

Research Article

An Intelligent Real-Time Traffic Control Based on Mobile Edge Computing for Individual Private Environment

Sa Math ¹, Lejun Zhang,² Seokhoon Kim ³, and Intae Ryoo ⁴

¹Department of Software Convergence, Soonchunhyang University, Asan-si, Chungcheongnam-do 31538, Republic of Korea

²Department of Information Engineering, Yangzhou University, Yangzhou 225127, China

³Department of Computer Software Engineering, Soonchunhyang University, Asan-si, Chungcheongnam-do 31538, Republic of Korea

⁴Department of Computer Engineering, Kyung Hee University, Gwangju-si, 17104 Gyeonggi-Do, Republic of Korea

Correspondence should be addressed to Seokhoon Kim; seokhoon@sch.ac.kr and Intae Ryoo; itryoo@khu.ac.kr

Received 10 August 2020; Revised 17 September 2020; Accepted 5 October 2020; Published 22 October 2020

Academic Editor: Xiaolong Xu

Copyright © 2020 Sa Math et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The existence of Mobile Edge Computing (MEC) provides a novel and great opportunity to enhance user quality of service (QoS) by enabling local communication. The 5th generation (5G) communication is consisting of massive connectivity at the Radio Access Network (RAN), where the tremendous user traffic will be generated and sent to fronthaul and backhaul gateways, respectively. Since fronthaul and backhaul gateways are commonly installed by using optical networks, the bottleneck network will occur when the incoming traffic exceeds the capacity of the gateways. To meet the requirement of real-time communication in terms of ultralow latency (ULL), these aforementioned issues have to be solved. In this paper, we proposed an intelligent real-time traffic control based on MEC to handle user traffic at both gateways. The method sliced the user traffic into four communication classes, including conversation, streaming, interactive, and background communication. And MEC server has been integrated into the gateway for caching the sliced traffic. Subsequently, the MEC server can handle each user traffic slice based on its QoS requirements. The evaluation results showed that the proposed scheme enhances the QoS and can outperform on the conventional approach in terms of delays, jitters, and throughputs. Based on the simulated results, the proposed scheme is suitable for improving time-sensitive communication including IoT sensor's data. The simulation results are validated through computer software simulation.

1. Introduction

The presence of the 5th generation (5G) communication network significantly aims to deliver an extremely fast communication speed, high reliability, and low end-to-end (E2E) delay in milliseconds (ms). Each base-station cell coverage provides local service within 100 meters, which attempts to offer strong connectivity, real-time (RT) communication, and Device-to-Device (D2D) and supports massive User Equipment (UE) connection as well [1, 2]. 5G provides huge bandwidth and ultra-low latency (ULL) ten times over 4G-LTE. Moreover, the 5G paradigm aims to support future user applications such as IoT traffic, Wireless Sensor Network (WSN) traffic, automotive transportation,

and gaming traffic. With this support and these contributions, the huge traffic is generated from the Radio Access Network (RAN) devices and goes through to the Evolved Packet Core (EPC) gateways such as a Service Gateway (S-GW) and Packet Data Gateway (P-GW). Nevertheless, there is still limited capacity in the EPC area while it is a common optical network. Therefore, EPC architecture keeps the similarity to the previous communication system which possibly arises traffic congestion problems and insufficient resources in the EPC area. To reduce the outgoing traffic to the remote network, local clouds have been proposed to establish local communication [3]. Currently, Mobile Edge Computing (MEC), Network Slicing (NS), Software-Defined Network (SDN), and Network Function Virtualization

(NFV) are necessary technologies which have been proposed to overcome the aforementioned troubles and challenges in 5G communication, to improve the network performance, and to take benefits from cost reduction [4, 5]. Figure 1 illustrates the typically 5G end-to-end (E2E) communication system architecture. The bottleneck network area is located in the fronthaul and backhaul, whenever the incoming traffic from the variety of Radio Remote Heads (RRH) surpasses the serving capacity, the network congestion will occur. The ULL perspective is required for RT communication, so it is obliged to cope with the existing problems in the EPC gateways. Due to the fact that the fronthaul (S-GW) and backhaul gateway (P-GW) share the same network architecture, in this paper, the gateway refers to the S-GW or P-GW. MEC servers were integrated into the gateway to cache the traffic sliced before forwarding to the remote network.

2. Related Works

2.1. Real-Time Communication. The time-sensitive communications refer to video conference, mobile video streaming, game streaming, voice over Internet protocol (VoIP), and other RT traffic running over an unreliable network transport protocol called user datagram protocol (UDP). These communication types required ULL (delay and jitter) for Round-Trip Time (RTT) and required sufficient communication bandwidth. Daily communication traffic typically comprised two classes, namely, time-sensitive and time-insensitive. Time-sensitive (RT) classes are conversation and streaming; therefore, these communication types are less restricted on the packet error ratio during communication but require extremely high network status. The time-insensitive class consists of two communication types including interactive and background traffic. These communication types required extremely high communication reliability and very low packet error ratio, since it sends information through transmission control protocol (TCP), so the retransmission packet will occur whenever the destination missed the previous packets. However, there is less restriction on communication latency and bandwidth. In nowadays communication environments, both RT and nonreal-time (NRT) are sent through the same network environments with insufficient dynamic resources management to meet the QoS perspective for each traffic class. During the last several years, RT communications take benefits from local cloud services.

2.2. Mobile Edge Computing (MEC). The local clouds (i.e., MEC, fog computing, and cloudlet) has been released to enhance QoS for various network applications such as the Internet of Things (IoT), Heterogeneous Internet of Things (HetIoTs), gaming, and other applications, especially for RT applications [6–9]. The presence of MEC establishes the intelligence network in the edge area [10–12]. This technology claims to gain higher communication bandwidth and provides ULL for real-time communication. Meanwhile, MEC consists of challenging issues in capacity limitation,

while it is required to offer heterogeneous services for massive users. Some applications with higher resource computation requirements are required to access remotely the MCC server. Moreover, privacy protection for the local cloud is required to be considered for safety communication and data integrity [13–16]. As shown in Figure 2, the caching method is enabled by MEC servers which synchronized with MCC servers, and the frequently requested contents or popularly used applications are targeted to be cached to MEC servers.

