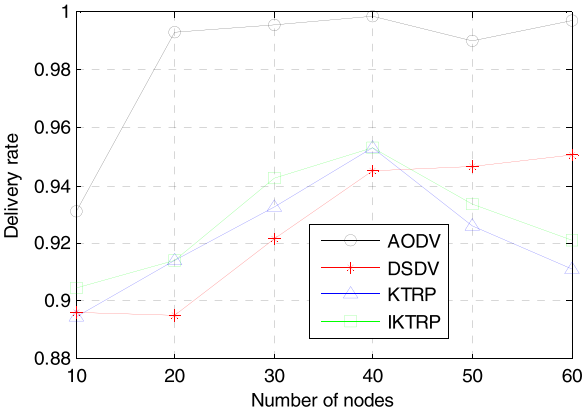

FIGURE 11. Packet delivery ratio with the number of nodes.

As show in Fig 12, when the number of nodes increases, the overhead of IKTRP is basically unchanged compared with the original KTRP protocol, but it is better than the existing DSDV. When the number of nodes is 50, the ratio of DSDV's routing overhead is close to 1. Routing overhead increases with the number of nodes. Because the more nodes there are, the more hello messages there are to maintain the tree topology. Moreover, when the nodes number increases, the sendip part of update message will increase and be forwarded to the root. Therefore, it has more routing overhead than AODV.

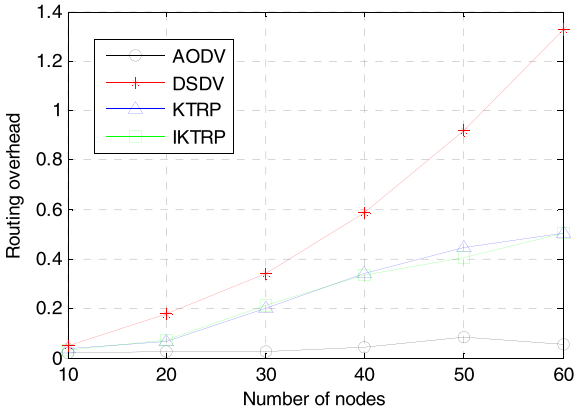

FIGURE 12. Routing overhead with the number of nodes.

From the results of the above three quantitative evaluation, we can draw the following conclusions: when the number of nodes in the region is between 30 and 40, the performance of our improved routing protocol is better, the delay and cost are relatively compromised, and the delivery rate remains above 90%.

5) DIFFERENT RATE OF MOVEMENT

From Fig. 13, when a node breaks its chain with its parent, the node does not immediately break the tree structure of the branch, but makes requests to rejoin the tree locally. Therefore, even if the node moves rapidly, the delay of IKTRP is lower than that of KTRP. Compared with AODV, IKTRP has better delay performance. Because it maintains route information in real time, and reduces the delay of route discovery. When the node moving rate increases, the time delay increases obviously, which indicates that the node may leave the kernel tree in the moving process. At this point, the node needs to rejoin the kernel tree, which resulting in an end-to-end delay.

In Fig 14, as the node moving rate increases, IKTRP delivery rate is better than the other protocols, because the improved maintenance strategy allows the node to join the kernel tree faster, which could reduce topology changes and

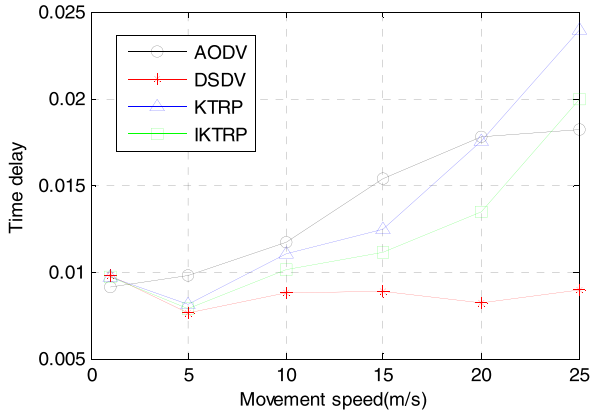

FIGURE 13. Delay with the velocity of nodes.

