symmetry

MDPI

Article

Research on Real-Time Communication Algorithm of Substation Based on Time-Sensitive Network

Beilei Wang [1], Yang Liu [2,3,4,*], Chenyang Guo [5], Yan Song [5], Jidong Wang [6], Jinchao Xiao [2] and Xiaoguang Chen [7]

1 Software College, Northeastern University, Shenyang 110000, China; wangbeilei@mail.neu.edu.cn
2 Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110000, China; xiaojinchao@sia.cn
3 Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China
4 Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
5 College of Physics, Liaoning University, Shenyang 110036, China; gcy@zdh.ac.cn (C.G.); song.yan@lnu.edu.cn (Y.S.)
6 Shenyang Sumboy Intelligent Imaging Technology Co., Ltd., Shenyang 110179, China; wjd002@sumboy.cn
7 Huawei Technologies Co., Ltd., Shenzhen 518129, China; ace.chen@huawei.com
* Correspondence: liuy@sia.cn

Abstract: A time-sensitive network (TSN) extends the conventional Ethernet to support time-sensitive data flow. Thus, it enables simultaneous transmission of high reliability (HR) flow, medium reliability (MR) flow, and low reliability (LR) flow on the same network, thereby improving the reliability of data transmission. A TSN is a symmetric network that connects sensors and other facilities. As a backbone network, it can efficiently connect the underlying sensors and other levels of facilities, as well as ensure the quality of service of the network. For modern supervisory control and data acquisition (SCADA) systems, several types of sensors are widely used. The acquisition cycle of sensors for different purposes varies significantly from milliseconds to seconds. Moreover, these data also have different real-time requirements. Based on satisfiability modulo theories (SMT), this study proposes a TSN routing and scheduling method by adding related scheduling constraints. Compared with other methods, the proposed method can realize the routing and scheduling of hybrid flow in a hyper period and consider MR flow and LR flow, which improves the feasibility and certainty of data flow interaction between substations.

Keywords: industrial internet; substation; routing and scheduling; time-sensitive network (TSN); real-time communication

check for updates

Citation: Wang, B.; Liu, Y.; Guo, C.; Song, Y.; Wang, J.; Xiao, J.; Chen, X. Research on Real-Time Communication Algorithm of Substation Based on Time-Sensitive Network. *Symmetry* **2022**, *14*, 1170. https://doi.org/10.3390/sym14061170

Academic Editor: Aviv Gibali

Received: 20 April 2022
Accepted: 1 June 2022
Published: 7 June 2022

1. Introduction

With the development of big data and edge computing, smart grids [1,2] have been promoted. The core of the smart grid provides a cyber-physical system (CPS) [3], which transmits the information collected by the sensing device in the power system to the controller. An increasing number of intelligent algorithms [4–7] have been introduced into this field. It applies the result of instruction analysis to the power field equipment to ensure the stability of the power field equipment. To realize the reliability of the CPS, we should design a real-time communication network to provide a guarantee for it. A time-sensitive network (TSN) is based on the conventional Ethernet. It combines clock synchronization and resource reservation to simultaneously transmit high reliability (HR) flow, medium reliability (MR) flow, and low reliability (LR) in the network; thus, it provides deterministic data interaction and delay for embedded instrument devices in the network [8]. However, a wide range of sensor networks may need to be connected to each other. TSN has good compatibility and scalability, and it is decentralized and symmetric because of the uplink and downlink as well as architectural designs. It can be used as a backhaul network or backbone network of multiple sensor networks or directly connect sensors, controllers, and actuators. For the data acquired from various sensors, different real-time requirements should be individually met. The TSN network architecture has significant natural advantages in meeting several real-time requirements of sensors, and it has wide adaptability and easy deployment. Artificial intelligence algorithms, such as Convolutional Neural Networks (CNN) and genetic algorithms, are often involved in network scheduling [9–13].

In the substation communication process based on IEC61850 [6,14], the data flow is divided into three types: express flow, moderato flow, and slow flow. The response delay corresponding to the three types of messages is 10–500 ms. To further improve the certainty of communication between stations, we introduced a TSN to reduce the aforementioned delay to a microsecond level. The three data flows correspond to HR flow, MR flow, and LR flow in our study.

To ensure the deterministic delay and jitter of HR flow, we should consider MR flow and LR flow in the routing and scheduling process to improve scheduling performance considerably. The joint routing and scheduling problem calculates the offline gate control list (GCL) of all flows [15] using an algorithm before deploying the network. When the network is running, the switch schedules the data flow through the scheduling list to realize the deterministic data flow transmission. Additionally, the GCL can be online, which could cause a certain degree of complex calculations and destroy the existing scheduling strategy, thereby generating a new GCL to satisfy the needs of new scheduling tasks.

The structure of the paper is as follows: Section 2 introduces related works on TSN routing and scheduling, which was the main topic of this study. Section 3 briefly describes the relevant standards of TSN and the substation communication configuration scheme based on TSN. Section 4 presents the proposed system model designs and constraints of the flow. Section 5 elaborates on the routing and scheduling processes of the algorithm, including preprocessing, scheduling calculation, and online monitoring. Section 6 comprehensively considers the influence of the flow, quantity, and scale of network topology on the algorithms' performance and compares it with other algorithms [16–18]. Section 7 summarizes the study.

2. Related Work

TSN routing and scheduling research can be divided into two types: offline and online scheduling [19]. Most researchers design offline schedules because the process is static, and all data flow, topology, scheduling constraints, and other data are known. In contrast, the online scheduling process is dynamic and variable, wherein dynamic adjustments are made according to changes in the data flow, network topology, and scheduling tasks. However, this process is often complicated.

Regardless of offline or online scheduling, the core task of the TSN scheduling algorithm is to calculate the GCL available for data scheduling. The solution method of the scheduling list is solved using approaches such as satisfiability modulo theories (SMT) [20], ILP [21], network calculus [18], or swarm intelligence optimization [17,22–24]. The authors of [17] proposes an improved ant colony algorithm, which aims to optimize the queuing delay in the data flow scheduling process and evaluates the average transmission delay and jitter of data flow scheduling by adding port and data flow constraints; the experimental results have achieved good performance indicators. The authors of [20] proposes an offline scheduling algorithm based on SMT. By adding data flow constraints, the maximum deadline of data flow transmission is used as a measurement index, and the schedulability of BE (best effort) flow is taken into account. In addition, the algorithm is also performing TSN simulation based on the INET framework of OMNET++, and the simulation results meet the performance indicators related to scheduling. The authors of [22] proposes an improved genetic algorithm, which considers both the routing and scheduling of data flows, effectively improves the scheduling performance of TT flows, and rationally optimizes network resource allocation. The authors of [24] proposes a hybrid genetic algorithm, which fully considers scheduling constraints, network topology scale, and data flow scale, effectively improving timeslot utilization and ensuring the real-time nature of data flow transmission. The scheduling constraint of the data flow in the solution process may be a single HR flow or the joint scheduling of HR and MR flow [16]. Finally, hardware or software verifies the algorithm according to a specific scenario, wherein the software verification uses a simulation program [25] or OMNET++ simulation software (omnetpp-5.5.1) [26] that simulates the hardware environment.

