J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr


# A Lightweight Software-Defined Routing Scheme for 5G URLLC in Bottleneck Networks☆

맛 사[1] 담 프 로 힘[1] 김 석 훈[1,2*]
Sa Math Prohim Tam Seokhoon Kim

ABSTRACT

Machine learning (ML) algorithms have been intended to seamlessly collaborate for enabling intelligent networking in terms of massive service differentiation, prediction, and provides high-accuracy recommendation systems. Mobile edge computing (MEC) servers are located close to the edge networks to overcome the responsibility for massive requests from user devices and perform local service offloading. Moreover, there are required lightweight methods for handling real-time Internet of Things (IoT) communication perspectives, especially for ultra-reliable low-latency communication (URLLC) and optimal resource utilization. To overcome the abovementioned issues, this paper proposed an intelligent scheme for traffic steering based on the integration of MEC and lightweight ML, namely support vector machine (SVM) for effectively routing for lightweight and resource constraint networks. The scheme provides dynamic resource handling for the real-time IoT user systems based on the awareness of obvious network statues. The system evaluations were conducted by utillizing computer software simulations, and the proposed approach is remarkably outperformed the conventional schemes in terms of significant QoS metrics, including communication latency, reliability, and communication throughput.

☞ keyword : Internet of Things, Quality of Service, Machine Learning, Mobile Edge Computing, Software-Defined Networking

## 1. Introduction

The 5th generation (5G) and 6th generation communication technologies are intended to perform agile response services for user devices with high capability to provide new radio (NR) services for massive user terminals. To respond to the enlargement of edge networks, millimeter-wave (mmWave) enables 5G communication with extensive benefits in terms of ultra-high mobility management (UHMM), ultra-high communication reliability (UHCR), ultra-high communication bandwidth (UHCBW), ultra-low latency (ULL), and massive user services and applications [1-3]. The rapid evolvement of massive heterogeneous Internet of Things (HetIoT) devices in communications generate immense traffic over networks. Bigdata transmission will be suffered at the bottleneck fronthaul and backhaul networks which are installed by optical environments, and insufficient scalability [4,5]. Future mobile services are based on cellular systems which a convergence of heterogenous radio gateway devices. New Radio technology enables massive of Remote Radio Head (RRH) in mesh communications. Ultra-high mobility is obligate for high-speed transportation application (Internet of Vehicles) and some of Autonomous Internet of Things (AIoT). These applications become the over the top (OTT) for current applicant and future mobile services research trends. The key quality of service (QoS) obligations are Ultra-high mobility, Ultra-high Reliable and Ultra-low Latency are under the research trends. In the cellular networks, the failure access and latency in the radio interface degrade the QoS of novelty mobile services and the huge challenging issues. Due to the 5G/6G radio access network (RAN) environments are integrated of high computing power

1 Department of Software Convergence, Soonchunhyang University, Asan, 31538, Korea
2 Department of Computer Software Engineering, Soonchunhyang University, Asan, 31538, Korea
* Corresponding author (seokhoon@sch.ac.kr)
[Received 8 October 2021, Reviewed 14 October 2021(R2 29 November 2021), Accepted 7 December 2021]
☆ This work was funded by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2020R1I1A3066543). In addition, this work was supported by the Soonchunhyang University Research Fund.
☆ A preliminary version of this paper was presented at KSII/APIC-IST 2021.


http://dx.doi.org/10.7472/jksii.2022.23.2.1

inherited from mobile cloud computing which located in remote area, a comprehensive opportunities and novelty technologies have been introduced. Moreover, the Intelligent RAN and Edge cloud bring the intelligent RAN resource management and orchestration and also intelligent services for mobile users. Self-Organizing Network (SON) takes advantages from Machine Learning for autonomous resource utilization, security, and increase speed of processing periods. The rest of the paper is organized as the follows. The section 2 presents the related work. Section 3 presents the proposed scheme. Furthermore, section 4 presents the experiment and result discussions. And, finally the paper conclusion is represented in the section 5.

