# Low-Latency Method and Architecture for 5G Packet-Based Fronthaul Networks

Yang Liu, Zunwen He, Yan Zhang, and Wancheng Zhang(✉)

Beijing Institute of Technology, Beijing 100081, China
zhangwancheng@bit.edu.cn

**Abstract.** The design of fronthaul link has become a challenging task in the 5th Generation Mobile Network (5G). In this paper, in order to reduce 5G fronthaul delay and jitter, a multi-thread scheduling receiving method based on interrupt and polling mode is introduced. Considering diverse application scenarios in 5G, we present a new scalable and flexible 5G fronthaul architecture to manage high data traffic flows efficiently. In the end, a hardware system is built to verify the method and fronthaul architecture. Experiments show that delay and reliability can meet the requirements of 5G fronthaul design. Besides, the designed architecture can be used in other distributed high-speed transmission systems to increase their flexibility and efficiency.

**Keywords:** 5G · Delay requirement · Multi-thread scheduling method · Fronthaul architecture

## 1 Introduction

Nowadays, with a continuous increase in data traffic and users, 5G has become a hot research topic. Compared with 4G Long-term Evolution (LTE) networks, 5G will provide three main applications: Enhanced Mobile Broadband (eMMB), Ultra-Reliable Low-Latency Communications (uRLLC), and Massive Machine-Type Communication (mMTC), which can support higher data rate, lower latency, and wider access [1]. Therefore, the strict requirements of time delay, reliability, bandwidth, and diversity service are huge challenges to mobile communication technology.

When it comes to solutions, one popular implementation is the Cloud Radio Access Network (C-RAN) architecture [2]. Unlike traditional radio access networks, C-RAN separates remote radio unit (RRU) from base band units (BBUs) to implement flexible and reconfigurable network, and the BBU is divided into a central unit (CU) and many distributed units (DUs) to flexibly adapt to network traffic changes in 5G. The BBU pool can share and dynamically allocate BBUs, offering energy and multiplexing gain.

Although C-RAN architecture can meet various needs of 5G well, some key obstacles still need be taken into consideration for its application. One is the fronthaul link, a low-latency and high-speed communication link between the RRU and BBU pool. Because of the centralization of BBUs, requirements of capacity and delay over the fronthaul link are becoming stricter, which is a key obstacle in the deployment of C-RANs. In 4G LTE, one RRU is connected to one BBU, so the bandwidth and data rate requirements are easy to meet. However, in C-RAN architecture, multiple RRUs


B. Li et al. (Eds.): IoTaaS 2020, LNICST 346, pp. 270–281, 2021.
https://doi.org/10.1007/978-3-030-67514-1_22

connected to one BBU multiple links are combined into one link, raising a huge challenge to 5G fronthaul bandwidth and delay [3]. At the same time, it is also difficult to manage multiple high data traffic flows efficiently. To solve these problems, several works have been carried out on the 5G fronthaul network. In [4], the authors discussed bandwidth usage of the fronthaul link and proposed two methods to reduce the use of bandwidth. The resource scheduling algorithm was studied in [5–8] based on multiple traffic to improve the delay of fronthaul. An erasure coding method was proposed in [9] for MAC frames to reduce delay. Authors in [10] and [11] paid attention to delay and jitter and proposed new fronthaul architecture.

However, most of studies (e.g., [5, 6, 8, 9] and [11]) were based on theoretical simulation and might ignore some influencing factors like clock synchronization and rate matching, which can occur during actual deployment in the 5G fronthaul link. Furthermore, [7–11] concerned delay in the multiple traffic scheduling algorithm and transmission process, but they did not consider delay optimization at the receiving side. In [12], the author proposed a poll-mode method at the receiving side to reduce delay. However, this method in [12] requires the central processing unit (CPU) to always listen, therefore CPU has no time to handle other things. In the 5G mMTC scenarios, this method cannot meet the requirements of intermittent data transmission and low power consumption. Moreover, these studies did not consider the effects of delay and packet loss rate simultaneously. In addition, aforementioned methods focus on a specific scene like optical transport network (OTN) or passive optical network (PON), so they are not able to be generalized to more application scenes. There remains strong demands to design the fronthaul to meet the requirements of delay and loss rate at the receiving side and to adapt to diverse scenes in 5G.

