

Intelligent Massive Traffic Handling Scheme in 5G Bottleneck Backhaul Networks

Prohim Tam¹, Sa Math¹, and Seokhoon Kim^{2*}

¹ Department of Software Convergence, Soonchunhyang University
Asan-si, Chungcheongnam-do 31538, Republic of Korea
[e-mail: prohimtam@gmail.com, mathsa98@gmail.com]

² Department of Computer Software Engineering, Soonchunhyang University
Asan-si, Chungcheongnam-do 31538, Republic of Korea
[e-mail: seokhoon@sch.ac.kr]

*Corresponding author: Seokhoon Kim

*Received January 10, 2021; revised February 10, 2021; accepted February 18, 2021;
published March 31, 2021*

Abstract

With the widespread deployment of the fifth-generation (5G) communication networks, various real-time applications are rapidly increasing and generating massive traffic on backhaul network environments. In this scenario, network congestion will occur when the communication and computation resources exceed the maximum available capacity, which severely degrades the network performance. To alleviate this problem, this paper proposed an intelligent resource allocation (IRA) to integrate with the extant resource adjustment (ERA) approach mainly based on the convergence of support vector machine (SVM) algorithm, software-defined networking (SDN), and mobile edge computing (MEC) paradigms. The proposed scheme acquires predictable schedules to adapt the downlink (DL) transmission towards off-peak hour intervals as a predominant priority. Accordingly, the peak hour bandwidth resources for serving real-time uplink (UL) transmission enlarge its capacity for a variety of mission-critical applications. Furthermore, to advance and boost gateway computation resources, MEC servers are implemented and integrated with the proposed scheme in this study. In the conclusive simulation results, the performance evaluation analyzes and compares the proposed scheme with the conventional approach over a variety of QoS metrics including network delay, jitter, packet drop ratio, packet delivery ratio, and throughput.

Keywords: Cloud Computing, Machine Learning, Massive Traffic, Resource Allocation, Software-Defined Networking

1. Introduction

In the perspective of future demands in real-time application services over the fifth-generation (5G) communication networks, the competence of intelligent resource allocation, control, and management is a significant characteristic to acquire for serving and reaching adequate Quality of Service (QoS) requirements [1, 2]. Fig. 1 illustrates the model environment in the future perspective which consists of massive multi-input and multi-output (MIMO), various types of communication technologies and devices, heterogeneous Internet of Things (HetIoT), diversified real-time 5G application scenarios, device-to-device (D2D) communication, and mobile edge computing (MEC) technology. Moreover, the extensive scale of cloud radio access networks (RAN) architecture allows centralized base-band units (BBU) pool deployment to process dynamic resource allocation for multiple distributed radio remote heads (RRHs), which reasonably generates sophisticated data flow for management and control purposes [3]. Therefore, due to these ultra-dense networks (UDN) scenario with massive multi-cell connectivity including macrocells, microcells, femtocells, and picocells, it makes the network resource management solution a challenging objective to accomplish. The real-time communication traffic over uplink (UL) transmission in 5G backhaul network environment requires appropriate communication and computation capacities for stable serving, which necessitates being not weakened by the excessive downlink (DL) transmission, particularly, extravagant peak hour caching content placement and update traffic flows [4]. In the meantime, the insufficiency of computation resources is feasible to take place in case there are inadequate distributed computing capabilities. Furthermore, nowadays network resource management and orchestration still have not reached the satisfied control level in terms of global view monitoring functionality, self-management, flexibility, scalability, and intelligibility, which possibly degrades the network performances and Quality of Experience (QoE) expectations.

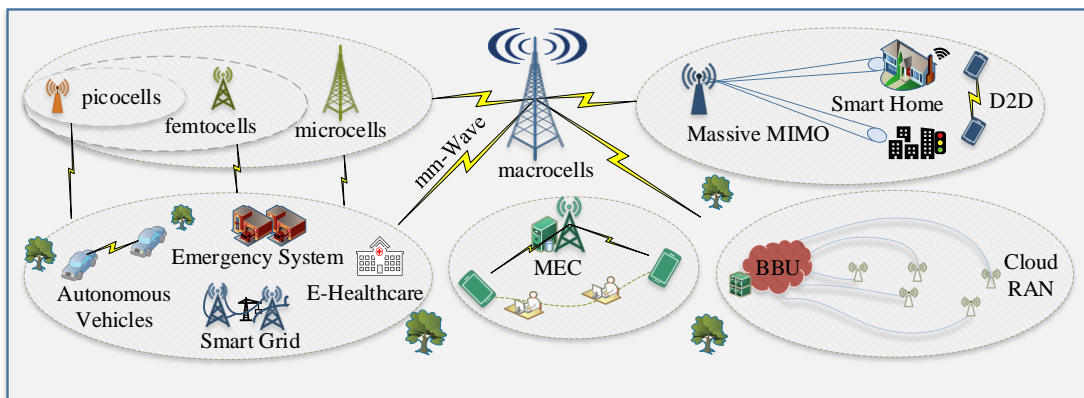


Fig. 1. Future perspective of model communication networks

To handle the above-mentioned difficulties, an optimal solution has to dynamically decouple the control plane and data plane for sufficient content request patterns gathering and global control purposes, partition the communication resource flexibly for adjusting between UL and DL transmission, and deal with the stability of computation resource in 5G core networks and backhaul networks conditions [5-9]. To enable the global view of the network control, forwarding devices, and application services, software-defined networking (SDN) is an optimal paradigm which provides a programmable and scalable architecture for central

orchestration with various available OpenFlow controllers (e.g., RYU, OpenDayLight, POX, NOX) [10-13]. Furthermore, as a major branch of artificial intelligence, machine learning algorithms feasibly overcome manifold challenges and provide significant end-goals such as content classification, prediction, user content request pattern analysis, continuous state/action improvement in caching scenarios, and particularly link bandwidth resource detection for handling real-time communication stability [14]. On top of that, MEC paradigm enables the extraction from mobile cloud computing (MCC) capacities to enhance the distributed network environment, which allows the edge network devices to obtain stable network resources, cache numerous contents with high hit probability, minimize the communication delay, and handle network congestion circumstances [15].

In this paper, intelligent resource allocation (IRA) and extant resource adjustment (ERA) schemes on UL and DL transmission approach with machine learning, SDN, and MEC paradigms for handling 5G real-time communications are proposed. The contributions of the paper are in the following three main procedures. Primarily, an SDN-oriented architecture for effective communication and computation network resource control is presented in the concept of multiple MEC servers to inspect and detect UL congestion statuses and DL resource usages, respectively. In the current network system, the main target is only on centralized control and consists of inadequate real-time solutions with complicated infrastructure between data and control flows [16]. Secondly, an intelligent SDN controller configuration is proposed in the concept of applying a machine learning algorithm, namely support vector machine (SVM), for detection and prediction purposes. The first function of the algorithm is to detect real-time UL traffic conditions and calculate the communication resource requirement. Additionally, the UL traffic can be detected as two conditions whether it requires extra serving resources or not. The second function of the algorithm is to predict the schedule of off-peak hour proactive caching for DL traffic recommendation. In the last procedure of contributions, MEC-driven framework is proposed at the bottleneck area of the backhaul network environment when the UL transmission reaches the conventional resource capacities limit. Moreover, to evaluate the proposed scheme, an end-to-end (E2E) simulation is conducted to illustrate the comparison with conventional schemes in terms of various QoS performances.

