electronics

MDPI

Article

A Network Scheduling Method Based on Segmented Constraints for Convergence of Time-Sensitive Networking and Industrial Wireless Networks †

Min Wei [1,*], Chang Liu [2], Jin Wang [2] and Shujie Yang [1]

1 School of Automation, China-Korea Belt and Road Joint Laboratory on Industrial Internet of Things, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; s200303048@stu.cqupt.edu.cn

2 State Grid Hubei Electric Power Research Institute, Wuhan 430077, China; liuchang.sgcc@foxmail.com (C.L.); syywangjin@foxmail.com (J.W.)

* Correspondence: weimin@cqupt.edu.cn

† This paper is an extension of the 2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Republic of Korea, 3–5 January 2023.

Abstract: In industrial applications, it is necessary to select different types of networks according to different communication requirements. To meet this requirement, a converged network of wired and wireless networks is frequently employed. Notably, fulfilling the end-to-end transmission requirements of converged networks is challenging. As a solution, converged-network scheduling methods have proved valuable. In this paper, a network scheduling method for the convergence of industrial wireless networks and time-sensitive networks is proposed. Additionally, the proposed method is tested and verified. The results show that the end-to-end average transmission delay is reduced and the jitter is acceptable.

Keywords: industrial wireless network; time-sensitive networking; scheduling method; greedy algorithm; dataflow

Citation: Wei, M.; Liu, C.; Wang, J.; Yang, S. A Network Scheduling Method Based on Segmented Constraints for Convergence of Time-Sensitive Networking and Industrial Wireless Networks. *Electronics* **2023**, *12*, 2418. https://doi.org/10.3390/electronics12112418

Academic Editors: Syed Muhammad Raza and Duc Tai Le

Received: 30 April 2023
Revised: 18 May 2023
Accepted: 24 May 2023
Published: 26 May 2023

1. Introduction

To meet the heterogeneous application requirements of future smart factories, a converged network of wired and wireless networks is needed. This converged network enables the entire industrial network to achieve both communication reliability and flexibility to meet the requirements of smart factories [1]. To meet the requirements of industrial automation, converged networks conventionally comprise several types of deterministic networks. Many deterministic industrial wireless networks exist, such as WIA-PA [2], WirelessHART [3] and ISA100.11a [4]. In the wire part, the time-sensitive network (TSN) is a deterministic industrial wired network that is introduced to the automation area. TSNs can facilitate the convergence of operational technology and information technology infrastructures to be shared across networks, enabling the coexistence of time-sensitive traffic (guaranteed delivery) with best-effort traffic (nonguaranteed delivery). However, the existing converged networks do not achieve efficient traffic scheduling between different networks [5]. Therefore, a method for achieving the efficient implementation of data scheduling between industrial wired networks and industrial wireless networks inside a converged network is urgently needed [6].

To date, many researchers have focused on the convergence of industrial heterogeneous networks using a software-defined network (SDN). To meet the QoS requirements, Yu [7] designed an allocation model that considers changing traffic based on the concept of SDNs and operates according to a newly designed network-resource allocation method. To achieve interoperability, Xue [8] proposed an interconnection scheme following the paradigm of software-defined time-sensitive networks (SD-TSNs), where the network is managed by a central entity with a network controller and a service orchestrator. Cruces [9], drawing on the concept of SDNs, designed a network architecture combining an industrial wireless network and a TSN. The solution to the scheduling problem of the TSN part and the industrial wireless network part in the industrial heterogeneous network is to use the existing scheduling technology. Wei [10] designed a converged network based on an SDN, which consists of a TSN and an industrial wireless network. Notably, the network designed by Wei could be effectively scheduled using the algorithm designed in this study. However, as some details of the algorithm remain inadequately considered, this extension supplements the existing issue.

Using SDNs, the data forwarding and network control of the industrial wireless network and industrial wired network can be separated, realizing dataflow scheduling with the converged network. This study focuses on the dataflow scheduling method of a converged network of industrial wireless and industrial wired networks. WIA-PA is selected as the industrial wireless network and the TSN is chosen as the industrial wired network, attributed to their excellent performance on determinism. This paper is structured as follows: In Section 2, an SDN-based converged-network architecture and a dataflow priority conversion mechanism between the two networks are proposed. In Section 3, the scheduling methods of WIA-PA and the TSN are presented, individually. A test platform is built and the average transmission delay and jitter of the converged network are tested and analyzed (see Section 4). Finally, after summary and analysis, the conclusions are drawn in Section 5.

2. SDN-Based Converged Network Architecture and Dataflow Priority Conversion Mechanism

The SDN-based converged-network architecture and dataflow priority conversion mechanism are discussed in this section. The performance of the scheduling method is based on these two factors.

2.1. SDN-Based Converged-Network Architecture

The proposed SDN-based converged-network architecture is shown in Figure 1.

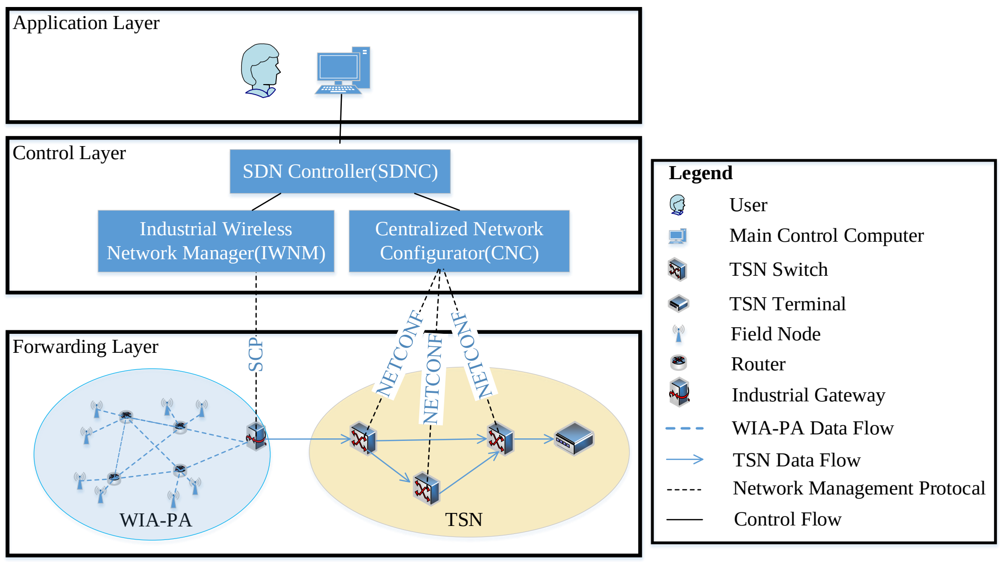

Figure 1. SDN-based converged-network architecture.