The caching methods are beneficial in latency reduction, gain a higher bandwidth, and save the resources at EPC for both user-plane and data-plane. Anyway, several challenges have been introduced in MEC employment, such as expanded RRH infrastructure, power consumption, resource management, and security problems [17]. Due to that a variety of user's information is stored in MEC in an edge network, the complicated security methods are required to enable trusted communications for edge networking [3, 8]. There exists a huge, especially, convergence of heterogeneous applications, services, and infrastructures in 5G edge networks, both physical and virtual. These network environments are not convenient to handle for both security and excellent network QoS [18, 19]. So, Network Slicing presents a novel opportunity to handle the issues by slicing the user applications into different groups. With machine learning algorithms combination, it is possible to facilitate Network Slicing in terms of classification of the complicated user information, applications, and devices [20, 21]. The user applications can be sliced by grouping the user applications which are sharing the same or similar resource requirement into the same group. The sliced applications are more convenient to control and provide flexible control and security configuration by the controller. Undoubtedly, Network Slicing is a key candidate to enhance future network QoS and network safety to meet the perspective of 5G technology [22, 23].

2.3. Software-Defined Network (SDN). SDN is a key adoption candidate to enable future networking driven to softwarization and intelligent networks. SDN provides a global view of network status and a completely programmable system at the control plane [24]. Also, SDN is a concept of decoupling a forwarding plane from the control plane. Plus, this separation gains more convenience in terms of flexibility and scalability, while the user-plane requires higher bandwidth and the control plane requires lower latency [25, 26]. The computing, routing, monitoring, scheduling, policy control, security, and load-balancing are performed by the SDN controller [27]. Not only can SDN, especially, be used to enhance the QoS for RT traffic, but also it can be used to enhance trusted communication based on blockchain [28]. The controller gathers information from the user-plane by the southbound interface and communication with the upper layer by the northbound interface. The communication interface between them is provided by the OpenFlow protocol [29]. Even though SDN could independently stand without other technologies getting involved, but the

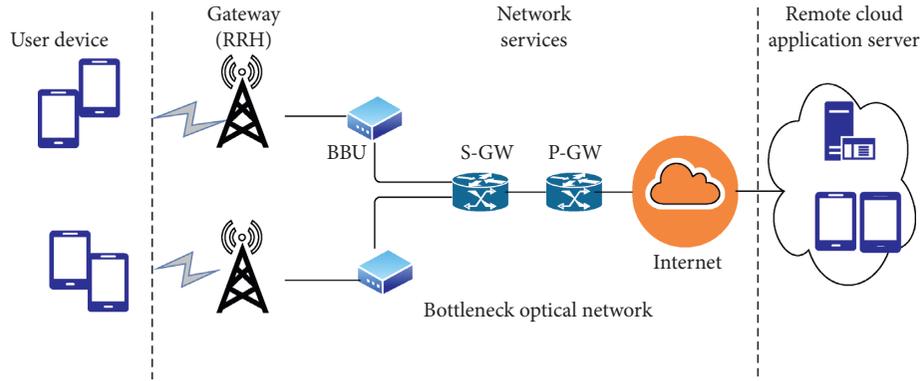


FIGURE 1: The typical future 5G communication architecture.

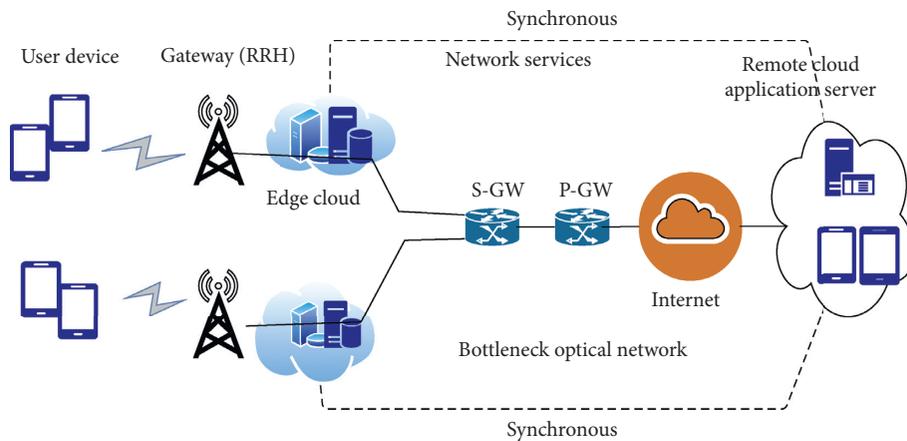


FIGURE 2: The typical future 5G communication architecture based on the MEC server.

integration of SDN and NFV presents a great opportunity to enhance virtualized computing in future network environments [30, 31]. This idea aims to provide virtualized resources to SDN entities and enables the controller to generate both physical and virtual resources. In the cloud systems, the converged SDN and NFV can be benefited in computing resources and dynamic resource configurations with a fault-tolerant technique [32, 33]. Based on this mention, the controller possibly generates the virtual controller and offloads from the physical to the virtualized for computing purposes.

2.4. The Proposed Intelligent Real-Time Traffic Control. The proposed method is to enhance the QoS for RT communications that can be caused by the limited resources at the backhaul gateway.

The proposed intelligent real-time traffic control handles the incoming traffic based on traffic classification and integration of the MEC server. Figure 3 shows the proposed network architecture by integrating the MEC server with the backhaul gateway. The MEC servers act as a caching server that buffers the incoming traffic, such as conversation, streaming, interactive, and background communication.

As formerly mentioned, the proposed scheme comprises three stages, namely, traffic classification, caching, and

controlling the classified traffic. The following expressions are details about the three stages above.

2.5. Traffic Classification and MEC Caching. In this paper, Network Slicing is referred to as the splitting of user traffic as the 4 different slices such as slice 1 for conversation, slice 2 for streaming, slice 3 for interactive, and slice 4 for background communication. Therefore, the classification process was based on each traffic characteristic such as a packet error ratio (PER), protocol data unit (PDU) size, and other QoS parameters.

