A Novel Real-Time Deterministic Scheduling Mechanism in Industrial Cyber-Physical Systems for Energy Internet

Yuhuai Peng, Alireza Jolfaei, *Senior Member, IEEE*, and Keping Yu, *Member, IEEE*

***Abstract*—As an effective distributed renewable energy utilization paradigm, a microgrid is expected to realize the high integration of the industrial cyber-physical systems (CPS), which has attracted extensive attention from academia and industry. However, the real-time interaction and feedback loop between physical systems and cyber systems have posed severe challenges to the reliability, determinacy, and energy efficiency of the multiway flow of information and communication transmission. In order to solve the problem of slot scheduling and data transmission (SSDT) in the microgrid, a novel real-time deterministic scheduling (RTDS) scheme for industrial CPS is proposed in this article. First, the SSDT is formulated as a multiway flow scheduling problem, and it is theoretically proved that the SSDT problem is NP-hard. Then, the RTDS scheme designs two heuristic algorithms: scheduling request preprocessing and greedy-based multichannel time slot allocation for an optimal scheduling solution. Practical experimental results demonstrate that the proposed RTDS scheme has significant advantages in packet loss rate, deadline guarantee rate, and energy consumption compared with the traditional schemes, and thus, is more suitable for deployment in microgrid systems.**

***Index Terms*—Deterministic scheduling, greedy algorithm, industrial cyber-physical systems (CPS), microgrid systems, optimal scheduling.**

I. Introduction

As THE basic unit of the energy Internet, a microgrid [1] can independently realize the generation, aggregation, storage, and sharing of renewable energy within the "local area network," form a closed loop of the multiway flow of information and energy, and ensure the efficient and sustainable supply of clean energy. As a more open, interconnected, and intelligent cyber-physical system (CPS), the microgrid will provide physical system functions such as voltage management, distributed generation, frequency control, and demand-side response [2], [3], as well as cyber system functions such as real-time energy information collection, processing, analysis, and decision making [4]. With the deployment of a large number of intelligent terminals [5] and the wide application of Big Data and artificial intelligence technology in the industrial CPS [6]–[8], the existing scheduling schemes are gradually difficult to meet the requirements of real time and reliability for data transmission of new services [9]. And it is difficult to realize the deterministic transmission of control instructions in the microgrid, thus affecting the stable operation of the microgrid. Therefore, an effective deterministic scheduling scheme for the CPS will improve the fine management level of microgrid operation and promote the further development of energy Internet.

A large number of sensors, intelligent terminals [10], and controllers are deployed at the underlying layer of CPS as key infrastructure. These devices will be mainly used for state monitoring, decision making, and action execution of the microgrid, realizing the fine control of power generation, transmission, and consumption. With the continuous increase of smart devices, the expansion of network scale and the emergence of high concurrent services, the real time and reliability of data transmission are gradually difficult to meet, which puts forward high requirements for the CPS [11]–[13], such as large bandwidth, massive connection, low energy consumption, and low delay. When a large amount of data are transmitted through the wireless channel at the same time, the wireless interference and collision probability will increase significantly. This will increase the packet loss rate, delay, and energy consumption of data transmission, and reduce the reliability and stability of the CPS. When the situation is serious, the microgrid system may be paralyzed [14].

In order to solve the wireless transmission problem of the CPS, a variety of scheduling methods have been proposed [15]–[18]. As a competition-based scheduling mechanism, the traditional carrier sense multiple access with collision avoidance (CSMA/CA) [19] method has been widely used in industrial scenarios. The scheduling method [20] based on fixed

Manuscript received August 26, 2021; revised November 1, 2021; accepted December 22, 2021. Date of publication December 31, 2021; date of current version May 6, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1702000, in part by the National Natural Science Foundation of China under Grant 61871107, and in part by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) under Grant JP18K18044 and Grant JP21K17736. Paper no. TII-21-3709. *(Corresponding author: Keping Yu.)*

Yuhuai Peng is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: pengyuhuai@mail.neu.edu.cn).

Alireza Jolfaei is with the College of Science and Engineering, Flinders University, Adelaide, SA 5042, Australia (e-mail: alireza.jolfaei@flinders.edu.au).

Keping Yu is with the Global Information and Telecommunication Institute, Waseda University, Tokyo 169-8050, Japan (e-mail: yukeping@fuji.waseda.jp).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3139357.

Digital Object Identifier 10.1109/TII.2021.3139357

time slot allocation can allocate fixed channels and time slots to each node in the network, effectively avoid collision and retransmission. In addition, the recently proposed multichannel protocol [21] allows devices to occupy multiple channels to communicate at the same time [22], which improves the communication rate and channel utilization. Considering that some data have strict survival time limits, scheduling strategies such as rate monotonic algorithm [23] and earliest deadline priority scheduling algorithm [24] are proposed to improve the real-time performance. However, most scheduling schemes ignore the strict deadline guarantee rate in deterministic transmission, which is difficult to meet the deterministic and real-time requirements of multiple services in the complex CPS.

In order to solve the problem of the real-time and deterministic data transmission, a real-time deterministic scheduling (RTDS) scheme for the CPS is proposed. The main contributions of this article can be summarized as follows.

1) Based on the features of the CPS, a three-layer network topology consisting of software-defined network (SDN) controller, switch, and microgrid smart devices is established. Also, the data transmission model of the CPS of the microgrid is given; then, the SSDT is formulated as a multiway flow scheduling problem. Through complexity analysis, it is theoretically proved that the SSDT problem is NP-hard.
2) In order to solve the SSDT problem, an RTDS scheme for the CPS of the microgrid is proposed. The RTDS designs two heuristic algorithms: scheduling request preprocessing (SRP) and greedy-based multichannel time slot allocation (GMT) for the optimal scheduling solution. The SRP can give priority to packets with the earliest expiration period, while the GMT ensures that these higher priority packets are greedily assigned to available time slots in advance. The working principle of the proposals is explained by an illustration. Moreover, the time complexity and optimality proof is also given.
3) Extensive practical experiments are carried out with the TI CC2530 module. Experimental results demonstrate that the proposed RTDS scheme is superior to traditional schemes in terms of packet loss rate, deadline guarantee rate, and energy consumption, and thus, is more suitable for deployment in microgrid systems.

The rest of this article is organized as follows. In Section II, related works are introduced. In Section III, the SSDT problem of the CPS is defined, and the complexity of the problem is analyzed. In Section IV, the RTDS scheme is proposed and the optimality of the scheme is proved. In Section V, the performance is analyzed. Finally, Section VI concludes this article.

II. Related Works

The real-time scheduling problem in the CPS has always been a research hotspot. In scheduling, there are two main solutions for channels and slots allocation: competitive and noncompetitive.

A. Competition-Based Algorithm

Doudou *et al.* [25] sorted out and classified competition-based algorithms. Regarding the research on competition-based scheduling methods, many scholars aim to improve the CSMA/CA algorithm. Ni *et al.* [26] proposed a discrete-time version of the CSMA/CA algorithm, which is based on the so-called Glauber dynamics generalization in statistical physics. This algorithm allows multiple links to update their state in a single time slot, and thus, can generate conflict-free transmission schedules. At the same time, the conflicts are clearly considered in the control phase of the protocol, thereby relaxing the assumption of the CSMA/CA under ideal conditions and achieving low latency while maintaining the throughput optimization characteristics. Ghosh *et al.* [27] proposed an improved CSMA/CA mechanism called the silent period CSMA/CA, which defines a dedicated silent period, and where nodes perform channel detection during the backoff phase without a separate dedicated sensing interval. Sahoo *et al.* [28] designed a MAC protocol in order to avoid data conflicts between nodes in a wireless sensor network. This protocol is based on the CSMA/CA mechanism, and a retransmission restriction is designed for nodes with a high packet-collision probability.