In this paper, a multi-thread scheduling receiving method is proposed, which combines the interrupt and polling methods. This method is suitable for diverse scenes because of generality at the receiving side. Based on this method, a new 5G fronthaul system architecture is designed. Based on the proposed method and architecture, a compact baseband processing unit is implemented, which can support Ethernet-based fronthaul and 10 Gigabit Ethernet interface. With radio frequency (RF) board as well as power and controller board integrated together, the baseband processing unit can support high energy efficiency and lightweight design.

In the remainder of this paper, Section 2 introduces the 5G fronthaul architecture. In Sect. 3, details of the receiving method and new fronthaul architecture are described. Section 4 introduces hardware implementation based on the proposed method and architecture. In Sect. 5 performance of our method is evaluated. Finally, conclusions are drawn in Sect. 6.

## 2 Fronthaul Architecture

Currently, the fronthaul link is realized through the Common Public Radio Interface (CPRI) by transporting IQ data samples. With the applications of 5G techniques like massive multi input multi output (MIMO), both the bandwidth and the number of antennas become very large. The CPRI interface bandwidth is proportional to numbers of antennas. For example, Table 1 shows the CPRI line rate requirement for

20 MHz/100 MHz bandwidth with different antenna numbers. Using the CPRI interface limits the development of centralized RAN because of the sharply increasing rate. When a large number of BBUs form a BBU pool, it needs huge fronthaul bandwidth, so CPRI is no longer applicable because of implementation difficulty. Furthermore, the transmission efficiency of CPRI is not high. CPRI transport is designed with a constant transmission rate and has nothing to do with actual network traffic. Even when there is no traffic on the network, the CPRI link rate is still fixed and not flexible. In order to promote the evolution of 5G C-RAN, the fronthaul interface needs to be redesigned to meet more stringent requirements of delay, data rate, and flexibility.

**Table 1.** CPRI line rate requirement.

| Channel BW/MHz | 2T2R /Gbps | 4T4R /Gbps | 8T8R /Gbps | 64T64R /Gbps |
|---|---|---|---|---|
| 20 | 2.4576 | 4.9152 | 9.8304 | 78.6432 |
| 100 | 12.288 | 24.576 | 49.152 | 393.216 |

As the fronthual link rate requirement continues to increase, the fronthaul architecture is changing. The architectures of traditional RAN, C-RAN, and 5G C-RAN is shown in Fig. 1. In the 5G C-RAN network, BBUs are divided into a CU and many DUs which connect RRUs over the fronthaul link to support flexible network architecture. It can be seen from the figure that receiving method is used for DU to reduce delay and fronthaul architecture design is between DU and RRU in 5G C-RAN to increase flexibility.

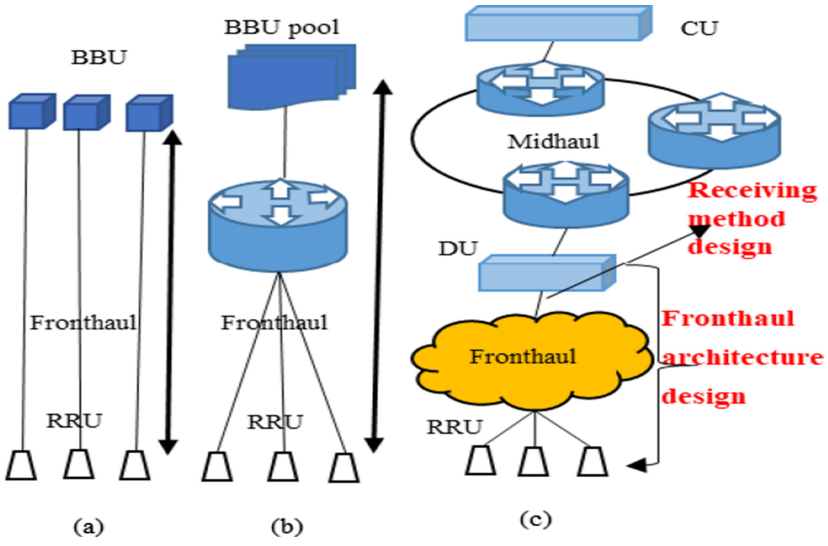


**Fig. 1.** Architectures of RAN, C-RAN and 5G C-RAN.

# 3 Fronthaul Design

## 3.1 Receiving Method Design

Figure 2 shows a general flow of receiving a packet. For high-speed data streams, not only CPU interrupt will be triggered frequently, but CPU usage will be too high. CPU may lose packets with a high error rate due to slow CPU processing. New methods need to be proposed to meet requirements of delay and accuracy [12].

