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

Deep Neural Network-Based Critical Packet Inspection for Improving Traffic Steering in Software-Defined IoT☆

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

ABSTRACT

With the rapid growth of intelligent devices and communication technologies, 5G network environment has become more heterogeneous and complex in terms of service management and orchestration. 5G architecture requires supportive technologies to handle the existing challenges for improving the Quality of Service (QoS) and the Quality of Experience (QoE) performances. Among many challenges, traffic steering is one of the key elements which requires critically developing an optimal solution for smart guidance, control, and reliable system. Mobile edge computing (MEC), software-defined networking (SDN), network functions virtualization (NFV), and deep learning (DL) play essential roles to complementary develop a flexible computation and extensible flow rules management in this potential aspect. In this proposed system, an accurate flow recommendation, a centralized control, and a reliable distributed connectivity based on the inspection of packet condition are provided. With the system deployment, the packet is classified separately and recommended to request from the optimal destination with matched preferences and conditions. To evaluate the proposed scheme outperformance, a network simulator software was used to conduct and capture the end-to-end QoS performance metrics. SDN flow rules installation was experimented to illustrate the post control function corresponding to DL-based output. The intelligent steering for network communication traffic is cooperatively configured in SDN controller and NFV-orchestrator to lead a variety of beneficial factors for improving massive real-time Internet of Things (IoT) performance.

☞ keyword : Deep Learning, Deep Packet Inspection, Network Functions Virtualization, Software-Defined Networking, Traffic Steering

1. Introduction

With the current development and implementation of newly modernized communication technologies and massive Internet of Things (IoT) devices, network transmission traffic in 5G network is increasing notably. The growth provides many challenges in terms of management, reliability, complexity, optimization, and security in the heterogeneous network environment which associates with various types of devices, protocols, data centers destination, application (e.g. conversation, gaming, streaming, interactive, background), and processing mechanisms [1,2]. By this diversification, the network architecture has to be flexible and extensible for handling the potential manifold issues. In these recent years, advanced enabler technologies have been deployed to tackle the existing challenges in both next-generation radio access network (NG-RAN) and 5G core (5GC), including mobile edge computing (MEC), software-defined networking (SDN), network functions virtualization (NFV), massive multiple-input and multiple-output (MIMO), millimeter wave (mmWave), device-to-device (D2D) communications, radio resource management (RRM) function, etc. [3,4].

European Telecommunications Standards Institute (ETSI) introduced the aspects of MEC integration and user plane/data plane split communication control as user plane function (UPF) in 5GC to be the centralized inspection for guiding the packet transmission traffic with adequate network functions (NFs) [5,6]. Moreover, the procedures of UPF are capable of flexibly handling the traffic with protocol data unit (PDU) session slices and Quality of Service (QoS) profiles from NG-RAN through NG-U/N3

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 25 June 2021, Reviewed 10 July 2021(R2 23 September 2021, R3 15 October 2021), Accepted 12 November 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 ICONI 2020.

http://dx.doi.org/10.7472/jksii.2021.22.6.1

interface in high complexity environment with well-defined functional parameters in the control entity [7]. To set up and administer the flow rules of traffic steering, the convergence of deep learning (DL), MEC, and SDN/NFV are significant to contribute the intelligent edge concept for orchestrating, monitoring, and managing the mobility pattern towards 5G service-based requirement in distributed networks.

With multiple incoming packets which have different conditions at the network devices, complexity and insufficient steering happen. The packet classification techniques are commonly used by port-based approach, deep packet inspection (DPI), or machine learning-based. For port-based technique, the uses of UDP and TCP port numbers determined the main application of the traffic, but presently, the dynamic ports are used by many other applications or platforms which is the main cause of ineffective port-based capability. Therefore, DPI is a better and more reliable method to classify and detect the traffic types. Nevertheless, there are remaining challenges such as long time-consuming processes, pattern-constrained, and high energy-consuming requirements. Thus, deep neural network (DNN) and DPI convergence becomes a sufficient approach.

