Hindawi
Wireless Communications and Mobile Computing
Volume 2024, Article ID 8882006, 9 pages
https://doi.org/10.1155/2024/8882006

Research Article

Hierarchical Cross Traffic Scheduling Based on Time-Aware Shapers for Mobile Time-Sensitive Fronthaul Network

Chang Liu,[1] **Yue Hong**,[2] **Jin Wang**,[1] **Chang Liu Sr.**,[1] **Li Tian**,[1] **and Jiangpei Xu**[1]

[1]*State Grid Hubei Electric Power Research Institute, Wuhan, China*
[2]*State Grid Hubei Electric Power Co., Ltd., Wuhan, China*

Correspondence should be addressed to Chang Liu; liuchang.sgcc@foxmail.com

Received 19 July 2023; Revised 15 March 2024; Accepted 2 April 2024; Published 15 April 2024

Academic Editor: Bithas Petros

To solve the problem of jitter and low network throughput caused by the impact of background flows on IQ traffic in mobile fronthaul network, this paper proposed a new scheduling model for background flows, named hierarchical crossover traffic scheduling mechanism based on time-aware shaper (HC-TAS) by improving the traditional counterpart. Then, in this new model, we designed an inbound scheduling algorithm based on frame length matching and an outbound scheduling algorithm based on queue status, making sure that smaller data frames will not be blocked by large data frames. This greatly improves the utilization of timeslots in the scheduling process and reduces the jitter impact of background flows. To verify its performance, we conducted experiments in a simulated fronthaul network conforming to IEEE 802.1CM. The experimental results show that, under the condition that the jitter is guaranteed to be zero, compared with two mainstream scheduling schemes, Comb-FITting and TAS + Preemption, our proposed scheme can achieve lower maximum end-to-end delay and higher link utilization. The proposed HC-TAS meets the requirements of low jitter and high bandwidth utilization in 5G fronthaul network, and the research results provide a technical basis for the application and development of general time-sensitive networks as well.

1. Introduction

In time-sensitive networks (TSN), traffic scheduling is one of the key technologies. At present, the TSN Working Group has completed the standardization of a series of traffic scheduling mechanisms, including IEEE 802.1Qav (credit value-based shaping), IEEE 802.1Qbv (gated scheduling), IEEE 802.3br and IEEE 802.1Qbu (frame preemption), IEEE 802.1Qch (cyclic queue forwarding), and IEEE 802.1Qcr (asynchronous shaping). Among them, the time-aware shaper (TAS) is the most promising. It adopts the gate control entries scheduling method, which is similar to the time division multiple access technology [1], and causes a very low jitter.

The traditional mobile fronthaul network is mainly responsible for the communication between remote radio units (RRUs) and building baseband units (BBUs). The cloud radio access network is used to virtualize the functions of BBUs so that they can transmit In-phase and Quadrature modulation streams (IQ streams) to multiple RRUs. BBU is divided into a distributed unit (DU) and a centralized unit (CU). From the RRU to the core network, it is also divided into fronthaul transmission, mid-haul transmission, and backhaul transmission. In 5G communication service [2], RRU and DU are connected through a fronthaul network. This architecture realizes the on-demand configuration of mobile fronthaul network resources through flexible and configurable CU/DU function separation, making it better adapt to personalized requirements of different scenarios. But at the same time, it puts forward strict delay requirements in such a complex traffic environment. The current cellular-based network deployment cannot meet the ultra-low delay, low jitter, and high-capacity requirements brought by the BBU division [3–5].

In this paper, we propose a hierarchical crossover time aware shaper (HC-TAS) traffic scheduling mechanism for 5G mobile fronthaul networks to achieve low latency, low jitter, and high link utilization. The main contributions are summarized as follows:

(i) On the basis of the traditional TAS, we add a cross-two-level queue scheduling for background flows to ensure normal transmission of different types of traffic.

(ii) An inbound scheduling algorithm and outbound scheduling algorithms are designed to effectively improve slot utilization and reduce the average latency of overall traffic.

2. Related Works

The TSN scheduling mechanisms are roughly divided into two categories, namely, synchronous traffic scheduling and asynchronous traffic scheduling [6–16]. At present, synchronous traffic scheduling mainly focuses on gate control lists (GCLs) and TAS-based joint routing. Asynchronous traffic scheduling includes credit-based shaper (CBS), asynchronous traffic shaper (ATS), and frame preemption scheduling.