The main contributions of this study are as follows:

(1) This study designed an offline and online TSN hybrid data flow routing and scheduling algorithm. Compared with previous studies [19], the offline and online scheduling algorithms designed in this study consider changes in the network topology and data flow. Previous studies [19] considered the change in network topology in the online scheduling mode and combined it with the offline mode to reconstruct a new network topology. The algorithm can prevent the failure of scheduling calculations caused by new equipment access;
(2) This study presents an integration method of TSN and IEC61850. Simultaneously, three types of messages in the substation: express flow, moderato flow, and slow flow, are mapped to HR flow, MR flow, and LR flow in the TSN network. The introduction of time-sensitive technology can effectively reduce the response time during flow transmission, from milliseconds to microseconds. In the context of energy interconnection, the introduction of a reliable TSN communication technology can improve the certainty of communication between substations;
(3) Finally, this study discusses the forwarding and dispatching problems of three different data flows, which is more in line with the real-time scenario of hybrid data flow transmission and is of great significance to research. Compared with previous studies [16,17,19], this study not only considers HR flow and MR flow but also considers the schedulability of LR flow. This is necessary for flow transmission between the substations.

3. Configuration of TSN Substation Communication

3.1. TSN Protocol

TSN contains an IEEE802.1 protocol cluster, which can provide the minimum delay for the data flow transmitted in a nondeterministic Ethernet and improve the certainty of the network. Sections 3.1.1–3.1.5 introduce the relevant standards of TSN technology, which is the basis for configuring TSNs. A specific TSN configuration scheme is introduced in Section 3.2.

3.1.1. IEEE802.1 AS

IEEE802.1 AS can realize high-precision clock synchronization for all time-sensing systems in VLAN. Based on IEEE1588, this standard provides clock synchronization for the TSN domains requiring high clock accuracy. As shown in Figure 1, each TSN domain contains end nodes and switch nodes, among which the best master clock can be selected through switching clock synchronization packets to achieve global clock synchronization.

3.1.2. IEEE802.1 Qat

The IEEE802.1 Qat stream reservation protocol (SRP) defines a set of resource reservation strategies that effectively solve the problem of quality of service (QoS) drop caused by resource competition for different data flow. The SRP consists of two parts: multiple registration protocol (MRP) and stream reservation. In the process of the flow, the destination node needs to initialize the MRP to establish multiple spanning tree forwarding paths between source and sink. The SRP is implemented on the reserved path in the receiving and forwarding of the data flow. Simultaneously, the subtrees of the spanning tree detect changes in nodes in the network topology and make dynamic adjustments to adapt to changes in the topology.

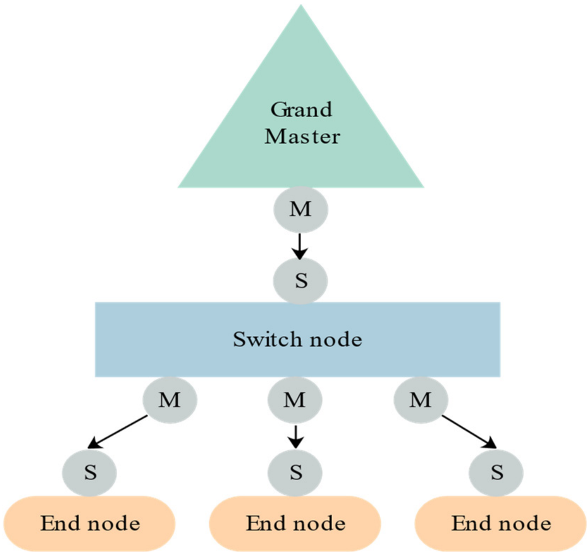

Figure 1. IEEE802.1 AS protocol.

3.1.3. IEEE802.1 Qbv

IEEE802.1 Qbv plan traffic scheduling enhancement protocol defines a time-aware shaper (TAS) mechanism. As illustrated in the following Figure 2, by defining different priority queues and adding queue gates controlled by a GCL in different transmission queues, it is realized that each provides a deterministic timeslot scheduling strategy for the flow in a transmission period.

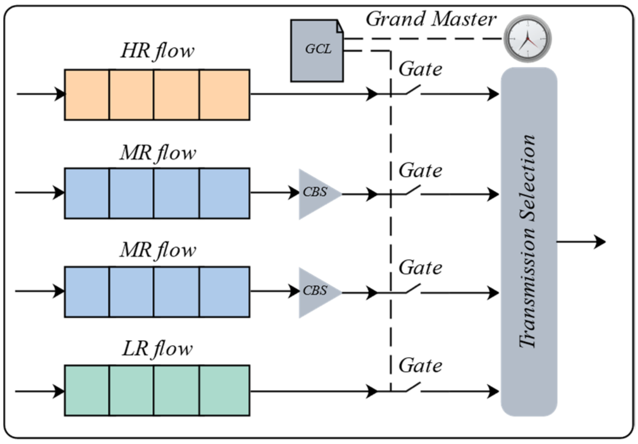

Figure 2. IEEE802.1 Qbv protocol time-aware shaper (TAS).

3.1.4. IEEE802.1 Qbu

The IEEE802.1 Qbu frame preemption protocol optimizes the time-sensitive data stream queuing problem caused by uninterrupted data stream transmission. When the data flow is greater than 124 B, the express frame can interrupt the slow frame [27].

3.1.5. IEEE802.1 Qcc

The IEEE 802.1 Qcc stream reservation enhancement and performance improvement protocol provides three configurations for the TSN. It includes a fully distributed model of independent deployment on the user and network sides, centralized deployment on the network side, distributed user/centralized network configuration model of distributed deployment on the user side, and centralized network configuration model of centralized deployment on the network and user sides.

3.2. Configuration of TSN Substation Communication

Figure 3 shows the steps of configuring the TSN, which are as followsFirst: during substation communication, the constructed TSN network topology may be a complex dynamic topology system, which may contain multiple subsystems. To achieve deterministic flow, it is necessary to ensure that the network clock is synchronized [28]. An excessive clock offset may affect the QoS of data flow. Second: provide the largest possible bandwidth reservation for data flow to ensure that the transmission of time-sensitive data flow is reliable [29]. Third: to reasonably allocate the timeslot in each scheduling period, the TAS of the substation node and switching node should be designed reasonably [30], and the GCL should be generated for scheduling. Data flow with different QoS requirements is distributed in the corresponding port. fourth: combined with scheduling requirements, time-sensitive data flow can preempt the transmission of low-priority data flow [31]. Finally: consider the actual size of the network and whether to store configuration information in each node in the topology or set up an independent configuration node. Generally, a complex network structure requires separate network configuration nodes to improve scheduling performance.

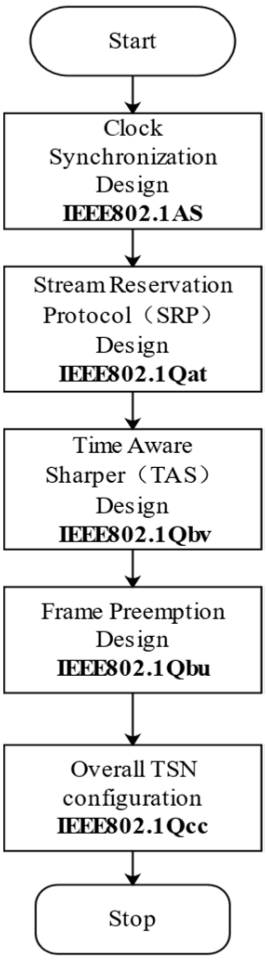

Figure 3. TSN configuration process chart.