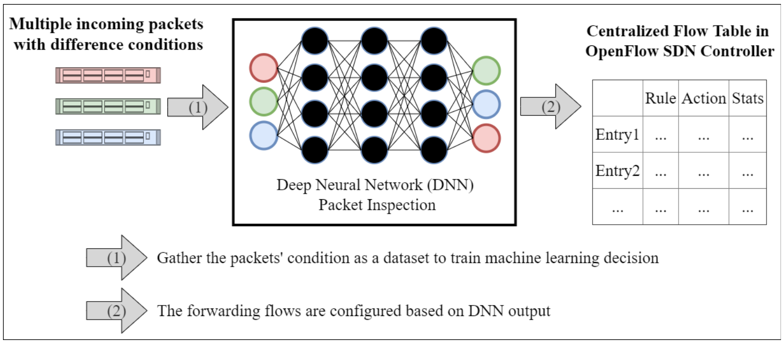
This article presents the system architecture of dynamic framework and programmable network automation with SDN/NFV interactions. SDN controller is used to interact with DNN-based mechanism in virtualized infrastructure manager (VIM) entity for reliable training and testing the model using the incoming OFPT_PACKET_IN messages. The packet features are gathered by the DPI paradigm. DNN-based packet classifications in terms of four target service-based criticalities are performed. Centralized SDN controller targets the outputs for priority determination.

2. Related Works

The evolution of SDN/NFV-enabled architecture has examined and developed advanced network architectures for improving the controller programmability, lessening the complexity of heterogeneity, and scaling the flexible architecture to reduce both capital and operating expense [8]. Furthermore, MEC resource pools for DL processing has further converged and extended to provide sufficient multi-round communications and computation resources, tailoring edge framework, and accurate evaluation on edge intelligence performance [9].

However, to obtain the global view of network statuses and controllability, SDN has drawn a multitude of concentrations to implement the decoupling of control plane and user plane in the network system for logical flow control and well-organized architecture [10]. The cluster nodes and heads in user plane consist of both hardware-based switches and virtual switches which are accountable for proactive flow configuration, revising, and routing based on centralized controllers instruction in the control plane dynamically. The user plane and control plane are communicated through southbound interface, which mostly used the OpenFlow protocol. To communicate between control plane and application plane, SDN paradigm introduces northbound interfaces as network service abstraction layer (NSAL) for service-based requirement interactivity by utilizing representational state transfer (RESTful) application programming interface (API). Moreover, SDN/NFV-based solution feasibly implements network service header protocol in control entities for service function chaining configuration and traffic steering enhancement [11]. To further apply network slicing, virtual network functions (VNFs) are enabled to create isolated, independent, and logical flow connectivity for end-to-end network [12]. Moreover, network slicing participates in supporting the high-mobility user in ultra-dense IoT networks, separating the resource allocation for traffic slicing with specified VNFs forwarding graph, and strengthening the end-to-end diversified transmission/received services to enhance traffic steering configuration.

With MEC supports, edge network gains faster speed, available storage, greater computing power capability, and higher possibility of offloading massive IoT tasks. Edge intelligence contributes minimal latency in computation offloading decision, allocating the resource, controlling overhead management, and handling corporate-server communication services [13]. However, MEC traffic steering also requires optimal DL algorithm design to detect and classify diversified traffic packet statuses for an accurate recommendation and further navigation. DNN, a DL algorithm, has overcome several challenges and provided



(Figure 1) The flow of DNN-based mechanism for critical packet inspection

sufficient services such as traffic classification, prediction, flow detection, clustering, pattern analysis, and continuous state improvement in edge networking applications [14,15]. By executing DNN to classify/detect the different types of critical packet conditions, the traffic steering is feasibly pre-configured in the SDN controller to identify routing optimization rules, develop more reliable packet transmission, and provide flexible management. Therefore, 5G traffic steering is considered to be advanced by DPI and DNN-based algorithm for SDN forwarding flow controller and virtualized network slicing with sufficient priorities.