The content of this paper is structured as follows. The related works are described in section 2. The details of the proposed schemes on UL and DL transmission are given in section 3. In section 4, the simulation environment, performance metrics, and results are discussed. Finally, a summarizing conclusion is presented in section 5.

2. Related Work

This section describes the existing researches and complementary paradigms regarding our proposed topic. The related work is organized into four parts including studies on SDN, MEC, machine learning, and working process of multiple technologies convergence for 5G backhaul network scenario.

SDN paradigm is a promising enabler technology towards a scalable and flexible solution for effective network management by decoupling data plane from control plane and providing a centralized control view of the network conditions [17]. With the complexity and heterogeneity of numerous smart devices, managing the communication and computation resources with traditional routing aspects in 5G backhaul network environment is utterly not satisfied [18]. Therefore, the emerging SDN will alleviate the drawback and presents the intelligent control system for adaptable routing optimization, cross-layer architecture, fault-tolerance, and load balancing in the core network. SDN-based controllers in distributed

networks are proposed to handle manifold shortcoming in HetIoT circumstances with wide-ranging area by optimizing path selection and developing Hybrid-Edge switches for the heterogeneity of end devices, and SDN controller capability is feasible to be configured adaptively for traffic flows offloading decision and selection among multiple controllers in a particular scenario to reduce the flow installation delay and provide optimal QoS performance [19, 20]. Moreover, the SDN-based technique has been applied to deal with network maintenance perspectives by detecting the deficiency possibility and rapidly utilizing the controller to update the optimal flow path installation without longer duration extensions in 5G backhaul network environment [21]. The interfaces between the application plane, control plane, and data plane allow the resourceful programmability to manage and orchestrate the backhaul network resource circumstances which outperform the traditional architecture based on manual control in proprietary and standard network devices [22]. To reach the reliable scalability, optimal network throughput, and secure connectivity in SDN architecture, network function virtualization (NFV) is applicable to enable and modify multicasting functionality of diversified application services [23]. However, SDN paradigm still requires enhancing multiple directions to cope with the future perspective requirement including controller placement, high-quality data plane information gathering, and adaptable flow installation rules for mission-critical and non-mission-critical applications.

The proficiency and plane separation techniques of SDN eases the implementation of machine learning algorithms which have been continuously applied to meet the communication and computation control in the backhaul network environment. Machine learning provides intelligent model construction and validation by using various algorithm options from supervised learning, semi-supervised learning, unsupervised learning, and reinforcement learning [24]. The model evolves into unified network adaption for problem formulation, data collection, and practical features extraction; therefore, the model can discover the hidden pattern of the historical network state environment towards specific requirement including massive traffic clustering, QoS and QoE classification, routing path recommendation, or resource allocation management [25]. To engage the network efficiency, machine learning is used to predict the MEC service allocation in 5G backhaul networks. Additionally, based on MEC paradigm, distributed caching storages are feasible to acquire for edge caching framework which alleviates the possible congestion in backhaul links by the duplication of numerous content requests. The convergence of these particular technologies activates plenty of functional supports to improve network QoE performance. Fig. 2 presents the working process of multiple technologies convergence including SDN, machine learning, and MEC. Every cluster node forwards the data to the other cluster node destination by D2D communication or to cluster heads (e.g., OpenFlow switch). When the controller gathers the information from end devices, OpenFlow, the first standard SDN communication protocol, queries the flow entries table of each cluster head by using PACKET_IN and PACKET_OUT messages [26]. Based on the requirement from application plane through the representational state transfer (RESTful) application programming interface (API) in terms of resource management, network monitoring, performance analysis, etc., SDN controller will collect the related data features to store in a virtualized database (e.g., hypervisor-based, virtual resource pool, or processing resource sharing entities) which are synchronously updated and processed [27]. By the existence of resourceful datasets, machine learning can output intelligent decisions in terms of prediction, recommendation, classification, detection, or inspection, for particular conditions. The tasks will be offloaded to MEC servers for computing purposes. When the computation is finished, SDN controller accordingly configures flow table rules, priority indication, and other data plane characteristics for optimal network performance.

