IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 23, 2021, accepted July 12, 2021, date of publication July 15, 2021, date of current version July 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097399

Real-Time Communication Model Based on OPC UA Wireless Network for Intelligent Production Line

ANYING CHAI[1,2,3], YUE MA[2,3], ZHENYU YIN[2,3], AND MINGSHI LI[1,2,3]

[1]Faculty of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[2]Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China
[3]Liaoning Key Laboratory of Domestic Industrial Control Platform Technology on Basic Hardware and Software, Shenyang 110168, China

Corresponding author: Zhenyu Yin (congmy@163.com)

This work was supported in part by the National Key Research and Development Project under Grant 2017YFE0125300, and in part by the Liaoning Revitalization Talents Program under Grant XLYC1802056.

ABSTRACT With the continuous development of intelligent manufacturing technology, the intelligent production line for the Industrial Internet of Things occupies an important position in the field of industrial intelligence. OPC UA is a standard for communication data exchange between intelligent production line devices. OPC UA can establish a unified information model for production line devices and improve the connectivity of heterogeneous networks. However, in industrial wireless network application scenarios, there are various types of sensing information and large amounts of data to be transmitted. Each type of data has different requirements for real-time. The traditional OPC UA communication method is difficult to achieve real-time and reliable transmission in intelligent production lines and can't meet the transmission requirements of time-sensitive data. In this paper, we propose a real-time communication model of the OPC UA wireless network for intelligent production lines (UAMPDS). IEEE 802.15.4e TSCH is used as the wireless communication infrastructure, and OPC UA protocols are fused to this model. Meanwhile, we design a load-aware time-slot scheduling algorithm to dynamically allocate and schedule time slots according to the network topology and traffic load of each node. We also implement a dynamic preemptive resource scheduling strategy based on service differentiation to ensure real-time transmission of time-sensitive data. The experimental results show that this model achieves differentiated services based on data with different latency tolerance and effectively reduces the transmission latency of real-time service data under the constrained network resources. It can avoid the starvation phenomenon of low priority queues, and improves the overall quality of service of intelligent production line communication networks.

INDEX TERMS Industrial wireless network, OPC Unified Architecture, real-time, service differentiation, preemptive.

I. INTRODUCTION

Since the beginning of the new century, networked information technology has continued to spread and develop [1]. Major changes have occurred in the field of intelligent manufacturing [2]. The advanced concept of intelligent manufacturing based on the deep integration of information technology and industrialization leads to another innovation of the global manufacturing development model [3], [4]. As the research focus of the new round of industrial revolution, real-time and reliable communication of intelligent production line network based on the industrial Internet of Things has received wide attention from scholars at home and abroad [5]. Using OPC UA [6] as an industrial IoT communication mode has become an important means to improve the connectivity of heterogeneous networks and promote manufacturing intelligence [7]–[9].

At present, industrial field networks have been initially completed to interconnect some industrial field devices. To a certain extent, this situation can meet the demand for interconnection and interoperability of intelligent production line networks [10]. However, there are fewer OPC UA use cases in industrial wireless networks [11]. Therefore, improving the network transmission quality [12] and ensuring the reliable

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco Rafael Marques Lima.

transmission of real-time data has become an important issue to be solved in the field of wireless communication, which uses OPC UA as the communication method.

There has been some work on network communication technology for OPC UA in the field of intelligent manufacturing. Researchers have proposed many methods to improve communication quality for industrial wired networks. These methods improve the real-time and effectiveness of data by establishing scheduling mechanisms or changing the underlying data transmission mechanisms of devices. However, there are few research cases of OPC UA for wireless network applications. Moreover, the types of devices in the intelligent production line environment for industrial IoT are complex and diverse. Different types of sensing information and device status information have various requirements on data transmission volume and real-time. Especially in intelligent production line wireless networks, the communication method using OPC UA as a framework has problems such as insufficient applicability and poor quality of service. And it cannot guarantee the differentiated transmission of data with different delay tolerance.

In this paper we propose a real-time communication model of the OPC UA wireless network (UAMPDS). The model incorporates the OPC UA protocol and uses IEEE 802.15.4e TSCH as the wireless communication mode for intelligent production line networks [32]. A load-aware time-slot scheduling algorithm is designed to achieve the dynamic allocation of network resources. Meanwhile, the dynamic preemptive resource scheduling strategy based on service differentiation is used to complete the priority transmission of various types of industrial data.

Section II introduces the related work in the field of communication for IIoT. Section III introduces the wireless network communication architecture based on OPC UA. section IV describes the real-time communication model of the industrial wireless network for the intelligent production line proposed in this paper. Section V presents the conclusion.

II. RELATED WORKS

At present, many scholars and institutions have conducted in-depth research in communication technology for intelligent manufacturing. And they have made a lot of research results in industrial IoT communication technology.

Some researchers have designed and implemented some industrial communication systems in conjunction with OPC UA. These research results have improved communication efficiency and reduced the transmission delay to different degrees. Girbea *et al.* [13] proposed a new architecture in industrial automation based on OPC UA, which contains an OPC UA server, multi-level service layer, and service constraint layer (CSP) with improved performance in system control and scheduling. Kim's team [14] designed a standalone wrapper based on OPC UA for industrial monitoring and control system by sharing memory and semaphore and other communication methods, which improved the system cross-platform and system throughput and effectively reduced the transmission data latency. Meanwhile, Kim *et al.* developed a PLCopen OPC-UA communication component, which has verified the feasibility of the component in a real environment [15]. Mahmoud *et al.* [16] conducted an in-depth study on OPC technology for communication in distributed control systems (DCS). Maia *et al.* [17] investigated the transmission performance of the OPC UA communication framework as an integration element between devices. They argued that OPC UA has a low traffic overhead.

Researchers have done a lot of work on scheduling algorithms. They have utilized many scheduling methods to improve the communication performance of industrial systems and have gotten good results. In addition, 5G networks have been used in industrial communication systems and are motivating new advances in the field. Nasser *et al.* [18] proposed a dynamic multi-level priority packet scheduling scheme (DMP) that uses a three-level priority queue to schedule different types of packets. This approach ensures minimum end-to-end delay for high-priority data and exhibits acceptable fairness for the lowest priority data. Palattella *et al.* [19] proposed a traffic-aware scheduling algorithm for the TSCH mode of the IEEE 802.15.4e standard. This algorithm extends graph-theoretic matching and coloring methods based on network topology and traffic load. In terms of performance, it reduces the wireless data transmission delay.

Xia *et al.* [20] proposed a data fusion scheme between OPC UA and 5G, which solved the transmission mapping problem between OPC UA and 5G cellular networks. Kim *et al.* [21] proposed a CPPS architecture using a centralized OPC UA server. This new architecture solves the problems of data loss and poor real-time performance of the communication system based on the traditional CPPS architecture. Hang *et al.* [22] proposed an NFV-enabled 5G paradigm for the industry. It fulfills the requirement of ultra-reliable and low-latency communications of 5G networks in industrial scenarios. Mekikis *et al.* [23] implemented a novel 5G network function virtualization enabled experimental platform. It supports the Tactile Internet characteristics in industrial environments and meets sub-millisecond end-to-end communication.

Liu *et al.* [24] implemented the digital twin of a factory using OPC UA technology for production lines. Ning [25] and Song [26] completed the application of OPC UA technology fused with industrial wireless sensor networks in an industrial environment, respectively. Wang *et al.* [27] proposed a smart factory implementation framework that constructs a flat network topology to meet the heterogeneous requirements of different devices in terms of real-time, reliability, security, and semantics. Wan *et al.* [28] proposed a delay-sensitive network bandwidth scheduling algorithm to improve the bandwidth transmission efficiency of different delayed flows. Min [29], Wang [30], and Yang [31] conducted in-depth studies on time slot scheduling in industrial wireless networks, respectively. They all improve the system data transmission performance to different degrees.

However, the above-mentioned studies cannot fully satisfy the real-time and reliable communication requirements of the IIoT intelligent production line systems. We seek to design a real-time communication model of the OPC UA wireless network (UAMPDS). UAMPDS incorporates the OPC UA protocol and uses IEEE 802.15.4e TSCH as the wireless communication mode for intelligent production line networks. Furthermore, the model contains a load-aware time-slot scheduling algorithm and a dynamic preemptive resource scheduling strategy based on service differentiation for high real-time transmission of time-sensitive data in the network. These methods effectively improve the quality of service in intelligent production line communication networks.

III. WIRELESS NETWORK COMMUNICATION ARCHITECTURE BASED ON OPC UA

We design the overall architecture of an intelligent production line network based on the traditional OPC UA communication architecture. The architecture divides the whole network system into industrial field device layer, data sensing layer, wireless communication gateway layer, and application management layer. It adopts a five-layer communication model of the industrial wireless network incorporating OPC UA. It can interface the underlying network with the Internet and provide communication services between the underlying device and the application management.