In this study, we aimed to design a reasonable time scheduling GCL for substation nodes and switching nodes based on the provided data flow. The timeslot utilization rate of each period was improved considerably while ensuring schedulability. Considering that our entire simulation process was a small-scale TSN, the configuration was performed on each node in the network. Simultaneously, we assumed that the clock of the simulated network topology was synchronized.

4. System Model

In Section 4.1, we establish a system, scheduling task, and traffic models. In Section 4.2, we add transmission constraints to the traffic model. The model and constraints established in Section 4 are applied to the routing and scheduling algorithm described in Section 5. For specific design methods, please refer to Sections 5.1–5.6.

4.1. System Model

For the convenience of representation, we used the tuple, $G(V, E)$, to represent the network topology [15]. Herein $V = E_n \cup S_n$ is a set of nodes in the topology graph, including substation node E_n and switching node S_n; E denotes a set of full-duplex links. In this study, the transmission rate of the full-duplex physical link was 1 Gbps. $G(T, F)$ represents the application [32], T represents the scheduling task running on the node, and F represents the virtual link connection established between tasks.

A real-time scheduling task is initiated by the substation node and arrives at the next substation node within the deadline. In practice, we try to allow more data flow to be scheduled within the deadline to improve the overall scheduling performance.

In the process of flow routing and scheduling, we considered three types of flow, including HR flow, MR flow, and LR flow. Under normal circumstances, HR flow has the highest QoS priority, and the unschedulability of data flow occurs primarily in the MR and LR flow. We used set S to represent the three types of traffic: $S = \left\{S^{TT}, S^{AVB}, S^{BE}\right\}$, where S^{TT} represents the HR flow set, S^{AVB} represents the MR flow set, and S^{BE} represents the BE flow set. The dictionary $S_i(S_i \in S^{TT}$ or S^{AVB} or $S^{BE})$ stores the corresponding attributes of the data flow, $S_i = \left\{f_{size}, f_{offset}, f_{period}, f_{deadline}\right\}$, where f_{size} represents the frame length, f_{offset} represents the offset of the frame, f_{period} represents the sending period, and $f_{deadline}$ represents the sending deadline.

The integration of network topology and application programs determined a reasonable routing and scheduling way for a flow. Routing determined the path of data flow routing between nodes. Scheduling determined whether the data flow could be scheduled based on the routing path and combined with parameters, such as the size of the data flow, transmission period, and deadline. These parameters were defined when designing the model of the flow attribute S_i. When the flow could not be scheduled, timeslot optimization was used to determine whether the data flow could arrive within the deadline.

4.2. Related Constraints

Here, we add transmission constraints based on the three types of flows. The application ensures that each node should reasonably allocate the timeslot of the task within a hyper period, define the relevant constraints and forwarding requirements, determine the switching node and the substation node, and send the data flow to the destination simultaneously.

Related constraints are as follows:

Constraint1: As shown in Figure 4 and Equation (1), because the transmission of the flow is periodic, transmission should be completed within this period. In addition, $\boldsymbol{t}_0$ to $\boldsymbol{t}_1$ represent a transmission period; the timeslot occupied by the data flow should be within this period to ensure that it does not overflow. Herein, $\boldsymbol{t}_{period}$ represents one transmission period and $\boldsymbol{t}_{frame}$ represents the specific timeslot allocated by the data flow in the period

$$\boldsymbol{t}_{period} \leq \boldsymbol{t}_{frame} \leq \boldsymbol{t}_{period+1} \tag{1}$$

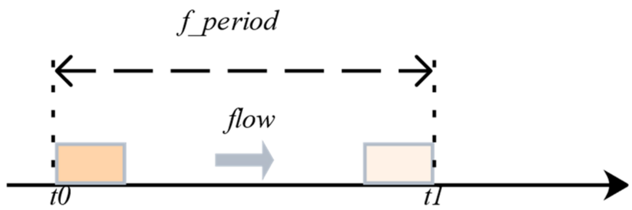

Figure 4. Flow should be constrained to be transmitted within the period.

Constraint2: As shown in Figure 5 and Equation (2), the transmission offset of the data flow should be between the minimum offset and the maximum period. This is considering that the periodically transmitted data flow will deviate from the fixed transmission timeslot owing to jitter. We must ensure that when an offset occurs, the data flow does not overflow the cycle owing to the offset. This may cause packet loss or unschedulability, which is very serious for the system. In addition, t_1 to t_2 represent the offset of the data flow and t_2 to t_3 represent the timeslot of the data flow. After determining the offset of the data flow, we should ensure that the data flow will not overflow its period. Here, t_{offset} represent data flow offset.

$$t_{offset} \leq t_{frame} \leq t_{period} \quad (2)$$

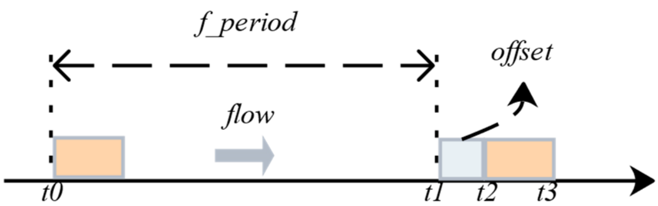

Figure 5. Under the effect of the offset, the transmission of the flow.

Constraint3: As shown in Figure 6, the transmission interval of the frame must be equal to the period. In addition, the data flow with t_0 as the transmission starting point should ensure that the transmission starting point of the data flow is t_1. Additionally, the transmission interval must be one period. Essentially, the data flow at t_1 should not be transmitted prior to t_1. The periodicity of the data flow transmission must be guaranteed.

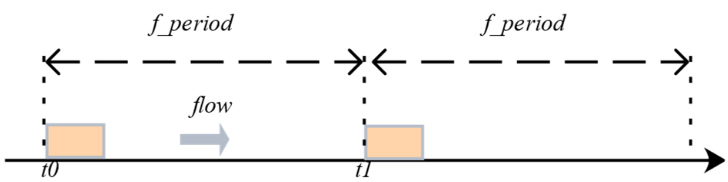

Figure 6. Transmission period constraints.

Constraint4: The link is exclusive to the frame during transmission, and only one flow is allowed on the link. Essentially, other data flow cannot be transmitted in a certain occupied timeslot;

Constraint5: As shown in Figure 7 and Equation (3), when the data flow passes through the switch, the transmission of the next data flow should be greater than the delay of the switch in the previous data flow. In addition, the delay caused by the data flow passing through the switch is from t_7 to t_8; therefore, we should wait for a processing delay before sending the next data flow. Essentially, the switch node in the figure will have a delay from receiving the data flow to transmitting the data flow. This issue must be considered during the simulation.

$$t_{frame+1} \geq t_{frame} + t_{processing} \quad (3)$$

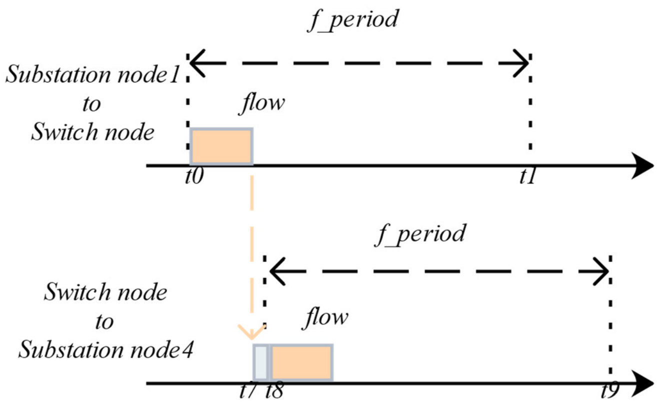

Figure 7. When there is an offset, the transmission period is constrained.