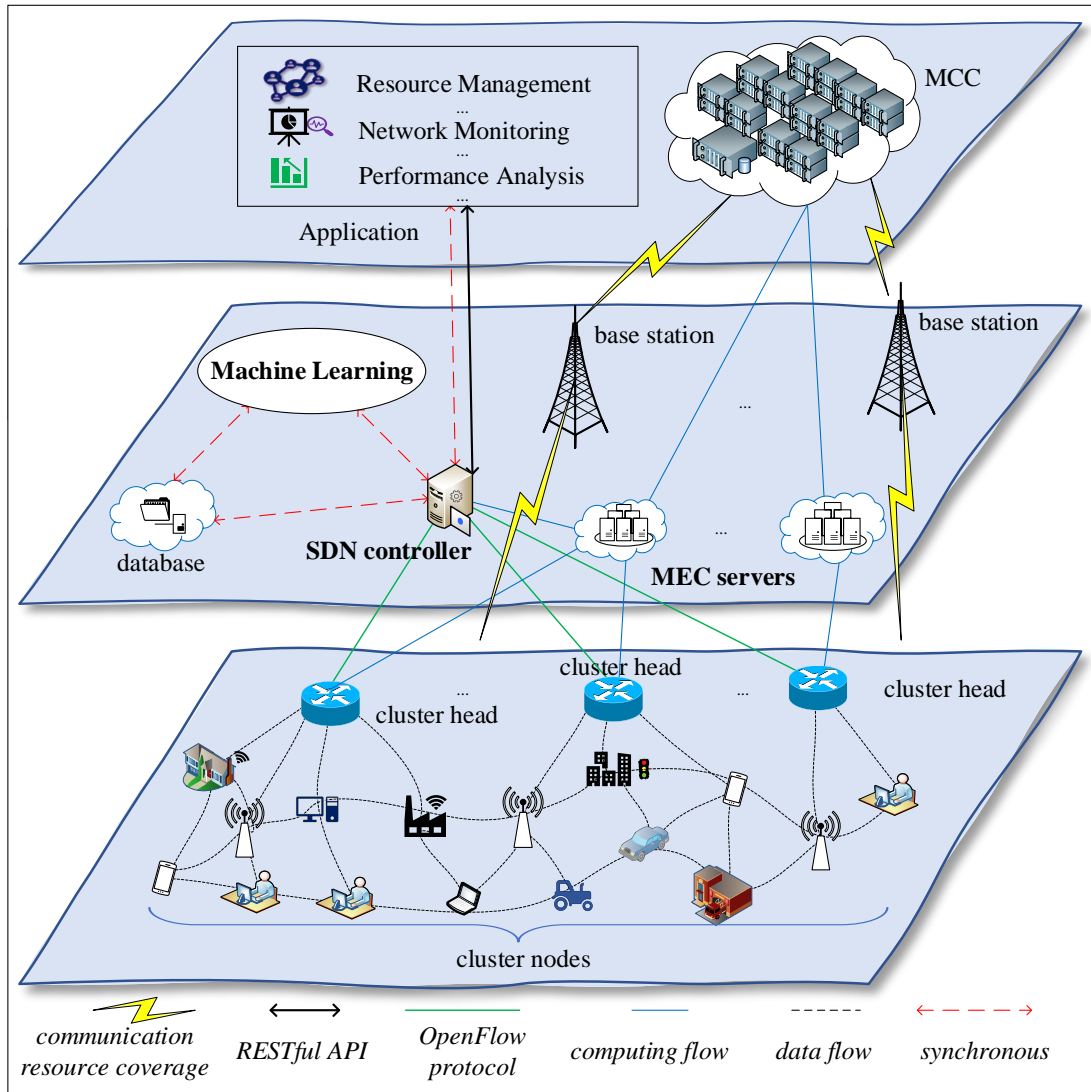


Fig. 2. The working flow of SDN, MEC, and machine learning convergence

3. Proposed Approach

In this section, we present detailed procedures on how to handle and allocate network resources for intelligent UL and DL transmission in 5G real-time communication scenarios. The proposed schemes can be forged in three main steps including an SDN-oriented architecture for sufficient and global network information gathering, an intelligent controller configuration for DL caching and UL real-time congestion detection, and the MEC-driven framework for sustaining network resources in 5G backhaul network environment.

3.1 SDN-oriented Architecture

To reduce the complexity of the network environment, SDN is required to modify and implement on top of the traditional network architecture. In this context, SDN gathers the real-time generated traffic from data plane to measure with the serving gateway (SGW) and packet

data network gateway (PGW) link characteristics in terms of bandwidth, queue size, delay, etc. By utilizing OpenFlow protocol, SDN-oriented architecture can adaptively manage the network environment and gather the information from end devices critically. In contrast with traditional architecture, the real-time traffic packets are not extracted by only just the header of packets but also various features including link bandwidth, delay, protocol data unit (PDU) sizes, congestion window, etc., by using the southbound interface (SBI) or control-data plane interface (C-DPI). Based on these features, the UL congestion level can be detected by the configured algorithm in SDN controller. Furthermore, the northbound interface (NBI) or control-application plane interface (C-API) also plays an essential role to import the congestion level characteristics as the class target using RESTful API. Fig. 3 illustrates the network scenario with SDN-oriented architecture. User devices are controlled and monitored by using reactive flow installation modes to set the traffic rules within SDN controller in terms of table id, idle timeout, hard timeout, priority numbers, match, and actions. By setting these attributes reactively, the controller can compute, detect, and predict the congestion possibility strictly. With the mobility of users, the DL caching transmission requires higher link bandwidth and throughput. Therefore, the caching servers are attached within the base stations for clustered contents storage. Next, the SGW and PGW transmissions perform significant roles for SDN controller to identify the bottleneck and congestion level in the backhaul network environment by control flow dispatch. On top of that, MEC server is allocated as a serving resource pool to handle high-level congestion purposes.

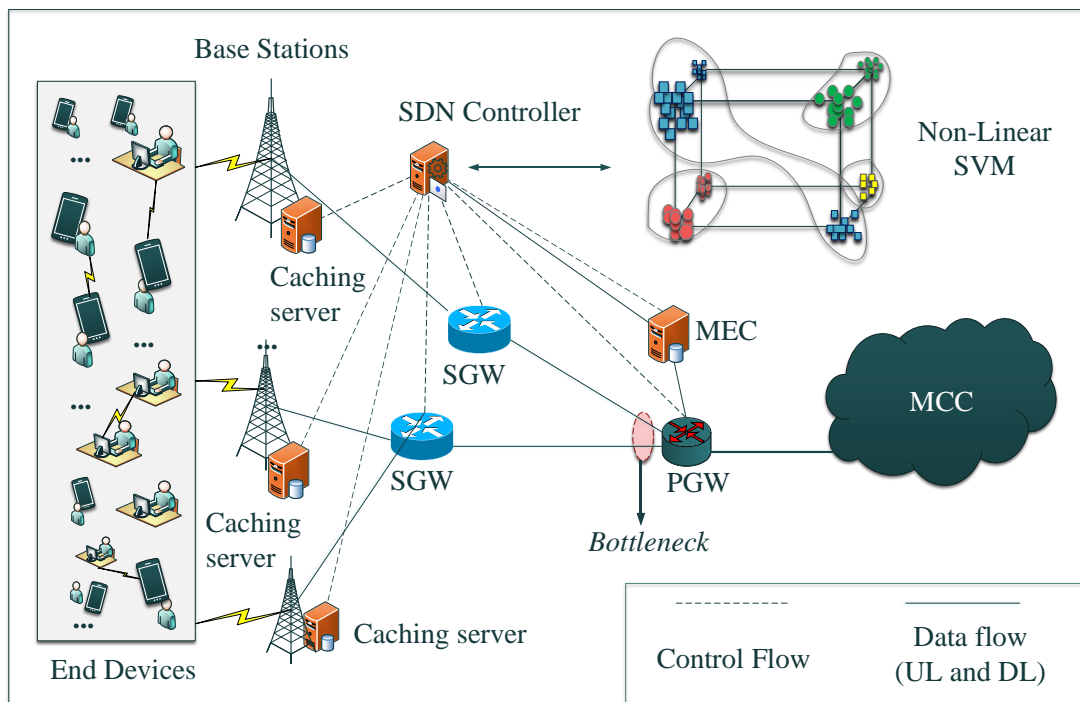


Fig. 3. SDN-oriented architecture

The level target of UL congestion has to be categorized intelligently, precisely, and rapidly in order to provide a satisfied and well-classified QoS for real-time communications. There are three main targets in this detection progress. Level-0 indicates the scenario that the UL transmission meets the resource capacity and does not require extra serving resources, in other

words, the DL transmission is in slight traffic flow. Level-1 expresses the overwhelmed and massive traffic flow of DL caching which severely degrades the real-time UL transmission, consequentially, it makes UL traffic requiring extra serving communication resources. In level-1 situations, SDN controller requires to proactively predict and recommend solutions to prevent further packet drop circumstances. For level-2, the entire network resources cannot handle the congestion statuses and necessitate to have an extraordinary computing entity.

3.2 Controller Configuration

To detect the level- n congestion statuses, a supervised machine learning algorithm, SVM, is applied to facilitate the SDN controller configuration. For each level, different actions are recommended to ensure that the proposed system can handle resource allocation in various circumstances. In level-0 condition, DL transmission is seldom endorsed to cache in off-peak hour schedule prediction for preventing future congestion. For level-1, DL transmission caused harmful consequences for real-time UL traffic, therefore, the resource adjustment and critical caching schedule, namely ERA approach, have to be well-configured as follows.

3.2.1 Extant Resource Adjustment for Real-Time UL Traffic

At this stage, based on real-time packet features captured by C-DPI and congestion targets captured by C-API, the labeled datasets were generated and synchronously updated in order to supervise the algorithm and improve the accuracy of resource adjustment procedures. Primarily, to inspect and classify the volume of real-time traffic, the model has to be intelligently trained, and the SVM kernel has to be accordingly selected to fit the non-linear data types. However, choosing the matching kernel is computationally discreet.