3. Deep Neural Network-Based Critical Packet Inspection

3.1 System Architecture

In 5G communication network environment, various packets in terms of destination fields, communication protocols, packet types, different QoS parameters, and other packet characteristics are received in a OFPT_PACKET_IN messages to central controller when the local cluster heads mismatch the flow entry matching. The DPI procedures are executed to extract the primary features for the training features, denoted as $\{x_1, x_2 \cdots, x_n\}$, in coarse-grained perspective. In this proposed scheme, n features are used within service-based QoS parameter requirement to identify the criticality levels. The experience replay of QoS performances, such as pattern analysis, packet delay, reliability, and packet loss, is appended for online networks to feed in input layers for enhancing the weight and biases. With sufficient algorithms, the proposed approach manages to gather big data of packet conditions and flow pattern as training dataset to develop the model. DPI requires the understanding of packet details with pre-determined pattern analysis of input user plane information from data service abstraction layer (DSAL) for identifying the destination that the flow rules forward to. Because of the rapid growth of application services, the pattern of packet becomes more challenging and problematic for inspection purposes. Big collection of packet traffic flows is required to complete for processing an accurate and reliable DNN-based mechanism. (Figure 1) illustrates the processing flow of selecting DNN algorithm to apply for packet detection and outputting a reliable clustering result for intelligent traffic steering configuration in OpenFlow SDN controller. DNN uses multiple neural networks with multiple hidden layers between input and output layers. By using multi-layer perceptron (MLP) algorithm, each neural part of hidden layers plays an important role to contain the value and analyze the data of the previous layer neural conditions for giving weighted metric summary reports. MLP allows the whole algorithm to enable non-linear or unlabeled packet flow patterns in real-time communications. After the model training stage, DNN is capable of generating an accurate and reliable result of packet classification, server recommendation, and application detection purposes. Thus, DNN results in terms of criticality levels, denoted as $Y = \{y_1, y_2 \cdots, y_m\}$, is crucial and cooperative for the next stage of SDN controller configuration. The flow entry installations are set based on DNN-based recommendation priority classes to create a reliable connectivity in distributed cluster head forwarding.

3.2 Algorithm Design

As discussion above, SDN control plane consists of centralized programmable controller (e.g., POX, NOX, RYU, OpenDayLight) which allows the proposed system to improve edge traffic steering by configuring and setting the flow table critically using OpenFlow protocol to interact with the data plane. By detecting the rule status, NFs virtually support for matching up the connectivity which has similar preference based on the historical dataset. At the final stage, edge traffic steering keeps synchronizing the forwarding rules with SDN controller continuously and updating the fault-tolerance or high availability discovery. Concurrently, DNN stores the historical flows and statuses for a better understanding of the surrounding network environment and advances the next action of the next state for better calculation and reliability. Algorithm 1 expresses the cooperation of DNN-based output with SDN post function with DPI in the proposed system.

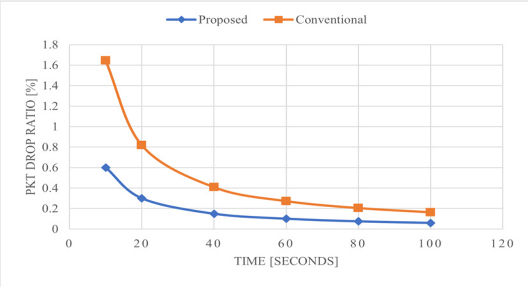
Algorithm 1 **Pseudocode for DNN-based Controller Post Function in DPI**

Require: $data_{training} = \{x_1, x_2, ... x_n, y_m\}$ denotes n of packet features in X with y_m target feature in the DPI
Ensure: Optimal packet classification into targets of $Y = \{y_1, y_2, y_3 .... y_m\}$ associating with k priorities
1: Initialize the learning rate (lr), number of epochs (n_{epoch}), batch size bs, communication interval, number of batch n_{batch}
2: Extend the class of base layer bl, middle layers mls, and output layer ol, from generalized DNN
3: **for** each epoch in range (n_{epoch}) **do**
4: Random the index, r_i, by arrange(length of X) and shuffle the random values
5: **for** each batch b in range(n_{batch}) **do**
6: Initialize mini-batch index by using r_i [$b \times bs: (b+1) * bs$]
7: Define the training (tr_{mb}) and validating (va_{mb}) mini-batch
8: Execute the forward propagation on tr_{mb}
9: Execute the backpropagation on va_{mb}
10: Update the parameters with defined lr
11: **end for**
12: Get the score metrics of the update parameters in ***get_error()*** and append to the list
13: **if** epoch % communication interval == 0 **do**
14: Store the error and current states for optimization
15: **end if**
16: **end for**
17: Execute ***get_accuracy()*** on training and validating data
18: **if** accuracy is satisfied **do**
19: Perform prediction after DNN ($bl \rightarrow mls \rightarrow ol$) model compiling
20: Transform the DNN class output (c) by bridging flow priority $k(c)$
21 Configure flow entry instruction by setting datapath id, table id, idle timeout, hard timeout, priority, match, and action.
22: **Else**
23: Re-execute line 2
24: **end if**