A. THE OVERALL ARCHITECTURE OF THE INTELLIGENT PRODUCTION LINE NETWORK

The network communication system for an intelligent production line is characterized by a variety of protocols and incompatibility. Different equipment manufacturers are using different communication modes. This phenomenon makes it difficult to form a converged and efficient information channel between industrial equipment. Especially in the application scenario where the data of the production line system has wireless transmission needs, it is even more impossible to meet the real-time and reliable communication of the equipment.

We integrate OPC UA into the industrial wireless network infrastructure. This architecture uses OPC UA to define the address space structure of various devices and design the address space model of the devices connected in the communication network. It provides communication channels for industrial devices and upper-layer applications and satisfies the demand for flat industrial networks. Figure 1 shows the overall architecture of wireless network communication for intelligent production lines. This architecture includes the industrial field device layer, the data perception layer, the wireless communication gateway layer, and the application management layer. Each layer transmits data based on a wireless network with OPC UA protocol.

The industrial field device layer consists of production line communication devices and industrial sensors. The intelligent production line system uses, RFID, industrial

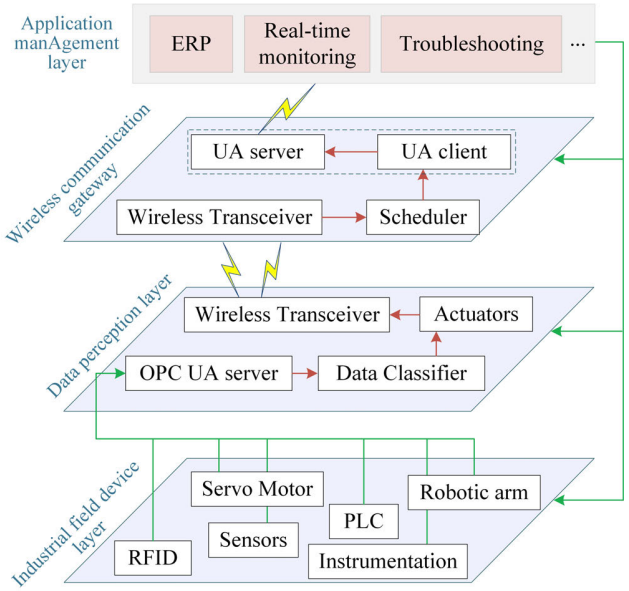

FIGURE 1. The overall architecture of industrial wireless network communication for intelligent production lines.

instruments, servo motors, and robotic arms to work together. At the same time, in order to ensure the healthy operation of the equipment during production, the entire production line contains wireless sensors of temperature, humidity, and vibration, which collect sensing information in real-time in order to sense the operating status of the equipment.

The data perception layer includes OPC UA communication modules, data classifiers, actuators, and wireless transceivers, which provide the system with functions such as data acquisition and signaling. OPC UA communication module can collect alarm information, equipment status information, and actual processing status information from servo motors, and robotic arms, and send them to the upper layer of the system. OPC UA communication module establishes the address space model of sensor network nodes and interconnects Internet applications with the underlying wireless sensor network [33]. The data classifier can classify the type of device data. The wireless communication gateway can further set the data priority according to the data type and provide the execution basis for the dynamic preemption scheduling algorithm. The actuator is the execution module of the data sensing layer. The main task is to parse and execute system commands. When the client wants to operate a specific device or obtain data from the specific device, the actuator will execute the instructions given by the client, complete the control tasks of the intelligent production line system. The wireless transceiver supports wireless transmission and reception of device data. It can complete the communication with the wireless gateway layer.

The wireless communication gateway layer includes modules such as the wireless transceiver, scheduler, and UA component. The UA component includes the UA server and UA client. The UA client collects and receives information from the data-aware layer devices by accessing the address space

of the embedded OPC UA server. The UA server establishes a unified information model for various devices in the network. The UA server is combined with the UA client to push data to application management. The scheduler is the data scheduling module of the system. It classifies the transmission by the data priority transmission rules defined by the system. This module uses a dynamic preemptive resource scheduling strategy based on service differentiation to achieve three priority data transmission. This strategy ensures high real-time transmission of delay-sensitive data in a wireless communication environment. The wireless transceiver can complete the transmission interface with the sensing data layer and complete the wireless transmission and reception of gateway data.

The application management layer consists of various upper-level systems for managing industrial equipment, such as ERP, safety monitoring systems, and fault diagnosis systems. They obtain the operating status of equipment, predict the probability of failure, and initially diagnose the cause of failure to unify the management of field-level equipment. At the same time, the systems can deliver control information to specific devices and complete real-time control of production line installations.

B. FIVE-LAYER COMMUNICATION MODEL FOR INDUSTRIAL WIRELESS NETWORKS BASED ON OPC UA

We adopt the five-layer model of industrial wireless network as the wireless communication infrastructure of intelligent production line network. This model will be divided into five layers from the bottom up. The task of the first four layers is the collection and transmission of network data. In these layers, the OPC UA security model is used to provide security for data transmission. The fifth layer is used to complete the unified management of sessions and semantic messages of OPC UA applications. Figure 2 shows the architecture of this model.

This model applies to intelligent production line industrial wireless networks. The physical layer adopts IEEE 802.15.4-2006 protocol and supports hardware devices with the low-power protocol. The data link layer uses a time slot and channel hopping (TSCH) scheme. This scheme can dynamically allocate and schedule time slots according to the network topology and traffic load. The network layer uses centrally controlled graph routing to improve network transmission efficiency. The transport layer uses TCP protocol and OPC UA security model to provide secure data transmission service. The application layer uses OPC UA to implement data collection and manage sessions. As the model layers work together, the devices communicate with each other wirelessly.

IV. REAL-TIME COMMUNICATION MODEL OF INDUSTRIAL WIRELESS NETWORK FOR INTELLIGENT PRODUCTION LINE

In this section, we focus on the real-time communication model proposed in this paper. We design a load-aware time-slot scheduling algorithm and a dynamic preemptive resource scheduling strategy based on service differentiation, based on the above wireless network communication architecture and the IEEE 802.15.4e TSCH model. At the same time, the model creates three different priority message queues, which store high, medium, and low priority data. The following is a detailed introduction.

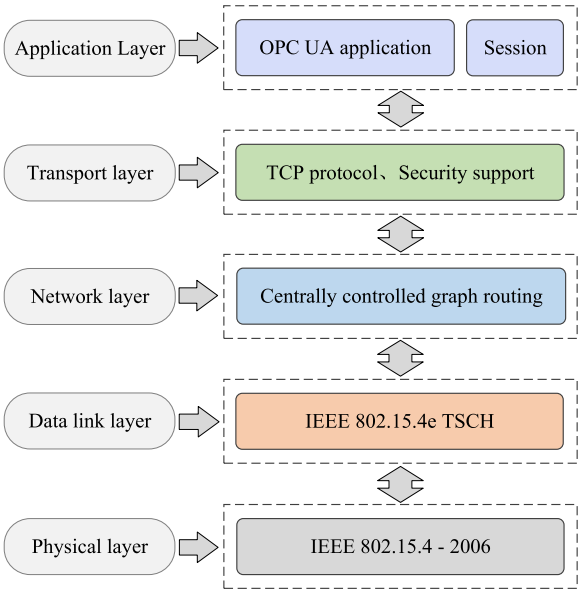

FIGURE 2. Five-layer model of industrial wireless network based on OPC UA.

A. DATA PRIORITY DEFINITION

The wireless communication gateway collects wireless device data through the data perception layer. The data classifier module has classified this data before it reaches the gateway.

According to the real-time requirements of each type of data, we divide the device data into high priority data ($HighPriority - 0, HPR - 0$), medium priority data ($MediumPriority - 1, MPR - 1$), and low priority data ($LowPriority - 2, LPR - 2$). Then, each priority data is cached into each message queue separately for differentiated scheduling. We define data prioritization rules for the model as follows.

1) $HPR - 0$: It is the highest priority of the system. Usually, it is defined for the highest priority scheduling of real-time data. This data contains OPC UA communication context messages and device alarm type messages. OPC UA communication context messages include ''HEL/ACK messages'' (establishing an initial connection), ''ERR messages'' (system connection error), and ''SC messages'' (establishing safety channels). Alarm messages consist of alarm parameters of all devices in the intelligent production line during operation. Both types of messages are time-delay sensitive data.

2) $MPR - 1$: It is the medium priority of the system. Typically, it is defined for medium priority scheduling. These

data consist of device status information, processing status information, and sensing information. They are second in rank only to connection-type messages and alarm-type messages. The purpose of this type of data is defined to guarantee that the upper layer applications have access to the underlying device operation data.

3) $LPR - 2$: It is the lowest-class priority of the system. Usually, it provides scheduling services for non-real-time data. This priority type serves for devices that deliver non-real-time media data. It satisfies the image data transmission needs of some applications in the upper layers of the system.