Subsequently, each slice of the traffic was cached to different MEC pools, and each MEC pool provides the buffer resources for queueing the incoming traffic to wait for serving. The traffic slicing can be made by employing the K -mean machine learning method as shown in Figure 4. Start with the determining number of groups $K = 4$ and then calculate the centroid. For the first time, the centroid has to select 4 different subsets as 4 classes randomly. In the next step, distance has to be calculated for each class. The distance can be calculated by using the Euclidean distance (ED) equation given as follows:

$$ED = \sqrt{(x - a)^2 + (y - b)^2}. \quad (1)$$

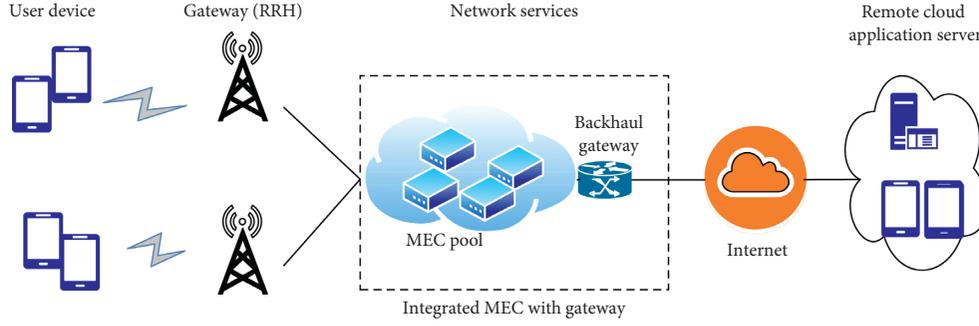


FIGURE 3: The proposed network architecture based on integrated MEC servers at the backhaul network.

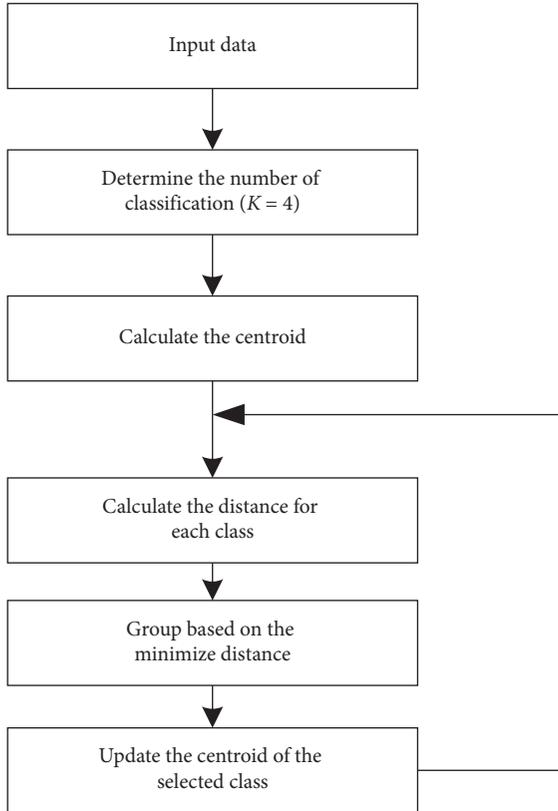


FIGURE 4: The diagram of traffic slice functions based on K-mean clustering flow.

The subset variables x , y , a , and b can present PDU size, PER, TCP, and UDP, respectively. As depicted in Figure 3, four MEC servers were integrated into the backhaul gateway to serve as buffers for the four traffic classes, so each traffic class has its individual MEC server. In this paper, the traffic classification was generated by computer software simulation. The conversation, streaming, interactive, and background traffic was generated based on its QoS parameters.

2.6. Management of the Classified Traffic. When the backhaul network becomes bad condition, it is required to serve real-time communication classes as the first priority. In the proposed scheme, the gateway has been configured to serve the cached traffic based on conditions of backhaul. The

scheme will not be crucial to use when the backhaul gateway is considered as a normal status; otherwise, it is critical to employ the scheme during the backhaul gateway assumed as the congestion state.

The backhaul network can be defined as the $M/M/1$ queue model as follows:

$$\partial = \frac{\lambda_c + \lambda_s + \lambda_i + \lambda_b}{\mu}, \quad (2)$$

where ∂ is denoted as the ratio of incoming traffic and serving rate and also represents the status of backhaul. λ_c , λ_s , λ_i , and λ_b are denoted as the incoming rate of conversation, streaming, interactive, and background traffic, respectively. μ represents the serving rate of the backhaul gateway.

The gateway condition was referred to the user-plane status as the forwarding traffic based on the controller. The controller handles each slice of traffic from MEC pools. The backhaul status can be analyzed based on ∂ , if $\partial \geq 1$ (the backhaul gateway resources are sufficient to handle for incoming traffic), and the status can be assumed as a normal condition. During the backhaul condition assumed as natural status (normal), the serving rule has been configured as default. The default rule handles the incoming traffic based on first come first serve (FCFS). So, the serving resources and rules are equalized for any incoming traffic classes. Thus, the real-time traffic will be dropped and low QoS will increase the waiting period in the MEC server. In another scenario, if $\partial < 1$ (this means that the serving resource of backhaul gateway μ is less than the incoming rate of user traffic λ), the network congestion in the system will occur, so the priority control of each traffic class has to be considered, as shown in Figure 5. RT traffic classes have to be considered as primary control rather than NRT. The scheme increases the communication rating of the RT traffic classes and reduces the communication rating of NRT traffic classes based on the increasing rate ratio of RT. The proposed scheme considers/classifies the cached traffic by four different classes as shown in Figure 6:

- (i) Conversation (RT) has been configured as a first primary class and the serving resources have to be increased more than the other communication classes

```

1:   while (communication ← true)
2:       monitoring ← true
3:       if ( $\partial \leq 1$ )
4:           IsConversation ← true
5:           if (state ← state 0)
6:               state ← state 1
7:           IsStreaming ← true
8:           else if (state ← state 0)
9:               state ← state 3
10:          IsInteractive ← true
11:          else if (state ← state 0)
12:              state ← state 4
13:          IsBackground ← true
14:          else if (state ← state 0)
15:              state ← state 2
16:          end if
17:      end if
18:  end while
    
```

FIGURE 5: The handling traffic based on the proposed scheme when the backhaul is considered as the bad statuses.