B. Noncompetitive Algorithms

Bhatia *et al.* [29] proposed a randomly distributed time-division multiple access (TDMA) scheduling algorithm to overcome this limitation. It can quickly generate a feasible schedule to deal with the relevant competition between nodes in a wireless network. Lin *et al.* [30] proposed a dynamic TDMA protocol based on the standard TDMA protocol and node neighborhood information. This protocol monitors the control messages that are periodically broadcast by neighboring nodes during the control cycle. Tan *et al.* [31] combined the cellular structure with the field bus and proposed a greedy scheduling (G-schedule) algorithm by using independent characteristics of multiple channels. This algorithm can effectively avoid intercell interference and proves the optimality of the algorithm. Ching *et al.* [32] defined a small time slot scheduling problem of wireless networks as an integer linear programming model and a small time slot as an atomic bandwidth allocation unit for data transmission between users and base stations.

However, most of the conducted studies have considered the competing scheduling requirements but not the performance and reliability of real-time data transmission in industrial network environments, and research on strict deadlines of data scheduling in microgrids is still inadequate.

III. Problem Statement

A. System Model

The proposed network topology is shown in Fig. 1, where it can be seen that it includes three main devices: controllers, switches, and sensors. Controllers and switches are connected in a wired manner forming a star network topology. Each switch and multiple sensors form a star topology called a cell, where

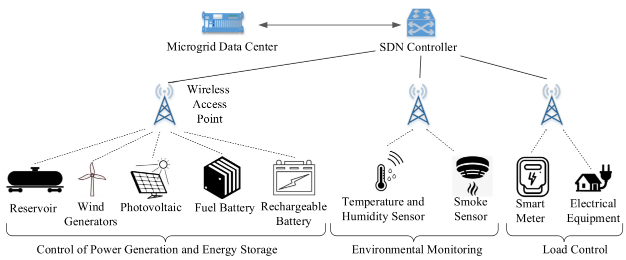


Fig. 1. Proposed network topology.

data transmission between the switch and sensors is performed wirelessly. The main functions of the controller include obtaining the topology information of the entire network, storing the conflict set and deadline data of each node, making a schedule for each sensor according to the stored information, and sending the schedule information to the switch. There is no data transmission relationship between switches. Each switch forwards data received from the sensors in its cell to the network controller. Therefore, the centralized control of the entire network is realized.

B. Problem Formulation

Suppose there are F channels in the network, and each channel is divided into T time slots of the same size, which is called a TDMA superframe; each data request occupies one time slot in a channel. There is only one controller for realizing the centralized control of the entire network. There are I switches under the management of the controller, denoted by $\mathrm{SW}_i (i \in \{0, 1, \ldots, I-1\})$. There are J sensors under the management of each switch, and s_{ij} represents the jth sensor under the ith switch control. Switch SW_i and sensor s_{ij} together form a cell C_i; and T_{ij} represents a data transmission period of the sensor s_{ij}. Each sensor sends a data transmission request at the beginning of the transmitting cycle, and this request must be transmitted to the receiver before the beginning of the next cycle; $s_{ij}.n$ represents the nth dataset sent by sensor s_{ij} (n is an integer greater than one). In this way, the request time RT_{ijn} of data $s_{ij}.n$ can be denoted as $(n-1)T_{ij}$, and its deadline is denoted by DL_{ijn}, where $DL_{ijn} = nT_{ij}$. ET_{ijn} represents the execution time of the nth dataset sent by the sensor s_{ij}. As mentioned previously, the data request must be sent to the receiving end before the start of the next cycle, which means that for the nth dataset sent by the sensor s_{ij}, the execution time ET_{ijn} must meet the following condition:

$$RT_{ijn} < ET_{ijn} \leq DL_{ijn}. \tag{1}$$

As long as there is a data request that does not satisfy (1), it is considered that overflow has occurred.

In addition, $W(i)$ represents the workload of C_i; that is, the size of the workload that C_i needs to transmit is $W(i)$. In C_i, the workload of the sensor s_{ij} is represented by $L_i(j)$. For the sensor s_{ij}, the workload in a cycle is represented by $w_{ij} (w_{ij} = 1)$; this means that the workload of a request is one, i.e., the size of the time slot required to transmit the data is one; $W(i)$, $L_i(j)$, and w_{ij} have the following relationship:

$$W(i) = \sum_{j=0}^{J-1} L_i(j) = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} w_{ij} = IJ. \tag{2}$$

TABLE I
DEFINITION OF SYMBOLS

symbol	definition	symbol	definition
F	The number of channels	RT_{ijn}	The request time of a dataset
T	Time slots in a TDMA superframe	DL_{ijn}	The deadline of a dataset
I	The number of switches	ET_{ijn}	the execution time of a dataset
SW_i	A switch	$W(i)$	the workload of a cell
J	The number of sensors connected to a switch	$L_i(j)$	the workload of a sensor
s_{ij}	A sensor	w_{ij}	A sensor workload in a cycle
C_i	A cell composed of a switch and its sensors	$itf(\omega)$	The wireless transmission interference set in a cell
T_{ij}	Data transmission period of a sensor	λ and ω	Two nodes in a cell
$s_{ij}.n$	The n^{th} dataset sent by a sensor		

For any cell C_i, define a wireless transmission interference set $itf(\omega) = \{\lambda | \omega \neq \lambda\}$, and wireless communication of λ can interfere the communication of ω, or vice versa. According to the definition of $itf(\omega)$, if λ interferes with ω, although ω does not necessarily interferes with λ, it is certain that the two nodes cannot occupy the same time slot in the same channel for communication, i.e., $\lambda \in itf(\omega) \Leftrightarrow \omega \in itf(\lambda)$. Meanwhile, there is an interference between all sensors in the cell of the same switch. In other words, for any cell, $\forall j, \delta \in \{0, 1, \ldots, J-1\}$ and $j \neq \delta$, it holds that $s_{ij} \in itf(s_{i\delta})$.

The RTDS scheme is designed to address the two following problems. Given F, T, I, J, itf, DL, and W, the two main problems are how to use the least scheduling workload W over F channels within T time slots and how to ensure that two interfering sensors (i.e., $s_{ij} \in itf(s_{i\delta})$) do not transmit data through the same channel in the same time slot. As mentioned before, a sensor's request must be sent to the receiver before its deadline DL is reached. For narrative simplicity, the SSDT problem is denoted as $SSDT(F, T, I, J, itf, DL)$.

An array $s[i][j][f][t]$ represents a scheduling table of a sensor s_{ij}. $s[i][j][f][t] = 1$ indicates that the sensor s_{ij} performs wireless data transmission in the *t*th time slot of the *f*th channel; other values mean that data cannot be transmitted on this time slot, "0" indicates this time slot is free, and "-1" indicates this time slot has been already occupied by other nodes, so the current node is prohibited from using it. The definition of the aforementioned symbols is listed in Table I.

The definition of effective scheduling is introduced to evaluate the effectiveness of scheduling results.

Definition 1 (Effective scheduling): Schedule is valid iff the following three conditions hold simultaneously.

Condition 1: $\forall s_{ij}, f, t (i, j, f, t \in N^+, 0 \leq i < I, 0 \leq j < J, 1 \leq f \leq F, 1 \leq t \leq T)$, such that $s[i][j][f][t] = 1$, we have $\forall \omega, \tau, s_{\omega\tau} \in itf(s_{ij}), s[\omega][\tau][f][t] \neq 1$.