The converged network consists of an application layer, a control layer and a forwarding layer. The composition and function of each layer are as follows.

1. Application Layer

The application layer consists of the user and the main control computer. The configuration of the SDN controller (SDNC) can be set by users according to their needs, using the application program on the main control computer.

2. Control Layer

The control layer consists of the SDNC, an industrial wireless network manager (IWNM) and a centralized network configurator (CNC). The secure copy protocol (SCP) is employed to establish communication between the IWNM and the industrial gateway so that WIA-PA can be scheduled. The scheduling scheme of the switches in the TSN is configured by the CNC and delivered through the network configuration protocol (NETCONF) so that the TSN can be scheduled. The IWNM and CNC in the converged network are configured by the SDNC according to the application request from the application layer.

3. Forwarding Layer

The forwarding layer consists of the field node, router, industrial gateway, TSN switch and TSN terminal. The field node is utilized to send data, the router, industrial gateway and TSN switch are employed to forward data and the TSN terminal is employed to receive data.

WIA-PA is controlled by the IWNM. The IWNM establishes communication with the industrial gateway through the SCP to model the WIA-PA system and design the time slot channel schedule for WIA-PA. WIA-PA works according to an already-designed schedule. The TSN is scheduled and managed by the CNC, which is responsible for calculating and configuring the data transmission scheme of the TSN switch. The information of the IWNM and CNC is aggregated to the SDNC so that the SDNC can obtain the global network topology information. Subsequently, the SDNC calculates the transmission priority of converged network dataflow to complete the scheduling of the converged network.

2.2. Dataflow Priority Conversion Mechanism

There are many kinds of dataflows with different end-to-end priorities in the network. Even the same kind of data may still have different end-to-end priorities. For example, although the alarm data that alerts of impending danger and the alarm data that warn that a danger has occurred are both alarm data, they should have different end-to-end priorities. A total of 16 end-to-end priority levels are defined in this study, ranging from 0 to 15. The lower end-to-end value represents the higher end-to-end priority. To simplify the mapping relationship, they are divided into eight classes.

According to different application scenarios of different applications [11,12], the dataflows in the converged network may be classified into four categories: security, control, monitoring and general, and the end-to-end time-delay requirements for the dataflow in class security, control, monitoring and general are set to (0, 50 ms), (50 ms, 90 ms), (90 ms, 150 ms) and (150 ms, +∞), respectively.

The same type of application dataflow still has different end-to-end priorities; therefore, there are 32 combinations in total. The delay requirements and end-to-end priorities of these 32 combinations are different and they should be assigned the priorities of the TSN PR_{TSN} and WN PR_{WN}. The WIA-PA dataflow priority (PR_{WN}) was originally defined in WIA-PA [13]. The data link sublayer of WIA-PA defines four priorities of dataflows, which are marked as 0, 1, 2 and 3 in this paper. The smaller value represents the higher priority. According to the standard of the TSN, the TSN dataflow priority (PR_{TSN}) has eight types (0–7). The smaller value represents the lower priority [14].

The mapping relations of the WIA-PA and TSN dataflow priorities under different delay requirements and priorities are shown in Table 1. The PR_{WN} and PR_{TSN} of one dataflow are set with the following method. To ensure the transmission quality of data streams with high end-to-end priority, considering that the TSN exhibits better delay performance than WN, when the delay requirements are the same, data streams with high end-to-end priority should use TSN resources extensively and dataflows with low end-to-end priority should use WN resources extensively.

Table 1. Mapping relations of the WIA-PA and TSN dataflow priorities under different delay requirements and priorities.

Delay Requirement (ms)	Priority	PR_{WN}	PR_{TSN}
$[0, 50)$	[0, 1]	0	7
	[2, 3]	0	6
	[4, 5]	1	5
	[6, 7]	1	4
	[8, 9]	2	3
	[10, 11]	2	2
	[12, 13]	3	1
	[14, 15]	3	0
$[50, 90)$	[0, 1]	0	7
	[2, 3]	0	6
	[4, 5]	1	5
	[6, 7]	1	4
	[8, 9]	2	3
	[10, 11]	2	2
	[12, 13]	3	1
	[14, 15]	3	0
$[90, 150)$	[0, 1]	0	7
	[2, 3]	0	6
	[4, 5]	1	5
	[6, 7]	1	4
	[8, 9]	2	3
	[10, 11]	2	2
	[12, 13]	3	1
	[14, 15]	3	0
$[150, +\infty)$	[0, 1]	0	7
	[2, 3]	0	6
	[4, 5]	1	5
	[6, 7]	1	4
	[8, 9]	2	3
	[10, 11]	2	2
	[12, 13]	3	1
	[14, 15]	3	0

According to the mapping relations, the priority of the dataflow in WIA-PA and the TSN is obtained by the SDNC using the delay requirement and delay input.

3. Scheduling Methods for WIA-PA and TSN

In this paper, dataflow scheduling methods are designed for WIA-PA and the TSN, respectively. Relying on PR_{TSN} and PR_{WN} obtained in Section 2, the dataflows are scheduled in WIA-PA and the TSN according to their scheduling rules without interfering with each other. Thereby, the scheduling optimization of the entire converged network is accomplished.

3.1. Dataflow Scheduling Method in WIA-PA

In this paper, a time slot channel resource scheduling method based on different scheduling constraints is designed and used to schedule the transmission of dataflows in WIA-PA. The related time slot conflicts situation is discussed in [15,16]. The symbols used in this method and their definitions are introduced in Table 2. There are nine steps in this method, starting from step 1 and ending in step 9; all the steps are introduced here.

Table 2. Symbol introduction of the dataflow scheduling method in WIA-PA.

Symbols	Description
$WN_{e,i}$	The node with ID i
$WN_{r,j}$	The router with ID j
WN_g	Industrial gateway
WN_{src}	Sending node at each hop
WN_{dst}	Receiving node at each hop
$(WN_{src}, WN_{dst})_m$	The sending and receiving nodes at the mth hop of each link
$S_0, S_1, \ldots, S_a$	Time slot number
$C_0, C_1 \ldots, C_b$	Channel number
L_k	Number of each link
L_k^m	The mth hop of L_k

Step 1:

All link sets in the network are calculated using the IWNM, after which the link set (LS) shown in Table 3 is obtained. According to the difference between the start and end, a complete data transmission link is divided into three link types. The first type is the node–router link: its LS has a combination of all the nodes and all the routers. The second type is the router–router link: its LS has a combination of the routers and all the routers. The third type is the router–gateway link: its LS has a combination of the routers and all the gateways.