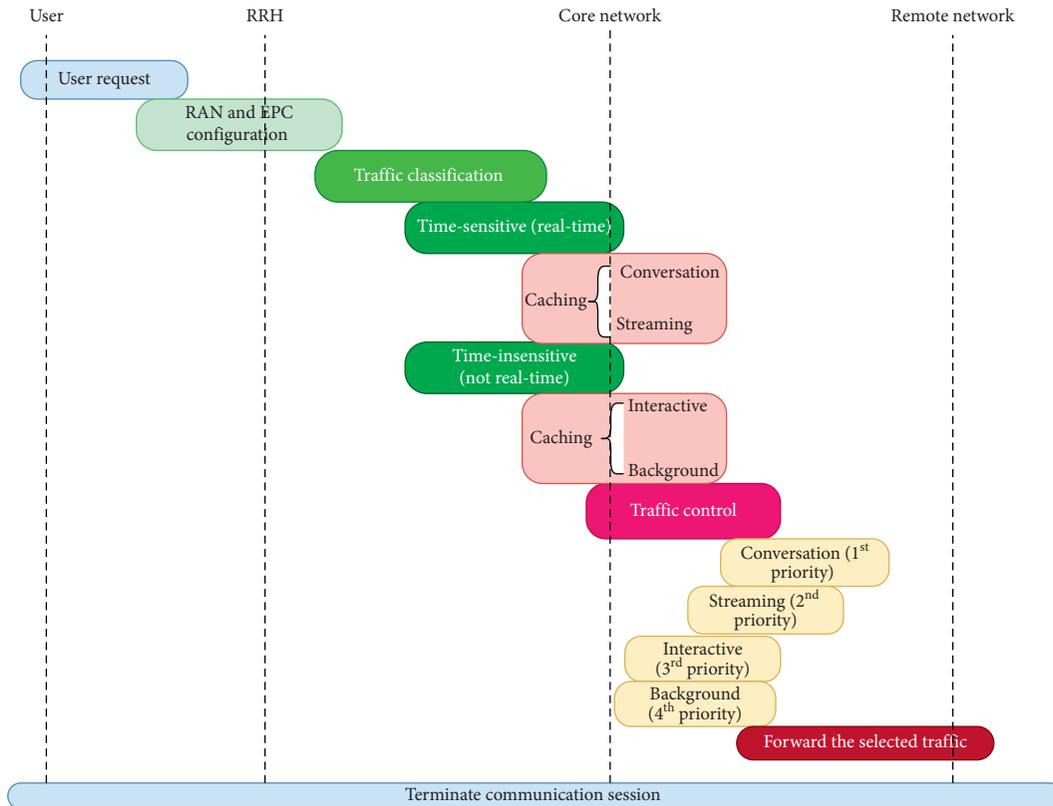


FIGURE 6: The proposed scheme for RT handling based on MEC caching.

(ii) Streaming (RT) has been configured as a second primary class and the backhaul resources will be increased to greater than the interactive and background communication but lower than conversation

(iii) Interactive (NRT) has been configured as a third priority class and the serving resources will be decreased to lower than conversation and streaming but greater than background communication

- (iv) Background (NRT) communication has been configured as the fourth priority and the backhaul resources are limited to lower than other communication classes

The increasing rates of conversation class based on the reducing rate of background communication class and the increasing rate of streaming class are based on the reducing rate of interactive communication class as shown in Table 1.

Moreover, the RT traffic in the MEC servers will be keeping adjusted similar to the normal network status, because, during the network limitation of backhaul resources, the algorithm is restricted to serving resources for NRT. In this scenario, NRT user traffic will be queued in longer periods, due to the fact that some of the NRT resources will be used for RT traffic. The scheme limits the NRT resources until the backhaul gateway becomes a normal status $\partial \geq 1$ and will configure the serving scheme to handle without restriction, as depicted in Figure 7.

2.7. Performance Evaluation

2.7.1. Analysis. The E2E latency occurs during packet transmission in the 5G communication system and can be written as T as follows:

$$T = T_{\text{RAN}} + T_{\text{Backhaul}} + T_{\text{Core}} + T_{\text{Transport}}, \quad (3)$$

where

- (i) T_{RAN} is the latency of packet transmission from UEs to eNB. This latency is mainly from the physical and data-link layer, such as time of negotiation, channel coding, modulation, cyclic redundancy check, and other duties consisted in the physical and data-link layer.
- (ii) T_{Backhaul} is the latency of the packet transmitted from eNB to the backhaul. The common connection between eNB and Core can be fiber optic or microwave link. The latency of the switching process can occur at the SGW.
- (iii) T_{Core} is the time of building the connection to the core gateway. The latency can be contributed by both the control plane and the user-plane. The control plane consists of the latency of various EPC entities such as Mobile Management Entity (MME), Home Subscriber Server (HSS), Policy and Charging Rule Function (PCRF), and the SDN controller.
- (iv) $T_{\text{Transport}}$ is the time taken by data transmission to the remote network; the latency is depending on the distance, link bandwidth, routing, and switching protocol:

$$T_{\text{RAN}} = t_0 + t_{\text{FA}} + t_{\text{TX}} + t_{\text{bsp}} + t_{\text{mpt}}, \quad (4)$$

where

- (i) t_Q is the waiting time of incoming traffic depending on λ , if $\lambda \geq \mu$ then t_Q is increased.
- (ii) t_{FA} is the latency that occurred by frame alignment.
- (iii) t_{TX} is the latency of transmission depending on radio channel condition, payload size, and transport protocol.
- (iv) t_{bsp} is the latency at the eNB.
- (v) t_{mpt} is the delay of at UEs and eNBs terminal; it is depending on the capacity of both terminals:

$$\begin{aligned} T_{\text{Backhaul}} &= t_Q + t_E + t_{\text{TX}} + t_S, \\ T_{\text{Core}} &= t_Q + t_E + t_{\text{TX}} + t_{\text{epc}} + t_{\text{SR}}, \end{aligned} \quad (5)$$

where

- (i) t_E is the time delay of the circuit through network devices.
- (ii) t_S is represented as the switching delay.
- (iii) t_{epc} is represented as the delay between communication interfaces of EPC entities such as MME, HSS, and PCRF. The communication delay between EPC entities will take a few microseconds.
- (iv) t_{SR} is represented the latency of the switching and routing periods.