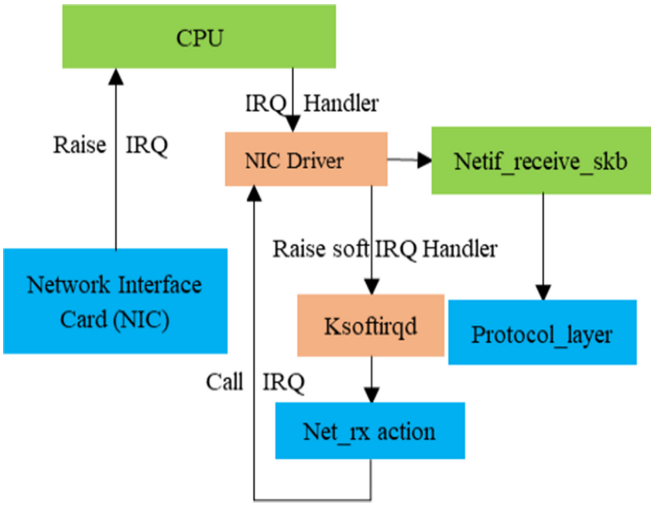


**Fig. 2.** General receiving procedure.

In this paper, a new method based on multiple-thread mode is proposed to solve above problems firstly. Delay and accuracy can be effectively improved by using multiple CPUs to simultaneously process data packet. Moreover, multi-queue mode, which receives packet data simultaneously, can be adopted to reduce receiving pressure. However, for resource-constrained BBU pools, an increasing number of threads not only reduces resource utilization but also affects reception performance. Thus, it is important to get appropriate numbers of threads and queues.

The flow diagram depicted in Fig. 3 illustrates the procedure of the receiving method with multi-thread interrupt mode. The details of the parameters are given as follows. T_num: thread numbers used in processing; Q_num: queue numbers used in receiving; T_delay: maximum allowable delay; hashtable: divide traffic to different queues for multi-thread reception according to a hash value. In this method we choose MAC address or IP address according to hashtable value; T_best and Q_best: numbers of threads and queues when delay is minimum; delay: delay measured during the receiving process; Flag: identification for suitable value. Multi-thread receiving method can be summarized in the following steps.