Table 3. Link sets (LS) of different link types.

Link Type	Link Set
From node to router	$(WN_{e,1} \to WN_{r,1}), \ldots, (WN_{e,i} \to WN_{r,1})$ $(WN_{e,i+1} \to WN_{r,2}), \ldots, (WN_{e,2i} \to WN_{r,2})$ $(WN_{e,2i+1} \to WN_{r,3}), \ldots, (WN_{e,3i} \to WN_{r,3})$ $\left(WN_{e,(j-1)i+1} \to WN_{r,j}\right), \ldots, \left(WN_{e,ji} \to WN_{r,j}\right)$
From router to router	$(WN_{r,1} \to WN_{r,2}), (WN_{r,2} \to WN_{r,1}),$ $\ldots, \left(WN_{r,1} \to WN_{r,j}\right)$
From router to gateway	$(WN_{r,1} \to WN_{r,2} \to WN_{r,3} \to WN_g),$ $\left(WN_{r,1} \to WN_{r,j} \to WN_g\right),$ $\left(WN_{r,2} \to WN_{r,1} \to WN_{r,j} \to WN_g\right),$ $\ldots, \left(WN_{r,j} \to WN_g\right)$

In Table 3, the symbol, "$\to$" indicates the direction of the dataflows. The sending node of the dataflows is on the left side of $\to$, whereas the receiving node is on the right.

Step 2:

According to the LS, the IWNM calculates the transmission path set (TPS) of every link shown in Table 4. A link corresponds to a TPS composed of several devices. A link starts from the TPS of the first device and ends at the TPS of the last device.

Table 4. Transmission path set (TPS) of every link.

Link	TPS
L_1	$\left(WN_{e,1} \to WN_{r,1} \to WN_{r,j} \to WN_g\right)$
L_2	$(WN_{e,1} \to WN_{r,1} \to WN_{r,2} \to WN_{r,3} \to WN_g)$
L_3	$\left(WN_{e,i} \to WN_{r,1} \to WN_{r,j} \to WN_g\right)$
...	...
L_k	$\left(WN_{e,ji} \to WN_{r,j} \to WN_g\right)$

Step 3:

To ensure that there is no conflict in the network, only one dataflow is allowed to be transmitted at the same time through the same channel and a time slot channel resource is designed in this study. In the form of the matrix shown in Figure 2, all the time slot resources and available channel resources in the WIA-PA are employed to establish a time slot channel resource scheduling table. In this matrix, a unique channel and a unique time slot can form a unique time slot channel resource. For example, the resource circled in red is uniquely identified by the channel with number 1 and the slot with number 1. All time slot channel resources are initialized to idle in this step.

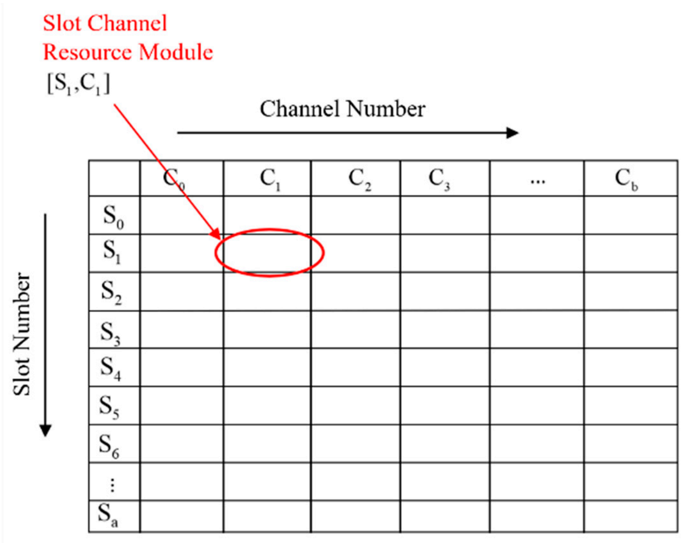

Figure 2. Time slot channel resource module scheduling table.

Step 4:

A transmission link is selected from the TPS in the order of the link sequence number and divided into several hops. The sending device of the first hop of any transmission link is the node and the receiving device of the last hop is the gateway. The sending and receiving devices of other hops are routers.

Step 5:

According to the time slot frequency hopping analysis of WIA-PA, to avoid conflict, four constraints are designed, as follows.

- Constraint 1: in the scheduling unit composed of any channel and time slot, only one link can transmit data at most.
- Constraint 2: in any time slot, the number of links for data transmission cannot exceed the number of channels.
- Constraint 3: in any time slot, a node can transmit data with only one neighboring node.
- Constraint 4: when data communication ends, all the data packets in the network must reach the industrial gateway.

According to these four constraints, each channel is traversed one by one, starting from the initial time slot, S_0. If all the time slot channel resource modules under the current time slot are empty, module $[S_0, C_0]$ is selected, where the first channel is located as the resource scheduling module of the first hop $(WN_{src} \to WN_{dst})_1$. Meanwhile, the source and destination nodes of this hop are recorded and the channel resource module of the time slot is marked as occupied.

Step 6:

The number of hops to be scheduled is updated to the next hop $(WN_{src} \rightarrow WN_{dst})_2$. Additionally, the time slot is updated to S_1. To ensure the balanced allocation of all channel resources, resource modules $[S_1, C_1]$ are selected for idle judgment and constraint judgment. If the result of the judgment is successful, the module becomes the resource scheduling module of the hop $(WN_{src} \rightarrow WN_{dst})_2$.

Step 7:

Step 6 is repeated until the time slot channel resource module is allocated for each hop of the current link. At this time, the scheduling of the current link ends. If the result of the judgment in step 6 is failure, then this step is skipped to step 8.

Step 8:

Each time slot channel resource module in time slot S_1 is traversed again, as are the idle judgment and the constraint judgment conditions. The first traversed module judged successful is selected as the resource scheduling module of the hop $(WN_{src} \rightarrow WN_{dst})_2$ until the time slot channel resource module is allocated for each hop of the current link. At this time, the scheduling of the current link ends.

Step 9:

The next link is selected from the TPS in sequence. Steps 4–8 operate in sequence until all links are scheduled.

After scheduling, the time slot channel resource scheduling result shown in Table 5 and the time slot channel resource module shown in Table 6, called by each transmission link, are given as output.