Constraint6: As shown in Figure 8, the delay in the entire transmission process of the flow transmitted between nodes should be less than the entire scheduling period of the application. Essentially, the hyper-period from t_0 to t_1 is composed of the period of data flow1 formed by t_0 to t_1 and the period of the flow2 formed by t_5 to t_6. Both flow1 and flow2 should not exceed the overall scheduling period of the data flow. Thus, all the data flow can be scheduled;

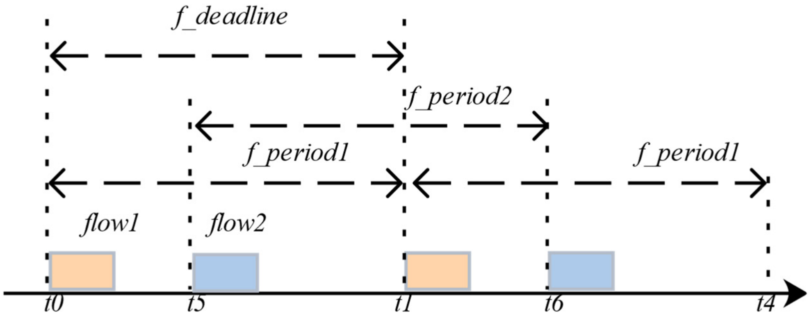

Figure 8. Hyper period constraints for hybrid data flow.

Constraint7: As shown in Equation (4), the $t_{scheduling}$ generated during the entire transmission process of the flow transmitted between nodes should be less than $t_{deadline}$; herein, $t_{scheduling}$ represents the total scheduling time (corresponding to t_1 in the figure), and $t_{deadline}$ represents the deadline (corresponding to t_1 in the figure):

$$t_{scheduling} \leq t_{deadline} \tag{4}$$

As shown in Equation (5), if there is no data flow queued in the transmission queue, the maximum offset of the data flow transmission is less than the t_{max1}.

$$t_{max\_offset} \leq t_{max1} = t_{period} - t_{frame} \tag{5}$$

As shown in Equation (6), if there are multiple data flows in the queue, the scheduling time of the flow is the task scheduling period, and the scheduling time of the next flow should be less than t_{max2}; herein, t_{max_offset} represents the maximum offset, t_{period} represents the data flow period, and $t_{queuing}$ represents the queuing delay.

$$t_{frame} \leq t_{frame+1} \leq t_{max2} = t_{period} + t_{queuing} + t_{max\_offset} \tag{6}$$

5. Routing and Scheduling Framework

The purpose of the algorithm is to generate a GCL that satisfies the constraints in [33,34]. Here, we propose a novel routing and scheduling approach architecture based on SMT [35] to generate an effective scheduling list. The framework included six parts:

mapping, stream processing, timeslot occupancy, task scheduling, communication scheduling, and dynamic monitoring. As shown in Figure 9, we used a process chart to describe offline and online routing and scheduling algorithms (Sections 5.1–5.6).

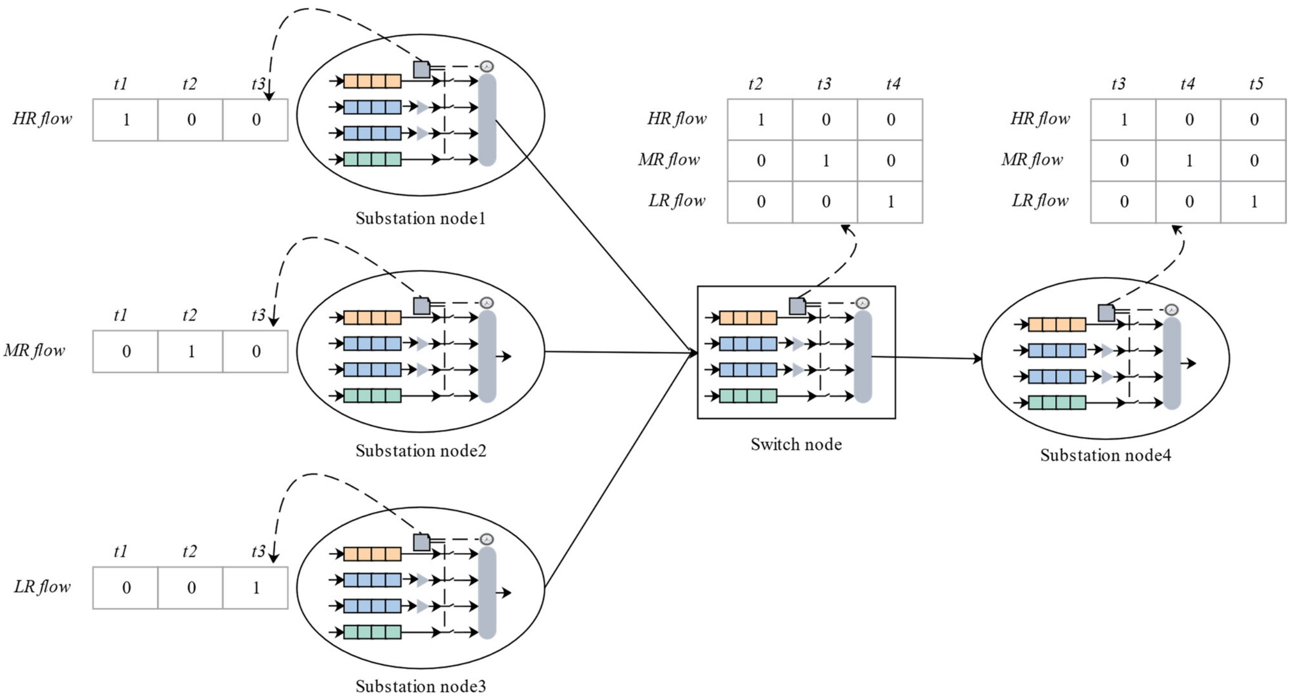

Figure 9. Scheduling instance.

As an example, Figure 10 briefly describes the computing tasks of the routing and scheduling frameworks. We assumed that the preprocessing work had been completed, including mapping, stream processing, and timeslot occupancy. The scheduling GCL was obtained through task and communication scheduling. The TAS running on each terminal node (substation node1–4) contained eight queues. However, we described only three queues in the figure to simplify the process. The substation nodes sent three data flows—HR, MR, and LR. Each substation node periodically sent data flow according to an offline dispatch GCL. One switching node received the mixed transmission data flow, combined the hyper period and scheduling constraints to schedule the task, scheduled the hybrid transmission data flow according to the calculated GCL, and transmitted it to substation node4.

5.1. Mapping

To enable the substation nodes in the network topology to perform scheduled calculation tasks, it was necessary to map the application program to the network topology. In the initial input file, the network topology and applications were separated. The network topology was written in XML, including all nodes (substation and switching nodes) and the connection status of the nodes; however, the nodes were not divided. This implies that we had to obtain the corresponding substation nodes and switch nodes through corresponding processing. By traversing the number of adjacent node connections of the node, we could obtain the substation and switching nodes, establish a link to them, and obtain the network topology with the node identification. Subsequently, the application contained scheduled calculation tasks that had to be executed on the substation node, and we could obtain the attributes of different types of data flow. The input application program was traversed, virtual links were established among the tasks of different substation nodes, and the tasks were initialized as well as stored in the corresponding substation nodes.