2.1. Synchronous Traffic Scheduling

(1) TAS: TAS-based traffic scheduling is based on IEEE 802.1Qbv. New researches on TAS focus on timeslot calculation and allocation of control data traffic (CDT). Through reasonable modeling and algorithm design, it can ensure the boundedness of CDT delay and reduce the scheduling delay and jitter. For example, Chang et al. [17] abstracted timeslot allocation as a combinatorial optimization problem and calculated slot allocation through satisfiability modulo calculation of static scheduling. Similar to other optimization algorithms, such as integer linear programing and genetic algorithms, it is of large complexity and small scalability and difficult to adapt to medium and large-size networks.

The introduction of a guard band in TAS leads to low bandwidth utilization. To handle that, Zhao et al. [18] mapped the slot allocation problem to no-wait job scheduling and proposed a tabu search algorithm and a schedule compression technology to reduce the guard bandwidth. Chitimalla et al. [19] proposed a time slot allocation algorithm called Comb-FITting (C-FIT) for low-jitter fronthaul network scheduling. Kim et al. [20] proposed a TAS-adaptive bandwidth-sharing mechanism to improve link utilization. However, these researches have not substantially solved the problem of low bandwidth utilization caused by guard bands.

(2) Joint route scheduling: By expanding TAS, Li et al. [21] proposed a joint routing method to optimize scheduling. It has higher network traffic schedulability and greatly reduces link load as well as end-to-end average delay of flows. Based on Li's work, Bush [22] introduced hops to further reduce the average delay. Krolikowski et al. [23] optimized the routing path of AVB flow using TAS to schedule CDT flow. However, these schemes also have large guard bands, so the path selection mechanism still has a great negative impact on scheduling delays.

2.2. Asynchronous Traffic Scheduling

(1) CBS: CBS schedules traffic by setting a credit value. Using network calculus, Yang et al. [24] deduced the waiting time of service according to the upper limit of CBS's credit value to calculate the bounded delay. Feng et al. [25] designed a specific shaping mechanism based on flow QoS by combining the strict priority algorithm and the CBS algorithm, in which the CBS algorithm is responsible for shaping ordinary flows. Maile et al. [26] proposed a scheme by combining CBS with TAS, which effectively meets the QoS requirements of industrial automation scenarios. Though these scheduling schemes can effectively reduce the delay, there exist jitters of high-priority streams caused by the continuous transmission of low-priority streams.

(2) ATS: ATS defines the priority of flows based on the urgency of the event. It uses rate control strategies for parallel scheduling of high-priority flows to improve network bandwidth utilization. Zhang et al. [27] proposed a deterministic transmittable time-based asynchronous scheduler based on an urgency-based scheduler algorithm. The high-priority flow is scheduled with a rate control strategy, which improves bandwidth utilization and reduces jitter. However, in the case of multirate traffic, its scheduling complexity is greatly increased.

(3) Frame preemption: Frame preemption is to prevent the continuous transmission of low-priority traffic from hindering the timely transmission of high-priority traffic. However, it is not able to prevent the influence of the background flow and eventually causes greater jitter [28]. At present, frame preemption-based scheduling is combined with other scheduling algorithms. For example, Bello et al. [29] proposed to combine the TAS mechanism with frame preemption to reduce the delay of high-priority flows and improve link utilization. However, the extra delay caused by the frame preemption mechanism will bring jitter to high-priority flows.

3. Proposed Technique

According to the enhanced common public radio interface specification, there are three types of flows in TSN networks:

(1) IQ flow: It is the main traffic in the form of in-phase and quadrature modulation (IQ) in fronthaul transmission, and it is usually of constant rate and periodicity.

(2) C&M: It refers to control and management (C&M) data. It is of an unsteady rate with no specific requirement for delay and no deterministic requirement for jitter.

(3) Synchronous information flow: It is used for time synchronization of CPRI frames. It is of constant rate with low delay and no deterministic requirement for jitter.

