

Efficient Resource Slicing Scheme for Optimizing Federated Learning Communications in Software-Defined IoT Networks[☆]

담 프 로 힘¹
Prohim Tam

맞 사¹
Sa Math

김 석 훈^{1,2*}
Seokhoon Kim

ABSTRACT

With the broad adoption of the Internet of Things (IoT) in a variety of scenarios and application services, management and orchestration entities require upgrading the traditional architecture and develop intelligent models with ultra-reliable methods. In a heterogeneous network environment, mission-critical IoT applications are significant to consider. With erroneous priorities and high failure rates, catastrophic losses in terms of human lives, great business assets, and privacy leakage will occur in emergent scenarios. In this paper, an efficient resource slicing scheme for optimizing federated learning in software-defined IoT (SDIoT) is proposed. The decentralized support vector regression (SVR) based controllers predict the IoT slices via packet inspection data during peak hour central congestion to achieve a time-sensitive condition. In off-peak hour intervals, a centralized deep neural networks (DNN) model is used within computation-intensive aspects on fine-grained slicing and remodified decentralized controller outputs. With known slice and prioritization, federated learning communications iteratively process through the adjusted resources by virtual network functions forwarding graph (VNFFG) descriptor set up in software-defined networking (SDN) and network functions virtualization (NFV) enabled architecture. To demonstrate the theoretical approach, Mininet emulator was conducted to evaluate between reference and proposed schemes by capturing the key Quality of Service (QoS) performance metrics.

✉ keyword : Deep Learning, Federated Learning, Internet of Things, Network Functions Virtualization, Software-Defined Networking

1. Introduction

Recently, the implementation of heterogeneous Internet of Things (IoT) is rapidly increasing with the goal of upgrading people's daily lives to a smart environment. IoT in 5G and future communication networks enables a variety of applications, including the Internet of Healthcare Things

(IoHT), Internet of Vehicles (IoV), Industrial IoT (IIoT), Internet of Agriculture Things (IoAT), and IoT on Shared Physical Infrastructure (IoT-SPI) [1]. However, due to the complex taxonomy of IoT devices, which includes a variety of communication protocols, interfaces, standard/proprietary types, resource constraints, sensing functionalities, and mobility, the prioritization of each application service has to be sliced and well-controlled in order to meet particular critical key performance indicator (KPI), Quality of Service (QoS) indicators, and Quality of Experience (QoE) expectations [2]. Multiple controllers allow softwarization capability and can exchange learning models via OpenFlow protocol with the programmability and scalability of the software-defined networking (SDN) paradigm. As a decentralized learning model, Algorithm 1 presents a brief method flow implementation of a machine learning algorithm, namely support vector regression (SVR), for IoT slices prediction [3, 4]. This model plays a vital role for local IoT nodes during peak hour intervals. However, SVR interacts with the central controller using deep neural

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 24 August 2021, Reviewed 7 September 2021, Accepted 14 October 2021]

☆ This work was supported by the MSIT(Ministry of Science, ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) in 2021“(2021-0-01399), and this research was funded by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2020R1I1A3066543).

☆ 본 논문은 2021년도 인터넷정보학회 춘계학술발표대회 우수 논문 추천에 따라 확장 및 수정된 논문임

networks (DNN) model for enhancing the precision and estimating weight initialization of each critical slice.

Algorithm 1 Brief pseudocode for SVR learning in decentralized controllers

Input: heterogeneous IoT packet inspection *data*
Ensure: slices prediction output and precision output for global model

- 1: Preprocess *data* using Binarizer, StandardScaler, and OneHotEncoder
- 2: **if** fit_transform(*data*) is true **do**
- 3: Train_test_split between training and testing subsets (70%, 30%)
- 4: Initialize *regressor_model* with RFB kernel and *accuracy_array*
- 5: **for** each training data **do**
- 6: Input the data into the *regressor_model*
- 7: Calculate score metrics and append to *accuracy_array*
- 8: **end for**
- 9: **if** average(*accuracy_array*) is satisfied **do**
- 10: **for** each testing data and “long-term incoming data” **do**
- 11: *regressor_model(data)* and return the slicing prediction
- 12: **end for**
- 13: **else**
- 14: Re-initialize *regressor_model* and re-train with modification
- 15: **end if**
- 16: **else**
- 17: Re-preprocess *data* with extra MinMaxScaler and Normalizer until the fit_transform(*data*) is true
- 18: **end if**