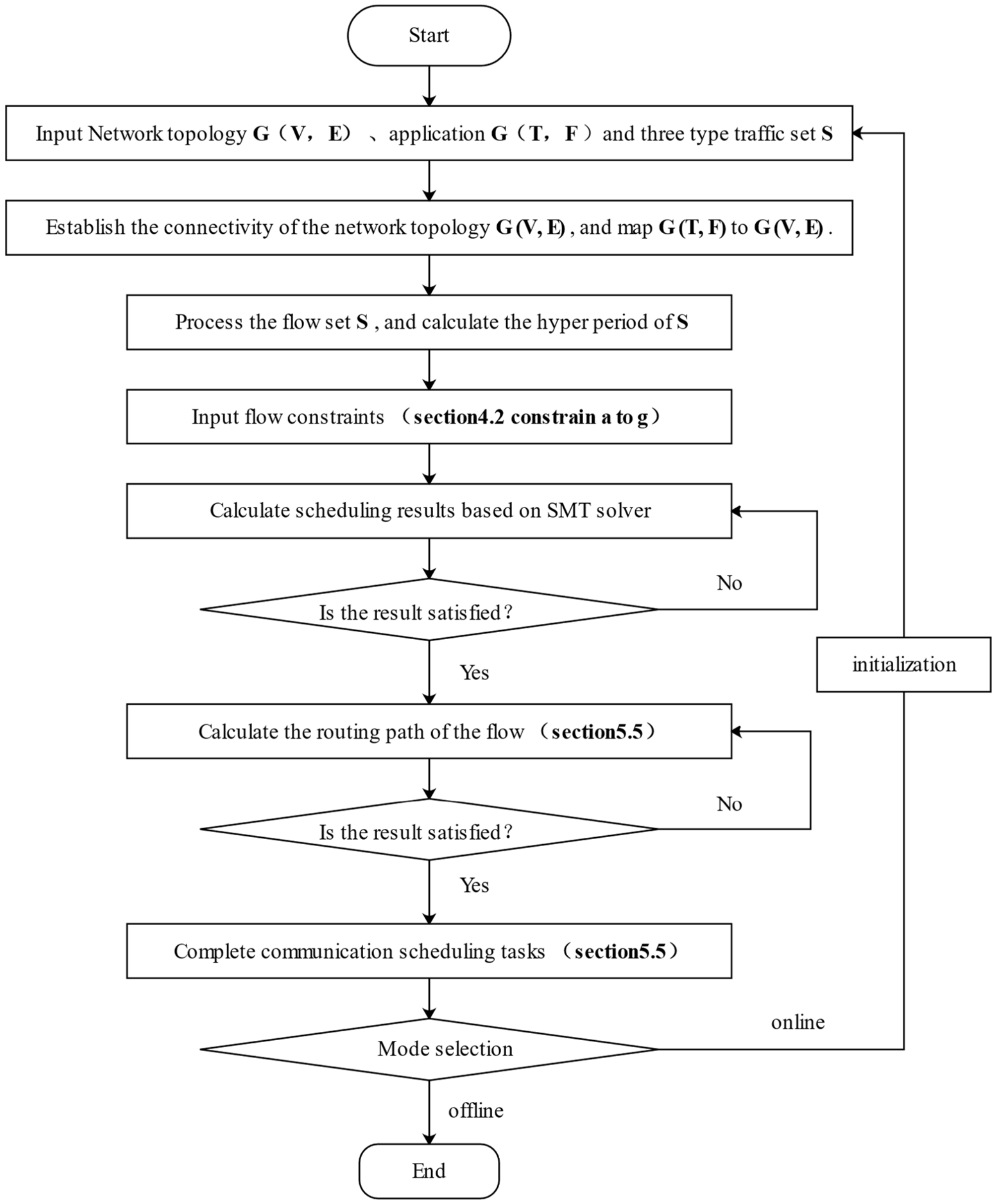

Figure 10. Overall process chart of the routing and scheduling algorithm.

5.2. Stream Processing

After mapping, we obtained a scheduling task that contained the attributes of the data flow. Thereafter, we processed the data flow of the scheduled computing task executed on the substation node. First, it was crucial to obtain the hyper period of the flow set. The hyper period was used to evaluate the feasibility index for a group of data flow to satisfy the schedules. As the name implies, if there are several flow tasks, L.C.M. of these tasks is named by hyper period. When a group of periodic flow can be scheduled in a hyper period, they repeat every other hyper period and run uninterrupted.

5.3. Timeslot Occupancy

We simulated the scheduling of eight periodic data flows to design an offline GCL. The timeslot occupancy was obtained by calculating the worst time occupied by the arrival time in the worst case of different data flow. Note that to increase the occupancy rate of the hyper period timeslot, we must compress the deadline of each data flow considerably to achieve compression of the overall data flow scheduling period. Finally, we used the SMT solver to add constraints to obtain the maximum timeslot occupancy value.

HR flow is the most stable, with an almost constant scheduling period and execution time. Therefore, the compression deadline discussed above refers to LR and MR flows because they are the key factors for evaluating scheduling performance.

5.4. Task Scheduling

Task scheduling determines the GCL on the substation node and reasonably arranges the timeslots occupied by different tasks. The hyper period of all data flow to be scheduled was obtained in the stream processing part. During task scheduling calculation, first, the data flow was divided. For non-HR flow, the offset and cutoff times were calculated again. This result was obtained in the timeslot occupancy part. Second, we traversed the task to calculate whether the data flow satisfied the bandwidth occupation. For a data flow that exceeded bandwidth occupation, reasonable timeslots were reallocated. Finally, we obtained the task scheduling strategy that satisfied the terminal node and timeslot allocation of each data flow.

5.5. Communication Scheduling

A communication schedule establishes communication and data stream transmissions between different nodes. First, we determined whether the offset and deadline between nodes and the period of the data flow satisfied the requirements. Second, the fusion of the mapping section to the topology map, application program, and substation node scheduling GCL was obtained from the task scheduling section as input. Third, the stream constraint conditions between nodes were established to determine whether the communication schedule was satisfied. Finally, the final scheduling link sequence was obtained through a communication scheduling calculation, and the communication schedule was completed.

5.6. Dynamic Monitoring

Dynamic monitoring provides flexibility and diversity in scheduling. We could monitor the connected switching nodes and substation nodes at any time and change the scale of the initial topology to achieve online scheduling.

In the topology initialization process, we input the newly accessed node information into the configuration file to update the network topology. The scheduler re-executed procedures 5.1 to 5.5 to calculate the schedule list that satisfied the constraints and used it as the new schedule criterion.

6. Experimental Result

To verify the superiority of the scheduling algorithm, Python was used to write the offline and online scheduling algorithms. The algorithm included network topology, task scheduling, and related constraints, and it was tested on a PC with a hardware specification of Intel Core i7-8750 H and 16 GB RAM. Moreover, Python supports SMT, ILP, and network calculus in the programming process. Although the abovementioned methods are effective for TSN data flow routing and scheduling calculations, SMT provides better support for Python, and the library functions are clear and simple, which is convenient for us to develop. Therefore, in the solution process, we used an SMT-based Z3 solver to complete the TSN data flow for routing and scheduling calculation tasks. Compared with ILP and network calculus, the execution process of the Z3 solver based on SMT is simple and clear and has a relatively short algorithm execution time.