MEC pool can be modeled as m/m/1 queue model. So, the average waiting time of user traffic is denoted as t_Q , where

$$t_Q = \frac{\lambda}{\mu(\mu - \lambda)}. \quad (6)$$

Then, the Round-Trip Time (RRT) of E2E delay is defined as $2 \times T$ approximately. The E2E delay and latency occurring in the communication system are well discussed in [34].

In the communication environments, delay D_t (at any time t) can occur constantly and vary based on the network statuses. So, the variance delays occur during the communication called jitters ΔJ , since the jitters at the time t are denoted as ΔJ_t and can be calculated as the following equation:

$$\Delta J_t = \sqrt{(D_t - D_{t-1})^2}, \quad \Delta J_t \geq 0. \quad (7)$$

According to equation (7), the communication jitters of the system at time t are denoted as $\Delta J(\text{syst})_t$ and can be formed as

$$\Delta J(\text{syst})_t = \sum_t^n \Delta J_t. \quad (8)$$

Based on equation (8), the average jitters of the four traffic classes in the system can be modeled as $\Delta J(c, s, i, b)_t$, where

TABLE 1: The configuration states for RT and NRT communication classes.

State	Rating control
State 0	None control (100%)
State 1	Increase rating (70%) for communication
State 2	Reduce rating (70%) for background
State 3	Increase rating (60%) for streaming
State 4	Reduce rating (60%) for interactive

```

1:   while (communication ← true)
2:       monitoring ← true
3:       if (∂ ≥ 1)
4:           IsConversation ← true
5:           if (state ← state 1)
6:               state ← state 0
7:           IsStreaming ← true
8:           else if (state ← state 3)
9:               state ← state 0
10:          IsInteractive ← true
11:          else if (state ← state 4)
12:              state ← state 0
13:          IsBackground ← true
14:          else if (state ← state 2)
15:              state ← state 0
16:          end if
17:      end if
18:  end while

```

FIGURE 7: The default handling incoming traffic at the backhaul gateway.

$$\Delta J(c, s, i, b)_t = \sum_{t=1}^n \frac{\Delta J(c)_t + \Delta J(s)_t + \Delta J(i)_t + \Delta J(b)_t}{4}. \quad (9)$$

$\Delta J(c)_t \geq 0$; $\Delta J(s)_t \geq 0$; $\Delta J(i)_t \geq 0$; and $\Delta J(b)_t \geq 0$, $t = 1, 2, 3, 4, \dots, n$, where $\Delta J(c)_t$, $\Delta J(s)_t$, $\Delta J(i)_t$, and $\Delta J(b)_t$ are the average jitters of conversation, streaming, interactive, and background, respectively.

The communication delays of the system at any time t are conveyed by $D(\text{syst})_t$ and can be formed as

$$D(\text{syst})_t = \sum_{t=1}^n \frac{D_t}{n}, \quad n \geq 0. \quad (10)$$

Corresponding to equation (10), the average delay of the system with four traffic classes can be determined as $D(c, s, i, b)_t$, where

$$D(c, s, i, b)_t = \sum_{t=1}^n \frac{D(c)_t + D(s)_t + D(i)_t + D(b)_t}{4}, \quad (11)$$

$D(c)_t$, $D(s)_t$, $D(i)_t$, and $D(b)_t$, are the average delays of conversation, streaming, interactive, and background, respectively.

2.7.2. Simulation Environments. The experiment was conducted by using a computer simulation program named network simulation version 3 (NS3) that was implemented

by using the C++ library. The simulation topology was composed of RAN area, fronthaul, and backhaul gateway. The RED-queue disc has been used to buffer the incoming traffic to represent the MEC server. The total simulation packets are 1025148, conversation packets are 458226, streaming packets are 532598, interactive packets are 22605, and background packets are 11719. Due to the communication link interval that was configured to 10 milliseconds, simulation times for each traffic class are 600 seconds. There are 8 user devices used for simulation, and the distances were configured with 15 meters between each other.

Figure 8 illustrates that the simulation stages were conducted for experimentation, such as initialization which initializes the state for simulation of conversation, streaming, interactive, and background traffic and will be generated for periodic communication. NCON is referred to as the configuration of network condition, and TCON is the handling of incoming traffic based on the proposed scheme. And finally, it is the collection of the simulated results.

3. Experiment Results and Discussion

In this paper, the system evaluations are based on the comparison between the proposed approach and the conventional approach. Evaluations are regarding the average E2E delays, average E2E jitters, and average throughputs of the individual of each RT communication (conversation and streaming) and total average values integrating RT and NRT in the communication system.

Figure 9 shows the comparative average delays of the proposed approach with the conventional approach. The revaluation results are related to the analysis in equation (11) and the average delays are compared by integrating each average delay value of the four communication classes $D(c)_t$, $D(s)_t$, $D(i)_t$, and $D(b)_t$, respectively. The graph shows that the average delays of the proposed approach are lower than the conventional approach. Referring to the graph, the average value of the proposed approach is mostly 0.01361326 seconds, while the average delay of the conventional approach is mostly 0.013676041 seconds. For the RT communication system, E2E delays have to be ultra low to perform the great QoS for each user. Typically, the backhaul traffic will be reduced rapidly during the increasing of forwarding rate of the RT traffic. The proposed approach can reduce the number of traffic queues in the MEC server and possibly reserve or reduce the MEC resources. With the possibility of the higher forwarding rate at the backhaul, the buffer resources will not be required and lessen the computing resources of network devices.

Figure 10 shows the comparison of average throughputs of the system between the proposed and conventional approaches. Based on the showing graphs, the proposed approach has higher communication throughputs than the conventional. The average throughputs are relying on the average E2E delays and PDU sizes. The throughputs can be calculated by division of PDU size with communication delays. In this paper, the PDU size was configured as constant; thus, the throughput will be varied based on communication delays. The evaluation was conducted by

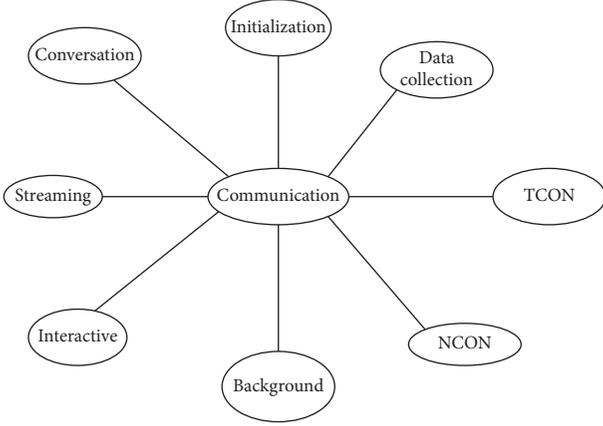


FIGURE 8: The simulation diagram.