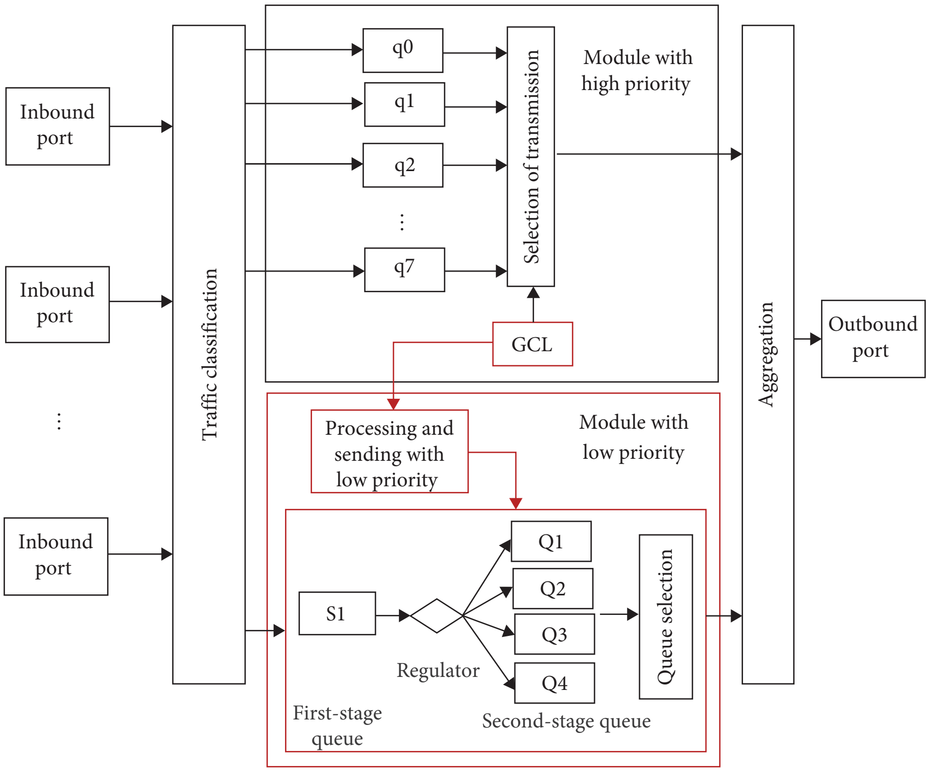

Figure 1: Structure of hierarchical cross traffic scheduling mechanism.

Usually, IQ flows are classified as high priority, C&M flows, and synchronous information flows are classified as low priority or background flows. The principle of traffic scheduling is to prioritize the transmission of high-priority traffic and, at the same time, to make background flow utilize the remaining bandwidth as much as possible.

Since the number of traditional TAS scheduling queues is eight, with one flow corresponding to one queue, it is not enough for a fronthaul network with a complex traffic environment. Besides, flows with multiple functions coexisting in one queue affect normal scheduling among them. This is because background flows will affect the throughput or jitter of high-priority flows due to their nonconstant rate characteristics.

To effectively and reasonably schedule background flows, we improve the TAS scheduling mechanism to introduce a hierarchical cross-scheduling model, shown in Figure 1. In Figure 1, all incoming queues first enter the "Traffic Classification" module, then are switched to the high-priority module based on the TAS or the low-priority module based on our proposed mechanism, respectively. The high-priority module includes a high-priority queue, a GCL, and a selection transmission module. The low-priority module includes a first-stage queue, a regulator, a second-stage queue, and a queue selection module. In Figure 1, q0, q1,…, and q7 represent a total of 8 levels of high-priority flows. All other traffic is of low priority, which will be further placed in four parallel waiting queues: Q1, Q2, Q3, and Q4.

Compared to the traditional TAS mechanism, the proposed HC-TAS has at least three advantages: First, there are more high-priority queues, so the schedulability of high-priority traffic scheduling is enhanced, and the delay and jitter are reduced. Second, different queues can be selected while reducing the impact of large bursts. By reasonably adjusting the forwarding order of data frames, bandwidth utilization can be improved, and the impact on high-priority traffic can be reduced. Third, to avoid the blocking effect caused by "first come, first serve" frames, we use a hierarchical queue to schedule frames according to our designed inbound queue and outbound queue algorithms in the low-priority processing module. By doing so, it can improve bandwidth utilization.

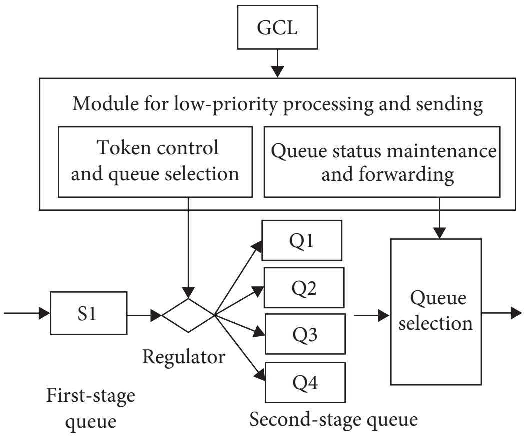

Figure 2: Detailed structure of low-priority module.