Condition 2: $\forall i, j \in N^+(0 \leq i < I, 0 \leq j < J), \sum_{f=1}^{F} \sum_{t=1}^{T} \text{cnt}(s[i][j][f][t]) = \sum_{i=0}^{I-1} W(i)$, where $\text{cnt}(x) = \begin{cases} 1 & \text{x = 1,} \\ 0 & \text{otherwise} \end{cases}$.

Condition 3: $\forall i, j, n \in N^+(0 \leq i < I, 0 \leq j < J, 1 \leq n \leq T/T_{ij}), RT_{ijn} < ET_{ijn} \leq DL_{ijn}$

The three conditions in Definition 1 can be explained as follows. *Condition 1* means that if node s_{ij} has chosen to transmit data in a certain time slot of a certain channel, that is, $s[i][j][f][t] = 1$, then node $s_{\omega\tau}$ that interferes with it will not be allowed to transmit data in this time slot of this channel, that is, $s[\omega][\tau][f][t] \neq 1$. The $cnt(x)$ in *Condition 2* is used to count the number of "1"s in the schedule, that is, to count how many data have completed transmission. If the statistical result is equal to the total load, it means that all data have been transmitted; otherwise, there are unscheduled nodes. Only when all data are successfully scheduled, the scheduling is considered effective. *Condition 3* stipulates that a node must complete data transmission between the request time and the deadline.

If there is such effective scheduling, the problem is considered schedulable; otherwise, it is considered nonschedulable.

C. Complexity Analysis

In the following, the complexity of the $SSDT(F, T, I, J, itf, DL)$ problem is analyzed. By proving that the subproblem of the SSDT problem is an NP problem, it is proved that the SSDT problem is an NP-complete problem [33].

Lemma 1 (NP-Completeness of SSDT it(1,3,I,J,itf,3)): Determining whether $SSDT(F, T, I, J, itf, DL)$ $(F \equiv 1, T \equiv 3, DL = T \equiv 3)$ is schedulable is NP-complete.

If it is proved that $SSDT(1, 3, I, J, itf, 3)$ is an NP-complete problem, then the more general problem $SSDT(F, T, I, J, itf, DL)$ must also be an NP-complete problem. To prove that a problem is NP-complete, first, it is needed to find a known NP-complete problem, and then, prove that the problem can be reduced to the known problem. The NP-complete problem chosen in this study is a three-coloring problem, $G(V, E)$. The proof process is as follows.

Proof: 1) For a subproblem $SSDT(1, 3, I, J, itf, 3)$, it can be verified whether the scheduling result is valid scheduling in polynomial time, indicating that the problem is an NP problem.

2) The following explanation proves that $G(V, E)$ can be reduced to $SSDT(1, 3, I, J, itf, 3)$. In the three-coloring problem $G(V, E)$ of a graph (denoted by G), $V = v_0, v_1, \ldots$ represents a set of vertices in the graph, and E represents a set of edges connecting the vertices in the graph; G is used to solve the following problem: for given sets of vertices and edges in a network, how to use three colors to color all vertices while ensuring that any two vertices connected with edges cannot be colored into the same color.

Given a graph three-coloring problem G, $SSDT(1, 3, I, J, itf, 3)$ problem can be constructed in polynomial time as follows.

a) The number of cells in a network is $I = |V|$.

b) For each cell C_i, $C_j \in itf(C_i)$ iff there is an edge in E connection between v_j and v_i in G.

c) For each cell C_i, the total workload is $W(i) = 1$.

Denote the problem generated by the aforementioned process as $SSDT_G$.

3) In the previous step, the input of G has been equivalent to the input of $SSDT_G$; then, it can be proved that their outputs are also equivalent. The proof needs to be divided into the following two parts: to prove that $SSDT_G$ is schedulable when G is three colorable, and to prove that G is three colorable when $SSDT_G$ is schedulable.

a) Proving that G has a colorable solution, it is proved that $SSDT_G$ is schedulable. When G has a colorable scheme, each vertex in G can be assigned a color from a set of $\{1, 2, 3\}$, and the colors of any two vertices connected by an edge cannot be the same. According to the coloring scheme of G, a scheduling result can be obtained. The specific process is as follows.

If each vertex v_i in G is colored as $c \in \{1, 2, 3\}$, the corresponding cell C_i in $SSDT_G$ will perform wireless data transmission in time slot $t \in \{1, 2, 3\}$. Since any two edge-connected vertices in G cannot be assigned the same color, in $SSDT_G$, any two interfering cells cannot be allocated to the same time slot for wireless data transmission, so *Condition 1* is met. In addition, each vertex in G is assigned a color, which means that each cell in $SSDT_G$ is assigned a time slot, so *Condition 2* is also met. Furthermore, all vertices in G can choose one of the three colors for coloring. Correspondingly, all nodes in a cell in $SSDT_G$ can choose one of the three time slots for wireless data transmission before the deadline, thus *Condition 3* is satisfied. Hence, it holds that $G \Rightarrow SSDT_G$.

a) Proving that $SSDT_G$ is schedulable, it is proved that G has a colorable solution. When $SSDT_G$ is schedulable, each cell is allocated a time slot from a set of $\{1, 2, 3\}$. Furthermore, two interfering cells cannot be allocated the same time slot, and requests sent by nodes in all cells should be scheduled before the deadline. According to the scheduling result of $SSDT_G$, the coloring scheme of G can be obtained, and the specific process is as follows.

For each C_i, if a time slot is allocated at $t \in \{1, 2, 3\}$, it means that the corresponding v_i in G is colored with $c \in \{1, 2, 3\}$. According to Definition 1, two interfering cells in $SSDT_G$ are assigned different time slots. Accordingly, any two vertices connected by edges in G are colored in different colors. Meanwhile, each C_i is assigned a time slot, and each v_i in G is colored. Therefore, coloring scheme G generated according to the scheduling scheme of $SSDT_G$ meets the requirements of three coloring. Hence, it holds that $SSDT_G \Rightarrow G$.

4) According to 3), $G \Leftrightarrow SSDT_G$ can be obtained; that is, the output results of $SSDT_G$ and G are the same. The three-coloring problem G can be reduced to $SSDT_G$, so the scheduling problem $SSDT_G$ is NP-hard. Since 1) has proved that $SSDT_G$ is an NP, then $SSDT_G$ must be NP-complete.

So far, it has been concluded that $SSDT(1, 3, I, J, itf, 3)$ is NP-complete, so Theorem 1 is defined according to Lemma 1.

Theorem 1 (NP-Hardness of SSDT(F,T,I,J,itf,DL)): Determining whether $SSDT(F, T, I, J, itf, DL)$ is schedulable is an NP-hard problem.

Algorithm 1: SRP.

int $sensorBuffer[T - \min T_{ij}][m] \leftarrow 0$;
for int $DL \leftarrow \min\{T_{ij}\}$ **to** $T - 1$ **do**
  for each dataset sent by all sensors $s_{ij}.n$ **do**
    if deadline of $s_{ij}.n$ is equal to DL **then**
      record $s_{ij}.n$ in $sensorBuffer[DL][m]$;
  for each element stored in $sensorBuffer[DL][m]$ **do**
    sort elements in ascending order of a cycle;
return $sensorBuffer$;

Algorithm 2: GMT.

int $s[I][J][F][T] \leftarrow 0$;
for int $visitDL \leftarrow \min\{T_{ij}\}$ **to** $T - 1$ **do**
  for $sensorBuffer[visitDL][0$ **to** $m]$ **do**
    loop:
    for int $f \leftarrow 0$ **to** $F - 1$ **do**
      for int $t \leftarrow 0$ **to** $T - 1$ **do**
        if $w_{ij} \leq 0$ **then**
          break loop;
        if $s[i][j][f][t] == 0$ **then**
          $s[i][j][f][t] \leftarrow 1$;
          $w_{ij} - -$;
          for each $s_{\omega}\tau \in itf(s_{ij})$ **do**
            $s[i][j][f][t] \leftarrow -1$;
    if $w_{ij} > 0$ **then**
      return $FALSE$;
return $s[I][J][F][T]$;

As per Lemma 1, a special case of $SSDT(F, T, I, J, itf, DL)$, $SSDT(1, 3, I, J, itf, 3)$, is already an NP-hard problem; thus, a more general problem, $SSDT(F, T, I, J, itf, DL)$, is also an NP-hard problem.