Table 5. Time slot channel resource scheduling result.

Channel	Time Slot	Resource
C_0	S_0	$L_1^1 : WN_{e,1} \rightarrow WN_{r,1}$
	S_1	——
	S_2	$L_2^1 : WN_{e,1} \rightarrow WN_{r,1}$
	…	…
	S_a	$L_{k-1}^5 : WN_{r,3} \rightarrow WN_g$
C_1	S_0	$L_5^1 : WN_{e,i+1} \rightarrow WN_{r2}$
	S_1	$L_1^2 : WN_{r,1} \rightarrow WN_{r,j}$
	S_2	$L_6^1 : WN_{e,i+1} \rightarrow WN_{r,2}$
	…	…
	S_a	——
…	…	…

Table 6. Time slot channel resource module used in each link.

Link	Time Slot Channel Resource Module
L_1	$[S_0, C_0]_1, [S_1, C_1]_2, [S_2, C_2]_3$
L_2	$[S_2, C_0]_1, [S_3, C_1]_2, [S_4, C_2]_3, [S_5, C_3]_4$
L_3	$[S_4, C_0]_1, [S_5, C_1]_2, [S_6, C_2]_3$
…	…
L_k	$[S_{a-2}, C_0]_1, [S_{a-1}, C_1]_2$

In Table 5, the symbol, "——," indicates that there is no corresponding item. In a specific time slot of a specific channel, the resource is allocated to a specific hop, indicating that only the data of this hop can be transferred in this time slot of this channel. For example, in time slot S_0, only the data from $WN_{e,1}$ to $WN_{r,1}$ can occupy channel C_0.

In Table 6, a specific link corresponds to several time slot channel resource modules, indicating that a dataflow occupies several different time slot channel resource modules to piece together a complete link to complete the data transmission. For example, the time slot channel resource module of L_1 is $[S_0, C_0]_1, [S_1, C_1]_2, [S_2, C_2]_3$. Thus, time slot S_0 and channel C_0 will be occupied to complete the first hop of L_1, time slot S_1 and channel C_1 will be occupied to complete the first hop of L_1 and time slot S_2 and channel C_2 will be occupied to complete the first hop of L_1.

According to Tables 5 and 6, the state of the device in WIA-PA is configured to realize the scheduling of the dataflow in WIA-PA.

3.2. Dataflow Scheduling Method in TSN

In this study, a greedy time slot allocation (GTSA) scheduling algorithm is proposed for TSNs.

According to the characteristics of the dataflows to be transmitted, a GTSA transmits each data frame of different priority dataflows within T_s to optimize each dataflow. The symbols in this method are introduced in Table 7 and the specific execution steps of this method are as follows.

Table 7. Symbol introduction of the dataflow scheduling method in TSN.

Symbols	Description
T_s	TSN scheduling period
f_i	The ith TSN dataflow
TR_i	The transmission speed of f_i
T_i	The transmission period of f_i
k_i	The number of frames transmitted in one scheduling period of f_i
$w_{tr}(i,j)$	The jth frame transmission window of f_i
$w_{tr}(i)$	The frame transmission window length of f_i
$w_{tr}(i,j).t_{start}$	The start time of the jth frame transmission window of f_i
δ	The time length of a single time slot
S	The total number of slots in the scheduling period
S_i	The number of time slots occupied by a single-frame transmission window
θ	The capacity of the transmission link
$F_{scheduled}$	List of TSN flows to be scheduled
L_i	The frame length of f_i
$L_{overhead}$	Fixed overhead of the frame transmission

Here, the frame transmission scheduling mechanism is introduced first and the specific execution steps of this method are shown.

3.2.1. Introduction to the Frame Transmission Scheduling Mechanism

The dataflow scheduling method in the TSN designed in this study is based on a frame transmission scheduling mechanism. Therefore, it is necessary to introduce a frame transmission scheduling mechanism first.

The transmission of converged network dataflows on the TSN side can be scheduled and configured by the CNC [17]. Based on the network topology information on the TSN side and the information on the dataflows to be scheduled, the CNC calculates the gate control list (GCL) [18] and subsequently configures it to the corresponding switch device.

The gating line of the switch output port periodically switches the gate according to the GCL scheduling scheme to ensure that each dataflow is transmitted in the appropriate time slot in the gating cycle period [19].

In the TSN switch, each outgoing port is designed with eight priority queues [20], corresponding to eight priorities, from 0 to 7, which are employed to transmit dataflows of corresponding priorities, among which priority 0 is the lowest and priority 7 the highest [21]. The TSN dataflow enters the corresponding priority queue for buffering according to the premarked priority label. When the TAS gate is opened, the data frame in the corresponding queue can be outputted [22].

A specific working example of priority filtering is shown in Figure 3. The dataflow mixed with different dataflows of different priorities originates from the ingress port, passes through the switching fabric and faces eight queues. Every queue is controlled by its gate. In Figure 3, only the queue of priority 7 is opened, whereas the other queues are closed. Thus, the dataflows of other priorities are blocked inside their queues and only the dataflow of priority 7 can come out from its queue and transmit toward the egress port.

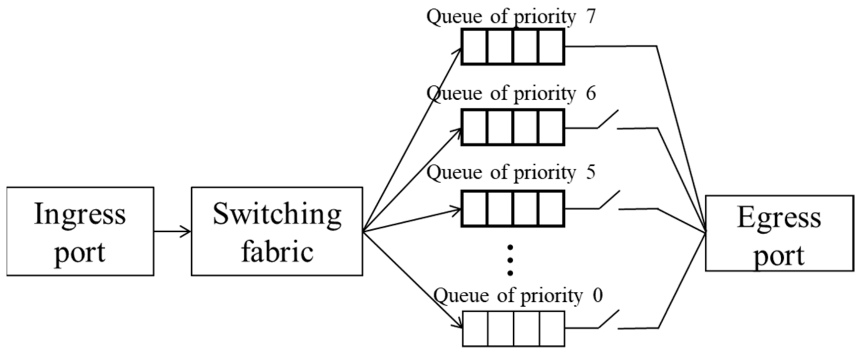

Figure 3. Schematic of the priority cluster in TSN.

The frame transmission window arrangement scheme for each dataflow at the output port of the TSN switch actually allocates an appropriate number of transmission time slots for the frame transmission window of each dataflow. The CNC calculates the frame transmission window scheduling scheme of each dataflow that meets the QoS [23] requirements by the scheduling algorithm and subsequently sends the calculated scheduling scheme to the TSN switch for configuration [24].