With difficulty in gathering privacy-sensitive IoT data and the high cost in raw data transmission towards the central cloud, federated learning inspires distributed on-device computation, which is practical for reducing big data IoT core network congestion and achieving acceptable accuracy on collaborative models [5, 6]. This converged machine learning technique enables iterative multi-model updates to aggregate the minimal local model errors and interact with neighboring node models for enhancing optimal precision satisfaction. At the beginning of the iteration, the global DNN model in the parameter server distributes the hyperparameters and model selection initialization in

$W_G(0)$ for a selected available client subset of overall participants, denoted as $C \subseteq K$ which consists of $\{1, 2, \dots, c\}$. In a particular client c , numerous discrete data batches denoted as $D_c = \{d_c(1), d_c(2), \dots, d_c(n)\}$ are generated or gathered for computing internal model. By using DNN algorithm, mean squared error (MSE) is used as a loss function of the model, denoted as $L(W_c)$, in decentralized and centralized models [7, 8]. At a particular t index iteration, a local model using $d_c(n)$ data batch is formulated based on $W_G(t)$ hyperparameters to optimize $L(w_c(t))$ values on local model of client c , $w_c(t)$. Within the argument of minimum, the parameters that achieve the lowest possible error from the beginning to t step are chosen for aggregation in the global parameter server. By following this procedure, global DNN algorithm can use maximum data distribution from stable IoT clients to perform specific services (e.g., classification) in management platform interfaces [9]. To ensure reliable multi-dimensional update interaction, prevent packet loss, and restrain accuracy decrement, an intelligent resource serving adjustment is required. Multi-access edge computing (MEC) and network functions virtualization (NFV) have to be emerged as functions and computation abstractions for reliable federated learning communications [10-13].

The paper is organized as follows. Section 2 presents system models of the proposed architecture. In section 3, the proposed efficient resource slicing scheme in federated learning communications is thoroughly described. Section 4 shows the simulation environment, performance metrics, and result discussions. In section 5, the conclusion is presented.

2. System Models

In this section, procedures of the proposed scheme and system architecture are presented. Decentralized SVR-based settings, centralized DNN-based global model, control slicing orchestration, and long-term self-sufficiency are the four primary steps of the proposed system model. The slices are primarily set into three conceptual conditions. (Figure 1) shows a flowchart for deploying virtual network functions (VNF) placement corresponding with SDN-based inspection and controlling resource orchestration for mission-critical,

mid-mission-critical, and non-mission-critical IoT application slices. Decentralized learning in hierarchically SDN distributed controllers is required to extract the core global model for local slicing prediction in heterogeneous IoT networks because numerous nodes with exclusive flow entries create a massive bottleneck of the system towards a single central control entity. The feature extraction of IoT node information is used to train a non-linear SVR model. SVR requires various criteria for forensics strategy, including historical training datasets, synchronized weight initialization, and target slices. With the capabilities of OpenFlow protocol, the southbound application programming interface (API) feasibly triggers the required data from infrastructure layer for virtual softwarization-enabled mechanism through device and resource abstraction layer (DAL) [14]. The intensive computing portions will be completed during off-peak hour intervals. During the bottleneck periods, the optimal model for the decision process in the central controller is initialized and transferred to the distributed units for local processing. The proposed scheme requires orchestrating the SDN/NFV-enabled resource placement and function chaining for each slice of federated learning communications based on collaborative model decision. With SDN controller as a virtualized infrastructure manager (VIM), the virtualization layer of NFV-enabled MEC physical resource pools are adjusted through network service abstraction layer (NSAL) accordingly. If the historical resource utilization outputs an evaluation of high resource-efficiency in SDN database, the

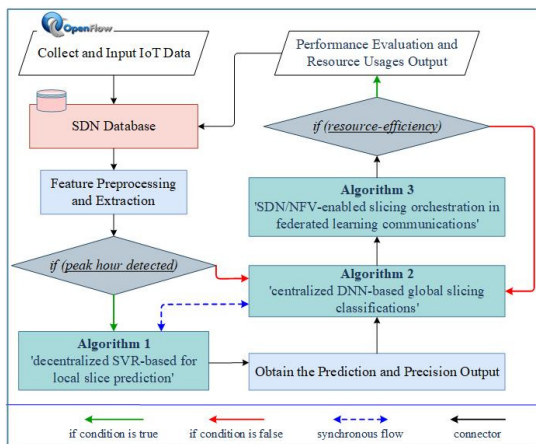
configured action will be stored with high expected long-term reward values. Therefore, the proposed scheme will exploit that action in the future. The features of IoT slice applications, including latency, packet loss, and example scenarios are defined for classifications [2, 15].