IV. Proposed RTDS Scheme

A. Algorithm Description

The first part of the proposed scheme is to preprocess the data request of a sensor. Each switch collects information on data to be sent by all sensors in its own cell, including the node number, transmission period, and interference set. The controller preprocesses data based on the information obtained from each cell. The controller groups and sorts packets to be sent according to different deadlines. Data with the same expiration date are stored in the same row of the preprocessing buffer list, and data of each row is sorted according to the data transmission cycle size. This can facilitate the allocation of time slots to packets with the earliest expiration period and the least available time slots in the channel time slot allocation stage. The pseudocode of the SRP algorithm, which returns a 2-D scheduling request preprocessing array, is given in Algorithm 1.

The second part of the proposed scheme extracts elements from $sensorBuffer$ and allocates multichannel time slots to each of the elements in turn using the greedy algorithm. When assigning a transmission time slot to a certain packet, only the influence of the previous assignment result on the current assigned node is considered. In other words, if there is an interference between the scheduled node and the current node, the time slot and channel can be reused. In this process, it is not considered whether the current scheduling result will affect the scheduling of future nodes. See Algorithm 2 for the pseudocode of the GMT algorithm.

After the GMT algorithm is completed, it returns to the schedule table of each node, and then, the controller sends the schedules to all sensor nodes by broadcasting. Finally, a node receives its schedule and completes data transmission according to the received scheduling information.

The time complexity of the scheme is analyzed in the following. The SRP algorithm needs to traverse the data transmission cycle, all switches and all sensors. Therefore, the time complexity of SRP is $O(A \cdot I \cdot J)$. The GMT algorithm needs to traverse all channels, superframes, and interference sets. Therefore, the time complexity of the GMT algorithm is $O(F \cdot T \cdot H)$. And

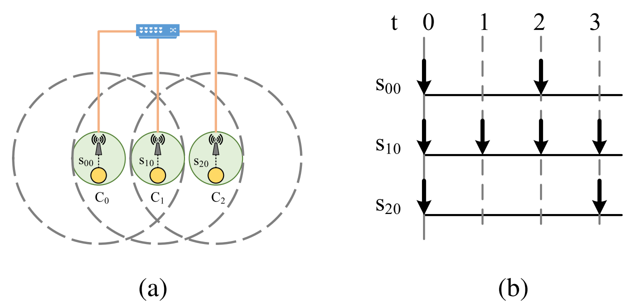


Fig. 2. RTDS scheme example. (a) Network topology diagram. (b) Periodic data request $itf(C_0) = \{C_1\}, itf(C_1) = \{C_0, C_2\}$, and $itf(C_2) = \{C_1\}$.

GMT needs to use the sensorBuffer element of SRP output. The SRP and GMT constitute the RTDS scheme. The time complexity of the RTDS scheme is $O(T \cdot I \cdot J \cdot F \cdot A \cdot H)$, where $A = \max\{T_{ij}\}, H = \max\{itf(S_{ij})\}$; obviously, $A \leq T$ and $H \leq IJ$. Then, the time complexity of the RTDS scheme is $O(T^2(IJ)^2F)$. Therefore, the RTDS is a pseudopolynomial-time scheme, and when T and F are fixed values, it represents a polynomial-time algorithm.

B. Example of RTDS

An example of the scheduling scheme in Fig. 2 is used to demonstrate the RTDS scheme. As shown in Fig. 2(a), in this example, there are three cells denoted by C_0, C_1, and C_2, each of which has only one sensor node denoted by, s_{00}, s_{10}, and s_{20}, respectively. The dotted line in Fig. 2(a) indicates that the data transmission range is an interference range, and the following interference relationships exist between cells, $itf(C_0) = \{C_1\}, itf(C_1) = \{C_0, C_2\}, itf(C_2) = \{C_1\}$. In other words, there is an interference between adjacent cells but not between nonadjacent cells. The transmission cycle of each node is displayed in Fig. 2(b). The cycles of the three nodes s_{00}, s_{10}, and s_{20} are $T_{00} = 2, T_{10} = 1$, and $T_{20} = 3$, respectively. A superframe period is the least common multiple of T_{00}, T_{10},

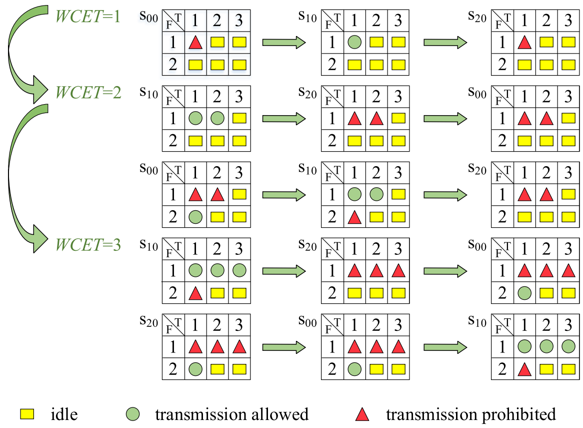


Fig. 3. RTDS scheme execution process.

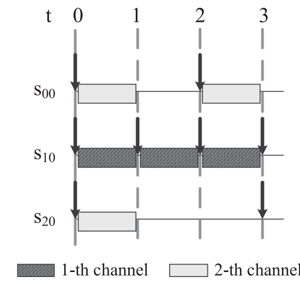


Fig. 4. Scheduling results obtained by the RTDS scheme.

and T_{20}, which is six in this example. For simplicity, only the previous time slot allocation is shown in Fig. 2(b). It is required that a node must complete the transmission before the start of the next cycle.

First, the SRP algorithm is run to preprocess data requests of all nodes to be scheduled. As shown in Fig. 2, data with a deadline of one have only $s_{10}.1$, but data with a deadline of two have $s_{10}.2$ and $s_{00}.1$. Since $T_{10} < T_{00}$, $s_{10}.2$ is stored before $s_{00}.1$; data with a deadline of three have $s_{10}.3$ and $s_{20}.1$. Furthermore, since $T_{10} < T_{20}$, $s_{10}.3$ is stored before $s_{20}.1$. The returned final preprocessing cache list is $sensorBuffer[1][m] = \{s_{10}.1\}$, $sensorBuffer[2][m] = \{s_{10}.2, s_{00}.1\}$, and $sensorBuffer[3][m] = \{s_{10}.3, s_{20}.1\}$. Next, the GMT algorithm is executed using the returned results. The scheduling process is shown in Fig. 3.

The scheduling process of the RTDS is as follows.