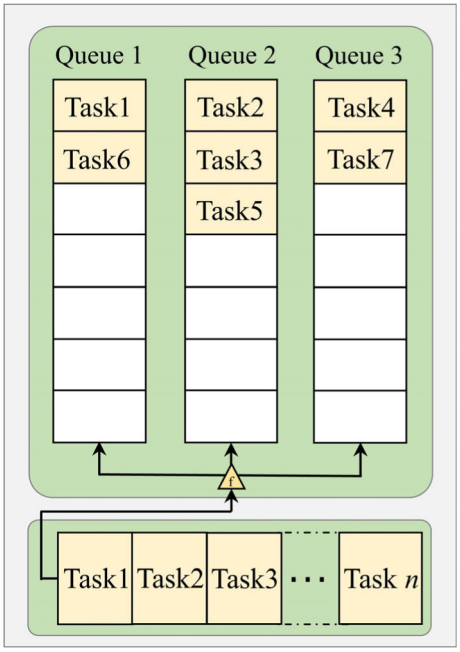

FIGURE 3. Multi-priority buffer queue service structure.

Figure 3 shows the multi-priority buffer queue service structure. The system stores three types of data in *Queue*1, *Queue*2 and *Queue*3 buffer queues according to their priority. In detail, *Queue*1 stores $HPR-0$ priority data, *Queue*2 stores $MPR - 1$ priority data, and *Queue*3 stores $LPR - 2$ priority data. *Queue*1 is the highest priority queue.

B. LOAD-AWARE TIME-SLOT SCHEDULING ALGORITHM

The load-aware time-slot scheduling algorithm is a centralized resource scheduling method with the wireless communication gateway as the master node. The master node has complete network topology information. In intelligent production line networks, most devices can communicate with the gateway directly. However, there are cases where some wireless sensor network nodes need multiple hops to make data reach the gateway. We fuse star topology and tree topology to represent the network topology as a graph.

We define the directed graph $G = (V, E)$, where $V = \{N_0, N_1, N_2, N_3, \ldots, N_n\}$, as shown in Figure 4. The N_0 node is the master node, which represents the wireless communication gateway. $|V| = n$, n is the total number of abstracted

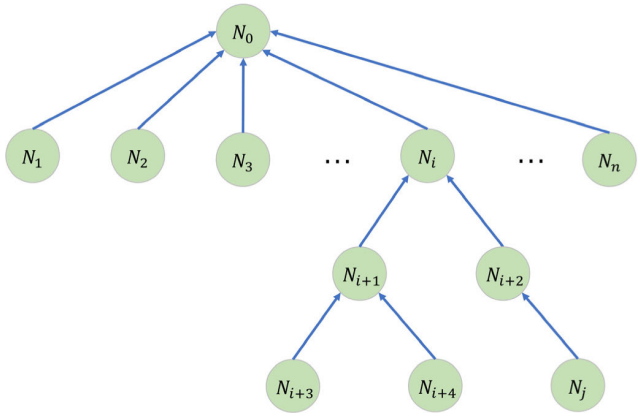

FIGURE 4. Wireless network topology.

nodes. E is the directed edge, which represents the connection relationship of all nodes in the graph. The variables i and j follow the constraints $1 \leq i \leq n-1$ and $1 \leq j \leq n-1$. We consider only the communication uplink. There is no communication between each node and its neighbor nodes on the same layer. Meanwhile, the wireless nodes eventually converge their data to the master node N_0.

In the model topology, most of the child nodes can push data directly to the master node N_0 through a single hop. However, some child nodes require multiple hops to communicate with the master node. This situation can easily lead to node duplex conflict problems [19].

Interference conflicts arise between nodes When the wireless network works normally. Therefore, we define $CFL(i)$ as a set of duplex-free conflicts under time slot i and $NIFL(i)$ as a set of interference-free conflicts. The two factors are incorporated into the model scheduling algorithm in the form of parameters to avoid problems such as communication inefficiencies caused by transmission conflicts during time slot scheduling.

We assume that there are N transmission nodes in the network, and the set of nodes is M. Each node can send data to the previous node and receive an acknowledgment message within a time slot. The model abstracts the wireless network as a directed graph $G = (V, E)$, where $G = (V, E)$, $|M| = |V| = n$.

NS represents the set of states of each node. The node states include active and dormant states. When a node is in the active state, it means that the node has data to send. When a node is in a dormant state, it means that the node has no data to send. Equation (1) is the description of the node state.

$$NS_i = \begin{cases} N & // \text{ Dormant state; node } i \text{ is not sending data.} \\ 1 & // \text{ Active state; node } i \text{ is sending data.} \end{cases} \tag{1}$$

Before the time slot allocation, the algorithm obtains the state and load of each node. By traversing the set of node states NS_i, the active state node in the global directed graph G can map to the entity device node M_j, which is added to the

waiting allocation list *TSAN*. According to the amount of data waiting to be sent, the model reorders the nodes in the *TSAN*. This step redefines the sending sequence for each device. When designing a single cycle timeslot scheduling table, this algorithm is better for nodes with higher load. It will prioritize these nodes to allocate time slots first. This method ensures the priority allocation and transmission of high load nodes in multi-time slot frame mode.

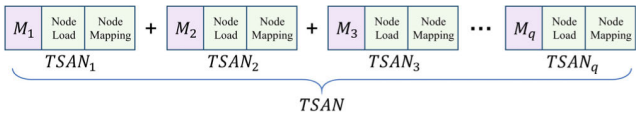

FIGURE 5. Model TSAN structure diagram.

After obtaining the waiting allocation list *TSAN*, the model redetermines the set of directed edges E_{alloc} of the scheduling nodes. This step avoids transmission conflicts and reduces errors in graph matching and vertex coloring. After matching and coloring, the model finally constructs the time-slot scheduling scheme *TSSP* within a single cycle. Figure 5 shows the *TSAN* list structure of the model. *NodeLoad* is the load of this node, and *NodeMapping* is the node in the directed graph G corresponding to the entity node.

This algorithm constructs the *TSSP* scheme mainly into two main processes, graph matching, and vertex coloring. The graph matching method aims to find a set of subgraphs in graph *G*, which has any two edges that are not the same vertex. Therefore, there is no duplex conflict at any node in this subgraph.

The following is a detailed The vertex coloring is mainly based on $CFL(i)$ to find the interference conflict-free set of the same channel that can be transmitted in the same time slot. This approach avoids interference conflicts for communication between network devices. A detailed description is given below.

The graph matching method quickly traverses the directed graph *G*. The implementation of graph matching relies on the *Hopcroft-Karp* method. According to the order of nodes in *TSAN*, multiple disjoint augmentation paths are found at once. Then, $CFL(i)$ adds multiple matching edges based on these augmented paths. At this point, the model initially determines the set of duplex-free conflicts.

The load-aware time-slot scheduling algorithm builds a set of interference conflict graphs $CI(i) = \{V_{CI}(i), E_{CI}(i)\}$ by $CFL(i)$, E_{alloc}, and the set of conflicting interference *IN* in the actual scenario, where $V_{CI}(i) \subset V$, $(n_i, n_j) \in E_{CI}(i)$.

According to the order of the nodes in *TSAN*, the high load nodes are colored first. The remaining nodes are divided into non-interference node group n_{nif} and interference node group n_{if}. The nodes within both n_{nif} and n_{nif} are sorted according to the load from high to low.

The algorithm continues to color the first node in n_{nif} with the same color. The set of links *colorscheme* with the same channel offset $offset_{chan}$ within time slot *i* is calculated. At the same time, the set of conflict-free links $NIFL(i)$ is determined.

The model assigns the links in *colorscheme* to the multi-time slot scheduling table under time slot *i*, respectively. Then the scheduling scheme *solution* under time slot *i* is calculated. Finally, the model adds *solution* to the time slot scheduling scheme *TSSP*. *TSAN* and the total amount of allocated data are updated. The model continues to perform the scheduling assignment task for time slot $i + 1$ until all nodes are assigned.

Algorithm 1 describes the working process of the load-aware time-slot scheduling algorithm. We analyze the scheduling algorithm using asymptotic analysis. Also, we calculate the time complexity of each function. Table 1 shows the results of these functions in terms of time complexity.

TABLE 1. Time complexity of each function.

Number	Function name	Time complexity
1	*NodeMapping*	$O(n)$
2	*SortLoad*	$O(nlogn)$
3	*DistributLink*	$O(n)$
4	*GraphMatching*	$O(m\sqrt{n})$
5	*ChannelInterfGraph*	$O(n)$
6	*GraphColoring*	$O(n^2)$
7	*FirstEdge*	$O(n)$
8	*Update*	$O(1)$

The time complexity is calculated by Equation (2), which follows the additive rule of asymptotic analysis. Therefore, the time complexity of this scheduling algorithm is $O(n^3)$.

$$T(n) = O(n^*n) + O(nlogn) + O(n) + O(n^*m\sqrt{n}) + O(n^*n) + O(n^*n^2) + c^*O(n^*n) + O(n^*1) \tag{2}$$

C. DYNAMIC PREEMPTIVE RESOURCE SCHEDULING STRATEGY BASED ON SERVICE DIFFERENTIATION

The wireless communication gateway receives the data from the data-aware layer and stores them in the buffer queue within the gateway in a classified manner. This layer contains three buffer queues, which store three kinds of priority data, respectively. Meanwhile, we design a dynamic preemptive resource scheduling strategy based on service differentiation to schedule the three queues. This approach ensures the real-time transmission of high real-time data. Figure 6 shows the resource scheduling framework of the wireless communication gateway.