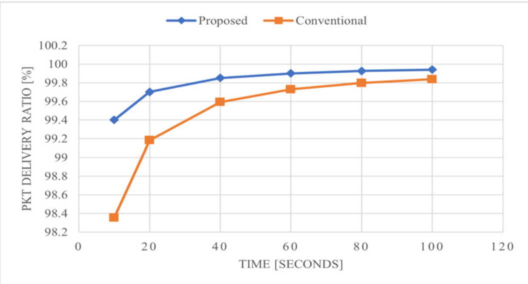
4 System Evaluation

4.1 Experimental Environment

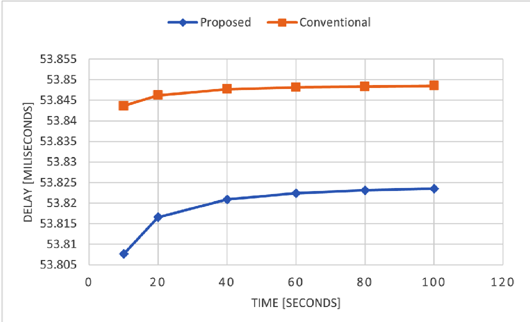
The end-to-end experiment was conducted by using computer software network simulator version 3 (NS3). There are 849,686 transmission packets from the radio network and the simulation topology was conducted within fronthaul, backhaul, and remote networks. The fronthaul gateway was supposed to be the serving severs and there are two servers in the simulation. Each server capacity was configured with different values, which server one and two were configured to 1000Mbps and 100Mbps for the serving data rate with 0ms and 200ms for the serving latency, respectively. The user devices were configured to have independent remote radio heads. In DNN implementation, Tensorflow and Keras are used within python programming language. The format of the input data is gathered as a csv file for conducting the training and testing simulation process. The architecture of network was configured as input layer with 4 main features $\{x_1, x_2, x_3, x_4\}$, 3 hidden layers, and output layer with 4 criticalities $\{y_1, y_2, y_3, y_4\}$. The main input features in this simulation were captured by NS3 simulator including transmitted bytes (TxBytes), received bytes (RxBytes), delay, and standard deviation (stdDev). TxBytes and RxBytes rows were numerical sizes of the packet. Delay was captured/gathered in the experiment at every configured access points. stdDev represents the formulation of variation, which mostly outputted an average of 0.000289s. The features were inputted for the classification procedure in terms of 4-level criticalities based



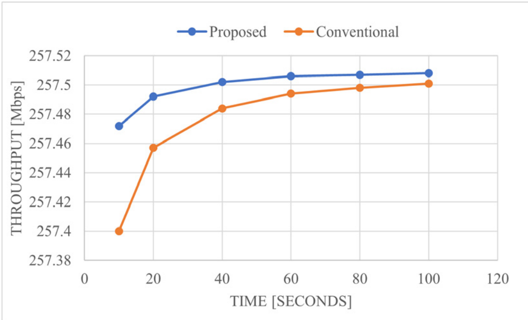
(Figure 2) The comparison of packet drop ratio between conventional and proposed scheme



(Figure 3) The comparison of packet delivery ratio between conventional and proposed scheme



(Figure 4) The comparison of delay between the conventional and proposed scheme



(Figure 5) The comparison of throughput between the conventional and proposed scheme

on 5G QoS identifier (5QI) 3GPP TS Release 17, which listed the sample services, error rate, packet delay budget, priority level, resource type, and default averaging window. In the proposed system, class 1 includes services of mission-critical conversational voice and video. Class 2 presents the sample services of real-time gaming, V2X message, and live-streaming.

3 consists of non-mission-critical push to talk. And class 4 includes the non-conversational video. Each class assigned weights on conditional packet delay, error rate, and priority level following the mentioned 5QI values. Moreover, to compare with the proposed scheme, conventional approach was conducted and formulated using shortest path algorithm to install the flow rules. QoS performance metrics, including packet drop and delivery ratios, delay, and throughput, are captured for comparison.

4.2 Results and Discussion

The results from the network simulator are a cooperative part of the methods based on the balancing of incoming traffic to fit with each serving server capacity. The proposed scheme contributed to the balancing of the incoming traffic to match with the obvious server condition and classification of the incoming traffic status based on DNN algorithm to predict the individual packet condition. Due to the scheme classification on the incoming packets into the different groups based on the server capacity, the recommendation of the specific server to be served/offloaded is configured. (Figure 2) shows the comparison of the packet drop ratio between proposed and conventional scheme. Based on the experiment output, the proposed scheme has reduced from an average packet drop ratio of 0.7% to 0.2166% compared to the conventional scheme. (Figure 3) illustrates the output of the end-to-end communication reliability of the proposed scheme, in terms of packet delivery ratio, compared to the conventional scheme. The proposed scheme is proved for handling the challenging issues of ultra-reliable real-time IoT communication perspective by achieving the average delivery ratio increment from 99.41% to 99.78% compared to the traditional approach.