1) *Step 1*. Each sensor s_{ij} has a 2-D scheduling table $s[i][j][f][t]$. Before starting scheduling, all states in $s[i][j][f][t]$ are initialized to the "idle" states; that is, all values of $s[i][j][f][t]$ are set to zeros.
2) *Step 2*. According to the preprocessing buffer list returned by the SRT, time slots are allocated for each row of data in the list in turn. When $DL = 1$, all time slots of schedule $s[1][0][f][t]$ are idle. Therefore, the first time slot of the first channel is assigned to $s_{10}.1$, and the corresponding position is marked as "transmission allowed," i.e., $s[1][0][0][0] = 1$. As $itf(C_1) = \{C_0, C_2\}$, the position in the scheduling table corresponding to the node that has interference with s_{10} should be marked as "transmission prohibited"; that is, $s[0][0][0][0] = -1$ and $s[2][0][0][0] = -1$.
3) *Step 3*. When $DL = 2$, the number of allocable time slots of $s_{10}.2$ is less than that of $s_{00}.1$. Therefore, $s_{10}.2$ is preferentially allocated time slots. The second time slot of the first channel is assigned to $s_{10}.2$, and the corresponding position in the schedule of s_{10} is marked as "transmission allowed," i.e., $s[1][0][0][1] = 1$. The corresponding positions in the dispatch table of sensors s_{00} and s_{20} that interfere with s_{10} are marked as "transmission prohibited," i.e., $s[0][0][0][1] = -1$ and $s[2][0][0][1] = -1$. Then, a time slot is assigned to $s_{00}.1$. The data are requested at a time of zero, so the available time slot is the first or second time slot. Since the previous assignment has marked the first and second time slots of the first channel as "transmission prohibited," only the first time slot of the second channel can be allocated to $s_{00}.1$.
4) *Step 4*. When $DL = 3$, $s_{10}.3$ is scheduled first. The third time slot of the first channel is assigned to $s_{10}.3$, and the corresponding position of the schedule is marked as "transmission allowed," i.e., $s[1][0][0][2] = 1$. The corresponding positions of sensors s_{00} and s_{20} that interfere with s_{10} are marked as "transmission prohibited," i.e., $s[0][0][0][2] = -1$, $s[2][0][0][2] = -1$. Then, time slots are assigned to $s_{20}.1$. The allocation method is the same as that of $s_{00}.1$ explained before. After allocation, $s[2][0][1][0] = 1$ and $s[1][0][1][0] = -1$.
5) *Step 5*. Nodes complete data transmission according to the scheduling table. The final scheduling result is presented in Fig. 4.

C. Optimality Analysis

The optimality analysis of the RTDS scheme is divided into two steps. First, it is discussed whether data overflow occurs during the execution of the RTDS scheme and whether all data can be scheduled between the request time and the deadline. Then, it is proved that the RTDS scheme has the least number of time slots required by nodes during the scheduling process.

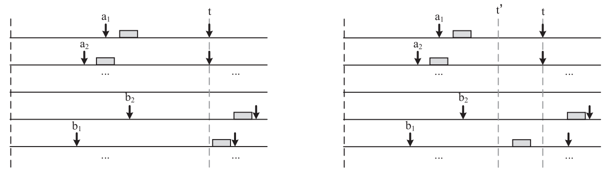
Theorem 2: For a data request sent by a given group of sensors, if and only if the following formula is satisfied, the data request can be scheduled according to the RTDS scheme as follows:

$$\sum_{i=0}^{I-1} \sum_{j=0}^{J-1} \frac{w_{ij}}{T_{ij}} \leq F. \tag{3}$$

Proof: 1) *Proof of necessity:* If the RTDS scheme can be used for scheduling, then (3) must be satisfied.

Suppose the number of sensors in a network is *m*. In the worst case, all cells will interfere with each other, i.e., all sensors will interfere with each other. In this case, the maximum number of required time slots is calculated by

$$T_{\max} = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} \frac{T}{T_{ij}} w_{ij} \tag{4}$$



(a) (b)

Fig. 5. Time slots in Cases 1 and 2. (a) Case 1. (b) Case 2.

where T represents the least common multiple of all T_{ij}, and $\frac{T}{T_{ij}}$ w_{ij} represents the number of time slots required by the sensor s_{ij} to schedule data periodically between zero and T.

If $T_{\max} > FT$, the number of time slots required for scheduling is larger than the number of time slots in the system. In this case, the scheduling must not be successful. In other words, if the scheduling is successful, then there is $T_{\max} \leq FT$, i.e., $\sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\frac{w_{ij}}{T_{ij}} \leq F$, which proves the necessity.

2) *Proof of sufficiency:* If (3) is met, then the RTDS must be used for scheduling.

Assume that (3) is satisfied, and the problem is nonschedulable. Then, an overflow will occur between zero and T. So, there must exist an overflow time $t \in [0, T]$, and the time slot is not idle before the overflow. To be specific, let $a_1, a_2, \ldots b_1, b_2$,... denote the request times of m datasets immediately prior to T, where the deadline for the data requested at a_1, a_2,... is t, and the deadline for the data requested at b_1, b_2,... is greater than t. Under these conditions, two cases should be examined.

Case 1. None of the datasets requested at b_1, b_2,... is transmitted before t, as shown in Fig. 5(a).

In the worst case, the number of time slots required to schedule m datasets between zero and t is calculated by

$$T_{\max} = \sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\lfloor\frac{t}{T_{ij}}\rfloor w_{ij}. \tag{5}$$

Since overflow occurs at t, the number of time slots should be greater than the number of time slots between zero and t as

$$\because (T_{\max} > Ft)\&(x > \lfloor x \rfloor)$$
$$\therefore \sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\frac{t}{T_{ij}}w_{ij} \geq T_{\max} > Ft$$
$$\therefore \sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\frac{w_{ij}}{T_{ij}} > F.$$

This contradicts the assumption made previously, so sufficiency in *Case 1* is proved.

Case 2. Some of data requested at b_1, b_2,... were transmitted before t, as shown in Fig. 5(b).

Since the overflow occurs at time t, there must be a time t' so that data requested at b_1, b_2,... will not be transmitted between t' and t. In other words, in this interval, only data whose deadline is less than or equal to t can be scheduled. Also, some of data requested at b_1, b_2,... will be scheduled at t'. This means that all data requested before t' whose deadline is less than or equal to t will be transmitted before t'. Therefore, between t' and t, the minimum number of required time slots is equal to

$$T_{\max} = \sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\lfloor\frac{t-t'}{T_{ij}}\rfloor w_{ij}. \tag{6}$$

As the overflow will occur at t, there is

$$\because \sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\lfloor\frac{t-t'}{T_{ij}}\rfloor w_{ij} > F(t-t')$$
$$\therefore \sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\frac{w_{ij}}{T_{ij}} > F.$$

This contradicts the above assumption, so sufficiency in *Case 2* is proved.

In summary, Theorem 2 is valid.

Theorem 3: If the problem $SSDT(F, T, I, J, itf, DL)$ is schedulable, for each result P returned by the RTDS scheme, let ϑ represents the number of required time slots; then, there must be no other algorithm to return the result as P', the number of time slots it needs is $\vartheta' < \vartheta$, i.e., the RTDS is an optimal scheduling scheme.

Proof: Data sent by sensors are sort according to the deadline size, and the smallest deadline is $T_{\min} = \min(T_{ij})$.

1) When $t = T_{\min}$, all sensors send their first dataset at a time of zero before t; some data have a deadline greater than t, and some data have a deadline equal to t. Data with a deadline greater than t will be scheduled at its deadline, so only data with a deadline equal to t will be considered in the following.

a) When $I = 1$, there is only one cell in a network, and it includes a switch and J sensors. There is a wireless interference between any two sensors, so the number of required time slots is calculated by

$$t(1) = \sum_{DL_{ij1}=T_{\min}} w_{0j}. \tag{7}$$

Suppose there is a scheduling method P', whose the number of required time slots is $t' < t(1)$. According to the definition of effective scheduling, this is obviously impossible because if such a method exists, there must be data that are unscheduled before the deadline.

b) Suppose that when $I = m$, the number of time slots $t(m)$ required by the RTDS is the least.

c) When $I = m + 1$, the number of time slots required by the RTDS is expressed as $t(m + 1)$. There are $(m + 1)$ cells in the network, which are denoted by $C_0, C_1, \ldots, C_m$. There are J sensors in each cell, each of which is denoted by s_{ij}. Depending on whether there is an interference between $C_(m + 1)$ and C_m, there are two possible situations.