According to the characteristic information of multiple dataflows to be transmitted, the GTSA scheduling algorithm in this study schedules each data frame of different priority dataflows to be transmitted to the link in a relatively appropriate time window within the scheduling period (T_s). The purpose of optimizing the transmission jitter of each dataflow is achieved using this method.

In the scheduling algorithm on the TSN side, the CNC calculates the TSN scheduling cycle based on the transmission cycle of all dataflows to be scheduled. Thereafter, the transmission of each TSN dataflow commences at a fixed time within each scheduling cycle. Assuming that the start time of the first scheduling cycle is 0 μs, then the end time of the current scheduling cycle is the start time of the next scheduling cycle [25]. Since the data link layer fragments the IP datagrams [26] into frames and subsequently transmits them [27], this study refers to the scheduling of TSN dataflows at the link layer as frame transmission scheduling.

A schematic of the frame transmission scheduling is shown in Figure 4. In one scheduling period, the TSN dataflow (f_1) has four frames ($f_{1,1}$, $f_{1,2}$, $f_{1,3}$, $f_{1,4}$) to be transmitted. In the scheduling period, the TAS gates (7, 6, 5) corresponding to different TSN dataflows (f_1, f_2, f_3) are combined into the final TAS gate scheduling result. Thereafter, the frame transmission of different TSN dataflows (f_1, f_2, f_3) is combined into the final frame transmission scheduling result. In Figure 4, the color green, orange and blue respectively represent dataflow with priorities 7, priorities 6 and priorities 5. Lines of different colors represent the states of different TAS gates for different dataflow. High lines represent that the gate is on, low lines represent that the gate is off.

The algorithm proposed in this paper is utilized to find the available frame transmission scheduling scheme. According to the available frame transmission scheduling scheme, the CNC generates a gating schedule and sends it to the switch.

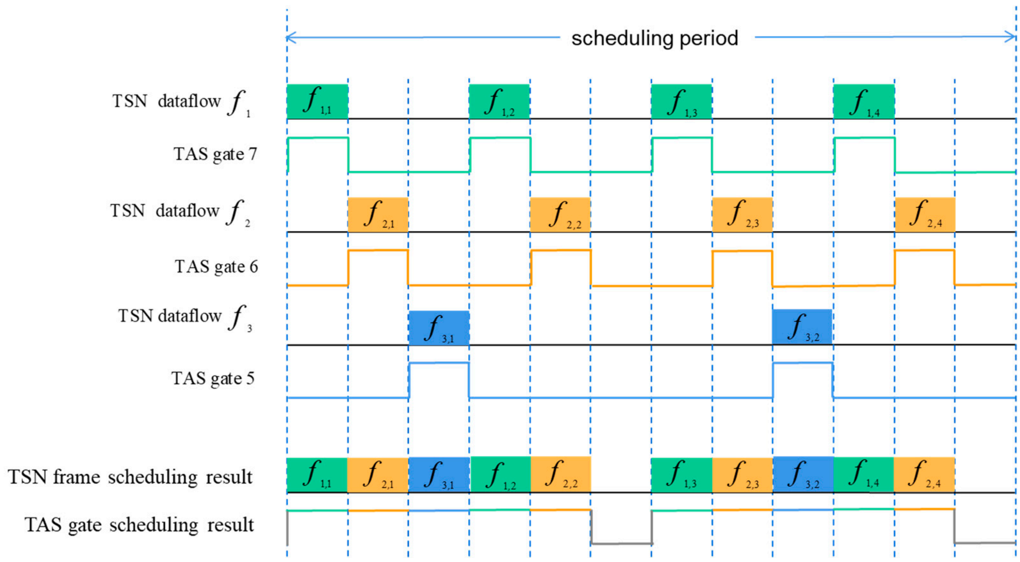

Figure 4. Schematic of the frame transmission scheduling.

3.2.2. Steps of the Dataflow Scheduling Method in TSN

The scheduling method in the TSN operates based on the technology of the frame transmission scheduling mechanism. There are 10 steps in this method, starting from step 1 and ending in step 10; all the steps are introduced here.

Step 1

According to PR_{TSN} of f_i, dataflows are sorted by the CNC, after which $F_{scheduled}$ is obtained. $F_{scheduled}$ consists of all the dataflows that need to be scheduled.

Step 2

With f_i input, T_s, S, δ, k_i and $w_{tr}(i)$ are computed according to (1), (2), (3), (4) and (5), respectively. Symbol $lcm()$ represents the calculation of the least common multiple.

$$T_s = lcm(T_1, T_2, \ldots, T_i), \tag{1}$$

$$S = lcm(TR_1, TR_2, \ldots, TR_i, \theta), \tag{2}$$

$$\delta = \frac{T_s}{S}, \tag{3}$$

$$k_i = \frac{T_s}{T_i}, \tag{4}$$

$$w_{tr}(i) = \frac{L_i + L_{overhead}}{TR_i}. \tag{5}$$

Step 3

All data frames to be transmitted in the scheduling period of each f_i are sorted by the CNC according to their frame transmission sequences (V_i) defined in (6):

$$V_i = \{0, T_i, 2T_i, 3T_i, \ldots, nT_i\}. \tag{6}$$

Step 4

All time slots to be allocated in T_s are initialized to idle state by the CNC and are numbered in order of time.

Step 5

For each f_i in $F_{scheduled}$, the time slots required for k_i data frame transmission windows need to be arranged within T_s. The time interval between adjacent data frames should be equal to T_i. According to (7), S_i of f_i is calculated using the CNC:

$$S_i = \frac{w_{tr}(i)}{\delta}. \quad (7)$$

Step 6

Among the unoccupied idle time slots, the time slots required by k_i frame transmission windows that satisfy the following two conditions are searched. If the time slots that meet the requirement are searched successfully, step 7 is implemented, otherwise, step 9 is implemented.

- Condition 1: each frame transmission window needs to be allocated with consecutive S_i idle time slots.
- Condition 2: the time interval between the adjacent frame transmission windows of f_i is equal to T_i of f_i.

Step 7

When the time slot satisfying two conditions of the jth frame transmission window of f_i is found in step 6, step 7 is performed. Among the time slots satisfying the above two conditions, the time slot with the smallest number is allocated to the k_i data frame transmission windows of f_i and marked as occupied.

Step 8

The jth frame transmission window of f_i, which is marked as $w_{tr}(i, j)$ is saved by the CNC. Thereafter, step 10 is implemented.