We used the network topology shown in Figure 11, which includes four substation nodes (Substation node1, Substation node2, Substation node3, and Substation node4) and a switch node. As summarized in Table 1, the scheduling task includes eight data flows, which are four HR flows (S1–S4), three MR flows (S5–S7), and one LR flow (S8). The MR flow was all Class A, and the default idle slope was 75%. Herein, R represents the sending path of the flow, T represents the period of the flows, D represents the deadline of the flows, and P represents the length of the flows. The transmission times of HR flow, MR flow, and LR flow were 6, 12, and 12 µs, respectively.

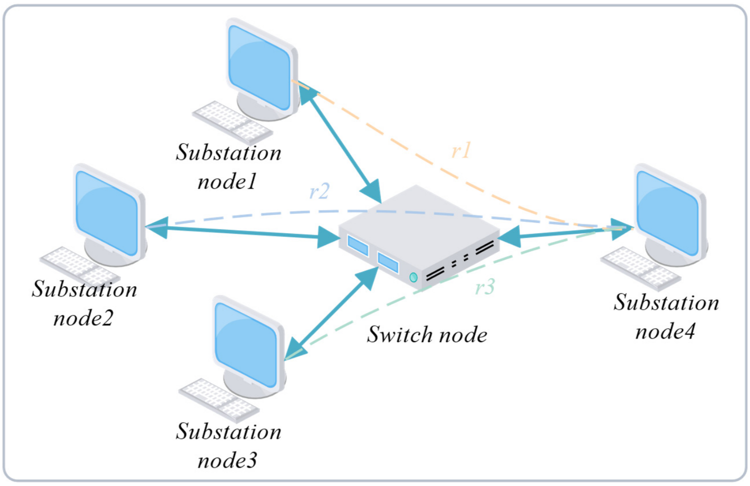

Figure 11. Network topology model.

Table 1. Application flow properties.

Flow	Type	R	T	D	P
S1	HR	r1	200 µs	200 µs	750 B
S2	HR	r1	200 µs	200 µs	750 B
S3	HR	r1	200 µs	200 µs	750 B
S4	HR	r1	200 µs	200 µs	750 B
S5	MR	r2	200 µs	200 µs	750 B
S6	MR	r2	200 µs	200 µs	750 B
S7	MR	r2	200 µs	200 µs	750 B
S8	LR	r3	200 µs	200 µs	750 B

The core task of routing and scheduling calculated the GCL, which is cyclic and can realize scheduling tasks for all data flow in a hyper-period. All data flows to be scheduled were sent to the link by the same port of the node.

We scheduled the data flow according to their priority and generated the GCL. In this case [16,17,22], we ran the algorithm 30 times and noted the average time. The overall TSN switch scheduling periods were 206.3, 209.4, and 211.3 µs, which caused the MR flow not to be scheduled, thereby resulting in scheduling failure. On the contrary, we ran our algorithm 30 times in the same case, and our result in an average scheduling period was 198.4 µs for the hybrid data flow, which satisfied the worst-case deadline, and all data flow could be scheduled.

For the purpose of verifying the general applicability of the scheduling approach, we changed the count of flow and the scale of the network topology, and we applied the algorithm to verify its scheduling performance. The scheduling results when the network topologies were 10, 20, 40, and 100, and when there were eight data flows are presented in Table 2 (Y means that data flow can be scheduled).

Table 2. The influence of network topology scale on scheduling.

Network Topology Scale	Number of Flow	Scheduling Performance
10	8	Y
20	8	Y
40	8	Y
100	8	Y

It can be observed from Table 2 that our algorithm can maintain the schedulability of the data flow in dynamically changing topologies, which improves the reliability of the algorithm. In addition, in the 10-node network topology, there were five source nodes, three destination nodes, and two switch nodes. The 20-node network topology contained 10 source nodes, five destination nodes, and five TSN switch nodes. In the 40-node network topology, there were 25 source nodes, 10 destination nodes, and five TSN switch nodes. The 100-node network topology contained 65 source nodes, 30 destination nodes, and five TSN switch nodes.

The network topology models of scales 10 and 100 are shown in Figure 12a,b, respectively. The network topology models of scales 20 and 40 were identical in the structure of the switch node. In four cases, switch node1 connected to the source node was selected as the test node, and the algorithm was run 30 times on average. The average scheduling period was 194.3 µs.

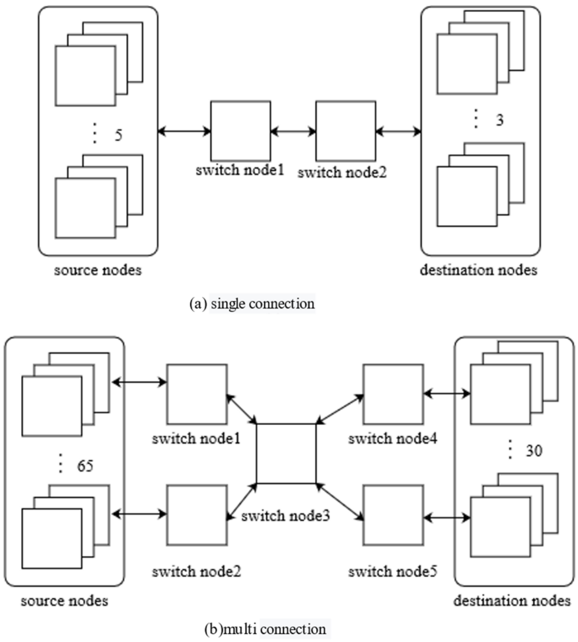

Figure 12. Network topology models of scales 10 and 100 (**a**) single connection; (**b**) multi connection.

In addition, as summarized in Table 3, when the number of flows is 8, 10, 20, and 40, and the network topology scale is 10, we verified the impact of the number of flows on the performance of the algorithm scheduling (Y means that data flow can be scheduled).

Table 3. The influence of the number of data flow on scheduling.

Number of Flow	Network Topology Scale	Scheduling Performance
8	10	Y
10	10	Y
20	10	Y
40	10	Y

As summarized in Table 3, we selected four different flow models, among which the data flow with a scale of 10 contained six HR flows, three MR flows, and one LR flow. The flow with a scale of 20 contained 12 HR flows, six MR flows, and two LR flows. The flow of size 40 contained 23 HR flows, 12 MR flows, and five LR flows. We ran the algorithm 30 times on average, and all data flows were scheduled.

In addition, we compared the algorithm execution times with those reported in the literature [16,17]. The execution time of the algorithm when the scale of network topology remained unchanged at 10, and the scales of data flow were 8, 10, 20, and 40 is shown in Figure 13. In a small-scale data flow, there was little difference in the execution times of the three algorithms. However, with an increase in the data flow scale, our algorithm (A) exhibited a better algorithm execution time than those in [16] (B) and [17] (C).