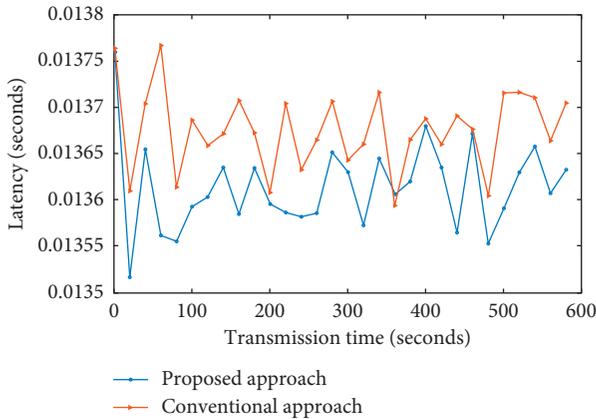


FIGURE 9: The comparison of average delays with integrating both RT and NRT.

calculating the average throughputs of conversation, streaming, interactive, and background communication and sum as a total average. The average E2E delays of the proposed approach are lower than the conventional, as shown in Figure 9.

The proposed scheme enhances the higher communication capacity of forwarding incoming traffic. So, the heavy user traffic in the backhaul gateway and bad statuses can be reduced. Moreover, this proposed approach is suitable to use for handling massive 5G user traffic, as well as enhancing QoS for RT communication classes.

Figure 11 shows that communication jitters of the proposed approach are lower if compared to the jitters of the conventional approach. The jitter evaluation is based on equation (9). The average jitters are compared based on each average of communication class, including average jitters of conversation, streaming, interactive, and background denoted as $\Delta J(c)_t$, $\Delta J(s)_t$, $\Delta J(i)_t$, and $\Delta J(b)_t$, respectively. Lower jitters are indicating the communication stability of the network. The communication jitter will occur when the backhaul network becomes congested. Consequently, the serving interval at gateway will be varied based on the obvious situations; when the network fluctuation of the serving times are higher, the communication QoS will be

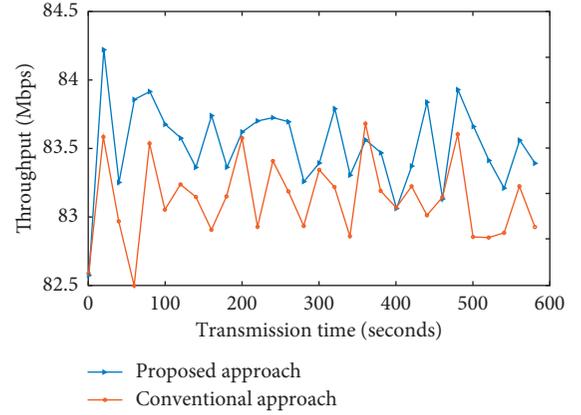


FIGURE 10: The average communication throughputs of the system.

decreased concurrently. In the RT communication, especially, ultralow communication jitters are required. The proposed approach provides ultra-low jitters in communication systems, so the E2E communication jitter will be consistent.

The comparison of E2E communication jitters of conversation and streaming is presented in Figures 12(a) and 12(b), respectively. Figures 12(a) and 12(b) illustrate that the E2E jitters of the proposed approach outperform the conventional for both conversation and streaming traffic classes. The jitter evaluation was analyzed based on equation (8). Based on the graphs in Figure 12(a), the jitters of conversation traffic class of the proposed scheme have been improved, because the proposed scheme is restricted on the time-insensitive user traffic serving rate and increased the serving rate for the conversation traffic class. Thus, the communication stability can increase. The streaming jitters are shown in Figure 12(b). The graphs show that the E2E jitters of streaming traffic class of the proposed scheme have been improved, while there are higher jitters occurring in the conventional approach. Based on the graphs, the proposed scheme is significant to control the serving resources to enhance the quality of services for time-sensitive communications.

The E2E delays comparison between the proposed and conventional schemes of conversation and streaming communication is exhibited in Figures 13(a) and 13(b), respectively. The evaluation graphs in both Figures 13(a) and 13(b) were analyzed based on equation (10), in the above section. As shown in the evaluation graphs, the proposed approach has lower communication delays for both conversation and streaming, while the conventional approach has higher communication delays for both conversation and streaming communications. Due to the scheme being targeted for RT classes, the serving rate of NRT classes has been restricted and increased the serving rate of RT classes. So, the waiting time of NRT classes will be increased while the waiting time of RT will be reduced. However, the network performance of time-insensitive traffic does not rely on communication times.

Figures 14(a) and 14(b) show the comparison of average throughputs between the proposed and conventional

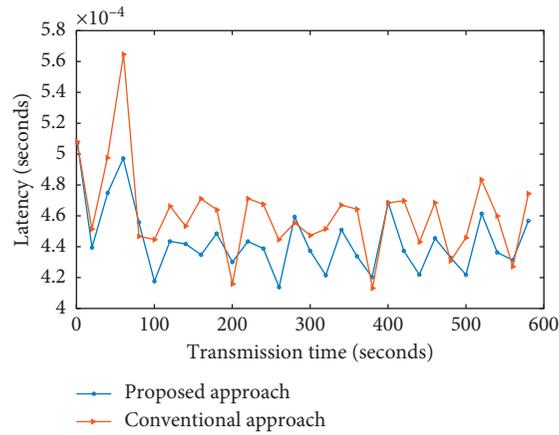


FIGURE 11: The comparison of average E2E communication jitters.