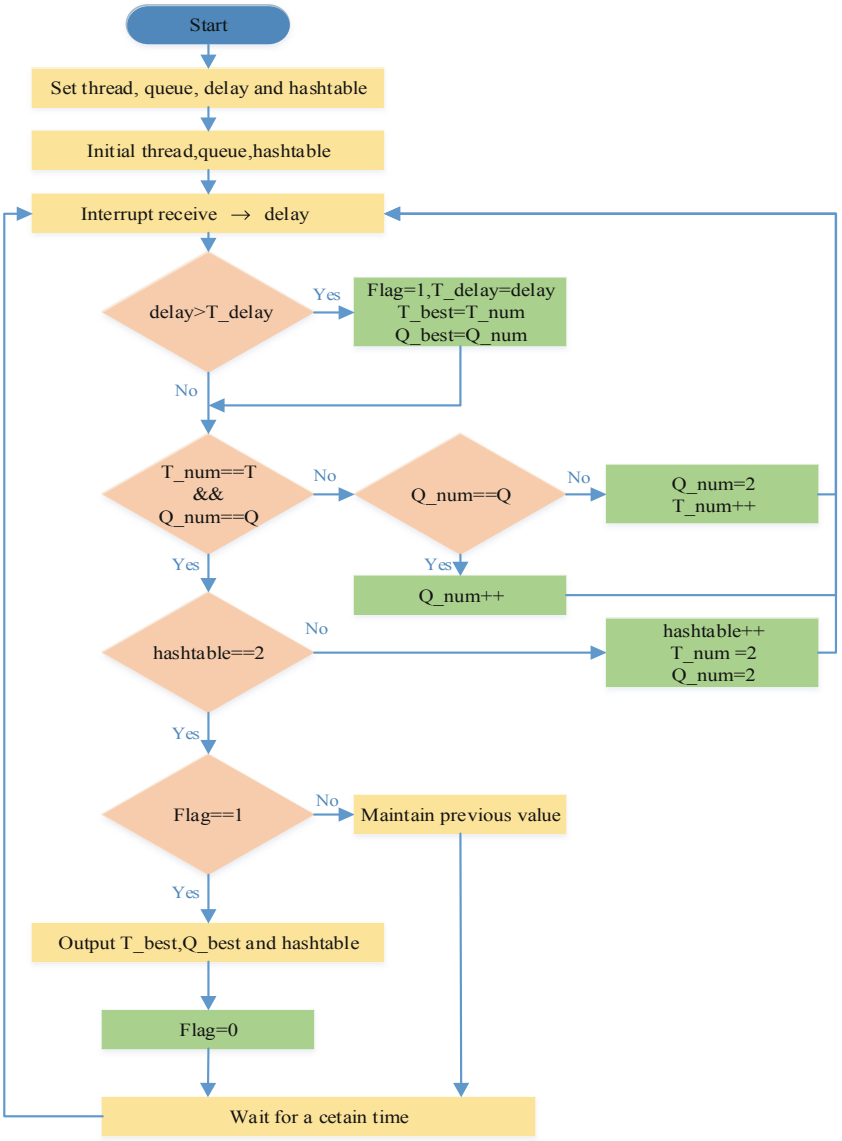


**Fig. 3.** Flow diagram of multi-thread receiving method.

**Step 0:** Set T_num, Q_num, T_delay and hashtable to T, Q, 250 microseconds and 2 respectively.

**Step 1:** Init T_num, Q_num, and hashtable to 2, 2, and 1.

**Step 2:** Measure delay with multi-thread and multi-queues.

**Step 3:** Compare measuring delay and setting delay to choose T_best and Q_best. Then, determine the delay for the next comparison. After T_num and Q_num reach T

and Q, hashtable value is changed to 2 to divide traffic according to MAC address rather than IP address.

**Step 4:** When traversal completes, suitable threads and queues can be determined according to T_best, Q_best, and Flag.

**Step 5:** After DU runs for a certain time with T_best, Q_best, and hashtable, this flow can be run again to obtain new T_best and Q_best.

In order to measure fronthaul delay, a call back mechanism is designed. When DU running on x86 server sends a specific command to RRU, it returns a specific frame data after baseband processing, so delay and loss rate can be measured.

The above method is implemented by using the multi-thread interrupt. Furthermore, a better method of receiving data packet combining interrupt and polling mode is proposed. The method of getting thread and queue numbers is consistent with the method in Fig. 3. When a data packet arrives at DU, the hardware interrupt is triggered. Then the polling mode is used to receive data packet for a setting time. At this time, multiple CPUs always monitor the channel. Packets sent to the NIC will be sent directly to multiple queues and received by multiple CPUs simultaneously rather than sent by a soft interrupt.. The polling time setting is related to different channel environments. For high traffic transmission, polling time can be set longer, whereas, for intermittent low traffic transmission, polling time needs to be reduced to increase resource utilization and reduce energy consumption.

### 3.2 Fronthaul Architecture Design

Based on proposed method, a new 5G fronthaul architecture is designed as illustrated in Fig. 4. In the new fronthaul design, the goal is that DU sends each data frame with a timestamp to RRU, then the data is written to corresponding random access memory (RAM) address according to DU timestamp. Finally, RRU will send data from the corresponding RAM address according to local RRU time. RRU does not care where data comes from and just sends data from specific address, so it is better for data traffic management through this design. In this way, multiple channels of data from different DUs can be received for transmission to increase flexibility and scalability. To implement the fronthaul design, some aspects should be considered.