Table 1. Optimal SVM kernels selection for SVC constructor

Algorithm 01	
Require:	Let $X' = \{x_1, x_2, x_3, \dots, x_m\}$ denotes m featuring numbers of training datasets $Y' = \{y_1, y_2, y_3, \dots, y_n\}$ denotes level- n congestion target values and (X, Y) are the testing datasets, respectively.
Ensure:	optimal kernel with satisfying precision
1.	Initialize One-Vs-All Classifier $h_{\theta}^{(n)}(X')$;
2.	Polynomial kernel with degree- d polynomials: $K(X', Y') = (X'^T Y' + c)^d$, $c \geq 0$
3.	Hyperbolic Tangent kernel with slope α and intercept β , where $\alpha = 1/n$ $K(X', Y') = \tanh(\alpha X'^T Y' + \beta)$, $\beta \geq 0$
4.	Gaussian kernel with adjustable sigma σ : RFB Kernel of $(X', Y') = e^{-\frac{\ X' - Y'\ ^2}{2\sigma^2}}$ alternatively, $K(X', Y') = e^{-\gamma \ X' - Y'\ ^2}$, $\gamma > 0$
5.	For each kernel: applying on testing prediction $K(X, Y)$ and append $h_{\theta}^{(n)}(x_m) = P(y = n x_m; \theta)$.
6.	Compute the $\max_n h_{\theta}^{(n)}(X')$;
7.	Select the optimal kernel to construct SVC

Therefore, **Table 1** illustrates the optimal SVM kernels selection for level- n congestion status identification. To train the support vector classifier (SVC), selecting an optimal kernel is highly significant to effectively fit the input data and output the highest possible precision.

Firstly, the total m numbers of training traffic datasets are denoted as X' ($x_1, x_2, x_3, \dots, x_m$), and the n level congestion targets are denoted as Y' ($y_1, y_2, y_3, \dots, y_n$) into the algorithms. The algorithm is well replacing the null values, dummy variables, and outliers. Next, the algorithm defined the class target to drop for the score calculation purpose. To construct an optimal One-Vs-All (OvA) classifier $h_{\theta}^{(n)}(X')$, which considers on multiple target classes n estimation over the probability that y_n belongs by the given X' and parametrized θ , three selected SVM kernels including polynomial, hyperbolic tangent, and gaussian radial basis function (RBF) kernels, were looped through and formulated concurrently to correspond with the OvA. Each output was appended to accumulate for performance evaluation. In this context of SVC, the non-linear transformation is highly essential, which is executed by nonparametric method of modifying into T maps based on the vector sets. For gaussian RBF kernel, the radial basis algorithm was used to enhance the transformation process, where γ represents the fraction of adjustable parameter σ with experimental estimation. Among each kernel output, only one function was selected to carry on the next stage by computing the maximum scaling of the OvA classifier on training datasets X' , based on kernelized module $K(X', Y')$, as $\max_n h_{\theta}^{(n)}(X')$. At last, the optimal kernel with satisfying precision for the training model and well-classified SVC were ensured.

$$tx_{rate} = \sum_{i=0}^n R_{(n)} - \sum_{i=0}^m dl_{(m)}, \forall R \in N^+ \text{ and } \forall dl \in N \quad (1)$$

After the congestion level was labeled, the action structure has to be configured for inspecting the communication resource availability, $R_{(n)}$, in the serving n computing entities, and detecting the irrelevant and insignificant DL transmission, $dl_{(m)}$, with total m amount. Therefore, the resource adjustment configuration, tx_{rate} , for real-time UL traffic can be formulated as (1).

3.2.2 Proactive Caching Schedules for Non-Real-Time DL Offloading

Based on the previous stage formulation, the congestion level status in the backhaul network communication is detected, therefore, in this stage, we ensure the DL caching schedule to mismatch with level 1 and 2 circumstances. Thus, a proactive caching scheme is applied to enable off-peak hour transmission, maximize peak hour network throughput, and contribute spectrum and energy effectiveness. The content criteria have to specifically identify with cache hit probability for each cluster of users. However, the major drawback is the caching performances which relied utterly on prediction precision. Accordingly, based on optimal SVM kernels selection flow, SVC decently supports to implement in this scenario. By modifying the features of training datasets, X' , with content request patterns of the user within each base station including popular preferences, social networking, historical interests, timestamps, mobility pattern, etc., the caching efficiency with SVM can be improved and served the demands on peak hour intervals. Fig. 4 illustrates the flowchart of predictive SVM scheduling on DL proactive caching transmission. By cooperating with the resource adjustment scheme, the proactive caching can assign weight score on off-peak hour interval which identifies mainly based on congestion level circumstances. Consequently, the user request pattern captured by C-DPI and edge caching servers is applied to the training model of SVC. If the data does not fit, each capture feature of the datasets will be processed and analyzed again. If the data fits, the time-interval targets will be generated and appended to the

lists of DL traffic availability. Among the particular time-interval, the congestion level is differentiated from one another. Therefore, the optimal time-interval is selected when the congestion values, namely *target_error*, are satisfied with the C-API requirement which can be nonexistent or close to zero. At last, the proactive caching schedules are well-prepared and well-configured in the proposed scheme to recommend for non-real-time DL content offloading transmission.