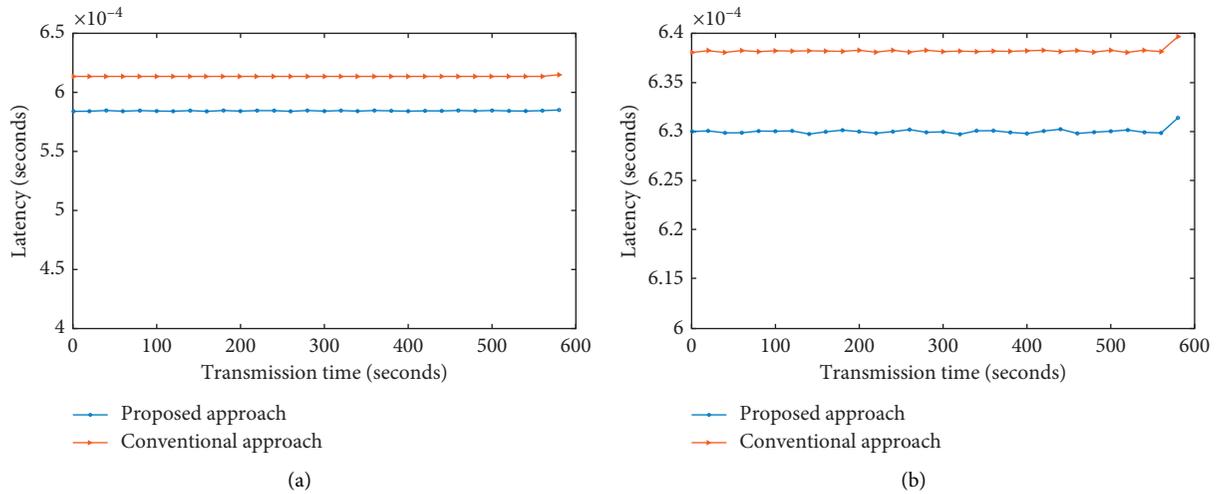


FIGURE 12: The comparison of average E2E communication jitters between the proposed and conventional approach of conversation (a) and streaming (b).

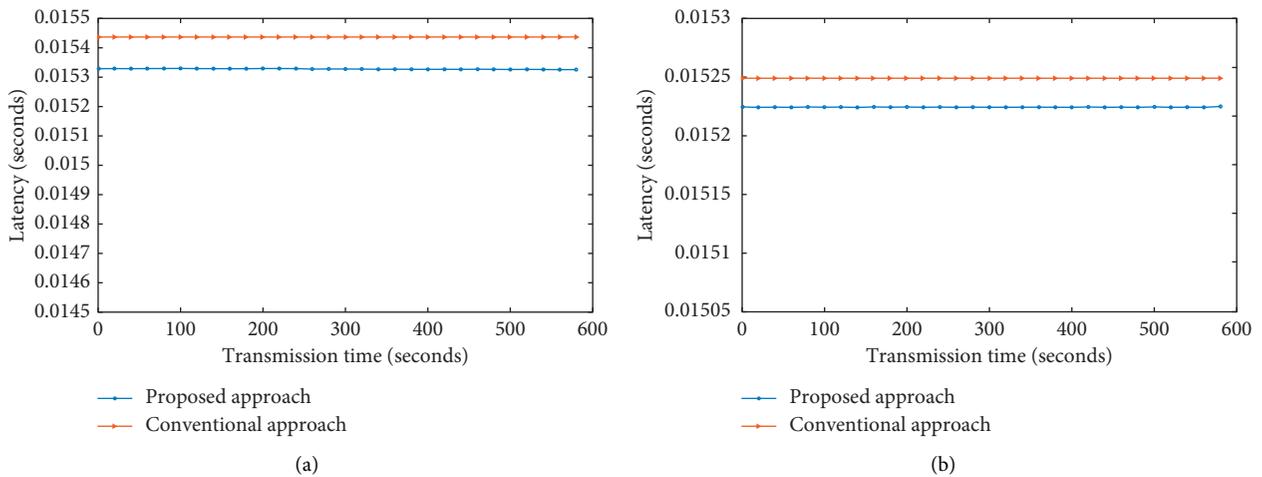


FIGURE 13: The comparison of average E2E communication delays between the proposed and conventional approach of conversation (a) and streaming (b) communication.

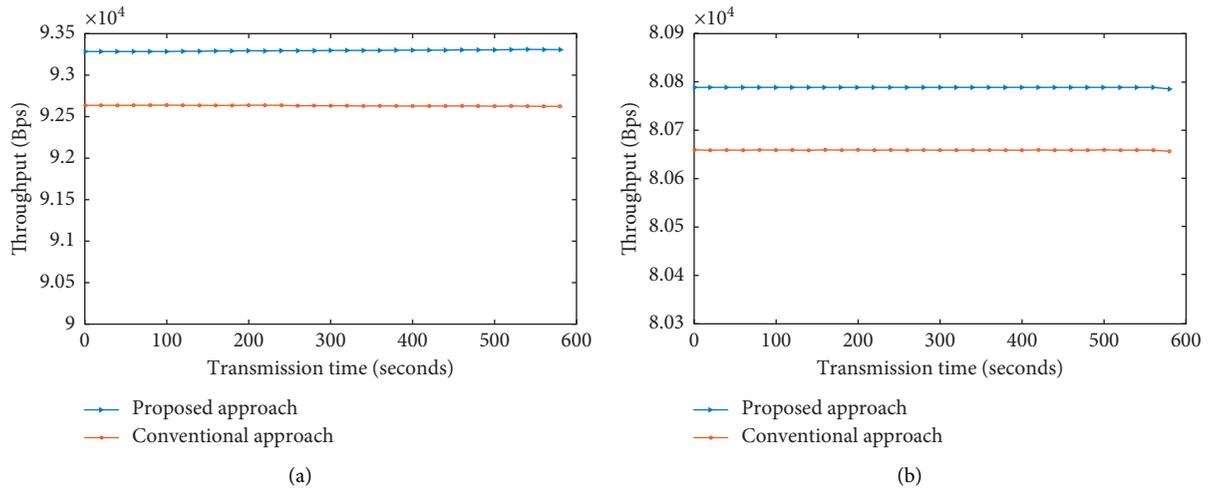


FIGURE 14: The comparison of average E2E communication throughputs between the proposed and conventional approach of conversation (a) and streaming (b) communication.

approaches for conversation and streaming communication. The graph shows that the proposed approach has higher communication throughputs for both conversation and streaming, while the throughputs of the conventional approach are lower. Because the E2E delays have been reduced in the proposed scheme, as shown in Figures 13(a) and 13(b) above, communication throughputs also improve relying on communication delays. Based on these evaluations, the proposed scheme enhances the communication throughputs for RT communication classes. According to the results in Figures 12–14, it is shown that the proposed scheme can be used to improve the network performance for RT communications. This proposed scheme, especially, meets the RT QoS perspectives and is suitable to be applied in bottleneck 5G backhaul gateway.