The gateway receives all device data through the wireless transceiver. And it can parse and map protocols using the integrated OPC UA service module. According to the defined

Algorithm 1 Load-Aware Time-Slot Scheduling Algorithm

Input: directed graph $G = (V, E)$, the set of nodes M, number of nodes N, the set of node states NS, The set of the number of node packets ND, amount of data that can be transmitted in a time slot frame TD, number of channels CH, the set of interfering conflicting nodes IN.

1. $i \leftarrow 0$; $dalloc \leftarrow 0$; // Initialize the model parameters (the number of cycles and the amount of transmitted data).
2. $TSAN \leftarrow []$; $TSSP \leftarrow []$; // Initialize the model parameters (wait for the allocation list and time slot scheduling scheme).
3. for $j \leftarrow 0$ to $N - 1$ do
4. if $NS_j = 1$ then
5. $M_j \leftarrow NodeMapping(NS_j)$; // The nodes in graph G are mapped to the production line physical devices.
6. $TSAN \leftarrow TSAN + M_j$; // Add the candidate nodes to the set of assigned nodes.
7. end if
8. end for
9. $SortLoad(TSAN)$; // Sort the nodes within the TSAN according to the node load.
10. $E_{alloc} \leftarrow DistributLink(TSAN)$; // Determine the set of directed edges of nodes that can be scheduled.
11. While $dalloc \neq TD$ do
12. $CFL(i) \leftarrow GraphMatching(G, ND, TD, N_0, i)$; // Graph matching operation. Find the model duplex-free conflict set.
13. $CI(i) \leftarrow ChannelInterfGraph(CFL(i), E_{alloc}, IN)$; // Create interference conflict maps.
14. $colorscheme \leftarrow GraphColoring(CI(i), ND, TSAN)$; // Node coloring in graph G.
15. $solution \leftarrow []$; // Initialize the scheduling scheme at time slot i.
16. for $k \leftarrow 1$ to CH do
17. $solution \leftarrow FirstEdge(colorscheme)$; // Calculate the scheduling scheme $solution$ at time slot i.
18. end for
19. $Update(dalloc, TSAN)$; // Update the list $TSAN$ and the value of the allocated data volume.
20. $TSSP \leftarrow TSSP + [(solution, i)]$; // Add $solution$ to the time slot scheduling scheme of the model.
21. $i = i + 1$;
22. end while
23. return $TSSP$;

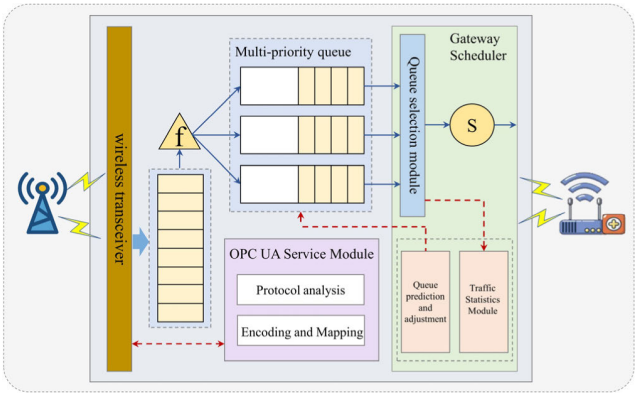

FIGURE 6. Wireless communication gateway resource scheduling framework.

priority rules, the model analyzes the type of each packet and stores the parsed data in a multi-priority buffer queue, which is scheduled by the gateway scheduler. The gateway scheduler follows a dynamic preemptive resource scheduling policy based on service differentiation. The queue selection module performs the queue scheduling task [35]. The details are as follows.

There are n data streams in the network sending data to the gateway. And we denote each data stream by f_i, where $0 \leq i \leq n$. Assume that the frame length of each data stream is fl_i and the flow rate is fr_i.

Then the sending cycle T_i can be calculated. Also aiming to avoid gateway data congestion, this method uses the least common multiple (T_g) of T_i as the value of the gateway scheduling cycle fl_i [34]. T_g is greater than the sending cycle of each data stream. Now, T_i and T_g can be expressed as:

$$\begin{cases} T_i = fl_i/fr_i \\ T_g = LCM(T_1, T_2, T_3, \ldots, T_n) \end{cases} \tag{3}$$

After calculating fl_i, the model cuts the scheduling cycle into k time slots, and each time slot represents a fixed time length L_{st}. Let the transmission rate of the output link be R and the amount of data transmitted by a single time slot be H (bit). Equation (4) can calculate the time slot length.

$$L_{st} = H/R \tag{4}$$

Also, equation (5) calculates the number of time slots k within T_g. This formula applies upward rounding. When the value cannot be rounded up, we treat the remainder as a time slot alone. Therefore, the last time slot length is less than the standard length L_{st}.

$$k = \lceil T_g/L_{st} \rceil \tag{5}$$

According to the queue priority, the gateway scheduler obtains the data volume of the three ready queues. They are denoted by $buff_1$, $buff_2$, and $buff_3$. When the highest priority HPR-0 queue has data to send, we use 123 to represent it. The model prioritizes the scheduling of this queue and calculates the number of time slots occupied by this queue in this cycle, as shown in Equation (6).

$$s_i = \left\lceil \frac{buff_i}{L_{st} \times R} \right\rceil \tag{6}$$

If $s_1 \geq k$, all time slots are assigned to $HPR-0$ queue. The data of queues $MPR-1$ and $LPR-2$ are temporarily stored at the gateway until the next scheduling cycle starts. This scheduling task ends.

If $s_1 < k$, there are enough time slots to allocate to $HPR-0$ queue in this cycle. The model allocates the remaining number of time slots r_1 to the other two queues in order of priority, where $r_2 = r_1 - s_2$.

When $r_2 > 0$ and $buff_3 > 0$, the model assigns r_2 to the $LPR-2$ queue for data transfer. The scheduling task ends.

The $HPR-0$ queue includes alarm messages, OPC UA communication connections, and security messages. When the system state is stable, the *HPR-0* queue is empty. If $buff_1 = 0$, the model does not perform the allocation operation for the $HPR-0$ queue and directly allocates the $MPR-1$ queue and $LPR-2$ queue.

If the model detects a real-time message in the $HPR-0$ queue during regular scheduling, it immediately saves the operating parameters and interrupts the $MPR-1$ or $LPR-2$ allocation operation. The $HPR-0$ queue seizes the time slot and continues to send messages. This approach ensures that high-priority data are real-time.

We consider that $LPR-2$ data may cause unallocated time slots in dynamic preemptive scheduling strategies. Too much data cache may lead to queue overload. These may lead to ''starvation'' of the $LPR-2$ queue. In this paper, we set two thresholds to dynamically regulate the time slot allocation process of the $LPR-2$ queue.

We set the queue threshold value δ and the queue stability value μ. When the $LPR-2$ queue load exceeds the threshold value δ, the model triggers a preemption operation. It interrupts the $MPR-1$ queue allocation task and immediately executes the $LPR-2$ queue allocation task until the queue load is not greater than μ. Then the regular scheduling task continues. Figure 7 shows the flow chart of the dynamic preemptive resource scheduling policy based on service differentiation.

Figure 8 shows an example of a dynamic preemptive resource scheduling policy based on service differentiation. In the figure, the scheduling period T_g is 8. In the first scheduling cycle, the gateway scheduler monitors that there is data in the $HPR-0$ queue, so the model allocates 2 time slots for $HPR-0$ priority data. (One time slot represents one time unit.) When the model finishes sending the $HPR-0$ queue data, the model starts scheduling the $MPR-1$ queue and allocates 3 time slots. Finally the model assigns the remaining 3 time slots to the $LPR-2$ queue. In the second scheduling cycle, the model performs regular allocation tasks for the $MPR-1$ queue and the $LPR-2$ queue.

In the third scheduling cycle, the gateway scheduler continues to listen to each queue. After judging that the $HPR-0$ queue is empty, the model directly allocates 3 time slots to the $MPR-1$ queue. Then, the model executes the allocation task for the $LPR-2$ queue. When the $LPR-2$ queue is transmitting data, the scheduler listens for the $HPR-0$ queue to receive real-time data. At this point, the preemption operation occurs. The model interrupts the task being executed and allocates the free time slot to the $HPR-0$ queue to transmit data. Then, the model performs the reallocation task. The shaded a is the $HPR-0$ queue wait time.

In the fourth scheduling cycle, we can see that the $MPR-1$ queue has more data than before. The model allocates 7 time slots for the $MPR-1$ queue. the $LPR-2$ queue gets only 1 time slot.

In the fifth scheduling cycle, the amount of cached data in the $LPR-2$ queue has exceeded the threshold value δ. A preemption operation is triggered. The model performs an interrupt to allocate the free time slot to the $MPR-1$ queue.