3.3 MEC-Driven Framework

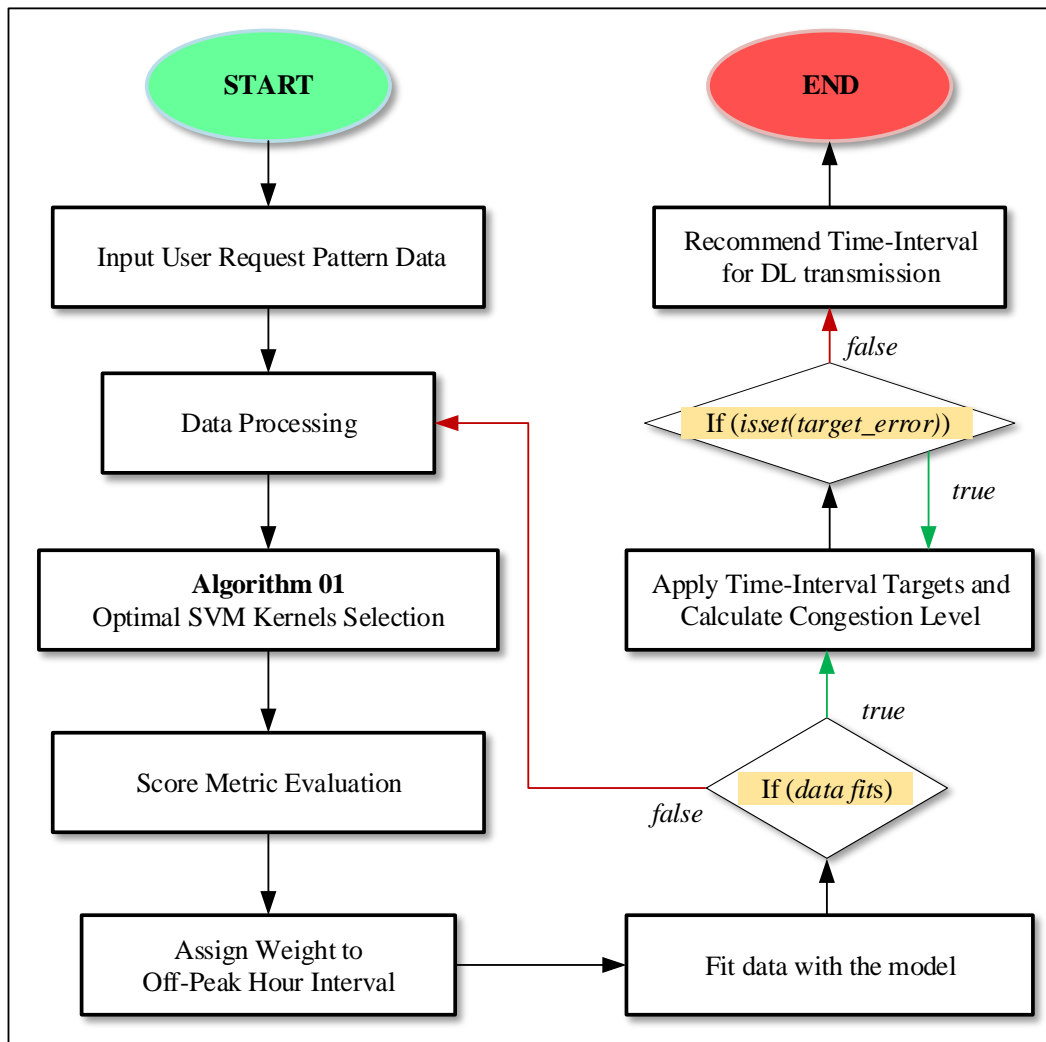


Fig. 4. Flowchart of off-peak hour interval recommendation for DL caching

To adapt and modify the ERA towards IRA scheme, MEC-driven framework is introduced. This approach is to handle the possible challenging drawback in 5G backhaul network environment where the bottleneck area is severely congested and requires absorbing extra serving communication and computation resources emergently from MEC entities. By converging with MEC paradigm, the approach detects the congestion levels and statuses in order to allocate the distributed entities where it predictively matters the most. Based on our SDN-oriented architecture, MEC entities are being placed with identical resource capacity,

$C_{(k)}^R$, where k denotes the index of entity numbers. Equation (2) presents the formulation by taking the ERA to modify with overall equipped $C_{(k)}^R$.

$$tx_{rate} = \sum_{i=0}^k C_{(k)}^R + \sum_{i=0}^n R_{(n)} - \sum_{i=0}^m dl_{(m)}, \forall C^R \& R \in N^+ \text{ and } \forall dl \in N \quad (2)$$

4. Performance Evaluation

This section discusses the mechanism to conduct the simulation results for the proposed schemes performance evaluation. Firstly, the simulation system is distinctly described. Next, performance metrics include the QoS parameters which we aim to compare with the conventional scheme. At last, the overall results are discussed in detail to anticipate the contribution of our proposed schemes.

4.1 Simulation System

The simulation system consists of three preeminent steps including real-time traffic generating, control system configuration, and performance metrics capture. To generate real-time traffics casually, a discrete-event network simulator for Internet systems, NS3, was used and simulated to 450 seconds (s). For control system configuration, an open-source machine learning library, Scikit-Learn, was utilized to perform overall SVM functions. The training datasets for user request pattern and UL detection levels are theoretically created by Python programming language. At last, controlled delay (CoDel) queue model was selected to apply in the simulation system for scheduling the network traffic, handling the buffer sizes, average queue length, packet drops probability, and capturing the potential QoS parameters.

4.2 Performance Metrics

In this section, the QoS metrics, which are used to evaluate the comparison between the proposed scheme and the conventional scheme, are introduced as follows:

- *delay* refers to the sum of delaying time that postpones from the sending node to the receiving node including processing delay, propagation delay, queueing delay, transmission delay, and control delay, which are denoted as $D_{(n)}^{proc}$, $D_{(n)}^{prop}$, $D_{(n)}^q$, $D_{(n)}^t$, and $D_{(n)}^{control}$, respectively. And n refers to the index number of queueing buffers that the packets pass.

$$delay = \sum_{k=1}^n (D_{(n)}^{proc} + D_{(n)}^{prop} + D_{(n)}^q + D_{(n)}^t + D_{(n)}^{control}) \quad (3)$$

- *jitter* refers to the sum of deviation between each delay difference, $D_{(n)}^{peak}$, which causes the instability of the network performance and mostly refers to the peak hour packet transmission in n times.

$$jitter = \sum_{k=1}^n D_{(n)}^{peak} \quad (4)$$

- PDr_{ratio} refers to the packet drop ratio, which is the percentage calculation between total packet lost, $totalPL$, and total packet successfully transmitted, $totalPT$.

$$PDr_{ratio} = \frac{totalPL}{totalPT} \quad (5)$$

- PDe_{ratio} refers to the packet delivery ratio which is calculated by the difference between the total ratio and PDr_{ratio} .

$$PDe_{ratio} = 1 - PDr_{ratio} \quad (6)$$

- Tp refers to the throughput rate at which the packets are being delivered successfully over a communication bandwidth channel, bw . The success rate denotes the efficiency of the transmission which can be calculated as the split of transmission time and the overall latency including transmission time itself, propagation time or broadcasting time, control time for SDN computing, and processing time at various gateways, which are denoted as $T_{(n)}^t$, $T_{(n)}^{prop}$, $T_{(n)}^{control}$, and $T_{(n)}^{proc}$, respectively. And n refers to the index number of queuing entities in the virtualized network simulation infrastructure.

$$Tp = \frac{\sum_{i=0}^n (T_{(n)}^t) \times bw}{\sum_{i=0}^n (T_{(n)}^t + T_{(n)}^{prop} + T_{(n)}^{control} + T_{(n)}^{proc})} \quad (7)$$

4.3 Results and Discussion

In this sub-section, the results of performance metrics and discussion about the conventional scheme, ERA, and IRA are described. The conventional scheme refers to the traditional network system in which the transmission rate, tx_{rate} , is randomly given. In contrast with ERA and IRA, the resource handling is intelligently adjusted and allocated as the above-mentioned sections. **Table 2** illustrates the numerical comparison between each QoS metric including average delay, average jitter, packet drop ratio, packet delivery ratio, and throughput.