The inbound queue and outbound queue algorithms work in the low-priority processing module and are marked in red in Figure 1. They adjust the forwarding order of background flows. Figure 2 shows the internal structure of the

```
Input:
{Slot1, Slot2,…, Slotn} Idle timeslots obtained from GCL list. Each timeslot includes end time, start time, and timeslot length;
Set S of frames to be transmitted in S1 queue;
output:
{frame.Q } Selected data frame entering queue Q
{W.start} Transmission time of data frame
WHILE S is not empty THEN
  IF the number of frame in S is 1
    Calculate the earliest Slot that can transmit the frame as W.start
    Update Slot
  ELSE
    Frame_number←get_Frame_number(S);
    // Determine the number of frames to be scheduled according to the number of waiting frames in S

      FOR ($i = 1$; i++; $i <$ Frame_number)
        Calculate the Slot set that can transmit the all frame;
      END FOR
      FOR ($i = 1$; i++; $i <$ Frame_number)
        $\text{Slot}_{\text{M}}$ ← get_feasible_slot(Slot)
        //Select the largest data frame to be scheduled, and select the transmissible slot with the most matching length
        Update Slot
        According $\text{Slot}_{\text{M}}$ to get W.start // W.start of each frame can be calculated
      END FOR
 // The above is to select the timeslot and get W.start. Next, determine the incoming queue Qm
      FOR ($i = 1$; $i$++; $i <$Q_number) // The secondary queue contains four queues. Now, determine Q.end.
        IF Q.end <time_now
      Q.end = time_now
        END IF
      END FOR
      FOR ($i = 1$; $i$++; $i <$ Frame_number)
      Select the queue of min (W.start-Q.end) as frame.Q; // Determine the incoming queue for each frame
      Q.end = W.start + frame; //Update Q.end
      END FOR
    END IF
END WHILE
```

Algorithm 1: Inbound Queue scheduling algorithm based on frame length matching.

low-priority processing module. Taking the GCL list and background data frames in buffer S1 as input, the submodule "token control and queue selection" uses an inbound queue algorithm to select a queue that can maximize the utilization of the available time slot. The regulator is used to process burst flows. The submodule "queue status maintenance and forwarding" uses an outbound scheduling algorithm to select a data frame from one of the queues (Q1–Q4) for forwarding with polling mode.

This mechanism performs scheduling in the low-priority queue according to the GCL list, frame size, and the queue status; thus, it conforms to the nonconstant rate characteristics of low-priority traffic. The inbound queue scheduling is based on frame length matching, and its details are introduced in Algorithm 1. The outbound queue scheduling algorithm is based on queue status, and its details are introduced in Algorithm 2. Key parameter symbols in Algorithm 1 and Algorithm 2 are shown in Table 1.

4. Experimental Results and Analysis

For TSN in 5G networks, there are typically three scheduling modes in the mixed environment of nonconstant rate flows, and IQ flows, namely, ATS + Preemption, TAS (including guard bandwidth), and TAS + Preemption. When IQ flow is slightly complex, ATS + Preemption has a large jitter. Therefore, we only compare our proposed HC-TAS scheme with the C-FIT (one mainstream TAS scheme) and the TAS + Preemption scheme. Below, we will refer to them as scheme HC-TAS, C-FIT, and Preemption, respectively.

```
Input:
{frame.Q} Queue that data frames need to enter
{W.start} Transmission time of each data frame
{slot} Time slot obtained by GCL
Output:
Q.status    Forwarding status of the queue
WHILE Q is not empty THEN
      // whether queue Q is empty, i.e., whether there is data waiting to be forwarded
  FOR ($i = 1$; $i$++; $i <= 4$) //Update Q.start
    Qi.start ← W.start_first (Qi)
        // W.start of the first data frame to be transmitted in each queue is Q.start
  END FOR
  $Q_k$ ← min (Qi.start); // Select the queue with the lowest value of Qi.start
  Calculate the number of frames that need to be transmitted continuously in $Q_k$ according to frame.Q.
  $Q_k$.end ← $Q_k$_finish ($Q_k$)
   //Determine $Q_k$ according to the number of frames that can be transmitted continuously
  WHILE (t_now >= Qk.start)
      $Q_k$.status = 1; // The queue status is 1, and forwarding is performed
      IF (t_now >= $Q_k$.end)
        $Q_k$.status = 0;
        break;
      END IF
   Update Q.start
END WHILE
```

Algorithm 2: Outbound algorithm based on queue status.

Table 1: Meaning of related parameters.