At 38 time units, the $LPR-2$ queue load is less than the queue stability value μ. At this point, the queue has already seized 5 time units of time slots. The shaded b is the waiting time for the $MPR-1$ queue. The model interrupts the $LPR-2$ scheduling task and starts to allocate time slots for the $MPR-1$ queue until the end of this scheduling cycle.

V. EXPERIMENTS

The integrated experimental platform of the IIOT intelligent production line focuses on the concept of customized, small-batch, multi-variety, and flexible production. It implements horizontal and vertical deep integration of the system. Figure 9 shows the integrated experimental platform of IIOT intelligent production line. The platform has complete intelligent production and processing functions oriented to the Industrial Internet of Things. It integrates modules such as task assignment, network transmission, and dynamic scheduling. This platform can deploy the wireless sensor network, and collect equipment status information.

We use OMNET++ 5.0 to establish the network communication simulation environment for the intelligent production line platform in Figure 9. The main functions of the OPC UA open-source communication protocol are implemented based on INET open-source framework for porting [36]. At the same time, OMNET++ is highly scalable. We develop an industrial wireless network simulation environment with OPC UA as the communication method. We simulate the actual application environment, which consists of various devices and a sensor network. The devices include air compressors, motors, etc. The sensor network contains temperature sensors, humidity sensors, and vibration sensors, etc.

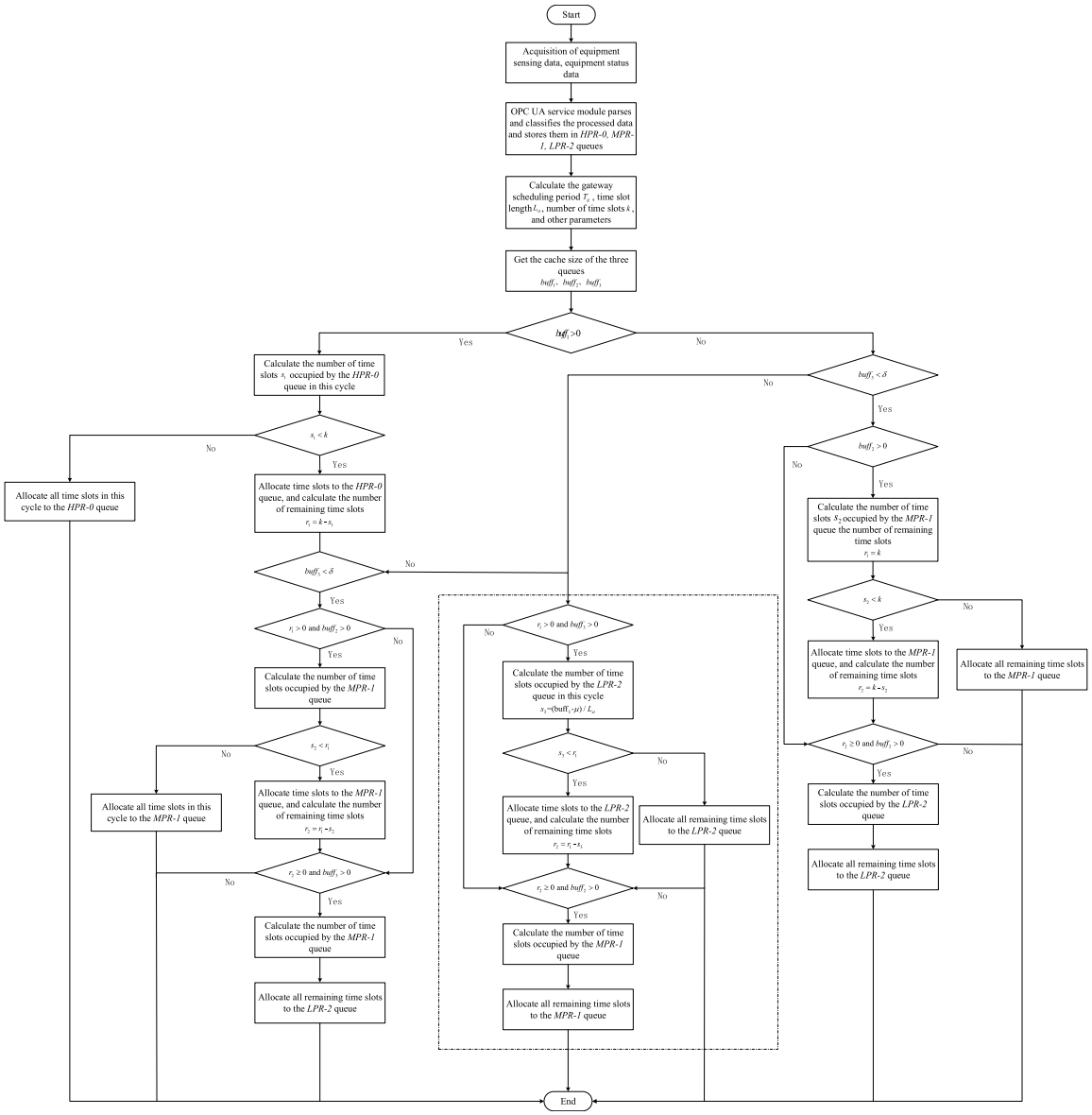

FIGURE 7. Flow chart of dynamic preemptive scheduling strategy based on service differentiation.

A. SIMULATION EXPERIMENT ENVIRONMENT

We designed eleven communication nodes in this simulation environment. These nodes include six nodes as wireless data sources, three nodes as data relay nodes, one master node as an industrial wireless gateway, and one node as a client. The wireless data sources include a robotic arm node, an air compressor node, a servo motor node, and three wireless sensor nodes.

The three device data sources can simulate data such as equipment status information and processing information in actual industrial scenarios. The three sensor data sources are used to simulate sensing data such as temperature, humidity and vibration in the network.

The other three relay nodes can simulate the multi-hop transmission of sensing data in a tree topology. Figure 10 shows a directed graph depicting the connection relationships of the nodes.

We select 4 channels as physical channels for simulated industrial wireless networks. Let the length of a single time slot in IEEE 802.15.4e TSCH mode be $4ms$. The maximum transmission rate of the underlying wireless device is $250Kbps$. The maximum transmission rate of the master node gateway is $10Mbps$. Table 2 describes each parameter of the model. We set two sets of conflicting interference nodes to verify the performance of the load-aware time slot scheduling algorithm. These two groups of nodes are N_2 and N_3, and N_6 and N_8, respectively. Figure 11 shows the matrix of connection relationships for each node in the algorithm.

The load-aware time-slot scheduling algorithm obtains a subset of four duplex-free conflicting edges CFL in

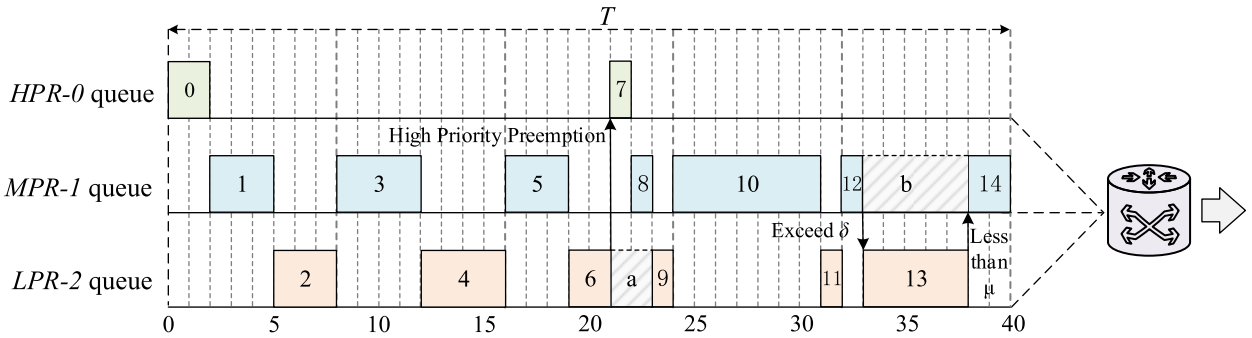

FIGURE 8. Model scheduling example.

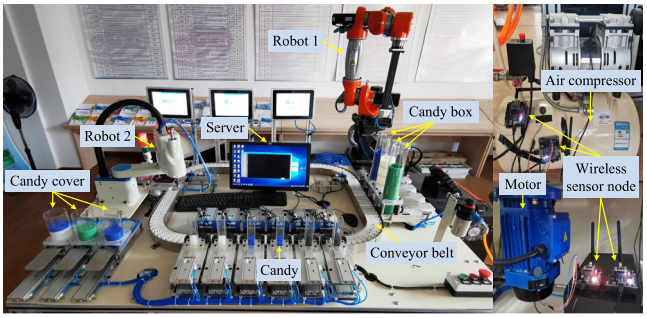

FIGURE 9. Integrated experimental platform of IIoT intelligent production line.