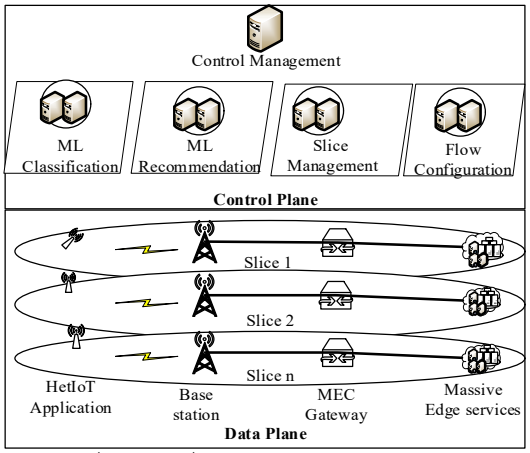
## 2. Related Work

The intelligent edge networks (IEN) aim to deliver intelligent services as same as the mobile cloud computing (MCC) environment while edge cloud will handle equivalent works as the MCC. It is required to consider the common key features, including E2E ULL, UHCR, data integrity, bigdata analytic, agile service response, and secured privacy [5-7]. ML and deep learning (DL) are under the family of artificial intelligence (AI) algorithms and anticipate handling the abovementioned challenges and edge network issues [7]. ML generally comprises of three types, such as supervised learning, unsupervised learning, and reinforcement learning (RL) [8-10]. These well-known ML are the components of mobile edge computing (MEC) based network to enable the intelligent edge computing paradigm. However, the training and inference of ML/DL in edge cloud requires numerous input dataset from devices, IoT services, resources, traffic, application, etc. IEN intends to offer agile resource offloading, UHMM, UHCR, edge fault-tolerance, load-balancing, and privacy.

Additionally, the appearance of classification approaches which are offered by lightweight ML algorithms meet the resource constraint application, such as real-time IoT which obligates the ultra-reliability low latency communications (URRLC) perspectives [10-12]. Support vector machine (SVM) approach works well with the less complexity network dataset and provides the robustness model for reliable classification [12]. The adoption of classification approaches in software-defined routing (SDR) for joined traffic aware routing will improve the routing experience and enhance QoS for time-critical applications. Moreover, the lightweight classification processes will be conducted in real-time and able to run over the low capacity computation unit and less computation overhead for both time and physical resources. Due to the control plane of software-defined networks (SDN) will be suffered from the physical resource limitation, the lightweight approach with less computation overhead will be applicable [13-15].

## 3. The Proposed Scheme

The proposed traffic steering is based on an amalgamation of classifications and recommendations by lightweight ML and MEC implementation. Figure 1 demonstrates the proposed network architecture which is composed of control plane (CP) and data plane (DP). The DP of the edge network consists of MEC gateways and massive edge servers which are jointly integrated. The MEC server takes the main advantage of caching the incoming traffic for computation of ML purposes. ML algorithms locate in the CP area to compute the cached IoT traffic in the MEC servers which are integrated with MEC gateways. The MEC servers at the CP contain the entire information of the incoming traffic from the MEC gateways and the statuses of the MEC servers from massive edge services.




(Figure 1) Proposed architecture

The proposed scheme contains five processes as follows:

Firstly, the caching of incoming traffic and server status: the SDN controller has to monitor and update the number of incoming traffic at the MEC server gateway and the information of the MEC server statuses based on the obvious conditions. Futhermore, the server capacity is determined by the serving capacity parameters in terms of serving rate, bandwidth, and computing delay of the server.

**Algorithm 1. Network statuses classifications**

**Input:** $x' = \{x_1, x_2, x_3, \ldots, x_n\}$ denotes number of $n$ features of training data, and $y' = \{y_1, y_2, y_3, \ldots, y_m\}$ denotes target features with $m$ different values.