A. Clock Synchronization

The strict clock synchronization between DU and RRU is required to meet clock requirements for transmission and reception. Unlike traditional CPRI link, delay can be estimated due to the fixed transmission rate. While the Ethernet-based is used, because of queue delay, frame packing delay and changed link rate, it's difficult to implement clock synchronization scheme between DU and RRU. In this design, in order to get synchronized, timestamps are maintained on both sides of gNodeB and baseband board to mark each local clock, in addition, data is sent according to the timestamp order. For the downlink, when gNodeB sends data frame, FPGA will put data into the specified address of ring RAM according to data frame timestamp. On the other hand, FPGA fetches data from the corresponding address according to local timestamp and sends it. Taking transmission and processing delay into consideration, gNodeB needs to transport data in advance to send correct radio data at the right time on the RF module.

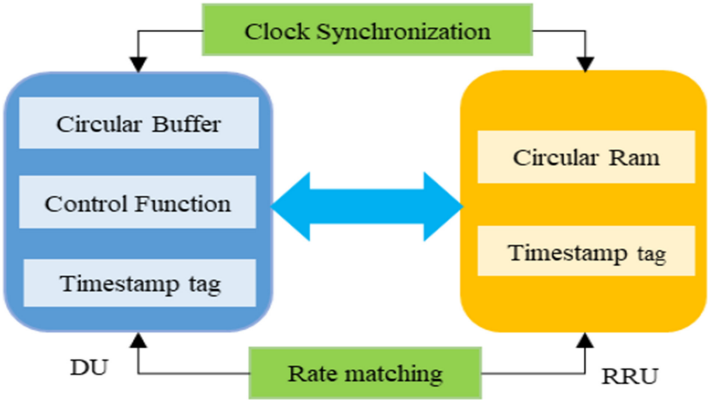


**Fig. 4.** Fronthaul architecture.

In this way, FPGA doesn't care where data comes from and just sends data according to local clock at FPGA. Therefore, it can provide a better way for scalability and management.

B. Rate Matching

At RRU, the size of circular RAM is limited, and the sending rate at DU and RRU are different. To prevent data at circular RAM from being overwritten, rate control is important for normal data transmission. RRU sends the local timestamp to DU, then DU compares its time with RRU timestamp to get rate relationship between DU and RRU. If time at DU is ahead a lot, which means sending rate at DU is faster and lots of data in circular RAM is waiting to be sent at RRU, DU should pause sending for a while to match rate.

C. Control Function

In addition to the ARM module (PS), gNodeB should also be able to configure the baseband processing unit, so the Ethernet-based network can simultaneously transmit user data as well as control data. The control data is mainly used to configure registers, such as address configuration, control register configuration for receiving and transmitting to achieve uplink and downlink switching. When the link has no network traffic, the corresponding receiving module can be turned off to save energy.

## 4 Hardware Implementation

In this design, ZYNQ 7000 chip is chosen as System on Chip (SoC) architecture, which includes PS and programmable logic module (PL). The block diagram of the baseband board is shown in Fig. 5, which is connected to Software-Defined Network (SDN) gNodeB running on the x86-64 server over the 10GE fronthaul link. The data transmission control process is realized by PS module, and the design of 10GE interface is completed on the FPGA side with data baseband processing function.

The RF chip AD9371 is configured by PS module. Unlike the traditional Low Voltage Differential Signal (LVDS) interface between FPGA and Analog-to-digital (AD) module, JESD204B high-speed serial interface is used between FPGA and AD module for data transmission because of the high sampling rate of AD module.