Situation 1: $C_m \notin itf(C_{m-1})$, then $\forall v(v < m)$ it holds that $C_m \notin itf(C_v)$. Therefore, sensors in C_{m+1} can share time slots and channels with sensors in the previous cells.

When $J = 1$, we have $t(m + 1) = \max\{w_{m0}, t(m)\}$

Case 1: The deadline of s_{m0} is $T_{\min}$. Assuming that there is a scheduling algorithm, the number of required time slots is t', and $t' < t(m+1) = \max\{w_{m0}, t(m)\}$.

If $w_{m0} < t(m)$, then $t' < t(m+1) = t(m)$; $t' < t(m)$ is impossible, so there is no such scheduling algorithm.

If $w_{m0} > t(m)$, then $t' < t(m+1) = C_{m0}$; $t' < C_{m0}$ is impossible, so there is no such scheduling algorithm.

Case 2: The deadline of s_{m0} is greater than $T_{\min}$, so $t(m+1) = t(m)$. Assuming that there is a scheduling algorithm, the number of required time slots is t', and $t' < t(m+1) = t(m)$. This situation does not exist.

Therefore, the number of required time slots when $J = 1$ is the least.

Suppose that when $J = k$, the number of time slots $t^k(m+1)$ required by the RTDS is the least.

When $J = k+1$, then $s_{mk} \in itf(s_{m(k-1)})$, which means that there is an interference between any two sensors, and the number of idle time slots is

$$\zeta = t^k(m+1) - \sum_{\Gamma} w_{m\mu}$$
$$\Gamma = (0 < \mu < k) \wedge (DL_{m\mu 1} = T_{\min}). \tag{8}$$

Suppose there is a time slot $t' < t^{k+1}(m+1)$ required by another scheduling algorithm.

If $\zeta \geq w_{mk}$, then $t^{k+1}(m+1) = t^k(m+1)$, which means $t' < t^k(m+1)$, and this is obviously impossible.

If $\zeta < w_{mk}$, then

$$\begin{aligned} t^{k+1}(m+1) &= t^k(m+1) + w_{mk} - \zeta \\ &= w_{mk} + \sum_{\Gamma} w_{m\mu} \end{aligned} \tag{9}$$

so

$$t' < w_{mk} + \sum_{\Gamma} w_{m\mu}. \tag{10}$$

In this case, there must be a sensor $s_{m\varepsilon} \in itf(s_{mk})$ such that $s[m][\varepsilon][f][t] = s[m][k][f][t] = 1$. Obviously, this situation does not exist, so the number of required time slots is the least when $J = k+1$.

Therefore, in *Situation 1*, when there is only one cell, the number of time slots required for scheduling with the deadline of less than $t = T_{\min}$ using the RTDS scheme is the least.

Situation 2: $C_m \in itf(C_{m-1})$, the number of idle time slots is

$$\xi = t(m) - \sum_{\Psi} w_{ij}$$
$$\begin{aligned} \Psi = &(0 < i < m) \wedge (0 \leq j < J) \wedge \\ &(C_i \in itf(C_m)) \wedge (DL_{ij1} = T_{\min}). \end{aligned} \tag{11}$$

Suppose the number of time slots required by another scheduling algorithm is $t' < t(m+1)$. Then, if

$$\xi \geq \sum_{\Omega} w_{ij}$$
$$\Omega = (0 \leq j < J) \wedge (DL_{mj1} = T_{\min}) \tag{12}$$

it holds that $t' < t(m+1) = t(m)$; obviously, such t' does not exist; and if

$$\xi < \sum_{\Omega} w_{ij} \tag{13}$$

then

$$t(m+1) = t(m) + \sum_{\Omega} w_{ij} - \xi = \sum_{\Psi} w_{ij}. \tag{14}$$

To satisfy $t' < t(m+1)$, there must exists $m,' j,' C_{m'} = itf(C_m)$ such that $s[m'][j'][f][t] = s[m][j][f][t] = 1$, so there must be no such t'.

Therefore, the result obtained by the RTDS scheme in *Situation 2* is the best.

Consequently, at $t = T_{\min}$, the scheduling result obtained by the RTDS scheme is optimal.

2) Suppose that when $t = e$, data that are requested before t and whose deadline is less than or equal to t can be scheduled, and the number of required time slots is the least.

3) When $t = e+1$, all data whose request time is less than and closest to $(e+1)$ are divided into two parts, one part of the data has a deadline of $(e+1)$, and the other part's deadline is greater than $(e+1)$. According to the method of 1), it can be proved that the number of time slots required for scheduling using the RTDS scheme is still optimal; thus, the RTDS scheme is the optimal scheduling algorithm.

Hence, Theorem 3 is proved.

V. Experimental Verification and Result Analysis

A. Experimental Setup

The CC2530 module was used as a sensor node to send packets, and a laptop computer was used as an SDN controller; CC2530 is a wireless radio frequency module that can be applied to 2.4-GHz IEEE 802.15.4 and ZigBee. In addition, four laptops and four CC2530 modules were used as four switches. A computer and a CC2530 formed a switch, and a switch and several CC2530 modules formed a cell. In each cell, three CC2530 modules with different operating frequencies were used to implement a multichannel experimental scenario; three CC2530 modules formed a sensor node. The experimental scenario included four data transmission channels for a node. Data were transferred between computers via an USB; one computer served as a controller, and another computer served as a switch. Each node transmits packets periodically, so the packets were generated periodically through simulation.

The four cells were denoted by C_0, C_1, C_2, and C_3. There was wireless interference between adjacent cells, but not between nonadjacent cells; that is, $itf(C_0) = \{C_1\}$, $itf(C_1) = \{C_0, C_2\}$, $itf(C_2) = \{C_1, C_3\}$, and $itf(C_3) = \{C_2\}$.

In simulation, 1000 time slots are set as a superframe, the length of each time slot is 2 ms. The actual wireless transmission rate of CC2530 was about 20 kb/s, and the time required to transmit one packet was 2 ms. The period of each sensor could be set dynamically. In the experiments, a random function was used to generate the transmission period of a sensor. The transmission period is an integer value that obeys the uniform distribution

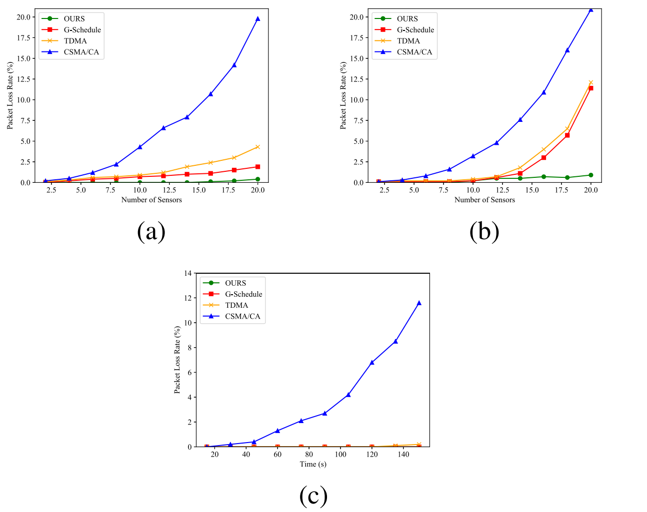


Fig. 6. Packet loss rate results. (a) $I \equiv 4, J = \{1, 2...20\}$ No interference. (b) $I \equiv 4, J = \{1, 2...20\}$ Interference exists. (c) $I \equiv 4, J \equiv 5$.