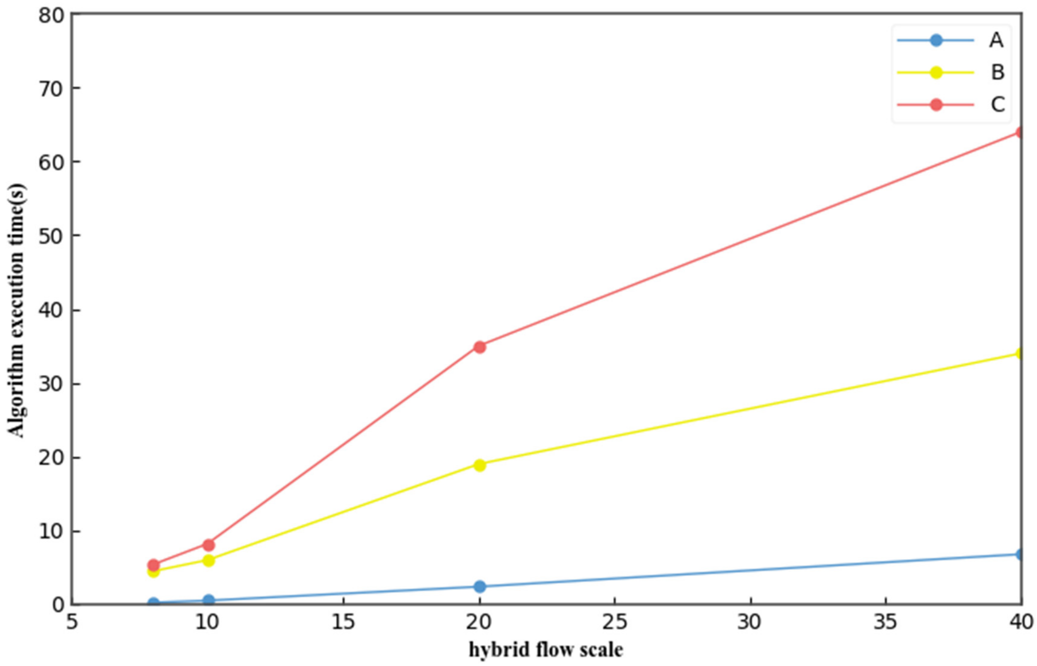

Figure 13. Impact of flow scale on algorithm execution time.

Similarly, the execution time of the algorithm when the data flow scale remains unchanged at eight and the data flow scales were 10, 20, 40, and 100 is shown in Figure 14. In small-scale networks, there is little difference in the execution time of the three algorithms, but with an increase in the network topology scale, our algorithm (A) exhibited a better algorithm execution time compared with those in [16] (B) and [17] (C).

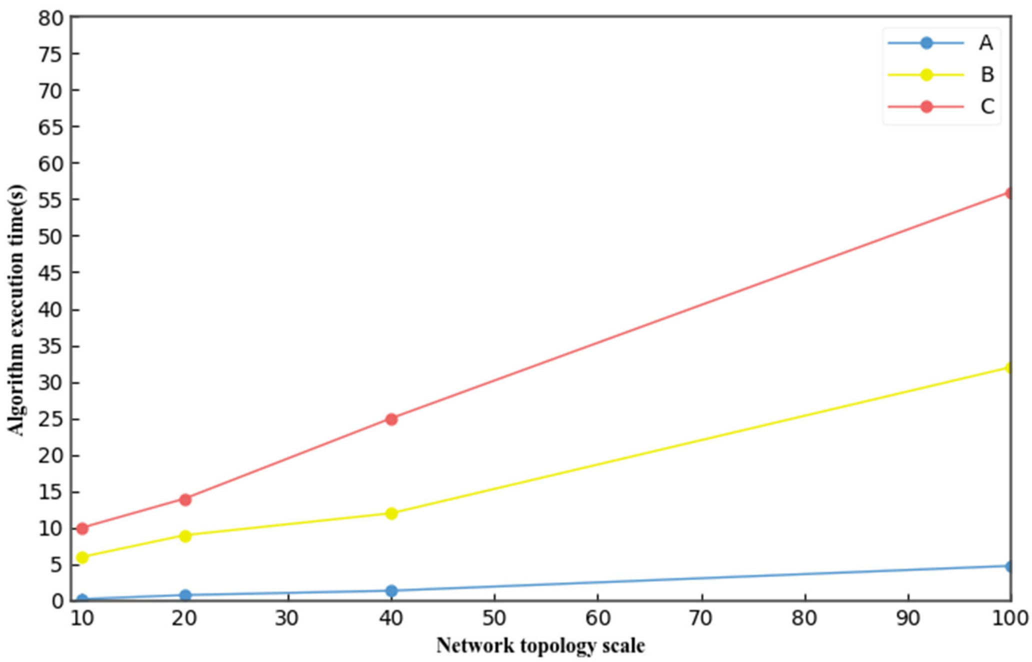

Figure 14. Impact of network topology scale on algorithm execution time.

7. Conclusions

In the research on TSN hybrid data flow transmission and scheduling, a design based on SMT is suitable for the GCL of TSN hybrid data flow transmission. The algorithm can realize both offline and online operating modes. When the network topology and scheduling data mode stream are fixed, a priori calculation can be used to solve the offline scheduling table to realize the scheduling of the data flow; the network topology and the scheduling data flow are not fixed. Additionally, our algorithm could identify new access devices by modifying the configuration file, and it could perform scheduling calculations to generate a new GCL.

Compared with other algorithms [16,17,23], the algorithm based on TSN substation communication considers HR flow, MR flow, and LR flow simultaneously and ensures that MR flow can be scheduled. In addition, our algorithm considered scheduling problems in online situations. The configurability of online scheduling can effectively solve the problem of scheduling table failure when topology changes. Considering the executing duration of the algorithm, our proposed algorithm can effectively reduce the executing duration of scheduling calculations and improve the overall availability of the algorithm in medium- and large-scale network topologies or flows.

The next step of this study is to apply the algorithm to a hardware platform for testing and to further improve the algorithm.

Author Contributions: Conceptualization, B.W.; methodology, Y.L.; software, C.G.; validation, Y.S.; formal analysis, J.W.; investigation, J.X.; resources, X.C.; data curation, Y.S.; writing—original draft preparation, C.G.; writing—review and editing, C.G.; visualization, Y.S.; supervision, Y.L.; project administration, B.W.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China under Grant 2018YFB2003502, and National Natural Science Foundation of China 92067110, and the 2020 industrial Internet innovation and development project—Industrial Internet identification data interaction middleware and resource pool service platform project, Ministry of industry and information technology of the China, and Ningbo Science and Technology project (2018B10089).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gungor, V.C.; Sahin, D.; Kocak, T.; Ergut, S.; Buccella, C.; Cecati, C.; Hancke, G.P. Smart grid technologies: Communication technologies and standards. *IEEE Trans. Ind. Inform.* **2011**, *7*, 529–539. [CrossRef]
2. Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4724–4734. [CrossRef]
3. Pop, P.; Raagaard, M.L.; Craciunas, S.S.; Steiner, W. Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks. *IET Cyber-Phys. Syst. Theory Appl.* **2016**, *1*, 86–94. [CrossRef]
4. Ma, L.; Cheng, S.; Shi, Y. Enhancing learning efficiency of brain storm optimization via orthogonal learning design. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 6723–6742. [CrossRef]
5. Ma, L.; Huang, M.; Yang, S.; Wang, R.; Wang, X. An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. *IEEE Trans. Cybern.* **2021**, *421*, 1–13. [CrossRef]
6. Ma, L.; Wang, X.; Wang, X.; Wang, L.; Shi, L.; Huang, M. TCDA: Truthful Combinatorial Double Auctions for Mobile Edge Computing in Industrial Internet of Things. *IEEE Trans. Mob. Comput.* **2021**, *426*, 1. [CrossRef]
7. Ma, L.; Li, N.; Guo, Y.; Huang, M.; Yang, S.; Wang, X.; Zhang, H. Learning to Optimize: Reference Vector Reinforcement Learning Adaption to Constrained Many-objective Optimization of Industrial Copper Burdening System. *IEEE Trans. Cybernetics.*. [CrossRef]
8. Song, Y.; Guo, C.; Xu, P.; Li, L.; Zhang, R. Research on routing and scheduling algorithms for the simultaneous transmission of diverse data flowing services on the industrial internet. *Sci. Rep.* **2021**, *11*, 18351. [CrossRef]