Step 9

When the time slot satisfying two conditions of the jth frame transmission window of f_i is not found in step 6, step 9 is performed. The location $j * T_i / \delta$ time slots away from the start of the first frame transmission window are selected. The consecutive S_i idle time slots in this location are allocated to $w_{tr}(i, j)$. If there are no S_i consecutive idle time slots at this location, they are searched for at the next location and allocated.

Step 10

Return to step 5; the next dataflow will be scheduled until all dataflows have been allocated with time slots.

According to the execution results obtained in the above steps, the CNC calculates the scheduling sequence table of all dataflows in the converged network at the output port of the TSN switch and configures the scheduling table to the TSN switch. Thereafter, according to the scheduling table calculated by the above CNC, the gate control queue of the TSN switch periodically opens and closes the gate of the corresponding queue to ensure that the data frames in the queue can be transmitted in the scheduled sending window. In this way, the scheduling of the TSN part is completed.

4. Results

In this study, a test verification platform for the converged network of WIA-PA and the TSN is built. The hardware and software employed in the platform are shown in Tables 8 and 9.

The platform is shown in Figure 5. Using the platform, the validity verification of the proposed method and the test and analysis of the end-to-end transmission delay and jitter are implemented.

Table 8. Hardware used in the test verification platform.

Hardware Name	Description	Number
WIA-PA Gateway	CQUPT developed	1
WIA-PA Router	CQUPT developed	2
WIA-PA Node	CQUPT developed	4
TSN Switch	KYLAND SICOM3000	2
TSN Adapter	ZY1 E314500	5
TSN Terminal	HP ProBook 650 G8 with Windows10 64 bit	2
Wireless Packet Sniffer	E18-2G4U04B 2.4GHz	1

Table 9. Software used in the test verification platform.

Software Name	Version
IAR Embedded Workbench	V7.1
Texas Instruments Packet Sniffer	V2.16.3
IntelliJ IDEA	V2020.2.1 ×64
Java JDK	Java8
Wireshark	Version 3.2.5

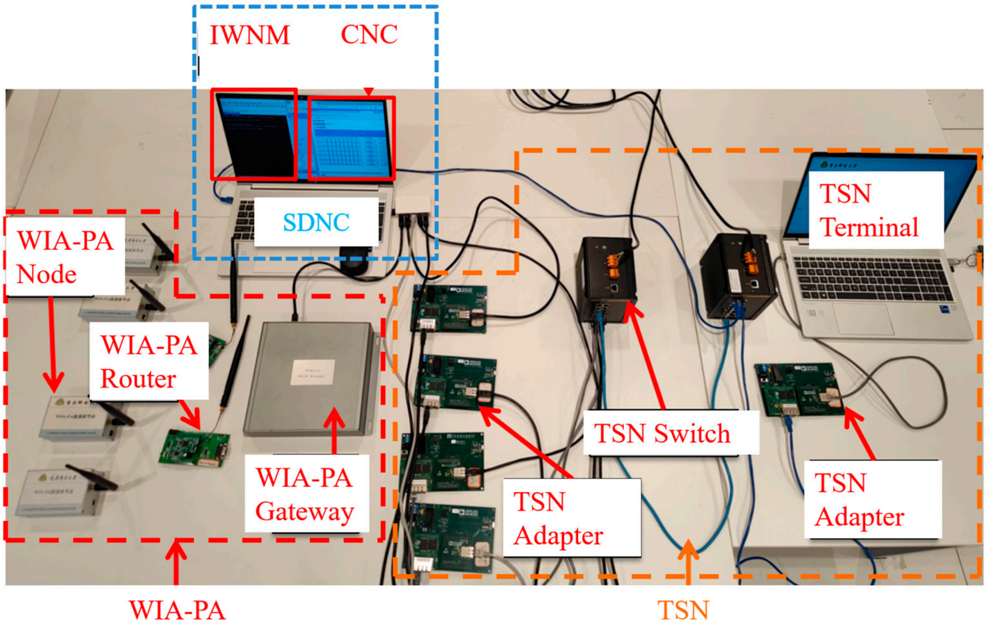

Figure 5. Test verification platform of the converged network of WIA-PA and TSN.

After the startup of the converged network, four different dataflows begin to transfer. From four different WIA-PA nodes (E1, E2, E3 and E4), the data are sent to a TSN terminal (TT). The data go through two different WIA-PA routers (R1 and R2), a WIA-PA gateway (G) and two different TSN switches (TS1 and TS2). The parameters of the dataflows in the network are shown in Table 10.

Table 10. Parameters of the dataflows in the network.

Dataflow	Path	Priority
f1	L1: E1 → R1 → G → TS1 → TS2 → TT	7
f2	L2: E2 → R1 → G → TS1 → TS2 → TT	6
f3	L3: E3 → R2 → G → TS1 → TS2 → TT	5
f4	L4: E4 → R2 → G → TS1 → TS2 → TT	4

4.1. Validity Verification of the Proposed Method

4.1.1. Validity Verification of the Proposed Method in WIA-PA

According to the dataflow scheduling method, the time slot channel of the corresponding node is configured in WIA-PA by its network manager.

When the converged network is running, the packet sniffer is employed to sniff the package in the dataflows. Some WIA-PA packages are sniffed. These packages constitute a one-way transmission process of dataflow f1 and they are shown in Figure 6.

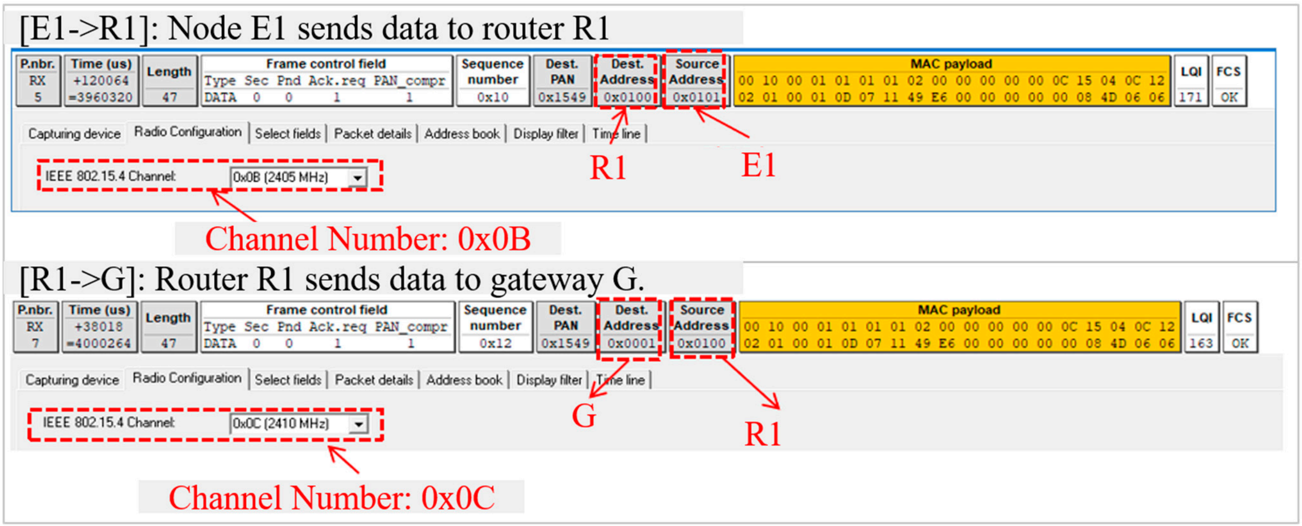

Figure 6. All packages sniffed in the one-way transmission process of a dataflow.