**Output:** Optimal learning model of the data to be classified, $X$.

1. Initialize model classifier $f_\theta^{(m)}(X)$;
2. For each input data:
3. SVM → Apply SVC constructor with RFB Kernel:
$$K(x', y') = e^{\left(-\frac{\| x^2 - y^2 \|}{2\sigma^2}\right)}$$
4. Fit each model with the data in parallel process: $f_\theta^{(m)}(x_n) = P(y_m = m | x_n; \theta)$
5. Output model selection decision with incoming data: $\max_m f_\theta^{(m)}(X)$;
6. Execute the performance and score metrics, *accuracy;*

Secondly, the traffic classification and server recommendation are obligated for heterogeneous MEC servers which have heterogeneous resource parameters, and there are different limitations services to the users.

Thirdly, the scheme adjusts the classified incoming traffic to meet the serving resources. So, different MEC capacities will receive a different number of incoming traffic. The recommendation processes in this paper are referred to the mapping between user traffic and server computation resources. SVM classifies the IoT traffic based on the capacity of the servers, while the number of IoT traffic classes are the same as the number of server classes. Each IoT traffic class will recommend to the server based on the corresponding server capacity. Furthermore, the server with high capacity will receive the traffic class with huge numbers, and the lowest-capacity server will receive the traffic class with the smallest number.

Fourthly, besides the learning of ML for the traffic and server-side, the flow configuration will be maintained and countered by the SDN controller. So, the forwarding flows configuration will be made based on the ML outcomes. Additionally, each IoT forwarding flow has to be updated whenever the destination server statuses have changed. The unbound works of the CP and DP will be benefited from the CP resource offloading times, learning, and the forwarding flows configuration. The updated periods can be migrated from the real-time DP communication.

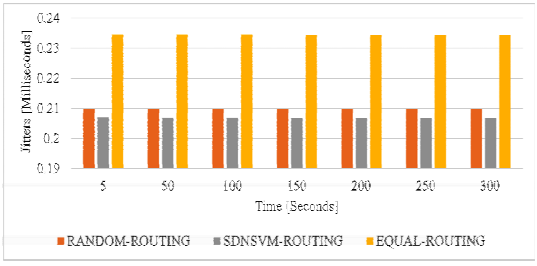
Finally, the MEC servers are proposed to integrate with the MEC gateway for buffered purposes. The forwarding flows are located in the MEC gateway which is configured by the SDN controller. The flow updates are based on the various incoming traffic and server capacity. If the incoming traffic is stable, the forwarding rules will also keep the stability from the update.

Algorithm 1 presents the flow procedure of multiple model calculations for fitting the data optimally. The concept of model and data parallelism allows the approach to ensure adequate computation and communication resource efficiency. The input requirement contains the well-processed dataset captured from the packet flows. The number of training data with various features is denoted as $x'$ to supervise the model. The target features are separated with different class values within $y'$. Towards the optimal learning model selection, the model classifier of the incoming packet, $f_\theta^{(m)}(X)$, is assigned and each algorithm processes simultaneously to reach the satisfying performance metrics. SVM algorithm initiates support vector classifier (SVC) from the support of radial basis function kernel, $K(x', y')$, with the free parameter$\sigma$ to deal with multidimensional non-linear circumstances. After each algorithm outcome is appended, the comparison with model classifier is fitted in parallel mechanism to determine the maximum performance, $\max_m f_\theta^{(m)}(X)$. After the optimal model is chosen, the score metrics will be evaluated spontaneously to prove the satisfactory level with Quality of Experience (QoE) expectation.

## 4. Experimentation and Results

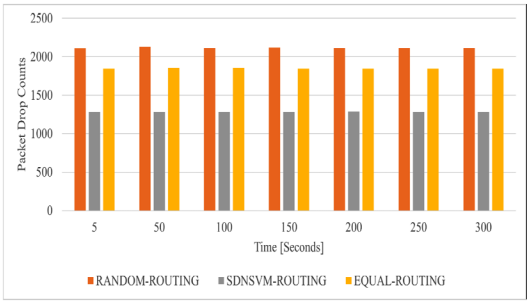
The system was simulated by comprising into two different scenarios, including the conventional schemes, including random-based handling (RANDOM-ROUTING) and equal-based routing (EQUAL-ROUTING) methods and proposed SVM-based software-defined routing (SDNSVM -ROUTING). There are 2,500 packets of the incoming traffic at the same time, and 5 MEC servers are used to experiment. 2,500 datasets were generated based on the assumed 5 different MEC server behaviors. The random and equal-cost based methods were simulated by selecting 2,500 packets randomly and configured to forward 500 packets equally for each serving MEC server, respectively. Due to the proposed scheme, the forwarding flow configuration was based on the output ML algorithm.