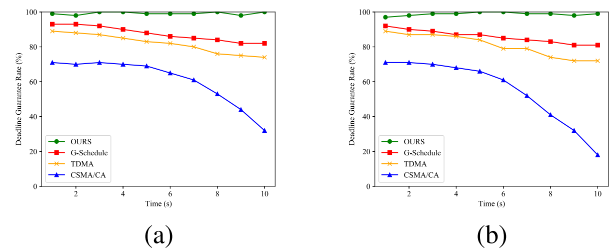


Fig. 7. Deadline guarantee rate results. (a) $I = 2$. (b) $I = 4$.

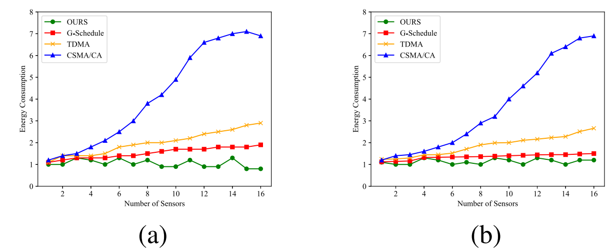


Fig. 8. Energy consumption results. (a) $PR \leq 20\%$. (b) $PR \leq 40\%$.

$U(50 \text{ ms}, 500 \text{ ms})$. Each group of experiments was repeated 30 times. After the abnormal data were removed, the remaining data were averaged. To verify the performance of the RTDS scheme, it was compared with the CSMA/CA, TDMA, and G-schedule schemes. The CSMA/CA and TDMA have been widely used in CPS as classical algorithms, and G-schedule is the latest and advanced algorithm.

B. Performance Analysis

First, the packet loss rate was analyzed. The packet loss rate was calculated by $PLR = \Sigma P_l / \Sigma P_r$, where ΣP_l represented the number of packets discarded due to data collisions in the network, and ΣP_r represented the number of packets sent by all nodes in the network. In the experiment, it was assumed that there was no wireless interference from the external environment. The experimental results for five sensors are shown in Fig. 6(c). The TDMA, G-schedule, and RTDS schemes could ensure that all packets reached the controller successfully. When the number of sensors in each cell was variable, test data were obtained every time a sensor was added. The experimental results of the packet loss rate are shown in Fig. 6(a). In contrast, the packet loss rate of the RTDS scheme did not change significantly with the number of sensor nodes. With the increase in the number of switches, the probability of collision of the CSMA/CA scheme increased, so the packet loss rate also increased. With the increased number of packets, the TDMA and G-schedule schemes were impossible to allocate a reasonable transmission time slot for each of the packets. Under ideal circumstances, the RTDS scheme could achieve a zero packet loss rate. However, in a real environment, there is additional interference with other networks (e.g., Wi-Fi) or physical obstruction in the external environment. As shown in Fig. 6(b), despite the environmental interference, the RTDS scheme performed better than the other three algorithms in terms of the packet loss rate.

Moreover, the deadline guarantee rate refers to the situation where tasks are met with deadlines, which reflects the robustness of a scheduling scheme. The specific calculation formula of the deadline guarantee rate is as follows: $\text{DGR} = \frac{\Sigma W^t}{\Sigma W^c} \times 100\%$, where ΣW^t represents the number of data samples that meet the deadline, and ΣW^c represents the number of requests issued by sensors. As shown in Fig. 7, the deadline guarantee rate of the CSMA/CA gradually decreased with the test time. This was because as the test time passed, the data amount sent by each sensor increased. Therefore, the amount of data scheduled before the deadline would decrease with the test time. Since the TDMA and G-schedule schemes did not consider the deadline of data during the scheduling process, with the test time, their deadline guarantee rates gradually decreased and were always lower than that of the RTDS scheme.

Finally, the energy consumption of the four algorithms is tested. Due to the deference of the interference of the environment and the working principle of the algorithms, part of the data needs to be retransmitted multiple times to complete the transmission, which increases the energy consumption of the data transmission process. In the experiment, the number of requests before the completion of the transmission is used as the unit of energy consumption. The number of times required to send packets before the deadline is evaluated under the condition of ensuring a certain packet loss rate. The maximum number of retransmissions is set to seven. The experiment intercepts the data in the first 10 s, and tests the energy consumption of each algorithm under the condition of ensuring that the packet loss rate is not greater than 20% or 40%. As shown in Fig. 8, the scheme of the RTDS has lower energy consumption than the other three algorithms. The more sensors, the higher the packet loss rate of the TDMA and G-schedule schemes. Therefore, to keep the packet loss rate below a certain level, more retransmissions are required, which leads to a higher energy consumption. With the increase of the number of sensor nodes, the collision probability also increases, and the number of retransmissions of the CSMA/CA scheme also increases. The RTDS scheme has a significant energy consumption advantage due to the lower retransmission times compared to other schemes.

From the aforementioned figures, it can be seen that with the increase in the number of sensors, the packet loss rate and energy consumption did not increase significantly. The deadline guarantee rate did not drop significantly either. Therefore, this scheme keeps satisfactory performance when the number of sensors continues to increase.

VI. Conclusion

In order to solve the real-time deterministic scheduling problem in microgrid, an RTDS scheme for CPS was proposed in this article. The two heuristic algorithms: SRP and GMT were designed to ensure that packets were transmitted before the deadline. Extensive practical experiments were carried out with TI CC2530 module. Experimental results demonstrated that the RTDS scheme was superior to CSMA/CA, TDMA, and G-schedule schemes in terms of packet loss rate, deadline guarantee rate, and energy consumption. The RTDS scheme can effectively improve the determinacy and real-time performance and reduce energy consumption of data transmission in microgrid.

References

[1] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 8, pp. 2072–2085, Aug. 2015.

[2] B. Fan, S. Guo, J. Peng, Q. Yang, W. Liu, and L. Liu, "A consensus-based algorithm for power sharing and voltage regulation in DC microgrids," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3987–3996, Jun. 2020.

[3] C. Zhang, Y. Xu, Z. Y. Dong, and L. F. Yang, "Multitimescale coordinated adaptive robust operation for industrial multienergy microgrids with load allocation," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3051–3063, May 2020.

[4] G. Dong and Z. Chen, "Data-driven energy management in a home microgrid based on Bayesian optimal algorithm," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 869–877, Feb. 2019.

[5] S. Chen *et al.*, "Internet of Things based smart grids supported by intelligent edge computing," *IEEE Access*, vol. 7, pp. 74089–74102, 2019.

[6] D. Ding, Q.-L. Han, Z. Wang, and X. Ge, "A survey on model-based distributed control and filtering for industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2483–2499, May 2019.

[7] Y. Bai, Y. Huang, G. Xie, R. Li, and W. Chang, "ASDYS: Dynamic scheduling using active strategies for multifunctional mixed-criticality cyber–physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5175–5184, Aug. 2021.

[8] L. Mo, P. You, X. Cao, Y. Song, and A. Kritikakou, "Event-driven joint mobile actuators scheduling and control in cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 11, pp. 5877–5891, Nov. 2019.

[9] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic, "High-reliability and low-latency wireless communication for Internet of Things: Challenges, fundamentals, and enabling technologies," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7946–7970, Oct. 2019.

[10] L. Liu *et al.*, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor–critic learning approach," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2342–2353, Feb. 2021.

[11] Q. Chen, X. J. Zhang, L. L. Wei, Y. S. Kwok, and S. Sun, "High reliability, low latency and cost effective network planning for industrial wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2354–2362, Dec. 2019.

[12] V. Huang, Z. Pang, C. Chen, and K. F. Tsang, "New trends in the practical deployment of industrial wireless: From noncritical to critical use cases," *IEEE Ind. Electron. Mag.*, vol. 12, no. 2, pp. 50–58, Jun. 2018.