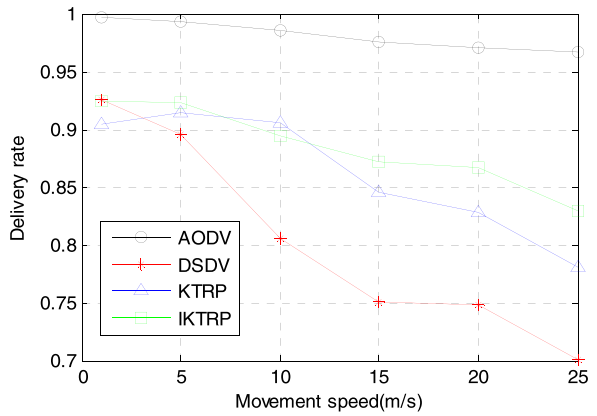

FIGURE 14. Packet delivery ratio with the velocity of nodes.

the probability that node cannot find the route. The faster the nodes move, the faster the network topology changes. The node leaves the kernel tree because of the changing in location. Then, it restarts the process of joining the kernel tree. During this process, if the data amount that needs to be sent is too large, which resulting data overflow in cache area, the packet will be lost. It shows that our routing protocol is suitable for fixed or slow moving scenarios.

Show in Fig. 15, with the increase of node movement rate, the routing overhead of IKTRP is still smaller than that of the original protocol, because the change of the maintenance strategy of the tree reduces the message overhead of the node rejoining the tree under this branch. However, in general, the total routing overhead is larger than that of AODV due to the regularly sending messages hello and update for maintenance. Therefore, the application of this routing protocol will be very limited in the case of fast node movement rate.

From the quantitative evaluation results of the main three protocols, it could be inferred that the performance of the proposed protocol is better within small moving rate (such as industrial production scenario).For example, when the node movement rate is within 5m/s, the packet delay is the minimum, and the delivery rate remains above 90%.

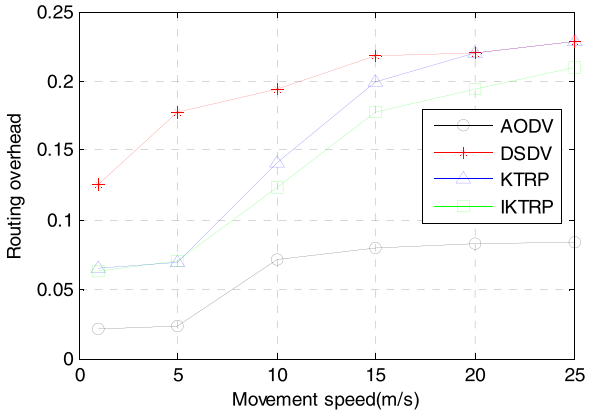

FIGURE 15. Routing overhead with the velocity of nodes.

VII. CONCLUSION

Facing with the actual needs of industrial Internet of things, we model, analyze and improve the kernel tree routing protocol based on deep learning theory. Firstly, a direct forwarding algorithm focusing on neighbor nodes was established to reduce the number of hops. When tree topology is maintained, packets can be directly sent to the destination node in the one-hop neighbor list. Secondly, to block duplicate rejoining requests, a stabilizing algorithm considering tree topology is designed to save protocol overhead. Thirdly, wired link priority policy is employed to improve link utilization and multicast mechanism is included to further decrease the delay. Simulation results show that the average end-to-end delay can reduced compared with the main existing protocols. It proves that the proposed protocol is suitable for industrial scenarios, which is able to provide strong communication guarantee and support for data interaction.

In the future, the number of sensors in IIoT would increase rapidly and the scale of the network continue to expand in the meantime, which is very easy to rise risk of data tampering and transmission interference. As our next work, preventing the protocol from being attacked, ensuring the high reliability and availability of protocol application will be paid more attention.