3 Proposed Approach

In this section, an efficient resource slicing scheme for optimizing federated learning communications in SDN/NFV-enabled system architecture is presented. To efficiently classify IoT slicing, algorithm 2 is employed as a function approximation with the provided system models for the centralized learning model using TensorFlow-based implementation. With DNN-based global slicing classifications, computation-intensive tasks are modified in off-peak intervals and synchronously interact with decentralized weigh initialization. With a system model of sufficient decentralized SVR-based controllers and centralized DNN-based IoT slice classifications, the optimal weight is exchanged; therefore, the reliability is significantly enhanced. With known slicing classes, the proposed softwarization-enabled scheme requires two primary procedures to sequentially consider, including algorithm flows of efficient resource slicing orchestration and optimized federated learning communication flows.

3.1 Flows of Efficient Resource Slicing Orchestration

In proposed software-defined IoT (SDIoT) networks architecture, multi-controllers are placed for adaptive scalability, high programmability, and global view. With NFV as functions abstraction, SDN controllers as a VIM and VNF are interacted through orchestration layers to ensure that the configured flow entry contains a sufficient actions and priority rule installations. The flow entry installation with reliable service chains is crucial for latency-efficient aspects. VNF deployment in a forwarding graph descriptor is essential to compensate its function backup instances and reliability. With conceptual slicing output, VNF forwarding graph (VNFFG) are rendered to create optimal service function chaining (SFC) with sufficient active primary functions in each VNF. With mission-critical slicing class,



(Figure 1) The flowchart of proposed system model

Algorithm 2 DNN-based global slicing classifications for computation-intensive tasks and weight interactions

Input: IoT data features of i general feature, denoted as $X = \{x_1, x_2, \dots, x_i\}$, and target feature $y_m = \{y_1, y_2, y_3\}$

Ensure: slice classification output and optimal weigh/biases

- 1: Preprocess X features towards *fit_transform()* execution
 - 2: Split the data between X_{train} , X_{test} , y_{train} , and y_{test}
 - 3: Initialize regularization (λ , α), *random_normal*(weights w_l , biases b_l) within $l \in Z^+$ of hidden layers
 - 4: **for** each hidden layer in *range*(l) **do**
 $layer_l = \text{Tanh}(\text{tf.add}(\text{tf.matmul}(layer_l, w_l), b_l))$
 - 5: (within TensorFlow platform implementation. *matmul*: matrix multiplication and *Tanh* activation function are used)
 - 6: Apply regularization on each weight
 - 7: **end for**
 - 8: Calculate the loss function with regularization values on last hidden layer $l[-1]$
 - 9: Use stochastic gradient descent (SGD) as optimization algorithm by *optimizer(a).minimize(loss)*
 - 10: $\equiv = \text{tf.global_variables_initializer}()$
 - 11: **if** training accuracy is satisfied **do**
 - 12: Create *builder* for *SavedModelBuilder()*
 - 13: *Session.run()* (\equiv) and cross-validation
 - 14: Use saved model to session run on $\{X_{test}, y_{test}\}$
 - 15: **if** testing accuracy is satisfied **do**
 - 16: Use saved model to run on future upcoming *data*
 - 17: Save classification outputs and optimal (w_l, b_l)
 - 18: **otherwise** re-run model and *optimizer()*
 - 19: **else**
 - 20: Re-execute from line 4
 - 21: **end if**
-

the proposed NFV orchestrator keeps the VNF connected by following the VNFFG descriptor in the form of SFC, where each VNF policy consists of extra capacity on virtualization deployment unit (VDU). The serving virtual machine execution is correspondingly increased. By isolating each descriptor on each slicing class, the network service requirements are well-classified and well-prioritized based on the initial VNF configuration with an adequate element management system. With the outputs of global slicing classifications, the proposed orchestrator efficiently configure the descriptor with cooperative prior insight to trigger the MEC resource pools in NFV infrastructure (NFVI).

3.2 Optimized Federated Learning Communication Flows

Algorithm 3 presents the proposed SDN/NFV-enabled virtual resource slicing orchestration to optimize federated learning communications. With efficient resource control, the local model updates from multi-dimensional clients can execute an instruction sets with optimal actions, counters, and priorities. The proposed forwarding rules ensure the reliability of local model $w_c(t)$ updates and global model $W_G(t)$ distributions in order to improve the precision.