By analyzing the above packages, it can be concluded that the dataflows in WIA-PA are transmitted according to the time slot channel configured in line with the proposed method.

4.1.2. Validity Verification of the Proposed Method in TSN

The four dataflows at the last stop arrive at the TSN terminal and their information is printed as shown in Figure 7. They are in the order of E1, E2, E3 and E4, which proves that the dataflow is scheduled in the order of priority.

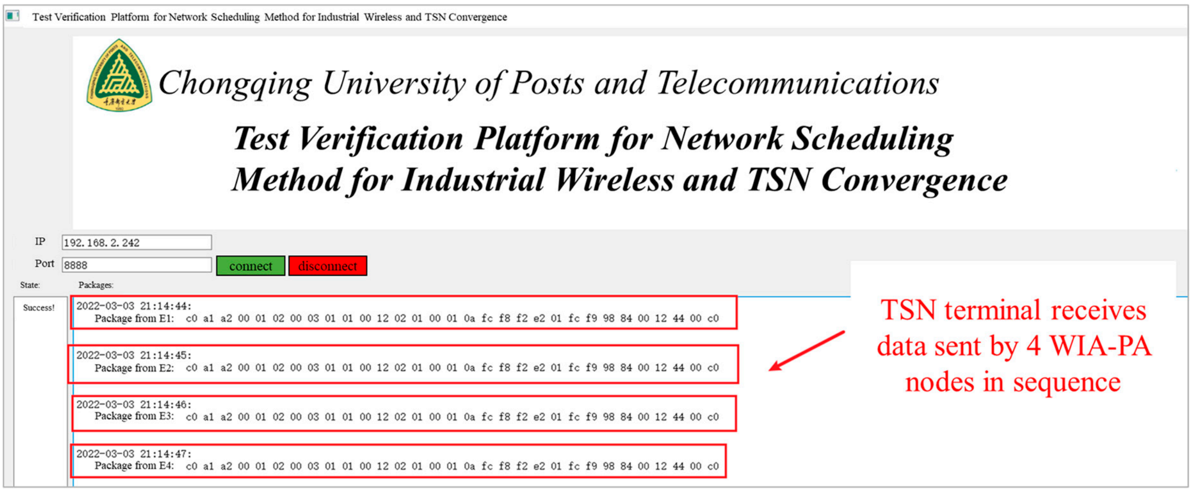

Figure 7. Four dataflows arriving at the TSN terminal.

4.2. Test and Analysis of the End-to-End Transmission Delay

The end-to-end time delays of the converged network with and without the proposed method are shown in Tables 11 and 12.

Table 11. The end-to-end time delay of the converged network without the proposed method.

Dataflows	Average Transmission Delay (ms)	Maximum Transmission Delay (ms)	Minimum Transmission Delay (ms)
f_1	55.872	56.717	54.891
f_2	54.895	55.827	54.103
f_3	54.965	56.215	54.171
f_4	55.782	56.911	55.112

Table 12. End-to-end time delay of the converged network with the proposed method.

Dataflows	Average Transmission Delay (ms)	Maximum Transmission Delay (ms)	Minimum Transmission Delay (ms)
f_1	36.635	36.912	36.316
f_2	36.213	36.335	35.829
f_3	35.916	36.178	35.516
f_4	36.582	36.835	36.102

By comparing the time delays, it can be concluded that the adoption of the proposed method significantly reduces the end-to-end time delay of the converged network.

4.3. Test and Analysis of Jitter

The jitter of the converged network composed of WIA-PA and the TSN with and without the proposed method is shown in Tables 13 and 14.

Table 13. The jitter of the converged network without the proposed method.

Dataflows	Jitter (ms)
f_1	1.826
f_2	1.724
f_3	2.044
f_4	2.044

Table 14. The jitter of the converged network with the proposed method.

Dataflows	Jitter (ms)
f_1	0.596
f_2	0.506
f_3	0.662
f_4	0.733

By comparing the jitter, it can be concluded that the adoption of the proposed method significantly reduces the jitter of the converged network.

5. Conclusions and Future Work

A network scheduling method for the convergence of industrial wireless networks and TSNs is proposed. The method consists of the SDN-based converged network architecture, dataflow priority conversion mechanism and respective dataflow scheduling methods in the industrial wireless network and the TSN. The scheduling method involves inputting the end-to-end device priority and end-to-end delay requirements, as well as obtaining the priority of the converged network dataflow on the industrial wireless network side and the TSN side through the priority mapping rule table. Based on this priority, the scheduling schemes of the dataflow on the wireless and TSN sides are designed, respectively.

The test and verification results revealed that the proposed method lowers the end-to-end time delay from 55 ms to 36 ms and lowers the jitter from 1.8 ms to 0.6 ms in converged networks. The experimental data proved that the proposed method is effective for the integration of the TSN and industrial wireless networks and that it improves the performance of the network.

Although this method has advantages, it still needs improvement. In future studies, the following areas require further research:

1. More extensive measurement and experimentation with different flow parameters and multiple parallel streams are required to demonstrate the proposed method in the future.
2. In this study, to facilitate the verification of slot-hopping channel scheduling on the wireless network side and gating scheduling on the TSN side, only four end-to-end dataflows were set up. In future studies, more nodes should be arranged on the wireless network to increase the number of end-to-end dataflows, bringing the test closer to the real industrial scene.
3. In this study, the scheduling method on the TSN side is based on a time-aware shaper (TAS). In future studies, the scheduling method of the TSN can also be studied by considering aspects of cyclic queuing and forwarding (CQF) and a credit-based shaper (CBS).
4. In this study, the role of industrial wireless networks is played by WIA-PA in the experiment; however, it has certain limitations. In future studies, other industrial wireless networks, such as WirelessHART and 6TiSCH, can be explored for the experiments to improve the universality of the proposed method.