Parameter	Meaning
Slot	Idle timeslot obtained from GCL list
S	Set of frames to be transmitted in S1 queue
Q.end	Time when the last frame in Q queue has been transmitted
W.start	Transmission time of data frame
Q.start	Transmission time of the first data frame in Q (determined according to W.start)
frame.Q	Queue Q that data frame selects to enter in

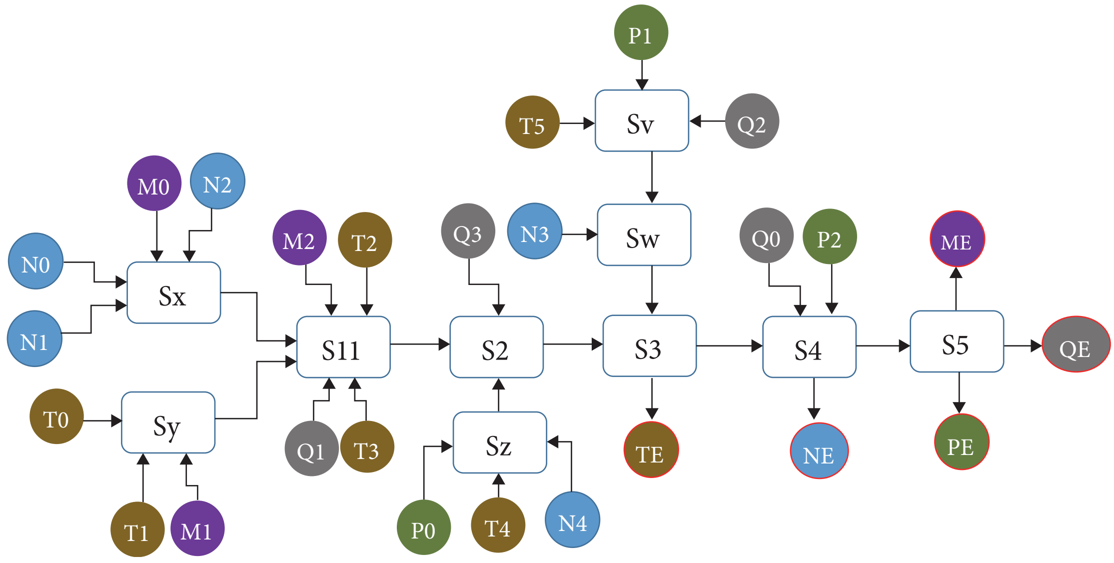

Figure 3: Hybrid simulation topology.

Table 2: Information of the flows.

Flow	Source node	Destination node
1	N0 ->	NE
2	N1 ->	
3	N2 ->	
4	N3 ->	
5	N4 ->	
6	P0 ->	PE
7	P1 ->	
8	P2 ->	
9	Q0 ->	QE
10	Q1 ->	
11	Q2 ->	
12	Q3 ->	
13	T0 ->	TE
14	T1 ->	
15	T2 ->	
16	T3 ->	
17	T4 ->	
18	T5 ->	
19	M0 ->	ME
20	M1 ->	
21	M2 ->	

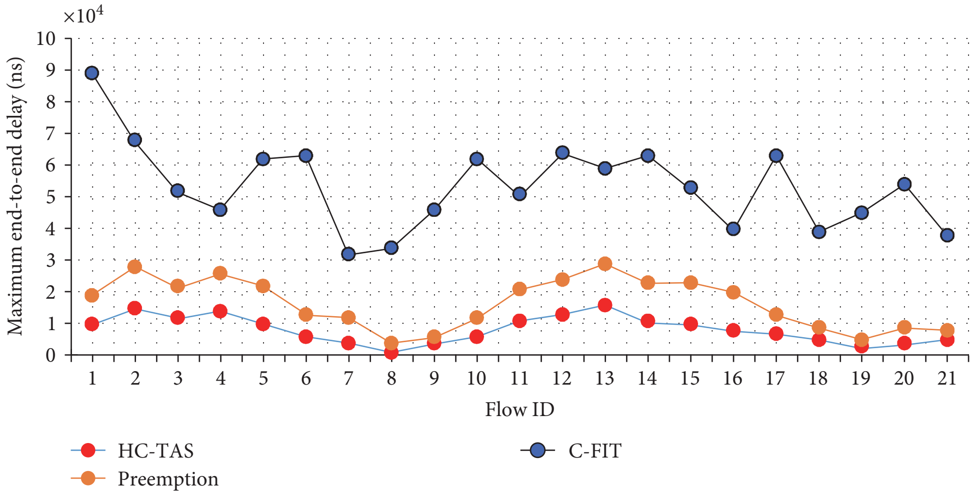

Figure 4: Maximum end-to-end delay.