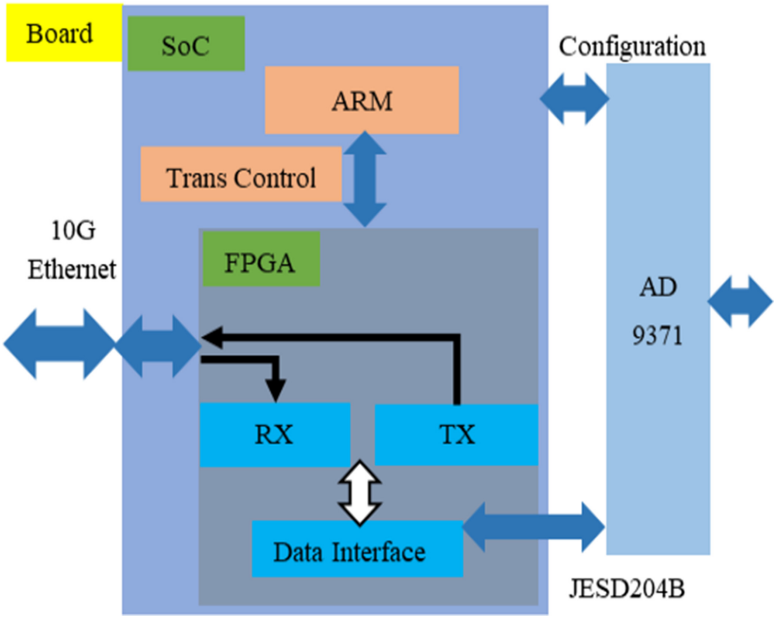


**Fig. 5.** Block diagram of the baseband board.

A simple sinusoidal signal test is shown in Fig. 6. In this design, the center frequency of AD9371 chip is set to 2.5 GHz and IQ sample data rate is set to 122.88 MHz. It can be seen from the figure that test result is consistent with theoretical result that there is a peak at frequency of 2.5 GHz.

## 5 Evaluation

Performance of the proposed method and architecture is evaluated in the section. CPRI basic frames are encapsulated in Ethernet payload before transmitted over the fronthaul link, so different Ethernet payload sizes will affect delay and packet loss rate.

Figure 7 shows the result of delay and packet loss rate with different Ethernet payload sizes (64, 256, 512, 1024, and 1480 bytes). As expected, with Ethernet payload sizes increasing, encapsulation delay increases. On the other hand, numbers of interrupts are decreased because of reduced packet numbers under the same traffic, so packet loss rate decreases. Because of increases in processing delay and queuing delay, fronthaul delay increases as the Ethernet payload size increases.