REFERENCES

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.
[2] W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, "BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors," *Inf. Sci.*, vol. 511, pp. 284–296, Feb. 2020.
[3] F. Song, Y.-T. Zhou, Y. Wang, T.-M. Zhao, I. You, and H.-K. Zhang, "Smart collaborative distribution for privacy enhancement in moving target defense," *Inf. Sci.*, vol. 479, pp. 593–606, Apr. 2019.
[4] W. Wang, Y. Li, X. Wang, J. Liu, and X. Zhang, "Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers," *Future Gener. Comput. Syst.*, vol. 78, no. 3, pp. 987–994, 2018.
[5] Z. Ai, Y. Liu, F. Song, and H. Zhang, "A smart collaborative charging algorithm for mobile power distribution in 5G networks," *IEEE Access*, vol. 6, pp. 28668–28679, Apr. 2018.
[6] M. Khakifirooz, M. Fathi, and K. Wu, "Development of smart semiconductor manufacturing: Operations research and data science perspectives," *IEEE Access*, vol. 7, pp. 108419–108430, 2019.

[7] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.
[8] K. Siddiquee, K. Andersson, F. J. M. Arrebola, Z. Abedin, and M. S. Hossain, "Estimation of signal coverage and localization in Wi-Fi networkswith AODV and OLSR," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 9, no. 3, pp. 11–24, 2018.
[9] F. Song, Z. Ai, Y. Zhou, I. You, R. Choo, and H. Zhang, "Smart collaborative automation for receive buffer control in multipath industrial networks," *IEEE Trans. Ind. Informat.*, to be published.
[10] C. Gaurav, J. Kim, and V. Sharma, "Security of 5G-mobile backhaul networks: A survey," *JoWUA*, vol. 12, pp. 41–70, 2018.
[11] A. A. Khan, M. H. Rehmani, and M. Reisslein, "Requirements, design challenges, and review of routing and MAC protocols for CR–based smart grid systems," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 206–215, May 2017.
[12] F. Song, Y.-T. Zhou, L. Chang, and H.-K. Zhang, "Modeling space–terrestrial integrated networks with smart collaborative theory," *IEEE Netw.*, vol. 33, no. 1, pp. 51–57, Jan. 2019.
[13] F. Song, M. Zhu, Y. Zhou, I. You, and H. Zhang, "Smart collaborative tracking for ubiquitous power IoT in edge-cloud interplay domain," *IEEE Internet Things J.*, to be published.
[14] X. Liu, J. Liu, S. Zhu, W. Wang, and X. Zhang, "Privacy risk analysis and mitigation of analytics libraries in the Android ecosystem," *IEEE Trans. Mobile Comput.*, to be published.
[15] S. Dhiviya, S. Malathy, and M. Monikha, "Enhancing the network lifetime using on demand tree based routing protocol for MANET," in *Proc. 4th Int. Conf. Comput. Commun. Autom. (ICCCA)*, Greater Noida, India, Dec. 2018, pp. 1–6.
[16] C. Wang, Y. Zhang, X. Wang, and Z. Zhang, "Hybrid multihop partition–based clustering routing protocol for WSNs," *IEEE Sens. Lett.*, vol. 2, no. 1, pp. 1–4, Mar. 2018.
[17] M. Carlier, C. M. Garcia Algora, A. Braeken, and K. Steenhaut, "Analysis of Internet protocol based multicast on duty–cycled wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 10, pp. 4317–4327, May 2018.
[18] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Mobile sink–based adaptive immune energy–efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4576–4586, Aug. 2015.
[19] Z. Zhou, B. Yao, R. Xing, L. Shu, and S. Bu, "E–CARP: An energy efficient routing protocol for UWSNs in the Internet of underwater things," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4072–4082, Jun. 2016.
[20] Y.-H. Chen, E. H.-K. Wu, and G.-H. Chen, "Bandwidth–satisfied multicast by multiple trees and network coding in lossy MANETs," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1116–1127, Jun. 2017.
[21] A. Bahramlou and R. Javidan, "Adaptive timing model for improving routing and data aggregation in Internet of Things networks using RPL," *IET Netw.*, vol. 7, no. 5, pp. 306–312, Sep. 2018.
[22] X. Li, B. Keegan, F. Mtenzi, T. Weise, and M. Tan, "Energy–efficient load balancing ant based routing algorithm for wireless sensor networks," *IEEE Access*, vol. 7, pp. 113182–113196, 2019.
[23] Y. Zhang, X. Zhang, S. Ning, J. Gao, and Y. Liu, "Energy–efficient multilevel heterogeneous routing protocol for wireless sensor networks," *IEEE Access*, vol. 7, pp. 55873–55884, 2019.
[24] J. R. Pullagura and D. Rao, "An efficient and reliable cooperative multicast routing based on hop tree in ad-hoc networks," in *Proc. 3rd Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Jan. 2016, pp. 1–5.
[25] W. Si, D. Starobinski, and M. Laifenfeld, "A robust load balancing and routing protocol for intra–car hybrid wired/wireless networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 250–263, Feb. 2019.
[26] C.-W. Wu, K.-J. Lee, and A. P. Su, "A hybrid multicast routing approach with enhanced methods for mesh–based networks-on-chip," *IEEE Trans. Comput.*, vol. 67, no. 9, pp. 1231–1245, Sep. 2018.
[27] X. Wang, H. Cheng, and Y. Yao, "Addressing-based routing optimization for 6LoWPAN WSN in vehicular scenario," *IEEE Sensors J.*, vol. 16, no. 10, pp. 3939–3947, May 2016.
[28] Z.-Y. Ai, Y.-T. Zhou, and F. Song, "A smart collaborative routing protocol for reliable data diffusion in IoT scenarios," *Sensors*, vol. 18, no. 6, p. 1926, Jun. 2018.