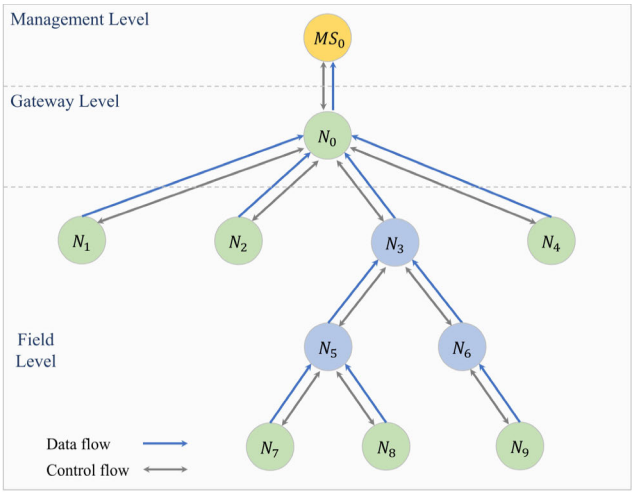

FIGURE 10. Intelligent production line network topology diagram.

the network by matrix matching. the set of *CFL* includes $\{e_1, e_5, e_9\}$, $\{e_2, e_6, e_7\}$, $\{e_3, e_8\}$, and $\{e_4\}$.

Then, the model excludes the known interference conflict set *CI*. According to the load size of each node, the algorithm starts to execute the data scheduling task. This experiment follows the algorithm rules and specifies that a subset e_4 occupies one channel for data transmission. In this way, the channel utilization is improved and the probability of channel congestion is reduced.

TABLE 2. Model parameter description.

Parameter name	Value	Symbol
Number of channels	4	*chnum*
Time slot length	4 *ms*	*stlen*
Number of time slots	250	*stnum*
Number of source nodes	6	—
Number of master nodes	1	—
Number of relay nodes	3	—
Queue threshold	50000	δ
Queue stability value	8000	μ
Device transmission rate	250 *Kbps*	*bntspeed*
Gateway transmission rate	10 *Mbps*	*gtwspeed*
Interference conflict nodes	$\{N_2, N_3,\}$, $\{N_6\, N_8\}$	—

	N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
N_0	0	0	0	0	0	0	0	0	0	0
N_1	1	0	0	0	0	0	0	0	0	0
N_2	1	0	0	0	0	0	0	0	0	0
N_3	1	0	0	0	0	0	0	0	0	0
N_4	1	0	0	0	0	0	0	0	0	0
N_5	0	0	0	1	0	0	0	0	0	0
N_6	0	0	0	1	0	0	0	0	0	0
N_7	0	0	0	0	0	1	0	0	0	0
N_8	0	0	0	0	0	1	0	0	0	0
N_9	0	0	0	0	0	0	1	0	0	0

FIGURE 11. Matrix diagram of network connection nodes.

B. MODEL PERFORMANCE ANALYSIS

We selected data about 50 cycles for the analysis. The utilization of each channel in each scheduling cycle and the average utilization of the cycle time slots are calculated [37].

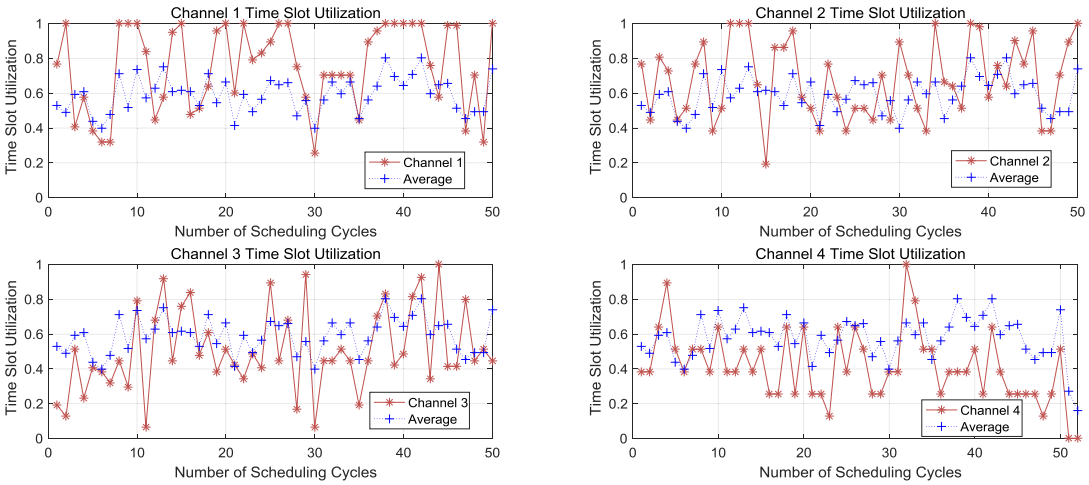

FIGURE 12. Comparison chart of each channel utilization rate and average utilization rate.

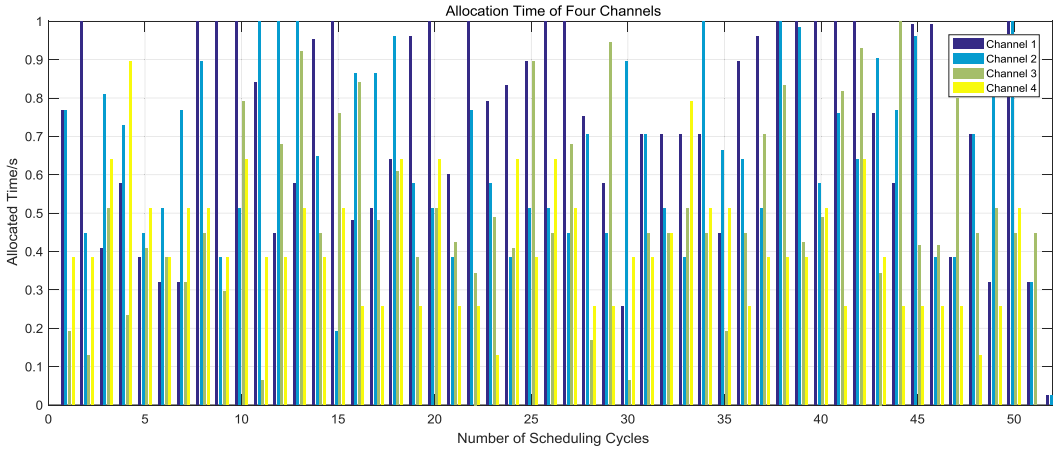

FIGURE 13. Actual allocated time for each channel.

They are given by Equation (7).

$$\begin{cases} CH\_UTIL_i = \dfrac{\sum_{k=1}^{m} node_k}{stlen \times stnum \times bntspeed} \\ AVG\_UTIL = \dfrac{\sum_{i=1}^{chnum} CH\_UTIL_i}{chnum} \end{cases} \tag{7}$$

Figure 12 shows the utilization and average utilization of each channel for the proposed model. As can be seen from the figure, the ratio of the average utilization of the four channels that is higher than half of the utilization of the model channels is 76%. We analyze the data of all runs of the model. The ratio of the utilization of channel 1 above the average utilization of all channels is 70%. The ratio of channel 1 utilization above half of the modeled channel utilization is 80%. The ratio of the utilization of channel 2 above the average utilization of

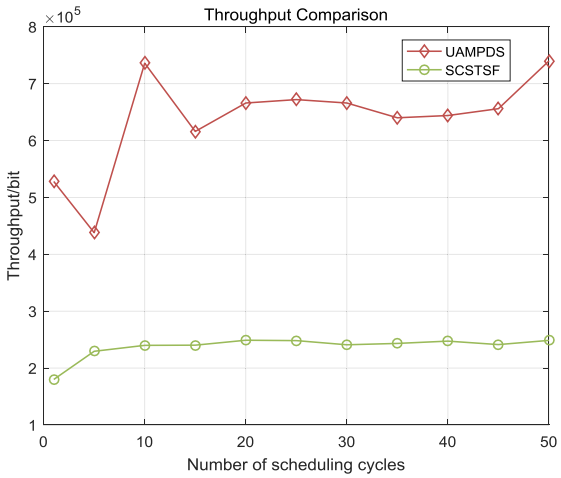

FIGURE 14. Throughput comparison chart.

all channels is 64%. The ratio of channel 2 utilization above half of the modeled channel utilization is 78%. The ratio of the utilization of channel 3 above the average utilization of