9. Zheng, Q.; Yang, M.; Yang, J.; Zhang, Q.; Zhang, X. Improvement of generalization ability of deep CNN via implicit regularization in two-stage training process. *IEEE Access* **2018**, *6*, 15844–15869. [CrossRef]
10. Zheng, Q.; Yang, M.; Tian, X.; Jiang, N.; Wang, D. A full stage data augmentation method in deep convolutional neural network for natural image classification. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 4706576. [CrossRef]
11. Liu, S.; Xiao, Z.; You, X.; Su, R. Multistrategy boosted multicolony whale virtual parallel optimization approaches. *Knowl.-Based Syst.* **2022**, *242*, 108341. [CrossRef]
12. Su, R.; Gu, Q.; Wen, T. Optimization of high-speed train control strategy for traction energy saving using an improved genetic algorithm. *J. Appl. Math.* **2014**, *2014*, 507308. [CrossRef]
13. Cauteruccio, F.; Fortino, G.; Guerrieri, A.; Liotta, A.; Mocanu, D.C.; Perra, C.; Terracina, G.; Vega, M.T. Short-long term anomaly detection in wireless sensor networks based on machine learning and multi-parameterized edit distance. *Inf. Fusion* **2019**, *52*, 13–30. [CrossRef]
14. Sidhu, T.S.; Yin, Y. Modelling and simulation for performance evaluation of IEC61850-based substation communication systems. *IEEE Trans. Power Deliv.* **2007**, *22*, 1482–1489. [CrossRef]
15. Gavriluţ, V.; Zhao, L.; Raagaard, M.L.; Pop, P. AVB-aware routing and scheduling of time-triggered traffic for TSN. *IEEE Access* **2018**, *6*, 75229–75243. [CrossRef]
16. He, F.; Zhao, L.; Li, E. Impact analysis of flow shaping in Ethernet-AVB/TSN and AFDX from network calculus and simulation perspective. *Sensors* **2017**, *17*, 1181. [CrossRef]
17. Wang, Y.; Chen, J.; Ning, W.; Yu, H.; Lin, S.; Wang, Z.; Chen, C. A time-sensitive network scheduling algorithm based on improved ant colony optimization. *Alex. Eng. J.* **2021**, *60*, 107–114. [CrossRef]
18. Zhao, L.; Pop, P.; Gong, Z.; Fang, B. Improving Latency Analysis for Flexible Window-Based GCL Scheduling in TSN Networks by Integration of Consecutive Nodes Offsets. *IEEE Internet Things J.* **2020**, *8*, 5574–5584. [CrossRef]
19. Yu, Q.; Wan, H.; Zhao, X.; Gao, Y.; Gu, M. Online scheduling for dynamic VM migration in multicast time-sensitive networks. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3778–3788. [CrossRef]
20. Houtan, B.; Ashjaei, M.; Daneshtalab, M.; Sjödin, M.; Mubeen, S. Synthesising schedules to improve QoS of best-effort traffic in TSN networks. In Proceedings of the 29th International Conference on Real-Time Networks and Systems, New York, NY, USA, 7–9 April 2021; pp. 68–77.
21. Falk, J.; Dürr, F.; Rothermel, K. Exploring practical limitations of joint routing and scheduling for TSN with ILP. In Proceedings of the 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Hakodate, Japan, 28–31 August 2018; pp. 136–146.
22. Pahlevan, M.; Obermaisser, R. Genetic algorithm for scheduling time-triggered traffic in time-sensitive networks. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; Volume 1, pp. 337–344.
23. Laursen, S.M.; Pop, P.; Steiner, W. Routing optimization of AVB flow in TSN networks. *ACM Sigbed Rev.* **2016**, *13*, 43–48. [CrossRef]
24. Bingqian, L.; Yong, W. Hybrid-GA based static schedule generation for time-triggered ethernet. In Proceedings of the 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), Beijing, China, 4–6 June 2016; pp. 423–427.
25. Li, Z.; Wan, H.; Deng, Y.; Zhao, X.; Gao, Y.; Song, X.; Gu, M. Time-triggered switch-memory-switch architecture for time-sensitive networking switches. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *39*, 185–198. [CrossRef]
26. Barzegaran, M.; Pop, P. Communication Scheduling for Control Performance in TSN-Based Fog Computing Platforms. *IEEE Access* **2021**, *9*, 50782–50797. [CrossRef]
27. Song, Y.; Guo, C.; Xu, P.; Wang, J. Design of Deterministic Transmission Framework for Distributed Power System Based on Digital Twin. In Proceedings of the 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2), Taiyuan, China, 22–24 October 2021; pp. 3391–3395.
28. Zhou, Z.; Lee, J.; Berger, M.S.; Park, S.; Yan, Y. Simulating TSN traffic scheduling and shaping for future automotive Ethernet. *J. Commun. Netw.* **2021**, *23*, 53–62. [CrossRef]
29. Nasrallah, A.; Thyagaturu, A.S.; Alharbi, Z.; Wang, C.; Shao, X.; Reisslein, M.; Elbakoury, H. Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS). *IEEE Access* **2019**, *7*, 44165–44181. [CrossRef]
30. Kim, H.J.; Choi, M.H.; Kim, M.H.; Lee, S. Development of an Ethernet-Based Heuristic Time-Sensitive Networking Scheduling Algorithm for Real-Time In-Vehicle Data Transmission. *Electronics* **2021**, *10*, 157. [CrossRef]
31. Bello, L.L.; Ashjaei, M.; Patti, G.; Behnam, M. Schedulability analysis of Time-Sensitive Networks with scheduled traffic and preemption support. *J. Parallel Distrib. Comput.* **2020**, *144*, 153–171. [CrossRef]
32. Vlk, M.; Hanzálek, Z.; Brejchová, K.; Tang, S.; Bhattacharjee, S.; Fu, S. Enhancing schedulability and throughput of time-triggered traffic in IEEE 802.1 Qbv time-sensitive networks. *IEEE Trans. Commun.* **2020**, *68*, 7023–7038. [CrossRef]
33. Yu, Q.; Gu, M. Adaptive group routing and scheduling in multicast time-sensitive networks. *IEEE Access* **2020**, *8*, 37855–37865. [CrossRef]
34. Bayrakdar, M.E. Exploiting cognitive wireless nodes for priority-based data communication in terrestrial sensor networks. *ETRI J.* **2020**, *42*, 36–45. [CrossRef]
35. De Moura, L.; Bjørner, N. Z3: An efficient SMT solver. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 337–340.