MINGQIANG ZHU was born in Dali, Yunnan, China, in 1984. He received the B.S. and Ph.D. degrees in electronic engineering from Beijing jiaotong University, China. From 2008 to 2011, he was an Assistant Engineer with the National Electrical and Electronic Experiment Center. Since 2011, he has been an Assistant Professor with the School of Electronic Information Engineering, Beijing jiaotong University. He holds one patent. His research interests include intelligent sensor technology, wireless sensor networks, industrial artificial intelligence, the industrial Internet of things, big data, and cloud computing technology. He is a Reviewer of the journal *Sensors*, *China Communications*.

LIU CHANG is with the Network Technology Research Institute, China Unicom. He has published many academic articles in well-known international journals. His current research interests include network modeling, IP and bearer technology, differentiated services, and user experience.

NAN WANG was born in Beijing, China, in 1991. He received the B.S. and M.S. degrees in communication and information systems from Beijing Jiaotong University, China. Since 2015, he has been an Assistant Engineer with Dawning Information Industry (Beijing) Company, Ltd. His research interests include intelligent sensor technology, wireless sensor networks, wireless communication transmission technology, broadband wireless ad hoc networks, and cloud computing technology.

ILSUN YOU (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively, and the second Ph.D. degree from Kyushu University, Japan, in 2012. From 1997 to 2004, he was at the Thin Multimedia Inc., Internet Security Company, Ltd., and Hanjo Engineering Company, Ltd., as a Research Engineer. He is currently an Associate Professor with the Department of Information Security Engineering, Soonchunhyang University. Especially, he has focused on 4/5G security, security for wireless networks & mobile Internet, the IoT security, and so forth, while publishing more than 180 articles in these areas. He has served or is currently serving as a Main Organizer of international conferences and workshops, such as MIST, MobiWorld, and MobiSec. He is a Fellow of the IET. He is the Editor-in-Chief of *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* (JoWUA). He is in the Editorial Board for *Information Sciences* (INS), *Journal of Network and Computer Applications* (JNCA), IEEE Access, *Intelligent Automation & Soft Computing* (AutoSoft), *International Journal of Ad Hoc and Ubiquitous Computing* (IJAHUC), *Computing and Informatics* (CAI), and *Journal of High Speed Networks* (JHSN).