We simulate a fronthaul network conforming to IEEE 802.1CM. The topology is shown in Figure 3. In Figure 3, a circle without an outer contour represents the node that generates a flow, while a circle with the same filling color and a red outer contour represents the receiving node of the flow. For example, P0, P1, and P2 are sending nodes. PE is their receiving node. Table 2 summarizes all the sending nodes (source nodes) and corresponding receiving nodes (destination nodes). All rectangles designate switches, among which S1–S5 are the central switches. The topology looks like a hybrid tree, consisting of a trunk (central switches), branches (Sv, Sw, Sx, Sy, and Sz), and leaves (nodes).

Two types of flows are available in the network, namely, CBR-type IQ flow and VBR-type background flow. Each sending node generates two types of streams in the same direction. Each link is a 10 Gbps full-duplex. The IQ flow is 1 Gbps, the frame sizes are 1,500 bytes, and the transmission interval is 12,000 ns.

4.1. Experimental Results of End-to-End Delay. Figure 4 depicts the maximum end-to-end delay of all IQ flows.

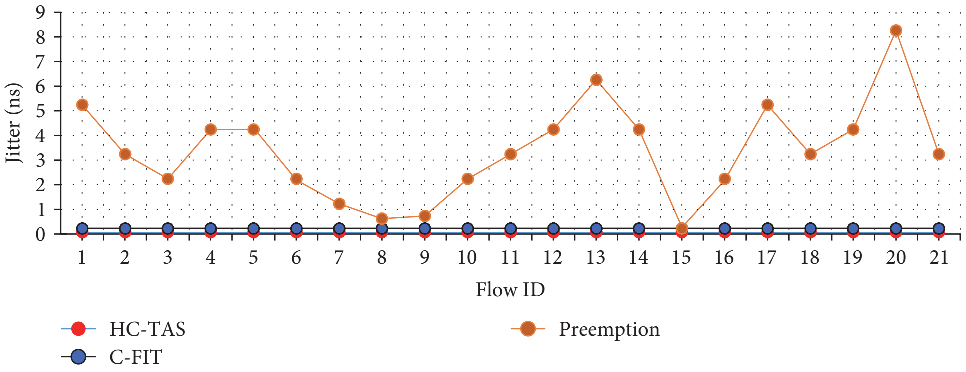

FIGURE 5: Jitter of flows.

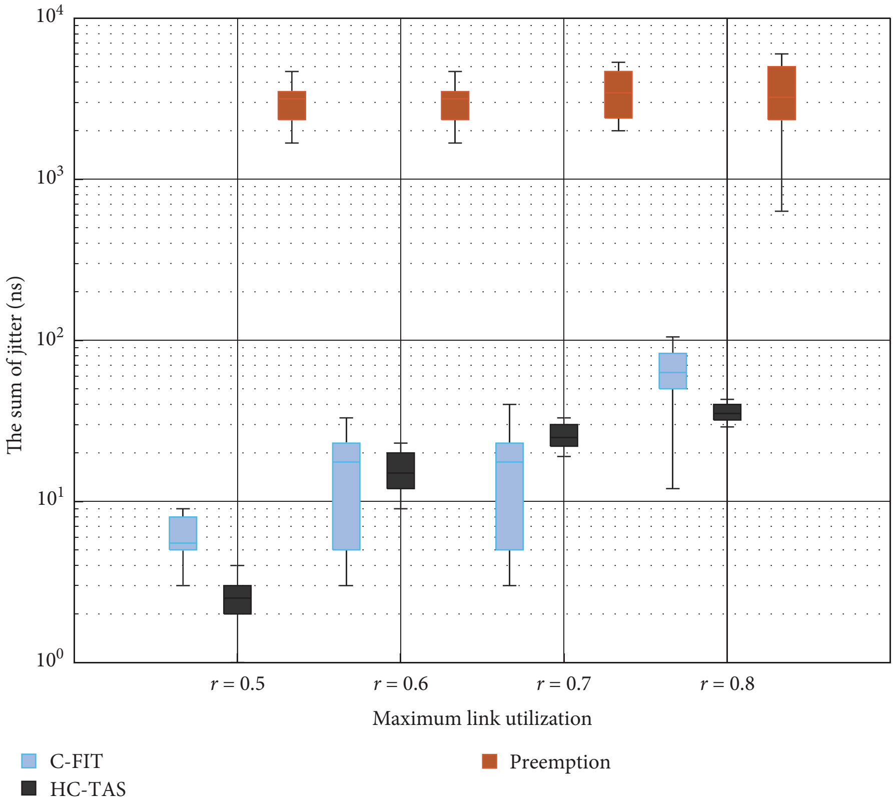

FIGURE 6: The sum of jitter distribution versus maximum link utilization.