Algorithm 3 Pseudocode on SDN/NFV-enabled virtual resource slicing orchestration to optimize federated learning

Input: slice classification outputs from decentralized or centralized controllers as target feature $y_m = \{y_1, y_2, y_3\}$

Ensure: optimal flow entry to execute model updates

- 1: Initialize the selected clients $C \subseteq K$, number of epoch n_e , $W_G(0)$, and learning rate α
 - 2: [Local client C]
 - 3: **for** each $d_c(n)$ in *range*(n) **do**
 - 4: Execute $L(w_c(n))$ with input of α and $W_G(0)$
 - 5: Find optimal local model by $w_c(t)^* = \text{argmin}(L(w_c(0)) \dots L(w_c(n)))$
 - 6: Update the local model parameters to central server
 - 7: **for** each new incoming OFPT_PACKET_IN **do**
 - 8: Perform/obtain the slicing target as y_m
 - 9: Adjust the actions and priorities accordingly to VNFFG mapping
 - 10: Detect y_1 for extra VNF backup instances
 - 11: Orchestrate the VDU and VM acceleration
 - 12: Execute the instruction sets
 - 13: **end for**
 - 14: [Parameter Server]
 - 15: **for** each iteration in *range*(t) **do**
 - 16: Perform averaging aggregation for future W_G
 - 17: Update the selected path for experience replays
 - 18: Distribute to global model to random subset C
 - 19: **end for**
-

Within iterations of federated learning communications, the proposed scheme performs the slicing classifications in

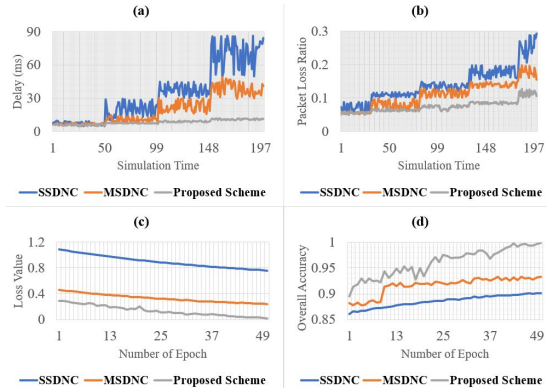
both decentralized and centralized controllers to determine the target features. With known classes, the scheme can obtain the prior insight for orchestration in terms of VNF resource placement. The rendered VNFFG is mapped towards sufficient resource in NFVI for each local model class by proposed SDN controller as a VIM. Therefore, the local client updates the optimal model $w_c(t)^*$ in a reliable manner. For parameter server, after the averaging aggregation is performed within each iteration, the selected path are updated in experience replays. Thus, the central federated learning server distributes the global model $W_G(t)$ with sufficient communication resource adjustment.

4 System Evaluation

4.1 Experimental Environment

SDN/NFV experiment was conducted by using Mininet emulator with RYU SDN controller and Mini-NFV framework [16-18]. 2 Ubuntu linux with 20.04.2 LTS are used as hosting data plane and remote SDN controller, respectively. Mininet 2.5 supports OpenFlow protocol in the experiment and primarily implements in the data plane host server. RYU 4.32 is used as a remote SDN controller which can configure the flow tables and post functions by Python programming language or RYU-based FlowManager application. Iperf 3.7 (cJSON1.5.2) is used to evaluate and capture the QoS performance metrics. Mini-NFV used OASIS TOSCA template to serve as a NFV MANO framework on Mininet emulator. Tensorflow-Federated is used to simulate the federated learning performance with MNIST handwritten dataset [19].

The simulation time is set to 200 seconds which transmit the payload sizes of 1,024 bytes. The network conditions are configured as IoT congested states. User datagram protocol (UDP) is indicated as the main communication protocol for time-sensitive aspects. The learning rate and number of episodes are 0.09 and 50, respectively. The reference schemes consists of single centralized controller and multiple controllers, denoted as SSDNC and MSDNC, which configure the flow tables based on shortest path algorithm.



