Cloud and Edge Collaborative Computing for Efficient 5G Optical Fronthaul Network Slicing

Bingchang Hua[1,2,3], Min Zhu[1,2*], Jiao Zhang [1,2], Yuancheng Cai[1,2], Mingzheng Lei[1,2], Yucong Zou[1,2], Aijie Li[1,2] and Zhiguo Zhang[3]

[1] *National Mobile Communications Research Laboratory, Southeast University, Nanjing, Jiangsu 210096, China*
[2] *Purple Mountain Laboratories, Nanjing, Jiangsu 211111, China*
[3] State Key Lab of Information Photonics and Optical Communications, *Beijing University of Posts and Telecommunications, Beijing, 100876, China*
*minzhu@seu.edu.cn

Abstract: We propose an efficient 5G optical fronthaul network-slicing solution based on cloud and edge collaborative computing. Simulation results show that the proposed solution significantly reduces the delay of each service type.
OCIS codes: (060.2330) Fiber optics communications; (060.4264) Networks, wavelength assignment.

1. Introduction

Novel network services and application scenarios are continuously emerging, including high-speed downloads, cloud services, autonomous driving, and more. Furthermore, the 5G network must satisfy the various requirements of three general business types: enhanced mobile broadband (eMBB), ultra-reliable low-latency communication (uRLLC), and massive machine-type communication (mMTC) [1]. The network system structure must offer considerable flexibility and self-adaptability, to adapt effectively to the diverse quality of service (QoS) requirements of different business types. Network slicing technology can abstract network resources and group them into multiple virtual resources, allowing the network infrastructure to be shared by different services [2]. By using specific user plane and user control plane functions, each network slice can be used to serve a specific business type. Simultaneously, to satisfy the QoS requirements of different service types in 5G networks (e.g., low latency and high reliability), edge computing has been proposed as a promising method capable of dramatically enhancing computing performance through deploying servers on the network edge [3-4]. Because edge computing is sensitive to deployment costs, the capacities of the mobile-edge computing (MEC) nodes are limited; this restricts the computing and communication capabilities of edge computing [5]. In recent years, the technologies of network slicing and MEC have been intensively studied. However, almost all of the studies have focused on network-slicing strategies and network bandwidth resource optimization; they rarely consider the multiple delay and reliability requirements of different 5G networks. The joint optimization problem of delay and network-resource utilization rates of multiple business types in 5G network have not been addressed in the open literature.

To solve the above network-slicing problem, this paper presents an efficient collaborative resource management (ECRM) solution based on cloud and edge computing; this solution can comprehensively consider the service delays and network-resource utilization rates. Simulation results show that the proposed solution greatly reduces the E2E delay of each service type, and the E2E delay of uRLLC services is stable at 2 ms.

2. System architecture and efficient collaborative resource management

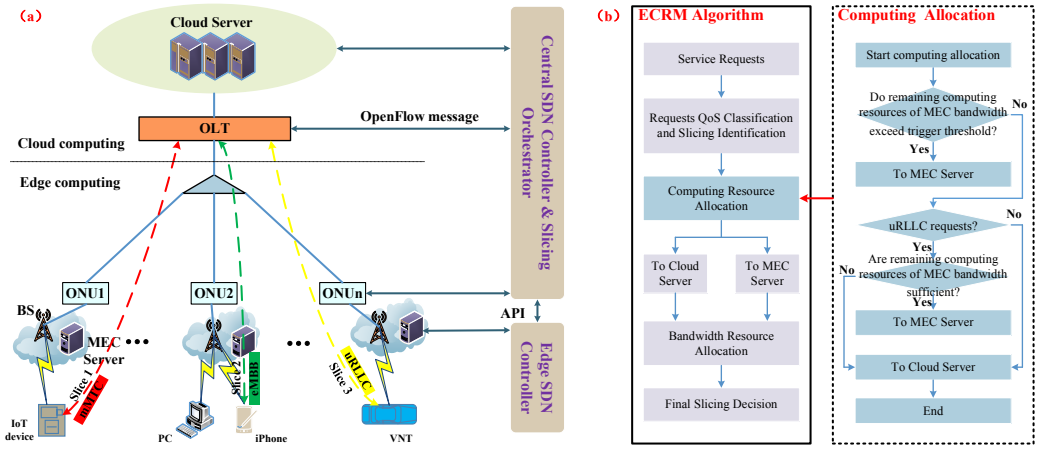

Fig.1. (a) TWDM-PON-based cloud-edge collaboration optical MFH network architecture; (b) Flow chart of the proposed ECRM algorithm.