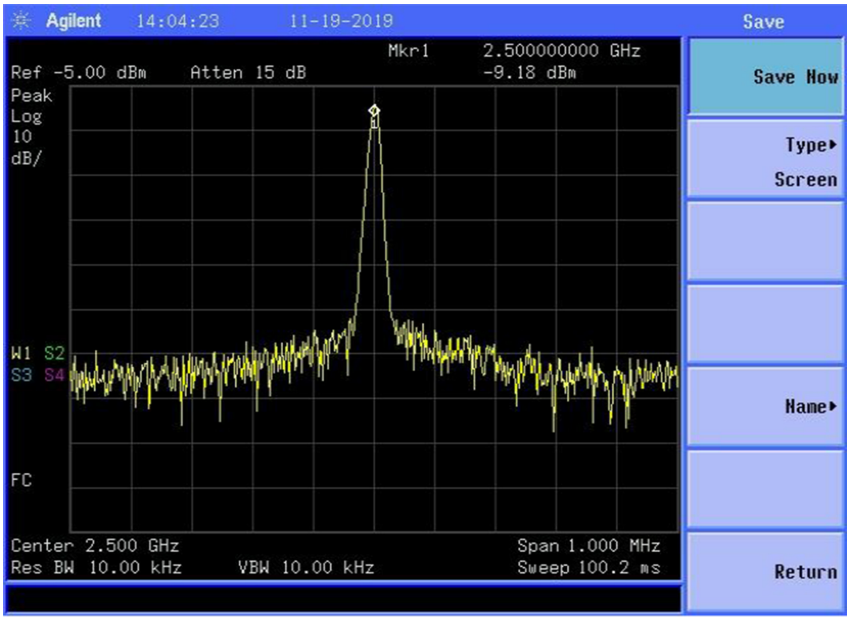


**Fig. 6.** The test result of spectrum analyzer.

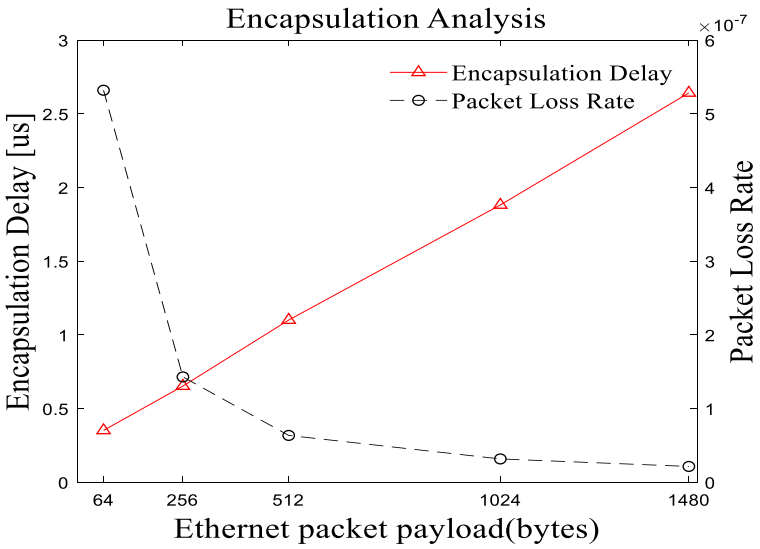


**Fig. 7.** Analysis of delay and packet loss rate.

Moreover, although encapsulation delay, queuing delay, and processing delay decrease with smaller numbers of payload sizes, a larger packet loss rate results in more retransmissions and has an impact on fronthaul delay. In 5G applications, it is

necessary to balance the effects of delay and packet loss rate. Appropriate payload sizes can be selected according to different application scenarios and service requirements to achieve optimal performance.

Delay cumulative distribution function (CDF) diagram is used during the experiment to compare delay performance from DU to RRU. Figure 8 shows the result of delay test using general interrupt, polling method and multi-thread method. The maximum delay of general interrupt is greater than 250 us. Compared with the polling method [12], multi-thread method has shorter delay, which is 160 us and 200 us respectively. Delay requirement of next-generation fronthaul Interface (NGFI) is 250 us when using options 7 and 8 [13], so multi-thread method can meet requirements well. Furthermore, the order of packet loss rate during reception is $10^{-8}$, which meets the requirement that NGFI packet loss rate does not exceed $10^{-7}$ [14].

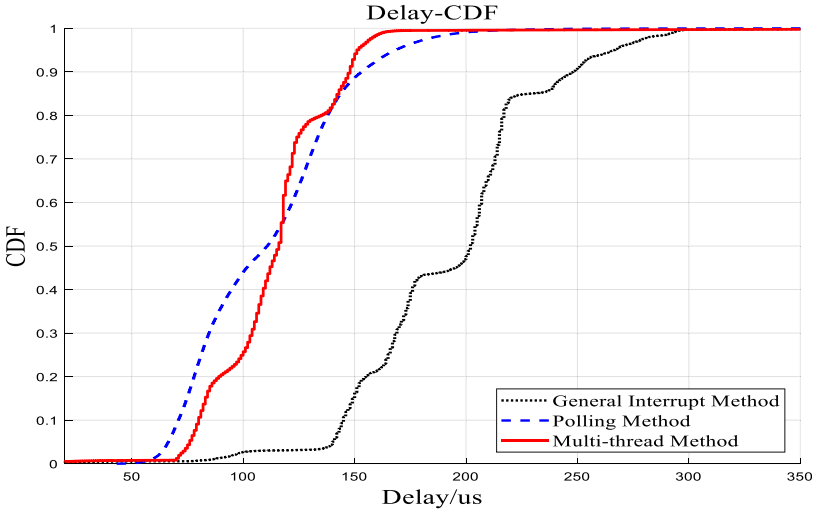


**Fig. 8.** Delay test result of interrupt, polling and multi-thread methods.

Figure 9 shows delay test result of new method and multi-thread method. By combining interrupt and polling method, lower delay can be achieved with a maximum delay of 110 microseconds, and this method can adapt various application scenarios by adjusting polling time. In addition, flexibility and energy efficiency can be easily achieved by using the method and fronthaul architecture.