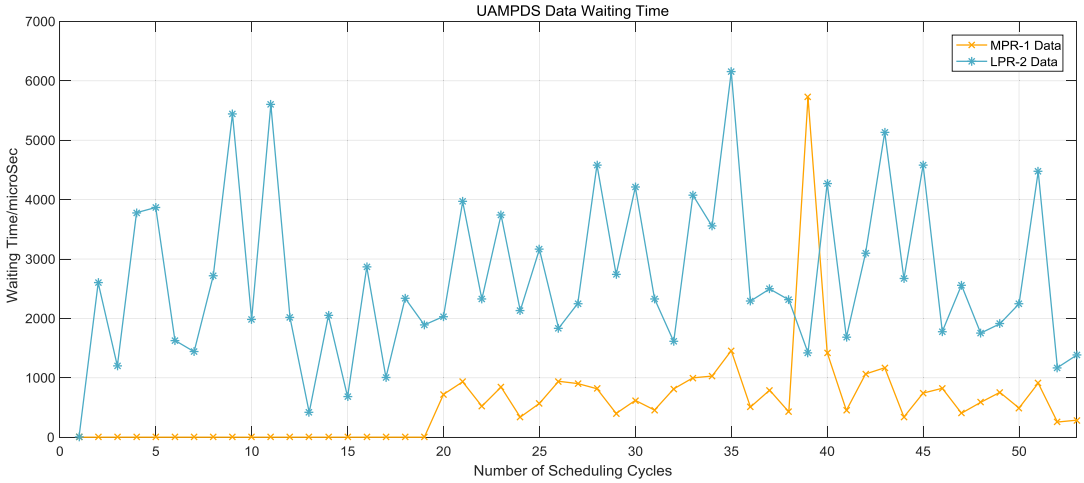

FIGURE 15. UAMPDS data waiting time.

all channels is 32%. The ratio of channel 3 utilization above half of the modeled channel utilization is 38%. The ratio of the utilization of channel 4 above the average utilization of all channels is 16%. The ratio of channel 4 utilization above half of the modeled channel utilization is 40%. As we can see, channel 1 has the highest utilization among all channels. All its time slots are used to transmit data in 17 cycles. Channel 2 has higher utilization than channel 3 and channel 4. Meanwhile, channel 3 performs better than channel 4 in terms of utilization. Since $\{e_4\}$ of *CFL* occupies channel 4 alone, channel 4 has the lowest utilization. Figure 13 shows the actual time values allocated for the four channels in each cycle.

Figure 14 shows the comparison of the UAMPDS model with the scheduling model for single channel and single time slot frames (SCSTSF) in terms of throughput. We have selected 11 cycles to compare the throughput profile of both models. The throughput is given by Equation (8) [26].

$$Throughput_i = \sum_{i=1}^{chnum} \sum_{k=1}^{m} node_k \tag{8}$$

After the first cycle, the throughput of the UAMPDS model fluctuates slightly due to the different amounts of data enerated by the source nodes in each cycle. the throughput of the SCSTSF model is always maintained at a lower level of transmission. We compare the two models under each cycle. The maximum throughput of UAMPDS is about 3.21 times higher than that of SCSTSF. And the average throughput of UAMPDS is about 2.69 times that of SCSTSF. Therefore, the UAMPDS model is much higher than the SCSTSF model in terms of throughput.

The UAMPDS model classifies data at the gateway layer and defines three priority levels of data. Meanwhile, *HPR-0* priority real-time data has the highest authority to send data,

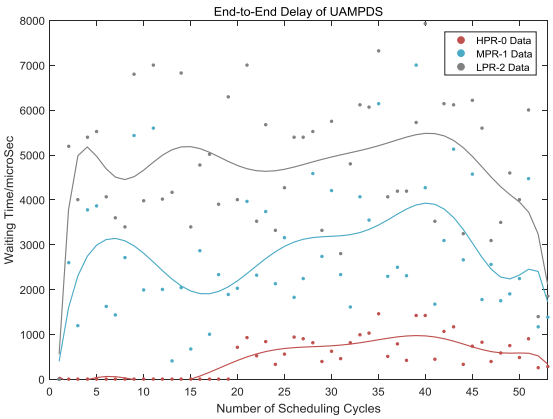

FIGURE 16. UAMPDS end-to-end delay statistics.

and *MPR-1* and *LPR-2* priority data can perform mutual preemption operations under certain conditions.

Figure 15 shows the waiting time for *MPR-1* and *LPR-2* priority data at each cycle. The source node in the network does not generate *HPR-0* data until cycle 20. The *MPR-1* data scheduling wait time is 0. And the *LPR-2* data is scheduled after the *MPR-1* data scheduling. The *LPR-2* data wait time is always higher than the *MPR-1* data.

After cycle 20, *HPR-0* data appear in the network. *MPR-1* data always occurs queue waiting in the following cycles, but its waiting time is lower than *LPR-2* data.

However, the *LPR-2* queue data cache exceeds the threshold value δ at cycle 39. Therefore, the dynamic preemption operation is triggered. The *LPR-2* data preempts the *MPR-1* data sending time until the *LPR-2* queue data cache is less than the stable value μ.

In Figure 15, the waiting time for *MPR-1* data increases rapidly, and it is higher than that for *LPR-2* data at cycle 39. This approach avoids the starvation phenomenon due to the continuity preemption of high-priority data.

It also protects the stability of the buffer queue in the gateway server. In short, this method guarantees the transmission performance of low-priority data.

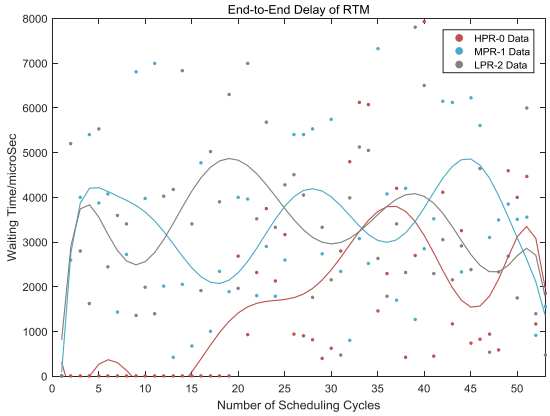

FIGURE 17. RTM end-to-end delay statistics.

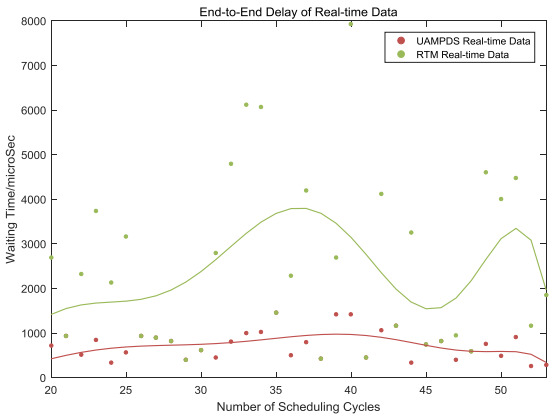

FIGURE 18. End-to-end delay statistics of real-time data.

We also complete the tests and statistics of end-to-end time delay and analyze these test data. Figure 16 and Figure 17 show the end-to-end delay of the UAMPDS model and RTM (Random Transmission Method), respectively. The RTM model is a random selection method for wireless node communication, which sends data without distinguishing the type of service [38]. The different colored scatter points represent the time delay values for various priority data, respectively. The three curves are the fitted curves of time delay for the three priority data.

Figure 16 shows the latency of the three types of UAMPDS data at each cycle. Before cycle 20, the network node does not generate *HPR-0* priority data. Therefore, there is no end-to-end delay for the *HPR-0* queue. The delay values of *MPR-1* and *LPR-2* priority data change with the cycle.

After cycle 20, as the number of *HPR-0* priority data gradually increases, the end-to-end delay of these data also gradually increases. But, we analyze the overall trend of the three fitted curves. As a whole, the end-to-end delay of *HPR-0* is always smaller than that of *MPR-1* and *LPR-2*. The delay value of *MPR-1* is smaller than that of *LPR-2*.

Figure 17 shows the end-to-end delay of the RTM model under the same transmission conditions. In terms of end-to-end delay, the ratio of *HPR-0* over both *MPR-1* and *LPR-2* is 67.65%. The ratio of *MPR-1* having higher latency than *LPR-2* is 50%.

From the curve trend, we can see that the RTM model has poor performance in data differentiation services, and RTM has difficulty in ensuring priority transmission of high-priority data. The UAMPDS model can better meet the transmission demand of high real-time data in the intelligent production line network environment.

We compare the latency of real-time data of both models of UAMPDS and RTM. As shown in Figure 18, we take cycle 20 as the starting point. At this moment, *HPR-0* priority data starts to appear in the network.

In terms of end-to-end delay, the UAMPDS model always has a smaller delay value for real-time data than the RTM model. In 50 cycles, the real-time performance of the UAMPDS model is about 5.59 times that of the RTM model. In terms of real-time performance, there is a gap between the two models. The transmission delay of the UAMPDS model is much lower than that of the RTM model.

Therefore, the UAMPDS model can meet the differentiated transmission of data in the intelligent production line network environment. It also ensures the priority processing of high-priority data. It is superior in the reliable transmission of real-time data in industrial networks.

VI. CONCLUSION

Aiming at the deficiencies in data transmission of wireless networks in the industrial IoT environment, a real-time Communication Model Based on OPC UA Wireless Network for Intelligent Production Line is proposed. In this study, we use IEEE 802.15.4e TSCH as the wireless communication infrastructure. Meanwhile, the model incorporates OPC UA to ensure real-time communication of devices in industrial networks. We design a load-aware time-slot scheduling algorithm and a dynamic preemptive resource scheduling strategy based on service differentiation. In addition, we also comprehensively test the performance of this communication model. It effectively reduces the transmission delay of real-time data under the constrained network resources. This approach guarantees the efficient transmission of real-time data and improves the overall service level of the intelligent production line communication network.