Fig. 1(a) shows a cloud-edge collaborative optical mobile fronthaul (MFH) network architecture, based upon a time and wavelength division multiplexed passive optical network (TWDM-PON). The architecture incorporates two different types of servers: a cloud server on the optical line terminal (OLT) side and a mobile-edge server on the base-station side. Owing to the large costs of deploying MEC servers on the base-station side, their resource capacity is limited. The resource capacity of the cloud server is much larger than that of the mobile-edge server; however, when the service is transmitted to the cloud server, it produces a certain propagation and queuing delay. The TWDM-PON-based cloud-edge collaborative optical MFH network operates through its central software-defined networking (SDN) controller and slice orchestrator (CSC&SO) and edge SDN controller (ESC) to achieve flexible resource management and control different network slices.

Fig. 1(b) presents a flow chart of the proposed ECRM algorithm. When the UE sends a service request, the ECRM algorithm first determine the service type according to the network-slice identification; subsequently, it performs the computing and bandwidth resource allocation. To alleviate the pressure of the MFH network bandwidth, each service request is processed preferentially at the MEC server. In addition, to meet the low-latency requirements, the MEC server reserves certain computing resources for subsequent delay sensitive service requests. Therefore, when allocating computing resources, it first determines whether the remaining computational capacity of the MEC server exceeds the trigger threshold; if it does, it puts the service request into the MEC server for processing; otherwise, it continues to judge the service type. For eMBB and mMTC services, the system puts the service request onto the cloud server for processing. For uRLLC services, it determines whether the MEC server has sufficient remaining resources to accept the service request; if it does, it puts the service request onto the MEC server for processing. Otherwise, the requested service is allocated to the cloud server. After computing-resource allocation has been completed, bandwidth resource allocation is performed, and the available link with the most remaining bandwidth is selected first. Notably, owing to the delay sensitive characteristics of uRLLC services, the priority allocation principle is adopted in bandwidth allocation. Finally, the network-slicing decision is completed, and the OpenFlow protocol is issued by the ESC to establish the path between the destination and source nodes. The edge-computing server trigger threshold can be set according to the traffic load.

To meet the QoS requirements of each service and achieve an optimal network-resource allocation strategy, the end to end (E2E) delay of each service must be calculated. The total E2E delay D_{e2e} of the network in this solution consists of four parts: the transmission delay D_{trans}, propagation delay D_{prop}, processing delay D_{proc} and queuing delay D_{queue}. The total E2E delay can then be expressed as follows:

$$D_{e2e} = D_{trans} + D_{prop} + D_{proc} + D_{queue}. \quad (1)$$

3. Results and discussion

Table 1. Simulation parameters

Parameter	Value
Number of cloud servers	1
Number of edge servers	16
Number of wavelengths	4
Bandwidth of wavelengths	25 Gbps
Cloud computation capacity	10×10^{13} CPU cycles
Edge computation capacity	2.5×10^{12} CPU cycles

The system simulation parameters are shown in Table 1. The number of service requests was 8000, of which eMBBs and uRLLCs each accounted for 20% and mMTCs accounted for 60%. The bandwidth requirements of eMBB, mMTC, and uRLLC services obeyed the Poisson distribution with averages of 50M, 1M, and 10M, respectively. To simplify the simulation system, the computing resources required for all the various types of service requests were 1000 CPU cycle/bit. Each request was randomly sent to each base station. In this paper, the total traffic load was simulated from 10Gbps to 100Gbps (the normalized traffic load from 0.1 to 1), by controlling the request arrival rate.

To compare and analyze the performance of the proposed solution, this study also simulated two comparison solutions. The first was the benchmark solution; here, all services were unified by cloud-server processing. The second was the integrated resource-management algorithm (INRM) [6]; the principle of this algorithm is that uRLLC services are preferentially processed by the edge server, and only moved to the cloud when the edge-computing resources are insufficient; meanwhile, the eMBB and mMTC services are always processed by the cloud-computing server. In the ECRM algorithm proposed in this paper, P was used to represent the percentage of edge-server computing resources, and the trigger threshold was $(1-p)\times Edge_{capacity}$, where $Edge_{capacity}$ is the computing-resource capacity of the edge server. The P-value of the ECRM solution in the simulation system was set to 0.7.

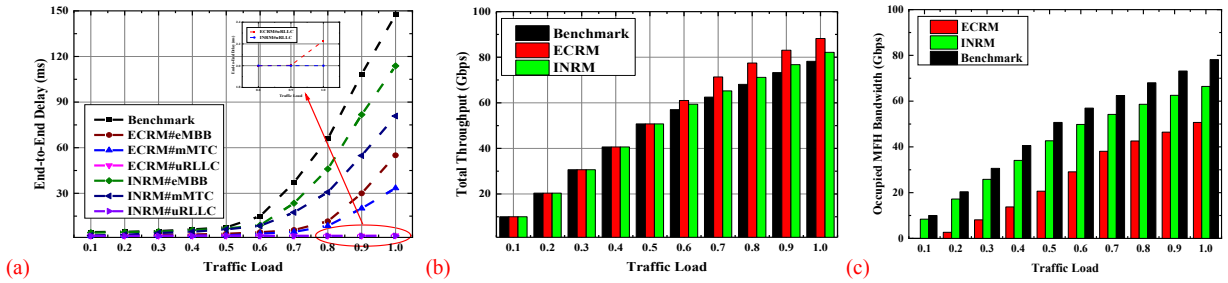

Fig. 2. (a) E2E delay of various service types; (b) total throughput of each solution; (c) Fronthaul network bandwidth occupied by each solution.