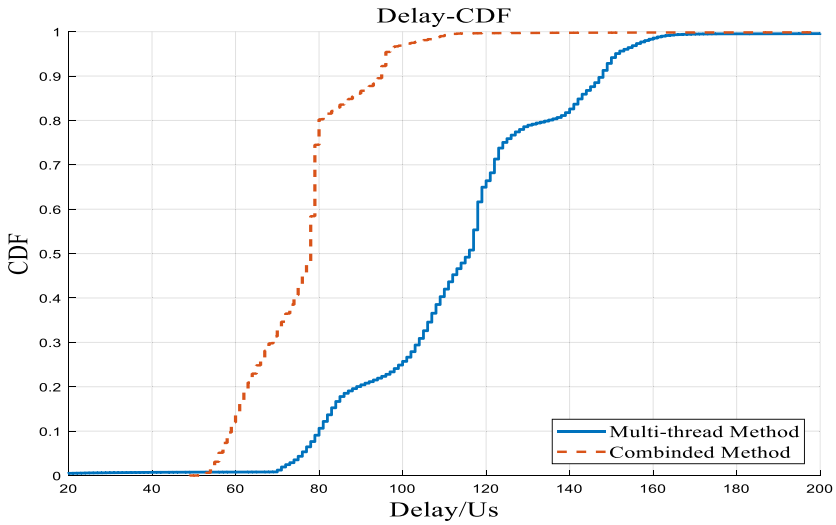


**Fig. 9.** Delay test result of multi-thread and combined methods.

## 6 Conclusion

In this study, a multi-thread scheduling receiving method was proposed based on interrupt and polling mode. Based on the method, new scalable and flexible 5G fronthaul architecture was presented to manage data traffic flow efficiently. In the end, an experimental 5G fronthaul platform was built to verify proposed method and architecture based on SDN/NFV technology and 10 Gigabit Ethernet. Delay and reliability can meet requirements of the 5G fronthaul link by using the method and new fronthaul architecture. Besides, the effect of Ethernet payload sizes on delay and packet loss rate was analyzed. With payload sizes increasing, encapsulation delay increased but packet loss rate decreased.

**Acknowledgement.** This work was supported by the National Key R&D Program of China under Grant 2020YFB1804901 and the National Natural Science Foundation of China under Grant 61871035.

## References

1. Chih-Lin, I., et al.: RAN revolution with NGFI (xhaul) for 5G. J. Lightwave Technol. **36**(2), 541–550 (2018)
2. Checko, A., et al.: Cloud RAN for mobile networks - a technology overview. IEEE Commu. Surv. Tutor. **17**(1), 405–426 (2015)
3. Chih-Lin, I., et al.: Toward green and soft: a 5G perspective. IEEE Commun. Mag. **52**(2), 66–73 (2014)
4. Hinrichs, M., et al.: Experimental investigation of new fronthaul concepts for 5G. Kurzfassung, pp. 56–60 (2017)

5. Liu, Y., et al.: Flow Scheduling with low fronthaul delay for NGFI in C-RAN. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pp. 1–6 (2019)
6. Halabian, H., Ashwood-Smith, P.: Capacity planning for 5G packet-based front-haul. In: IEEE Wireless Communications and Networking Conference, WCNC, April 2018, pp. 1–6 (2018)
7. Chitimalla, D., et al.: 5G fronthaul-latency and jitter studies of CPRI over ethernet. J. Opt. Commun. Network. **9**(2), 172–182 (2017)
8. Tonini, F., et al.: A traffic pattern adaptive mechanism to bound packet delay and delay variation in 5G fronthaul. In: 2019 European Conference on Networks and Communications (EuCNC), vol. 2, pp. 416–420 (2019)
9. Mountaser, G., et al.: Reliable and low-latency fronthaul for tactile internet applications. IEEE J. Sel. Areas Commun. **36**(11), 2455–2463 (2018)
10. Mountaser, G., et al.: Latency bounds of packet-based fronthaul for cloud-RAN with functionality split. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pp. 1–6 (2019)
11. Waqar, M., et al.: A transport scheme for reducing delays and jitter in ethernet-based 5G fronthaul networks. IEEE Access **6**(2018), 46110–46121 (2018)
12. Su, S., Wang, W.: 5G fronthaul design based on software-defined and virtualized radio access network. In: 2019 28th Wireless and Optical Communications Conference (WOCC). WOCC 2019, pp. 1–5 (2019)
13. GPP TR38.801: Study on new radio access technology: radio access architecture and interfaces. V1.2.0 (2017-02) (2017)
14. Zhiling, Y., et al.: White paper of next generation fronthaul interface (v1.0). China Mobile Research Institute (2015)