Our HC-TAS has the best performance. The HC-TAS usually requires more resources and might get blocked longer at one hop, but it is more advantageous in a decentralized network environment with more stream aggregation. The preemption scheme blocks a large amount of IQ data due to unpredictable preempt action, so it causes the increase of end-to-end delay of IQ flows. Since the C-FIT ensures low jitter by sacrificing latency performance, its maximum end-to-end delay of all IQ flows is longer.

4.2. Experimental Results of Jitter. Figure 5 shows the jitter of each flow. It is seen that both HC-TAS and C-FIT can reach zero jitters. Figure 5 also shows that the performance of the Preemption scheme is unstable, *i.e.*, the jitter of each flow is different due to unstable transmission and unpredictable schedule action. Figures 4 and 5 show that our HC-TAS not only provides relatively low latency but also achieves lower jitter.

4.3. Experimental Results of Link Utilization. Now, let's assess the relationship between maximum link utilization and jitter. In this experiment, there are more than 2,000 IQ data flow rates, randomly generated from 0.5 to 2 Gbps. Figure 6 illustrates the relation between jitter and maximum link utilization (note that the vertical axis is a logarithmic scale). As the maximum link utilization increases, the jitter of all three schemes increases. However, for the same maximum link utilization, the Preemption scheme has much more severe jitter. Both HC-TAS and C-FIT have low jitter, and the sum of jitter of HC-TAS remains more concentrated than C-FIT.

5. Conclusions and Future Work

To improve the schedulability and bandwidth utilization of background flows in TSN's fronthaul network, we first proposed a hierarchical cross-traffic scheduling mechanism (HC-TAS) by modifying the structure of traditional TAS. Then, for the background flow, we designed an inbound scheduling algorithm based on frame length matching and an outbound scheduling algorithm based on queue status. The performance of the HC-TAS scheduling mechanism is verified through the simulation experiments. Compared with the mainstream schemes, C-FIT and Preemption, our proposed HC-TAS not only provides relatively low latency but also achieves lower jitter, and its sum of jitter remains more concentrated.

However, some future works still need to be done. The two queues in our proposed HC-TAS model are used to cache background flows. Although the C&M flow and synchronous information flow in the mobile fronthaul network belong to the same background flow, there are still differences. For example, although the C&M flow is not a high-priority flow, it includes a medium-priority part and a low-priority part. How to carry out more detailed and effective differential scheduling will be our future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the Science and Technology Project of the State Grid Corporation of China, with grant number 5108-202222039A-1-1-ZN.

References

[1] Y. Yu, S. C. Liew, and T. Wang, "Non-uniform time-step deep Q-network for carrier-sense multiple access in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2848–2861, 2021.

[2] L. B. D. Silveira, H. C. de Resende, C. B. Both, J. M. Marquez-Barja, B. Silvestre, and K. V. Cardoso, "Tutorial on communication between access networks and the 5G core," *Computer Networks*, vol. 216, Article ID 109301, 2022.

[3] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, and G. Fettweis, "Fronthaul and backhaul requirements of flexibly centralized radio access networks," *IEEE Wireless Communications*, vol. 22, no. 5, pp. 105–111, 2015.

[4] N. J. Gomes, P. Sehier, H. Thomas et al., "Boosting 5G through ethernet: how evolved fronthaul can take next-generation mobile to the next level," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 74–84, 2018.

[5] I. A. Alimi, A. L. Teixeira, and P. P. Monteiro, "Toward an efficient C-RAN optical fronthaul for the future networks: a tutorial on technologies, requirements, challenges, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 708–769, 2018.

[6] G. Liu, H. Zhang, R. Shang, Z. Wu, and Q. Song, "Hierarchical optimization scheduling of active demand response for distribution networks in 5G base stations," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 8666749, 15 pages, 2022.

[7] A. Singhal, S. Varshney, T. A. Mohanaprakash et al., "Minimization of latency using multitask scheduling in industrial autonomous systems," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1671829, 10 pages, 2022.

[8] M. N. M. Haque, Y.-D. Lee, and I. Koo, "Deep learning-based scheduling scheme for IEEE 802.15.4e TSCH network," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 8992478, 17 pages, 2022.

[9] D. Liu, X. W. Liu, and C. L. Ma, "Multiconstraint-oriented tactical communication network planning problem," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1857304, 9 pages, 2022.