Table 2. The comparison between the QoS metrics of each scheme

	Conventional	ERA	IRA
Average delay (ms)	196.8832	140.2473	118.9146
Average jitter (ms)	0.323	0.3067	0.3067
Packet drop ratio	0.003229 (0.32%)	0.001289 (0.12%)	0.001089 (0.1%)
Packet delivery ratio	0.9968 (99.68%)	0.9987 (99.87%)	0.9989 (99.89%)
Throughput (Mbps)	799.5819	799.597	799.5987

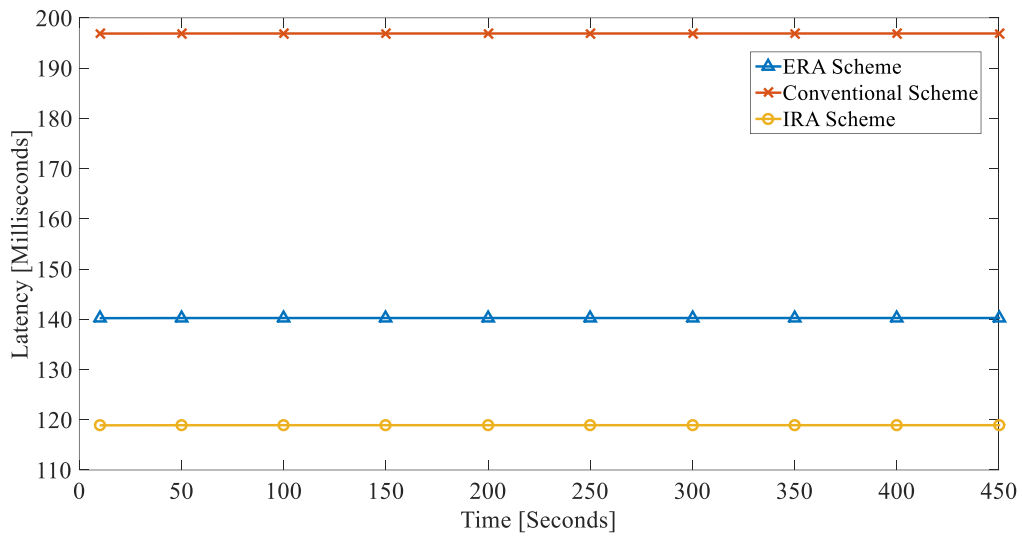


Fig. 5. Comparison of average delay

In terms of the average delay in packet transmission, the conventional scheme reached 196.8832 milliseconds (ms) which was 28.7662% and 39.6014% higher than ERA and IRA, respectively. It was proven that if the network resources were being allocated based on ERA and IRA procedures, the average E2E latency of a packet was reduced by 56.6359 ms and 77.9686 ms, respectively, which can be beneficial to the overall network performances. Fig. 5 presents the graphical line chart of each scheme based on simulation time, 0 to 450 s, and delay gaps in ms. Each scheme had a stable flow from time to time which is convenient to examine. In 5G backhaul networks, the congestion is feasible to occur and leads to numerous packet drops. It is caused by heterogeneous and massive traffics which is concurrently generated by various smart devices, therefore, a single ms of average packet transmission is highly crucial to the overall QoE. IRA achieved the smallest average delay and stable performance in this circumstance.

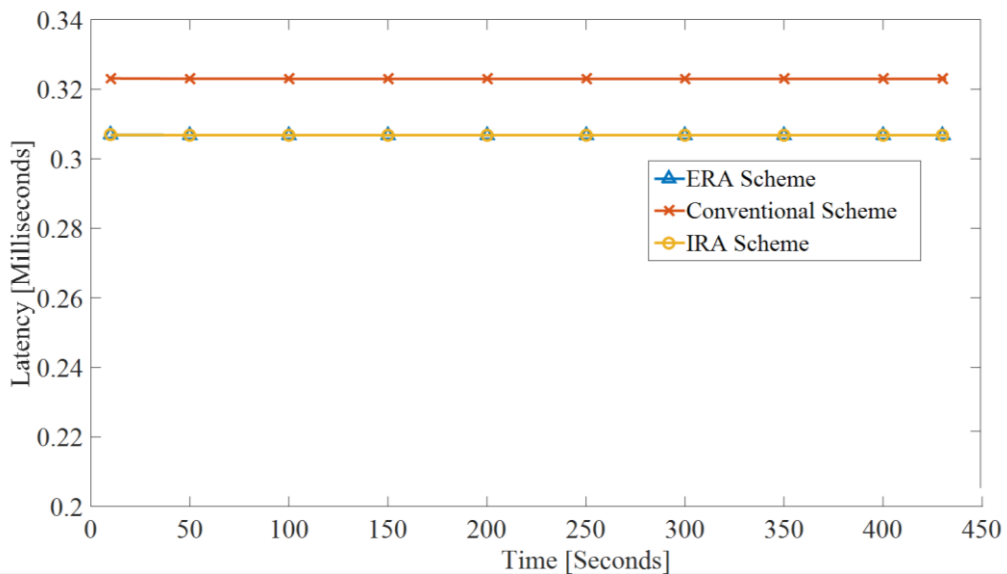


Fig. 6. Comparison of average jitter

The average jitter comparison is illustrated in **Fig. 6**. A variation of each delay that occurs during the transmission is a potential metric to measure the scheme performance. Jitter is possibly caused by signal interference, weak hardware performance, invalid queueing, false configuration, collisions, or network congestion in backhaul networks. In another word, the higher jitter gets, the worse congestion happens. In the future perspective of communication networks, it is crucial to minimize the jitter due to the variety of real-time conversations, video conferences, streaming, or emergency system which the drop action leads to severe and critical loss. In jitter evaluation, ERA and IRA scheme were identical, which caused both lines overlaying with each other from 0 to 450 s simulation time. Comparing with the conventional scheme, ERA and IRA were 5.0464% or 0.0163 ms decreasing in one average traffic which is indeed significant to consider.

The packet drop and packet delivery ratio detailed comparison of complete simulation duration are presented in **Fig. 7** and **Fig. 8**, respectively. The great amount of packet drop counts severely degrades the overall network performance and generates dissatisfaction for the user demands. At the first 50 s of communication, the drop ratio of the conventional scheme reached 1.24% packet loss, which was an extremely high number. And, from 50 s to 450 s, the drop ratio was 0.093%. Otherwise, on average, ERA possibly alleviated 60%, from 0.003229 to 0.001289, of the drop ratio in the conventional approach. Beyond that, the packet drop ratio of IRA averagely decreased by 66% and 15.52% of the conventional scheme and ERA, respectively. Namely, the IRA approach provided the minimal possibility of packet loss which is highly crucial for all communication protocols, particularly, user datagram protocol (UDP). For transmission control protocol (TCP), the great packet drop ratio will lead to throughput reduction and latency increment in terms of packet retransmission. As a consequence, for the packet delivery ratio, the success transmission rate had been illustrated accordingly.