However, we only evaluated the performance of the model through a simulated experimental environment. It is somewhat different from the application in real production line scenarios. In future research, we will apply this model to the IIOT intelligent production line platform. At the same time, we will improve the algorithm in terms of time complexity. The model will further reduce the model time loss while meeting the requirements of complex application scenarios in the production line. It will provide a theoretical basis

and algorithmic support for an intelligent production line communication system.

ACKNOWLEDGMENT

(Anying Chai and Zhenyu Yin contributed equally to this work.)

REFERENCES

[1] M. H. Zhang, "The world's manufacturing industry is facing major adjustments," *China Electron. Post*, vol. 7, no. 3, pp. 1–3, 2019.

[2] K. D. Thoben, S. Wiesner, and T. Wuest, "'Industrie 4.0' and smart manufacturing-a review of research issues and application examples," *Int. J. Automat. Technol.*, vol. 11, no. 1, pp. 4–16, 2017.

[3] J. H. Yan and B. L. Li, "Research hotspots and trend analysis of intelligent manufacturing," *Chin. Sci. Bull.*, vol. 65, no. 8, pp. 684–694, 2020.

[4] Y. F. Wang and X. Z. Zhou, "A review of research on domestic and international intelligent manufacturing," *Forum Sci. Technol. China*, vol. 10, no. 4, pp. 154–160, 2016.

[5] Y. Wei and P. Xiang, "Application of intelligent production line system integration," *Mech. Elect. Eng. Technol.*, vol. 48, no. 8, pp. 125–128, 2019.

[6] *OPC Unified Architecture Specification Part 1: Overview and Concepts Release 1.04*, B. S. Inst., U.K., 2017.

[7] L. Han, M. L. Li, L. Zhuo, H. Yang, and X. Zhang, "Interpretation of, industrial Internet of Things white paper," *Inf. Technol. Standardization*, vol. 2017, no. 12, pp. 30–34, 2017.

[8] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[9] K. Kim, I. Kim, and J. Lim, "National cyber security enhancement scheme for intelligent surveillance capacity with public IoT environment," *J. Supercomput.*, vol. 73, no. 3, pp. 1140–1151, Mar. 2017.

[10] F. Palm, S. Gruner, J. Pfrommer, M. Graube, and L. Urbas, "Open source as enabler for OPC UA in industrial automation," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Luxembourg, Luxembourg, Sep. 2015, pp. 1–6.

[11] X. Li, D. Li, J. Wan, A. V. Vasilakos, C. F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless Netw.*, vol. 23, no. 1, pp. 23–41, 2017.

[12] E. Al-Masri, "QoS-aware IIoT microservices architecture," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Seattle, WA, USA, Oct. 2018, pp. 171–172.

[13] A. Girbea, C. Suciu, S. Nechifor, and F. Sisak, "Design and implementation of a service-oriented architecture for the optimization of industrial applications," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 185–196, Feb. 2014.

[14] W. Kim and M. Sung, "Standalone OPC UA wrapper for industrial monitoring and control systems," *IEEE Access*, vol. 6, pp. 36557–36570, 2018.

[15] W. Kim and M. Sung, "OPC-UA communication framework for PLC-based industrial IoT applications," in *Proc. 2nd Int.Conf. Internet–Things Design Implement.*, Pittsburgh, PA, USA, Apr. 2017, pp. 327–328.

[16] M. S. Mahmoud, M. Sabih, and M. Elshafei, "Using OPC technology to support the study of advanced process control," *ISA Trans.*, vol. 55, pp. 155–167, Mar. 2015.

[17] R. F. Maia, Â. J. Bálsamo, G. A. W. Lopes, A. A. Massote, and F. Lima, "Evaluation of OPC-UA communication in an autonomous advanced manufacturing cell implementation," *Gestão Produção*, vol. 27, no. 4, p. e5414, 2020.

[18] N. Nasser, L. Karim, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1448–1459, Apr. 2013.

[19] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, "Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks," in *Proc. IEEE 23rd Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sydney, NSW, Australia, Sep. 2012, pp. 327–332.

[20] W. X. Xia, L. L. Pang, D. D. Li, W. M. Zhang, and Z. W. Jia, "A new industrial IoT solution: Application of OPC UA and 5G in industrial test bed," *Manuf. Automat.*, vol. 42, no. 7, pp. 74–78, 2020.

[21] J. Kim, G. Jo, and J. Jeong, "A novel CPPS architecture integrated with centralized OPC UA server for 5G-based smart manufacturing," *Procedia Comput. Sci.*, vol. 155, pp. 113–120, 2019.

[22] H. Huang, W. Miao, G. Min, J. Tian, and A. Alamri, "NFV and blockchain enabled 5G for ultra-reliable and low-latency communications in industry: Architecture and performance evaluation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5595–5604, Aug. 2021.

[23] P.-V. Mekikis, K. Ramantas, A. Antonopoulos, E. Kartsakli, L. Sanabria-Russo, J. Serra, D. Pubill, and C. Verikoukis, "NFV-enabled experimental platform for 5G tactile Internet support in industrial environments," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1895–1903, Mar. 2020.

[24] L. Liu, H. Du, H. Wang, and T. Liu, "Construction and application of digital twin system for production process in workshop," *Comput. Integr. Manuf. Syst.*, vol. 25, no. 6, pp. 1536–1545, 2019.

[25] Y. Ning, "Design and implementation of industrial wireless sensor network supporting OPC UA," Ph.D. dissertation, Beijing Jiaotong Univ., Beijing, China, 2018.

[26] T. Song and K. Li, "Data communication technology and applicationins for intelligent manufacturing workshops based on OPC UA," *China Mechanical Engineering*, vol. 31, no. 14, pp. 1693–1699, 2020.

[27] S. Wang, J. Ouyang, D. Li, and C. Liu, "An integrated industrial Ethernet solution for the implementation of smart factory," *IEEE Access*, vol. 5, pp. 25455–25462, 2017.

[28] J. Wan, J. Yang, S. Wang, D. Li, P. Li, and M. Xia, "Cross-network fusion and scheduling for heterogeneous networks in smart factory," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6059–6068, Sep. 2020.

[29] M. H. Min, Z. J. Yang, Z. S. Li, and Z. F. Liu, "Research on multi-slot frame scheduling algorithm in industrial IoT applications," *Comput. Eng.*, vol. 42, no. 11, pp. 15–21, 2016.

[30] L. Wang, F. Liu, X. Wang, and Y. Shi, "Diff-service minislot scheduling alogrithm in IEEE 802.16 mesh mode," *Comput. Eng. Appl.*, vol. 34, pp. 105–108, Mar. 2006.

[31] X. Yang, "Design and implementation of IEEE 802.15.4e TSCH timeslot scheduling based on contiki," Ph.D. dissertation, Univ. Electron. Sci. Technol. China, Chengdu, China, 2015.

[32] M. R. Palattella, N. Accettura, L. A. Grieco, Boggia, G. M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3655–3666, Oct. 2013.

[33] M. H. Schwarz and J. Börcsök, "A survey on OPC and OPC-UA: About the standard, developments and investigations," in *Proc. XXIV Int. Conf. Inf., Commun. Automat. Technol. (ICAT)*, Sarajevo, Bosnia and Herzegovina, Oct./Nov. 2013, pp. 1–6.

[34] P. Y. Wang, "Research and simulation of schedule algorithm in time sensitive networking," Beijing Univ. Posts Telecommun., Beijing, China, 2018.

[35] G. C. Buttazzo, M. Bertogna, and G. Yao, "Limited preemptive scheduling for real-time Systems. A survey," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 3–15, Feb. 2013.

[36] F. Palm, S. Grüner, J. Pfrommer, M. Graube, and L. Urbas, "Open62541: Der offene OPC UA stack," in *Proc. Jahreskolloquium 'Kommunikation in der Automation'*, Aachen, Germany, Nov. 2014, pp. 1–9.

[37] W. L. Wang, Y. F. Cen, and X. W. Yao, "Muti-priority hybrid slot transmission method based on IEEE802.11e EDCA," *Comput. Sci.*, vol. 40, no. 12, pp. 156–159, 2013.

[38] C. Y. Liu and L. C. Zhang, "Dynamic multi-priority scheduling for cyber-physical systems," *Comput. Sci.*, vol. 42, no. 1, pp. 28–32, 2015.

ANYING CHAI was born in1992. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences. His current research interests include Industrial Internet of Things and network communication scheduling.

YUE MA was born in 1960. He is currently a Researcher with the Shenyang Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include intelligent manufacturing, computer application technology, and numerical control technology.

ZHENYU YIN was born in 1979. He is currently a Researcher with the Shenyang Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include advanced manufacturing, Industrial Internet of Things, embedded technology, and numerical control technology.

MINGSHI LI was born in 1991. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences. His current research interests include Industrial Internet of Things and intelligent manufacturing.