The QoS evaluations for massive real-time IoT communications, including the proposed and conventional schemes, in terms of E2E communications jitters, packet drop counts, E2E packet drop ratio, E2E communication reliability, and average communication throughput, are expressed in Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6, respectively. In future communication systems, radio networks will generate bigdata of HetIoT traffic that causes the network congestion in the radio interface between IoT devices and RRH. The real-time IoT user will suffer from the loading delay that occurs during MAC scheduling. Consequently, to guarantee E2E QoS, a real-time IoT network is required to handle both congestions at radio gateways and MEC gateways. The proposed scheme provides the systematic handling of IoT traffic resources in both radio and MEC gateways.




(Figure 2) The comparison of E2E communications jitters between proposed and conventional schemes.

Figure 2 depicts the comparisons of communciation jitters in the network system between the RANDOM-ROUTING, SDNSVM-ROUTING, and EQUAL-ROUTING at the metrics of 0.209979857 milliseconds, 0.2068964 milliseconds (Proposed approach), and 0.234466657 milliseconds, respectively. As predicted in the given graphs, the proposed SDNSVM-ROUTING contains lowest jitters metrics in each communication time. The lowest communication jitter shows the optimal network stability. Moreover, the E2E communication latency can be lessened when the poor conditions at bottleneck areas can be handled. In real-world HetIoT systems, massive requests will be required to wait for available resources. Due to the limitation of radio resources and edge gateways, the stability of the RAN environments will be insufficient for massive IoT traffic. These significant issues were solved by the proposed scheme which relied on ML algorithms for classifying DP resources and recommending the incoming traffic according to obvious serving entity statuses. So, the systematic handling for both radio and edge gateways are accomplished. Consequently, the E2E optimal resource utilization overcomes the ULL latency for both communication latency.

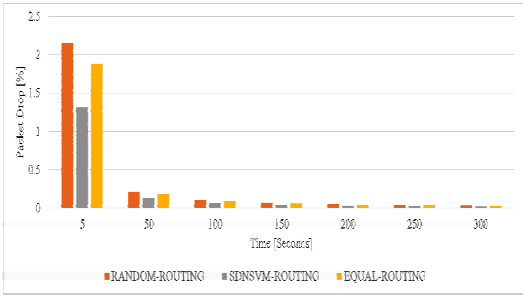



(Figure 3) The comparisons of Proposed and Conventional Schemes in terms of packet drop counts.

Naturally, the packet drop will be lessened whenever the congestions can be reduced and the number of packet drop will be arising rely on the communication conditions, as shown in Figure 3. The proposed scheme outperformed the

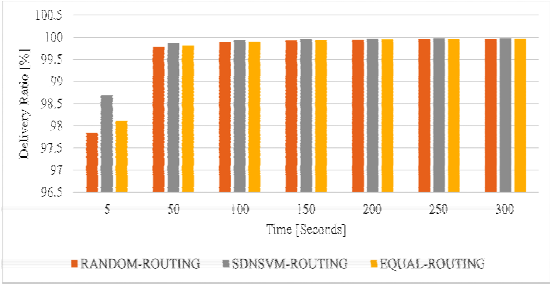
conventional approaches in any conditions. Each of the following scheme shares different metrics RANDOM-ROUTING, SDNSVM-ROUTING, and EQUAL-ROUTING at 2115.571429, 1285.714286 (proposed scheme), and 1848.857143, respectively. The graphs show that the proposed scheme remarkably outperforms the conventional approaches, thus, the proposed approach is suitable for URLLC applications.