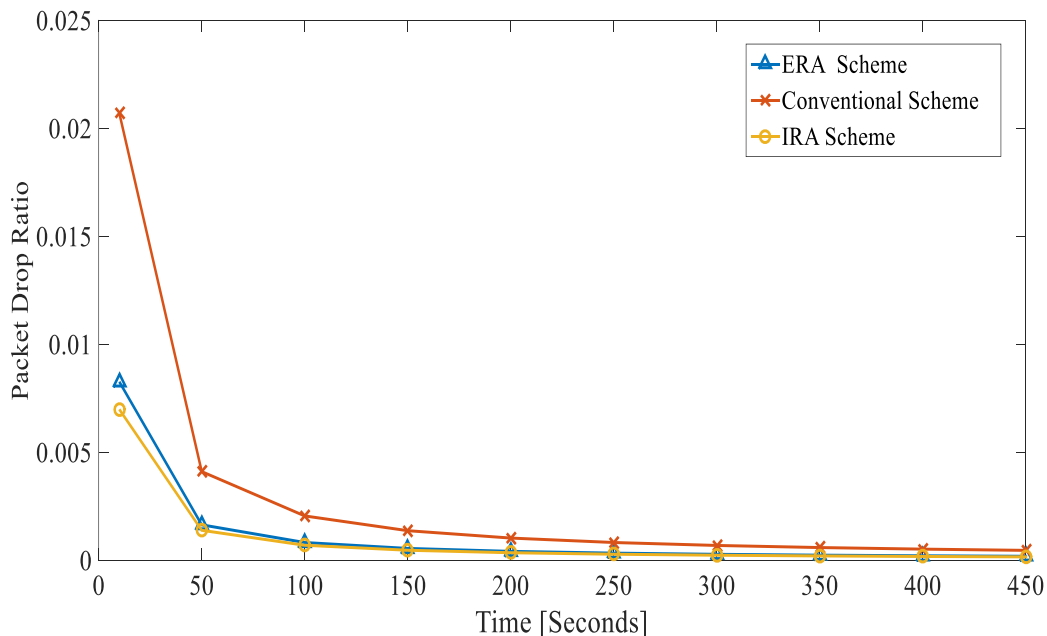


Fig. 7. Comparison of packet drop ratio

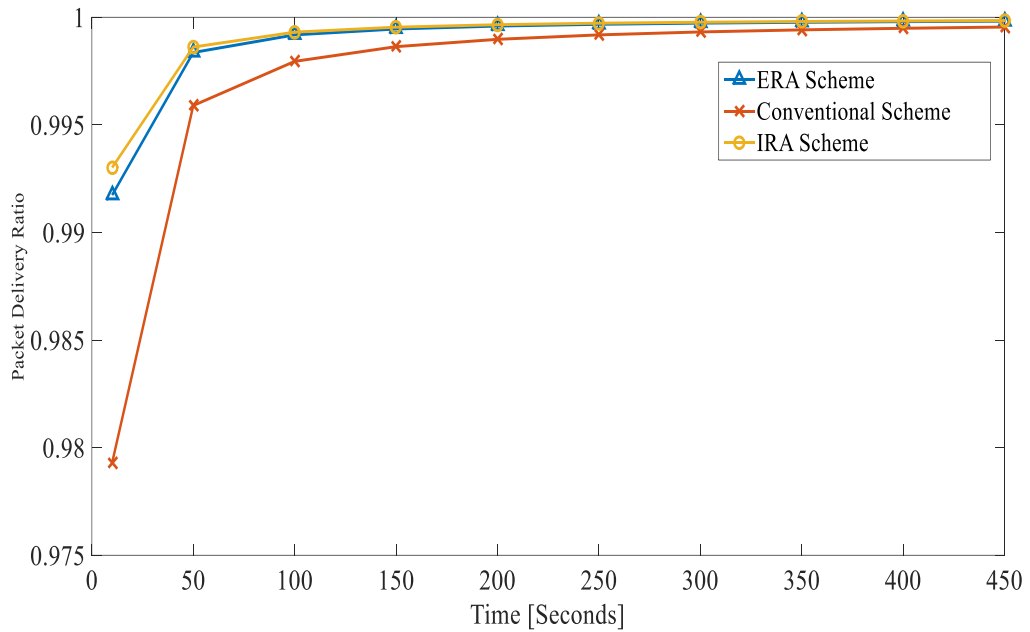


Fig. 8. Comparison of packet delivery ratio

The performance of throughput is presented in **Fig. 9**. The network throughput of the conventional scheme, ERA, and IRA are 799.5819, 799.597, and 799.5987 Megabits per second (Mbps), respectively. In terms of throughput stability and improvement, IRA contributed the utmost efficiency which was followed by ERA and the conventional scheme. With higher network throughput achievement, the communication resource (e.g., link bandwidth) in 5G backhaul networks can be utilized thoroughly.

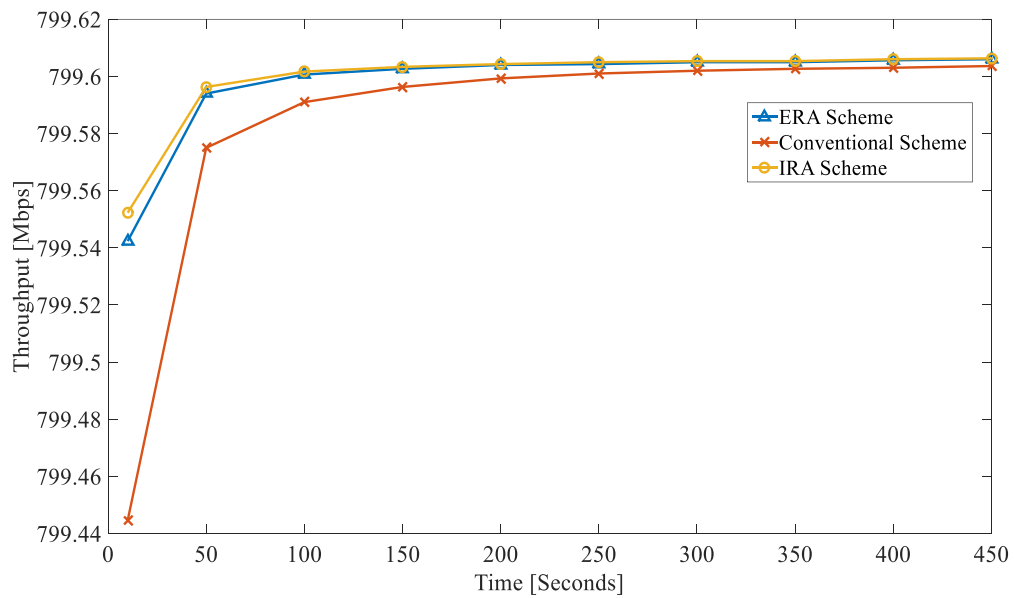


Fig. 9. Comparison of throughput

5. Conclusion

In this article, we proposed a novel intelligent resource handling scheme for allocating the network resources and controlling massive communication traffic in backhaul scenarios. By emerging SDN-oriented architecture, ERA scheme preserved the extant communication and computation resources for primarily handling the real-time UL traffic. Simultaneously, for DL traffic, a machine learning algorithm, SVM, was converged with the proposed method to observe the user request pattern and recommend the off-peak hour interval for non-real-time proactive caching transmission. With the MEC-driven framework, IRA scheme was modified on top of ERA in order to enlarge computing resources for dealing with severe network congestion statuses. To demonstrate the proposed scheme performance, a simulation system was conducted to capture various QoS metrics for comparison purposes. Finally, our finding contributes towards a key enabler approach to proactively cache the content with the highest hit probability, alleviate the high possibility of packet loss in UL transmission, increase communication throughput, and efficiently allocate the resource for mission-critical applications in 5G backhaul networks. This scheme mainly considered comprehensive real-time applications; therefore, the deep packet inspection for resource awareness will conduct in the future study to integrate with the distinct QoS communication requirement.

Acknowledgement

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-0-00403) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation), and this work was supported by the Soonchunhyang University Research Fund.