The average end-to-end communication delay of the proposed and conventional scheme are illustrated in (Figure 4). In the experimental environment, the proposed scheme reduced the average communication delay from 53.8466ms to 53.8183ms compared to the conventional scheme. The reduction of communication delay is significant for mission-critical massive machine-types communication in IoT network due to the essential of ultra-low latency prerequisite. Moreover, IoT power consumption can be accordingly reduced based on the reduction of the communication delay.

The communication throughput of the proposed and conventional scheme is shown in (Figure 5), which illustrates the enhancement and efficiency of the data delivery rates summation from an average of 257.4666Mbps towards 257.4983Mbps. Therefore, the potential of forwarding user traffic and network congestion can be improved.

5 Conclusion

Multi-perspective traffic handling methods are required to overcome massive mission-critical IoT transmission in order to meet the prerequisite of 5G service-based communication. Therefore, this paper presents an adaptive real-time IoT service handling based on DNN algorithm for improving end-to-end network QoS. The proposed method focuses on the classification of incoming traffic into different action groups and the capacity of serving servers. The scheme has two main contributions including balancing the incoming traffic to meet the capacity of MEC servers and inspecting the different QoS profiles from multiple critical IoT packets. SDN/NFV-enabled control is utilized to set the priority rules and flow table management of each IoT criticality level. With DPI and DNN-based outputs, the steering configuration is enhanced.

In future works, the simulation of SDN/NFV environment will be included with enhanced virtual link connectivity. The DL algorithms for critical packet classifications will consider the fine-grained aspect and in-depth structure modifications.

Reference

[1] H. Lee and S. Lee, "Analysis of E2E Latency for Data Setup in 5G Network," Journal of Internet Computing and Services, Vol. 20, No. 5, pp. 113-119, 2019. http://dx.doi.org/10.7472/jksii.2019.20.5.113

[2] H. Lu, X. He, M. Du, X. Ruan, Y. Sun, and K. Wang, "Edge QoE: Computation Offloading With Deep Reinforcement Learning for Internet of Things," IEEE Internet of Things Journal, Vol. 7, No. 10, pp. 9255-9265, 2020. http://dx.doi.org/10.1109/JIOT.2020.2981557

[3] S. Kim and D. Kim, "Adaptive data transmission method according to wireless state in long range wide area networks," Computers, Materials & Continua, Vol. 64, No. 1, pp. 1–15, 2020. http://dx.doi.org/10.32604/cmc.2020.09545

[4] P. Tam, S. Math, and S. Kim, "Intelligent Massive Traffic Handling Scheme in 5G Bottleneck Backhaul Networks," KSII Transactions on Internet and Information Systems, Vol. 15, No. 3, 2021. http://dx.doi.org/10.3837/tiis.2021.03.004.

[5] S. Kekki et al., "MEC in 5G networks," ETSI White Paper, No. 28, pp. 1–28, 2018. https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf

[6] H. Hamzah, D. Le, M. Kim, and H. Choo, "Mobility-Aware Service Migration (MASM) Algorithms for Multi-Access Edge Computing," Journal of Internet Computing and Services, Vol. 21, No. 4, pp. 1-8, 2020. http://dx.doi.org/10.7472/jksii.2020.21.4.1

[7] "5G; NG-RAN; PDU Session User Plane protocol," ETSI TS 138 415, Vol. 15.0.0, 2018. https://www.etsi.org/deliver/etsi_ts/138400_138499/138415/15.00.00_60/ts_138415v150000p.pdf

[8] S. Math, P. Tam, and S. Kim, “Intelligent Media Forensics and Traffic Handling Scheme in 5G Edge Networks,” Security and Communication Networks, Vol. 2021, 2021.
http://dx.doi.org/10.1155/2021/5589352

[9] M. Mishra, M. Kang, and J. Woo, “Leveraging Big Data for Spark Deep Learning to Predict Rating,” Journal of Internet Computing and Services, Vol. 21, No. 6, pp. 33-39, 2020.
http://dx.doi.org/10.7472/jksii.2020.21.6.33

[10] 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

[11] 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

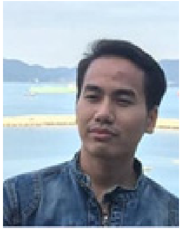
[12] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, “Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions,” IEEE Communications Surveys & Tutorials, Vol. 20, No. 3, pp. 2429-2453, 2018.
http://dx.doi.org/10.1109/COMST.2018.2815638

[13] 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.

[14] 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

[15] 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 ◑



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

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

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