[13] F. Ding, G. Zhu, M. Alazab, X. Li, and K. Yu, "Deep-learning-empowered digital forensics for edge consumer electronics in 5G hetnets," *IEEE Consum. Electron. Mag.*, to be published, doi: 10.1109/MCE.2020.3047606.

[14] S. Marzal, R. González-Medina, R. Salas-Puente, G. Garcerá, and E. Figueres, "An embedded internet of energy communication platform for the future smart microgrids management," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7241–7252, Aug. 2019.

[15] D. G. Zhang, S. Zhou, and Y. M. Tang, "A low duty cycle efficient MAC protocol based on self-adaption and predictive strategy," *Mobile Netw. Appl.*, vol. 23, no. 4, pp. 828–839, 2018.

[16] G. Luo, J. Li, L. Zhang, Q. Yuan, Z. Liu, and F. Yang, "sdnMAC: A software-defined network inspired MAC protocol for cooperative safety in VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 2011–2024, Jun. 2018.

[17] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: A priority-enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 824–835, Feb. 2014.

[18] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C.-W. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2021.3079574.

[19] H. Baek and J. Lim, "Time mirroring based CSMA/CA for improving performance of UAV-relay network system," *IEEE Syst. J.*, vol. 13, no. 4, pp. 4478–4481, Dec. 2019.

[20] Z. Wang, F. Yu, J. Tian, and Z. Zhang, "A fairness adaptive TDMA scheduling algorithm for wireless sensor networks with unreliable links," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 1535–1552, 2015.

[21] R. D. Gomes, C. Benavente-Peces, I. E. Fonseca, and M. S. Alencar, "Adaptive and beacon-based multi-channel protocol for industrial wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 132, pp. 22–39, 2019.

[22] T. Xie, H. Zhao, J. Xiong, and N. I. Sarkar, "A multi-channel MAC protocol with retrodirective array antennas in flying ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1606–1617, Feb. 2021.

[23] A. Díaz-Ramírez, P. Mejía-Alvarez, and L. E. Leyva-Del-Foyo, "Comprehensive comparison of schedulability tests for uniprocessor rate-monotonic scheduling," *J. Appl. Res. Technol.*, vol. 11, no. 3, pp. 408–436, 2013.

[24] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "Security-aware task scheduling with deadline constraints on heterogeneous hybrid clouds," *J. Parallel Distrib. Comput.*, vol. 153, no. 1, pp. 15–28, 2021.

[25] M. Doudou, D. Djenouri, N. Badache, and A. Bouabdallah, "Synchronous contention-based MAC protocols for delay-sensitive wireless sensor networks: A review and taxonomy," *J. Netw. Comput. Appl.*, vol. 38, pp. 172–184, 2014.

[26] J. Ni, "Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 825–836, Jun. 2012.

[27] C. Ghosh, H. A. Safavi-Naeini, S. Roy, K. Doppler, and J. Stahl, "QP-CSMA-CA: A modified CSMA-CA-based cognitive channel access mechanism with testbed implementation," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw.*, 2012, pp. 501–509.

[28] P. K. Sahoo and J. P. Sheu, "Design and analysis of collision free MAC for wireless sensor networks with or without data retransmission," *J. Netw. Comput. Appl.*, vol. 80, pp. 10–21, 2016.

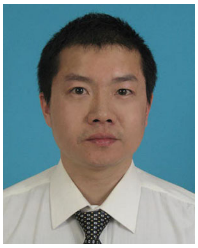
[29] A. Bhatia and R. C. Hansdah, "RD-TDMA: A randomized distributed TDMA scheduling for correlated contention in WSNs," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, 2014, pp. 378–384.

[30] C. Lin, X. Cai, Y. Su, P. Ni, and H. Shi, "A dynamic slot assignment algorithm of TDMA for the distribution class protocol using node neighborhood information," in *Proc. 11th IEEE Int. Conf. Anti-Counterfeiting, Secur., Identification*, 2017, pp. 138–141.

[31] A. Tan, Q. Wang, G. Nan, Q. Deng, and X. S. Hu, "Inter-cell channel time-slot scheduling for multichannel multiradio cellular fieldbuses," in *Proc. IEEE Real-Time Syst. Symp.*, 2015, pp. 227–238.

[32] C. T. Chiang, H. C. Chen, W. H. Liao, and K. P. Shih, "A decentralized minislot scheduling protocol (DMSP) in TDMA-based wireless mesh networks," *J. Netw. Comput. Appl.*, vol. 37, pp. 206–215, 2014.

[33] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1983.



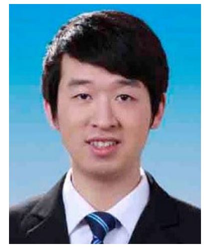
Yuhuai Peng received the Ph.D. degree in communication and information systems from Northeastern University, Shenyang, China, in 2013.

He is currently an Associate Professor with Northeastern University. His research interests include Internet of Things, industrial communication networks, health monitoring, etc.

Alireza Jolfaei (Senior Member, IEEE) received the Ph.D. degree in applied cryptography from Griffith University, Gold Coast, Australia, in 2015.

He is currently an Associate Professor in Networking and Cyber Security with the College of Science and Engineering, Flinders University, Adelaide, Australia. He has previously been a Faculty Member with Macquarie University and Federation University, Australia, and Temple University, USA. He has participated in several projects involving different aspects of cyber security. On these topics he has authored and co-authored more than 100 papers appeared in journals, conference proceedings, and books. His main research interests include cyber and cyber-physical systems security.

Dr. Jolfaei was the recipient of multiple awards for Academic Excellence, University Contribution, and Inclusion and Diversity Support, including the prestigious IEEE Australian Council Award for his research paper published in the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. He served as the Chairman of the Computational Intelligence Society in the IEEE Victoria Section and also as the Chairman of Professional and Career Activities for the IEEE Queensland Section. He has served as the Associate Editor of IEEE Journals and Transactions, including IEEE INTERNET OF THINGS JOURNAL, IEEE SENSORS JOURNAL, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. He has served as a program Co-Chair, a Track Chair, a Session Chair, and a Technical Program Committee Member, for major conferences in cyber security, including IEEE International Conference on Computer Communications and Networks and IEEE International Conference on Trust, Security and Privacy in Computing and Communications. He was the General Chair of the 6th IEEE International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications, Fiji. He is a Distinguished Speaker of the Association for Computing Machinery on the topic of Cyber Security.



Keping Yu (Member, IEEE) received the M.E. and Ph.D. degrees in global informaiton and telecommunication studies from Waseda University, Tokyo, Japan, in 2012 and 2016, respectively.

From 2015 to 2019, he was a Research Associate, and from 2019 to 2020, he was a Junior Researcher with the Global Information and Telecommunication Institute, Waseda University, where he is currently a Researcher (Assistant Professor). His research interests include smart grids, information-centric networking, the Internet of Things, artificial intelligence, blockchain, and information security.

Dr. Yu has hosted and participated in more than ten projects, is involved in many standardization activities organized by ITU-T and ICNRG of IRTF, and has contributed to ITU-T Standards Y.3071 and Supplement 35. He is an Associate Editor for IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY, *Journal of Intelligent Manufacturing*, and *Journal of Circuits, Systems and Computers*. He has been a Lead Guest Editor for *Sensors*, *Peer-to-Peer Networking and Applications*, *Energies*, *Journal of Internet Technology*, *Journal of Database Managemen*t, *Cluster Computing*, *Journal of Electronic Imaging*, *Control Engineering Practice*, and *Sustainable Energy Technologies and Assessments* and a Guest Editor for *IEICE Transactions on Information and Systems*, *Computer Communications*, *IET Intelligent Transport Systems*, *Wireless Communications and Mobile Computing*, *Soft Computing*, and *IET Systems Biology*.