Figure 4 illustrates the comparison metrics of RANDOM-ROUTING, SDNSVM-ROUTING, and EQUAL-ROUTING approaches in terms of E2E packet drop ratio in series of 0.381792479%, 0.232464271%, and 0.333917716%, respectively. Based on the given graphs the proposed scheme consists lowest E2E communication drop ratio in any network situation. In real-word, IoT communicates over 5G communications, therefore, the mission-critical IoT applications will suffer from highly vulnerable packet drop ratios. Especially, lightweight WSN devices that have limited resources and communication over unreliable transmission protocols. The proposed scheme provides the ideal traffic steering for E2E networks and will be significant for future HetIoT applications.

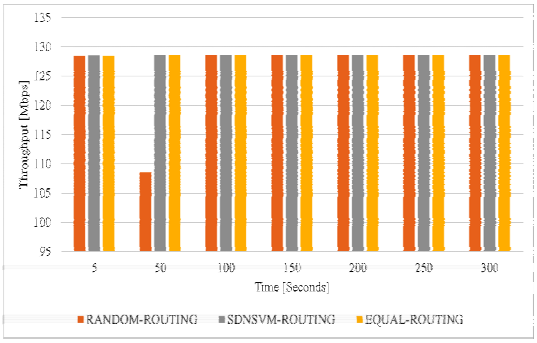



(Figure 4) The Packet drop ratio in percents comparisons between proposed and conventional schemes

Due to the proposed scheme reduces E2E packet drop ratio, Figure 5 illustrates the comparisons of communication reliability. The average E2E communication reliability of RANDOM-ROUTING, SDNSVM-ROUTING, and EQUAL-ROUTING with the metric of 99.61820752%,99.76753573%, and 99.66608228%, respectively. And, Figure 6 illustrates the communication throughput between the proposed and conventional schemes. The proposed scheme outperformed the throughput metric over the conventional scheme. While Each of the following scheme shares different metrics RANDOM-ROUTING, SDNSVM-ROUTING, and EQUAL-ROUTING at 125.6839429 Mbps, 128.5457714 Mbps, and 128.5426571 Mbps, respectively.




(Figure 5) The E2E communication reliability (Delivery ratio) in percents comparison between proposed and conventional schemes




(Figure 6) Average throughput comparison between proposed and conventional schemes

## 5. Conclusion

This paper presented a lightweight traffic steering scheme for massive real-time IoT traffic with multiple edge servers. The proposed schemes deliver systematic resource handling

based on the recommendation of the ML algorithm. The IoT traffic will be balanced for each of the edge gateways relied on its serving capacity. Moreover, the scheme improves the main key factors of user QoS in terms of reducing E2E communication delays and jitters, increasing communication reliability, and improving communication throughput, therefore, the stability of the communication systems will be significantly improved. Moreover, the paper provided the lightweight methods which meet the perspective of fronthaul MEC network infrastructure and real-time IoT applications. Since the MEC network environments suffer from resource limitation, the real-time IoT applications required a lightweight handling method to perform ULL resource offloading and computing.

## 참고문헌(Reference)


[1] P. Popovski et al., "Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC)," IEEE Transactions on Communications, vol. 67, no. 8, pp. 5783-5801, Aug. 2019. http://dx.doi.org/10.1109/TCOMM.2019.2914652

[2] M. Aazam, K. A. Harras, and S. Zeadally, "Fog Computing for 5G Tactile Industrial Internet of Things: QoE-Aware Resource Allocation Model," IEEE Transactions on Industrial Informatics, vol. 15, no. 5, pp. 3085-3092, May 2019. http://dx.doi.org/10.1109/TII.2019.2902574

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4724-4734, Nov. 2018. http://dx.doi.org/10.1109/TII.2018.2852491

[4] M. Agiwal, A. Roy, and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 1617-1655, 2016. http://dx.doi.org/10.1109/COMST.2016.2532458

[5] P. Tam, S. Math, A. Lee, and S. Kim, "Multi-Agent Deep Q-Networks for Efficient Edge Federated Learning Communications in Software-Defined IoT" CMC-Computers, Materials & Continua, vol. 71, no. 2, pp. 3319–3335, 2022. http://dx.doi.org/10.32604/cmc.2022.023215