[10] Ó. Seijo, J. A. López-Fernández, and I. Val, "w-SHARP: implementation of a high-performance wireless Time-Sensitive Network for low latency and ultra-low cycle time industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3651–3662, 2021.

[11] Y. Yuan, X. Cao, Z. Liu, C. Chen, and X. Guan, "Adaptive priority adjustment scheduling approach with response-time analysis in Time-Sensitive Networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8714–8723, 2022.

[12] Z. Pang, X. Huang, Z. Li et al., "Flow scheduling for conflict-free network updates in time-sensitive software-defined networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1668–1678, 2021.

[13] Y. Liu, Y. Zhou, J. Yuan, and L. Liu, "Delay aware flow scheduling for time sensitive fronthaul networks in centralized radio access network," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2992–3009, 2020.

[14] F. Luque-Schempp, L. Panizo, M.-D.-M. Gallardo, P. Merino, and J. Rivas, "Toward zero touch configuration of 5G non-public networks for time sensitive networking," *IEEE Network*, vol. 36, no. 2, pp. 50–56, 2022.

[15] Q. Lyu, Y. Zhou, Q. Fan et al., "JOSP: joint optimization of flow path scheduling and virtual network function placement for delay-sensitive applications," *Mobile Networks and Applications*, vol. 27, pp. 1642–1658, 2022.

[16] C. Zhang, Y. Wang, R. Yao et al., "Packet-size aware scheduling algorithms in guard band for time sensitive networking," *CCF Transactions on Networking*, vol. 3, pp. 4–20, 2020.

[17] S.-H. Chang, H. Chen, and B.-C. Cheng, "Time-predictable routing algorithm for Time-Sensitive Networking: schedulable guarantee of Time-Triggered streams," *Computer Communications*, vol. 172, pp. 183–195, 2021.

[18] L. Zhao, P. Pop, and S. Steinhorst, "Quantitative performance comparison of various traffic shapers in Time-Sensitive Networking," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2899–2928, 2022.

[19] D. Chitimalla, K. Kondepu, L. Valcarenghi, M. Tornatore, and B. Mukherjee, "5G fronthaul–latency and jitter studies of CPRI

over ethernet," *Journal of Optical Communications and Networking*, vol. 9, no. 2, pp. 172–182, 2017.

[20] M. Kim, D. Hyeon, and J. Paek, "eTAS: enhanced time-aware shaper for supporting nonisochronous emergency traffic in Time-Sensitive Networks," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 10480–10491, 2022.

[21] J. Li, S. K. Bose, and G. Shen, "Cooperative resource scheduling for Time-Sensitive Services in an integrated XGS-PON and Wi-Fi 6 network," *IEEE Communications Letters*, vol. 26, no. 6, pp. 1338–1342, 2022.

[22] S. F. Bush, "Toward efficient Time-Sensitive Network scheduling," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 1830–1842, 2022.

[23] J. Krolikowski, S. Martin, P. Medagliani et al., "Joint routing and scheduling for large-scale deterministic IP networks," *Computer Communications*, vol. 165, pp. 33–42, 2021.

[24] D. Yang, K. Gong, J. Ren, W. Zhang, W. Wu, and H. Zhang, "TC-Flow: chain flow scheduling for advanced industrial applications in Time-Sensitive Networks," *IEEE Network*, vol. 36, no. 2, pp. 16–24, 2022.

[25] Z. Feng, Q. Deng, M. Cai, and J. Li, "Efficient reservation-based fault-tolerant scheduling for IEEE 802.1Qbv Time-Sensitive Networking," *Journal of Systems Architecture*, vol. 123, Article ID 102381, 2022.

[26] L. Maile, K.-S. J. Hielscher, and R. German, "Delay-guaranteeing admission control for Time-Sensitive Networking using the credit-based shaper," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1834–1852, 2022.

[27] X. Zhang, W. Hou, L. Guo, Q. Zhang, P. Guo, and R. Li, "Joint optimization of latency monitoring and traffic scheduling in software defined heterogeneous networks," *Mobile Networks and Applications*, vol. 25, pp. 102–113, 2020.

[28] Specification and Management Parameters for Interspersing Express Traffic, "IEEE standard 802.1802.3br," 2024, [Online April], https://ieeexplore.ieee.org/document/7592835/.

[29] L. L. Bello, M. Ashjaei, G. Patti, and M. Behnam, "Schedulability analysis of Time-Sensitive Networks with scheduled traffic and preemption support," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 153–171, 2020.