First, we simulated the performance of the three solutions in terms of delay. Fig. 2 (a) depicts the curve of the E2E delay with respect to traffic load; the benchmark solution exhibits the highest E2E delay. Compared with the INRM solution, the E2E delay for eMBB and mMTC in the ECRM solution was significantly reduced; however, the uRLLC service was improved. Besides, the E2E delay of uRLLC in the ECRM solution stabilized at approximately 2 ms. The relationship between the total throughput of each solution and the traffic load is shown in Fig. 2 (b). It can be observed that when the normalized traffic load exceeds 0.5, the total throughput improvement of the ECRM solution is as much as 15% compared to the benchmark solution, while the total throughput improvement of the INRM solution is 5%. Because the network is congested at this time, both the INRM and ECRM solutions use edge servers to offload the uRLLC traffic, effectively reducing the pressure on the MFH network bandwidth. The ECRM solution also offloads eMBB and mMTC traffic when the remaining resources of the edge-computing server exceed the trigger threshold; this makes the total throughput greater under the same normalized traffic load. This results is also verified from the curve of the fronthaul network bandwidth occupied by each solution, depicted in Fig. 2 (c)

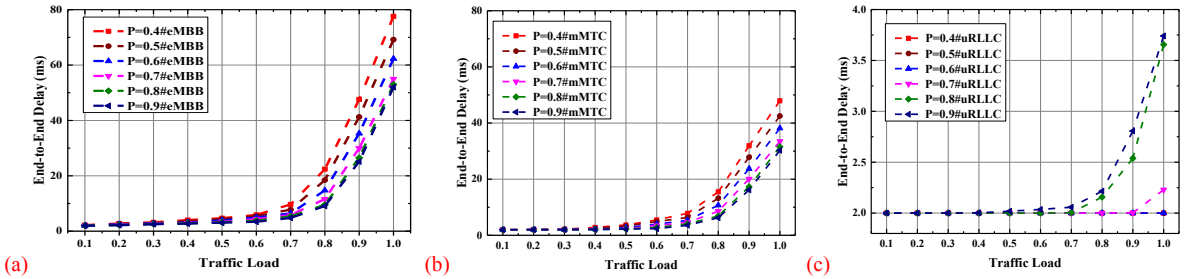

Fig. 3. E2E delay of (a) eMBB, (b) mMTC, and (c) uRLLC services under ECRM solution with different *P*-values.

To verify the performance of the proposed solution under different trigger thresholds, corresponding simulations were conducted. Fig. 3 (a-c) depict the E2E delay of eMBB, mMTC, and uRLLC services under different *P*-values of the ECRM solution, respectively. The E2E delay of eMBB and mMTC services decreases with increasing *P*-value, whereas the E2E delay of uRLLC services increases under an increase in *P*-value. The analysis for eMBB and mMTC services suggests that under an increase of *P*-value, the edge-computing server trigger threshold decreases, the resources used by the edge-computing server increase, and more traffic is offloaded to the edge, reducing the delay of these service requests. In contrast, as the *P*-value for uRLLC services increases, the edge-computing server trigger threshold decreases, and the edge-computing resources reserved for uRLLC services are decreased.

4. Conclusions

In this paper, we proposed an ECRM solution based on cloud-edge computing. Depending on the QoS requirements of the chosen service, the proposed solution provides appropriate network resources through network-slicing technology. Simulation results show that the proposed solution greatly reduces the E2E delay of each service type, and the E2E delay of uRLLC services is stable at 2 ms. Compared with the traditional benchmark solutions, the proposed solution achieves a maximum of 15% network throughput improvement.

5. References

[1] A. A. Gebremariam, et al. Proc. IEEE Int. Conf. Commun., Kansas, MO, USA, 2018: 1–6.
[2] O. Sallent, et al. IEEE Wireless Commun **24**.5 (2017): 166–174.
[3] Y. Sahni, et al. IEEE Access **5** (2017):16441-16458.
[4] Xiaomin Chang, et al. IEEE International Conference on Acoustics, Barcelona, Spain, 2020: 8981-8985.
[5] V. L. Nguyen, et al. IEEE Network **32**.1 (2018): 118-124.
[6] C. Song, et al. Journal of Optical Communications and Networking **11**.4 (2019): B60-B70.