Author Contributions: Conceptualization, M.W.; Methodology, C.L.; Software, S.Y.; Validation, J.W.; Formal analysis, J.W.; Writing—original draft, M.W.; Writing—review & editing, C.L.; Visualization, S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science and Technology project of State Grid Corporation of China, grant number 5108-202222039A-1-1-ZN.

Data Availability Statement: The data can be shared up on request.

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

References

1. Ji, L.; He, S.; Wu, W.; Gu, C.; Bi, J.; Shi, Z. Dynamic network slicing orchestration for remote adaptation and configuration in industrial IoT. *IEEE Trans. Ind. Inform.* **2021**, *18*, 4297–4307. [CrossRef]
2. Wang, Q.; Jiang, J. Comparative examination on architecture and protocol of industrial wireless sensor network standards. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2197–2219. [CrossRef]
3. Jin, X.; Kong, F.; Kong, L.; Liu, W.; Zeng, P. Reliability and temporality optimization for multiple coexisting WirelessHART networks in industrial environments. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6591–6602. [CrossRef]
4. Tanyakom, A.; Pongswatd, S.; Julsereewong, A.; Rerkratn, A. Integration of WirelessHART and ISA100. 11a field devices into condition monitoring system for starting IIoT implementation. In Proceedings of the 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Kanazawa, Japan, 19–22 September 2017.
5. Raviv, L.O.; Leshem, A. Joint Scheduling and Resource Allocation for Packets with Deadlines and Priorities. *IEEE Commun. Lett.* **2022**, *27*, 248–252. [CrossRef]
6. Schüngel, M.; Dietrich, S.; Ginthör, D.; Chen, S.P.; Kuhn, M. Heterogeneous synchronization in converged wired and wireless time-sensitive networks. In Proceedings of the 2021 17th IEEE International Conference on Factory Communication Systems (WFCS), Linz, Austria, 9–11 June 2021.
7. Yu, P.; Yang, M.; Xiong, A.; Ding, Y.; Li, W.; Qiu, X.; Cheriet, M. Intelligent-driven green resource allocation for industrial Internet of Things in 5G heterogeneous networks. *IEEE Trans. Ind. Inform.* **2020**, *18*, 520–530. [CrossRef]
8. Xue, J.; Shou, G.; Li, H.; Liu, Y. Enabling deterministic communications for end-to-end connectivity with software-defined time-sensitive networking. *IEEE Netw.* **2022**, *36*, 34–40. [CrossRef]
9. Cruces, C.; Torrego, R.; Arriola, A.; Val, I. Deterministic hybrid architecture with time sensitive network and wireless capabilities. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018.

10. Wei, M.; Yang, S. A Network Scheduling Method for Convergence of Industrial Wireless Network and TSN. In Proceedings of the 2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea, 3–5 January 2023.
11. Wei, M.; Li, C.; Li, C. An IPv6 internet accessing architecture and approach for industrial wireless network. In Proceedings of the 2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM), Taichung, Taiwan, 4–6 January 2020.
12. Lee, K.; Park, T.; Kim, M.; Chwa, H.S.; Lee, J.; Shin, S.; Shin, I. MC-SDN: Supporting mixed-criticality scheduling on switched-ethernet using software-defined networking. In Proceedings of the 2018 IEEE Real-Time Systems Symposium (RTSS), Nashville, TN, USA, 11–14 December 2018.
13. Jin, X.; Guan, N.; Xia, C.; Wang, J.; Zeng, P. Packet aggregation real-time scheduling for large-scale WIA-PA industrial wireless sensor networks. *ACM Trans. Embed. Comput. Syst. (TECS)* **2018**, *17*, 1–19. [CrossRef]
14. Houtan, B.; Ashjaei, M.; Daneshtalab, M.; Sjödin, M.; Mubeen, S. Work in progress: Investigating the effects of high priority traffic on the best effort traffic in tsn networks. In Proceedings of the 2019 IEEE Real-Time Systems Symposium (RTSS), Hong Kong, China, 1–4 December 2019.
15. Singh, A.P.; Luhach, A.K.; Gao, X.Z.; Kumar, S.; Roy, D.S. Evolution of wireless sensor network design from technology centric to user centric: An architectural perspective. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1550147720949138. [CrossRef]
16. Kala, S.M.; Reddy, M.P.K.; Musham, R.; Tamma, B.R. Interference mitigation in wireless mesh networks through radio co-location aware conflict graphs. *Wirel. Netw.* **2016**, *22*, 679–702. [CrossRef]
17. Pop, P.; Raagaard, M.L.; Gutierrez, M.; Steiner, W. Enabling fog computing for industrial automation through time-sensitive networking (TSN). *IEEE Commun. Stand. Mag.* **2018**, *2*, 55–61. [CrossRef]
18. 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]
19. 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]
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, Nantes, France, 7–9 April 2021.
21. Finzi, A.; Mifdaoui, A.; Frances, F.; Lochin, E. Network calculus-based timing analysis of AFDX networks with strict priority and TSN/BLS shapers. In Proceedings of the 2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES), Graz, Austria, 6–8 June 2018.
22. Houtan, B.; Bergström, A.; Ashjaei, M.; Daneshtalab, M.; Sjödin, M.; Mubeen, S. An automated configuration framework for TSN networks. In Proceedings of the 2021 22nd IEEE International Conference on Industrial Technology (ICIT), Valencia, Spain, 10–12 March 2021.
23. Karakus, M.; Durresi, A. Quality of service (QoS) in software defined networking (SDN): A survey. *J. Netw. Comput. Appl.* **2017**, *80*, 200–218. [CrossRef]
24. Nsiah, K.A.; Alkhouri, K.; Sikora, A. Configuration of wireless TSN networks. In Proceedings of the 2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Dortmund, Germany, 17–18 September 2020.
25. Syed, A.A.; Ayaz, S.; Leinmüller, T.; Chandra, M. Dynamic scheduling and routing for TSN based in-vehicle networks. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Virtual, 14–23 June 2021.
26. Vanhoef, M. Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), New York, NY, USA, 11–31 August 2021.
27. Ojewale, M.A.; Yomsi, P.M.; Nikolić, B. Multi-level preemption in TSN: Feasibility and requirements analysis. In Proceedings of the 2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC), Nashville, TN, USA, 19–21 May 2020.