References

- [1] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A Survey on 5G Usage Scenarios and Traffic Models," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 905-929, 2020. [Article \(CrossRef Link\)](#)
- [2] J. Yao, T. Han, and N. Ansari, "On Mobile Edge Caching," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2525-2553, 2019. [Article \(CrossRef Link\)](#)
- [3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016. [Article \(CrossRef Link\)](#)
- [4] D. Kim and S. Kim, "Gateway Channel Hopping to Improve Transmission Efficiency in Long-range IoT Networks," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 3, pp. 1599-1610, 2019. [Article \(CrossRef Link\)](#)
- [5] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive Transmission Optimization in SDN-Based Industrial Internet of Things With Edge Computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1351-1360, June 2018. [Article \(CrossRef Link\)](#)
- [6] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, "Machine Learning Meets Computation and Communication Control in Evolving Edge and Cloud: Challenges and Future Perspective," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 38-67, 2020. [Article \(CrossRef Link\)](#)
- [7] E. Kim and S. Kim, "An Efficient Software Defined Data Transmission Scheme based on Mobile Edge Computing for the Massive IoT Environment," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 2, pp. 974-987, 2018. [Article \(CrossRef Link\)](#)

- [8] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big Data Analytics, Machine Learning, and Artificial Intelligence in Next-Generation Wireless Networks," *IEEE Access*, vol. 6, pp. 32328-32338, 2018. [Article \(CrossRef Link\)](#)
- [9] S. Math, P. Tam, A. Lee, and S. Kim, "A NB-IoT data transmission scheme based on dynamic resource sharing of MEC for effective convergence computing," *Persnal Ubiquitos Computing*, 2020. [Article \(CrossRef Link\)](#)
- [10] Z. Zaidi, V. Friderikos, Z. Yousaf, 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. [Article \(CrossRef Link\)](#)
- [11] T. Das, V. Sridharan, and M. Gurusamy, "A Survey on Controller Placement in SDN," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 472-503, 2020. [Article \(CrossRef Link\)](#)
- [12] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A Survey of Networking Applications Applying the Software Defined Networking Concept Based on Machine Learning," *IEEE Access*, vol. 7, pp. 95397-95417, 2019. [Article \(CrossRef Link\)](#)
- [13] J. Lei, Y. Wang, and Y. Xia, "SDN-Based Centralized Downlink Scheduling with Multiple APs Cooperation in WLANs," *Wireless Communications and Mobile Computing*, vol. 2019, 2019. [Article \(CrossRef Link\)](#)
- [14] S. K. Singh and A. Jukan, "Machine-learning-based prediction for resource (Re)allocation in optical data center networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D12-D28, Oct. 2018. [Article \(CrossRef Link\)](#)
- [15] C. Zhao, Y. Cai, A. Liu, M. Zhao, and L. Hanzo, "Mobile Edge Computing Meets mmWave Communications: Joint Beamforming and Resource Allocation for System Delay Minimization," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2382-2396, Apr. 2020. [Article \(CrossRef Link\)](#)
- [16] S. Math, L. Zhang, S. Kim, and I. Ryoo, "An Intelligent Real-Time Traffic Control Based on Mobile Edge Computing for Individual Private Environment," *Security and Communication Networks*, vol. 2020, 2020. [Article \(CrossRef Link\)](#)
- [17] H. Alshaer and H. Haas, "Software-Defined Networking-Enabled Heterogeneous Wireless Networks and Applications Convergence," *IEEE Access*, vol. 8, pp. 66672-66692, 2020. [Article \(CrossRef Link\)](#)
- [18] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457-7469, Aug. 2020. [Article \(CrossRef Link\)](#)
- [19] R. K. Das, N. Ahmed, F. H. Pohrmen, A. K. Maji, and G. Saha, "6LE-SDN: An Edge-Based Software-Defined Network for Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7725-7733, Aug. 2020. [Article \(CrossRef Link\)](#)
- [20] S. Bera, S. Misra, and N. Saha, "Traffic-Aware Dynamic Controller Assignment in SDN," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4375-4382, July 2020. [Article \(CrossRef Link\)](#)
- [21] C. Ren, S. Wang, J. Ren, and X. Wang, "Traffic Engineering and Manageability for Multicast Traffic in Hybrid SDN," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 6, pp. 2492-2512, 2018. [Article \(CrossRef Link\)](#)
- [22] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An Efficient Computation Offloading Management Scheme in the Densely Deployed Small Cell Networks With Mobile Edge Computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651-2664, Dec. 2018. [Article \(CrossRef Link\)](#)
- [23] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Efficient NFV-Enabled Multicasting in SDNs," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2052-2070, Mar. 2019. [Article \(CrossRef Link\)](#)
- [24] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," *IEEE Network*, vol. 33, no. 5, pp. 156-165, 2019. [Article \(CrossRef Link\)](#)

- [25] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine Learning for Networking: Workflow, Advances and Opportunities," *IEEE Network*, vol. 32, no. 2, pp. 92-99, Apr. 2018. [Article \(CrossRef Link\)](#)
- [26] Z. Shah, "Mitigating TCP Incast Issue in Cloud Data Centres using Software-Defined Networking (SDN): A Survey," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 11, pp. 5179-5202, 2018. [Article \(CrossRef Link\)](#)
- [27] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li, and J. Rao, "Dynamic Flow Migration for Embedded Services in SDN/NFV-Enabled 5G Core Networks," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2394-2408, Apr. 2020. [Article \(CrossRef Link\)](#)



Prohim Tam received the B.S. degree in the Department of Management of Information Systems from Paragon International University, Cambodia, in 2019. Currently, he is a master's candidate in the Department of Software Convergence, Soonchunhyang University, Republic of Korea. His research interests include future access networks, software-defined networking, network function virtualization, artificial intelligence in bigdata transmission, mobile edge computing, and the Internet of Things.



Sa Math received the B.E. in Department of Telecommunication and Electronic Engineering, from Royal University of Phnom Penh, Phnom Penh, Cambodia, in 2018. Currently, he is doing a Ph.D. in Department of Software Convergence Soonchunhyang University, Asan, Republic of Korea. His research interests on 5G Distributed and Core networking, Internet of Things, Quality of Service, Software Defined Mobile edge computing, and Machine Learning.



Seokhoon Kim received the B.E. and Ph.D. in computer engineering from Kyunghee University, Korea, in 2000 and 2004, respectively. From 2004 to 2006, he was with IPOne, Inc., Seoul, Korea, where he led various projects as a research engineer. From 2006 to 2009, he was a research engineer at Neowave, Inc., Anyang, Korea, where he developed Mobile WiMAX (IEEE 802.16) devices. He was an Assistant Professor in Department of Mobile Communications Engineering at Changshin University, Changwon, Korea. Since March 2016, he has been with the Department of Computer Software Engineering, Soonchunhyang University, Asan, Korea, where he is currently an Assistant Professor. His research interests comprise Cloud and Mobile Edge Computing, Internet of Things, Heterogenous Internet of Things, Software Defined Networking and Network Function Virtualization, Mobile System and Communications, and Machine Learning based on Bigdata.