4. Conclusions

The 5G backhaul gateway consists of massive incoming traffic from heterogeneous devices with a variety of communication traffic. Thus, it is necessary to handle the communication traffic based on each traffic class, especially for RT communication that required ultra-low latency and higher communication rates more than the NRT traffic classes. The proposed approach handles the incoming traffic based on classifying the user traffic into four different classes, including conversation, streaming, interactive, and background communication. The MEC servers have been used to integrate with the backhaul gateway to buffer each of the traffic classes individually. Each communication class has an individual MEC server. When the backhaul is considered as a bad status, the proposed approach will be used to handle by giving more communication rates to RT (conversation and streaming) and reducing the communication rates of NRT based on the increasing ratio of the RT communication. Based on the simulation results, the proposed approach enhances the QoS over the conventional approach for RT communication in terms of reducing jitters, delays, and

enhancing higher communication throughputs. This approach is suitable for enhancing QoS for RT in bottleneck 5G backhaul network environments and the privacy protection for each communication class based on Network Slicing. Finally, for further research, we aim to integrate more effective methods to enhance the massive user traffic in the bottleneck area.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Korea Electric Power Corporation (Grant no. R18XA02), and this work was supported by the Soonchunhyang University Research Fund.

References

- [1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [2] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3098–3130, 2018.
- [3] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, "Trust-oriented IoT service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2020.
- [4] C. Xue, C. Lin, and J. Hu, "Scalability analysis of request scheduling in cloud computing," *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 249–261, 2019.

- [5] D. Shen, J. Luo, F. Dong, and J. Zhang, "VirtCo: joint coflow scheduling and virtual machine placement in cloud data centers," *Tsinghua Science and Technology*, vol. 24, no. 5, pp. 630–644, 2019.
- [6] L. Liu, X. Chen, Z. Lu, L. Wang, and X. Wen, "Mobile-edge computing framework with data compression for wireless network in energy Internet," *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 271–280, 2019.
- [7] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, "Dynamic mobile crowdsourcing selection for electricity load forecasting," *IEEE Access*, vol. 6, pp. 46926–46937, 2018.
- [8] Y. Xu, L. Qi, W. Dou, and J. Yu, "Privacy-preserving and scalable service recommendation based on SimHash in a distributed cloud environment," *Complexity*, vol. 2017, Article ID 3437854, 9 pages, 2017.
- [9] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2616–2624, 2017.
- [10] X. Xu, H. Cao, Q. Geng, X. Liu, F. Dai, and C. Wang, "Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment," *Concurrency and Computation: Practice and Experience*, 2020.
- [11] L. M. Pham and T. Nguyen, "Flexible deployment of component-based distributed applications on the cloud and beyond," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 3, pp. 1141–1163, 2019.
- [12] J. W. Jang, S. Kwon, S. Kim, J. Seo, J. Oh, and K. Lee, "Cybersecurity framework for IIoT-based power system connected to microgrid," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2221–2235, 2020.
- [13] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlation graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, 2019.
- [14] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified LSH-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, 2020.
- [15] L. Liu, Y. Du, and Q. Fan, "A constrained multi-objective computation offloading algorithm in the mobile cloud computing environment," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 9, pp. 4329–4348, 2019.
- [16] W. Tang, B. Qin, Y. Li, and Q. Wu, "Functional privacy-preserving outsourcing scheme with computation verifiability in fog computing," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 1, pp. 281–298, 2020.
- [17] A. Rasheed, P. H. J. Chong, I. W. Ho, X. J. Li, and W. Liu, "An overview of mobile edge computing: architecture," technology and direction," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 10, pp. 4849–4864, 2019.
- [18] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Generation Computer Systems*, vol. 88, pp. 636–643, 2018.
- [19] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.
- [20] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: a systematic survey," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2019.
- [21] J. Xie, F. Richard Yu, T. Huang et al., "A survey of machine learning techniques applied to software defined networking (SDN): research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, 2019.
- [22] M. Finsterbusch, C. Richter, E. Rocha, J. Muller, and K. Hanssgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1135–1156, 2014.
- [23] L. Velasco, L. Gifre, J.-L. Izquierdo-Zaragoza et al., "An architecture to support autonomic slice networking," *Journal of Lightwave Technology*, vol. 36, no. 1, pp. 135–141, 2018.
- [24] F. A. Lopes, M. Santos, R. Fidalgo, and S. Fernandes, "A software engineering perspective on SDN programmability," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1255–1272, 2016.
- [25] C. Song, M. Zhang, Y. Zhan et al., "Hierarchical edge cloud enabling network slicing for 5G optical fronthaul," *Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B60–B70, 2019.
- [26] Z. Zaidi, V. Friderikos, Z. Yousef, S. Fletcher, M. Dohler, and H. Aghvami, "Will SDN be part of 5G?" *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3220–3258, 2018.
- [27] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3075849, 8 pages, 2018.
- [28] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "Be-Come: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.
- [29] J. Li, X. Shen, L. Chen, J. Ou, L. Wosinska, and J. Chen, "Delay-aware bandwidth slicing for service migration in mobile backhaul networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B1–B9, 2019.
- [30] D. A. Chekired, M. A. Togou, L. Khoukhi, and A. Ksentini, "5G-Slicing-enabled scalable SDN core network: toward an ultra-low latency of autonomous driving service," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1769–1782, 2019.
- [31] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.
- [32] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.
- [33] X. Li, C. Guo, L. Gupta, and R. Jain, "Efficient and secure 5G core network slice provisioning based on VIKOR approach," *IEEE Access*, vol. 7, pp. 150517–150529, 2019.
- [34] S. Kim and W. Na, "Safe data transmission architecture based on cloud for Internet of things," *Wireless Personal Communications*, vol. 86, no. 1, pp. 287–300, 2015.