[6] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450-465, Feb. 2018. pp. 2462-2488, 2020. http://dx.doi.org/10.1109/JIOT.2017.2750180

[7] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge Computing in Industrial Internet of Things: Architecture, Advances and Challenges," IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2462-2488, 2020. http://dx.doi.org/10.1109/COMST.2020.3009103

[8] 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. http://dx.doi.org/10.3837/tiis.2018.02.027

[9] P. Tam, S. Math, and S. Kim, "Efficient Resource Slicing Scheme for Optimizing Federated Learning Communications in Software-Defined IoT Networks," Journal of Internet Computing and Services, vol. 22, no. 5, pp. 27–33, Oct. 2021. http://dx.doi.org/10.7472/JKSII.2021.22.5.27

[10] P. Tam, S. Math, C. Nam and S. Kim, "Adaptive Resource Optimized Edge Federated Learning in Real-Time Image Sensing Classifications," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 10929-10940, 2021. http://dx.doi.org/10.1109/JSTARS.2021.3120724

[11] J. Xie 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. http://dx.doi.org/10.1109/COMST.2018.2866942

[12] H. Hantouti, N. Benamar, T. Taleb, and A. Laghrissi, "Traffic Steering for Service Function Chaining," IEEE Communications Surveys & Tutorials, Vol. 21, No. 1, pp. 487-507, 2019. http://dx.doi.org/10.1109/COMST.2018.2862404

[13] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, “Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions,” IEEE Communications Surveys & Tutorials, Vol. 20, No. 3, pp. 2429-2453, 2018.
http://dx.doi.org/10.1109/COMST.2018.2815638

[14] J. Liu, H. Huang, Y. Zhong, J. He, T. Huang, Q. Xiao, and W. Jiang, “Joint wireless and computational resource allocation for ultra-dense mobile-edge computing networks,” KSII Transactions on Internet and Information Systems, Vol. 14, No. 7, pp. 3134-3155, 2020.
http://dx.doi.org/10.3837/tiis.2020.07.022

[15] H. Zhang, H. Zhang, K. Long, and G. K. Karagiannidis, “Deep Learning Based Radio Resource Management in NOMA Networks: User Association, Subchannel and Power Allocation,” IEEE Transactions on Network Science and Engineering, Vol. 7, No. 4, pp. 2406-2415, 2020.
http://dx.doi.org/10.1109/TNSE.2020.3004333

[16] D. Kim, J. Park, Y. Lee, and S. Kim, “Network virtualization for real-time processing of object detection using deep learning,” Multimedia Tools and Applications, 2020.
http://dx.doi.org/10.1007/s11042-020-09603-0

## ◐ Authors ◑

**맛 사(Sa Math)**
2014: Royal University of Phnom Penh (Cambodia), Telecommunication and Electronic Engineering (Bachelor Degree)
2014 ~ Present: Soonchunhyang University (Korea), Software Convergence (Combined Master and Ph.D.)
Research Interests: 5G, IoT, QoS, SDN, MEC, and Machine Learning.
E-mail : mathsa98@gmail.com

**담 프 로 힘(Prohim Tam)**
2015: Paragon International University (Cambodia), Management of Information System (Bachelor Degree)
2015 ~ Present: Soonchunhyang University (Korea), Software Convergence (Combined Master and Ph.D.)
Research Interests: 5G Service-Based Orchestration, QoS, SDN/NFV, Edge Intelligence, and IoT.
E-mail : prohimtam@gmail.com

**김 석 훈(Seokhoon Kim)**
2000: Kyunghee University (Korea), Computer Engineering (Bachelor Degree)
2004: Kyunghee University (Korea), Computer Engineering (Master and Ph.D. Degree)
2016 ~ Present: Professor at Department of Computer Software Engineering, Soonchunhyang University, Korea
Research Interests: 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.
E-mail : seokhoon@sch.ac.kr