(Figure 2) The result comparison in terms of (a) delay, (b) packet loss ratio, (c) loss value, and (d) overall accuracy

4.2 Result and Discussions

The proposed scheme tackled intelligent resource slicing management by using proposed decentralized controllers to predict the incoming traffics and setting the forwarding rules to optimal serving gateways in peak hour intervals. The centralized controller is used to optimize the weight and classify the slicing using DNN-based algorithm with computation-intensive manner in off-peak hour intervals. (Figure 2) presents the result comparison between proposed and reference schemes in terms of (a) delay, (b) packet loss ratio, (c) loss value, and (d) overall accuracy. By applying the efficient resource slicing management scheme, the average delay of federated learning iteration was reduced 11.309ms and 25.427ms compared to MSDNC and SSDNC, respectively. The packet loss ratio of proposed scheme is 0.07527% within 200 seconds simulation, which is 0.03401% and 0.06577% lower than MSDNC and SSDNC, respectively. With high packet loss of local model updates, the accuracy of federated learning is greatly decreased. However, the proposed scheme achieved a closing loss value of 0.01636, which is 0.2237 and 0.7399 lower than MSDNC and SSDNC, respectively. The overall accuracy performance of proposed scheme is 0.9986 or 99.86%, which is 0.06553% and 0.09764% higher than MSDNC and SSDNC, respectively. With the proposed control, the performance of federated learning communications is notably enhanced.

5 Conclusion

The trade-off between communication-critical and computation-intensive was balanced by properly utilizing decentralized and centralized controllers in a particular interval setup. The scheme utilizes SDN controllers as a VIM to orchestrate the optimal forwarding graph for each slice criticality. The properties and capabilities of VDUs, connection points, and virtual links were considered in NFV MANO framework simulation. Federated learning communications was improved by slicing the local model update classes, forwarding the global model distributions optimally, and adjusting a sufficient NFV-enabled MEC capacity to ensure the reliability and model performance.

Reference

- [1] "Overview of the Internet of Things, Y.4000/Y.2060 (06/2012)," ITU, Geneva, Switzerland, ITU-Recommendation Y.2060, p. 53, 2016.
https://www.itu.int/rec/dologin_pub.asp?lang=e&cid=T-REC-Y.2060-201206-1!!PDF-E&type=items
- [2] S. Sinche et al., "A Survey of IoT Management Protocols and Frameworks," *IEEE Communications Surveys & Tutorials*, Vol. 22, No. 2, 2020.
<http://dx.doi.org/10.1109/COMST.2019.2943087>
- [3] Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research* 12, pp. 2825-2830, 2011.
<https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>
- [4] S. Math, P. Tam, and S. Kim, "Intelligent real-time iot traffic steering in 5g edge networks," *Computers, Materials & Continua*, Vol. 67, No.3, pp. 3433 - 3450, 2021.
<http://dx.doi.org/10.32604/cmc.2021.015490>
- [5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, Vol. 54, pp. 1273 - 1282, Apr. 2017.
<https://arxiv.org/abs/1602.05629>
- [6] S. Math, P. Tam, and S. Kim, "Reliable Federated Learning Systems Based on Intelligent Resource Sharing Scheme for Big Data Internet of Things," *IEEE Access*, Vol. 9, pp. 108091-108100, 2021.
<http://dx.doi.org/10.1109/ACCESS.2021.3101871>
- [7] Mohammed Aledhari, Rehma Razzak, Reza M. Parizi, and Fahad Saeedi, "Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Access*, Vol. 8, pp. 140699-140725, 2020.
<http://dx.doi.org/10.1109/ACCESS.2020.3013541>
- [8] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A Survey on Federated Learning for Resource-Constrained IoT Devices," *IEEE Internet of Things Journal*, Jul. 2021.
<http://dx.doi.org/10.1109/JIOT.2021.3095077>
- [9] C. Jung, "Prioritized Data Transmission Mechanism for IoT," *KSII Transactions on Internet and Information Systems*, Vol. 14, No. 6, pp. 2333-2353, 2020.
<http://dx.doi.org/10.3837/tiis.2020.06.002>
- [10] A. Abid, M. F. Manzoor, M. S. Farooq, U. Farooq, and M. Hussain, "Challenges and Issues of Resource Allocation Techniques in Cloud Computing," *KSII Transactions on Internet and Information Systems*, Vol. 14, No. 7, pp. 2815-2839, 2020.
<http://dx.doi.org/10.3837/tiis.2020.07.005>
- [11] D. Li, J. Lan, and Y. Hu, "Central Control over Distributed Service Function Path," *KSII Transactions on Internet and Information Systems*, Vol. 14, No. 2, pp. 577-594, 2020.
<http://dx.doi.org/10.3837/tiis.2020.02.006>
- [12] H. Choi, S. M. Raza, M. Kim, and H. Choo, "UDP Flow Entry Management for Software-Defined Networking," *Journal of Internet Computing and Services*, Vol. 22, No. 2, pp. 11-17, 2021.
<http://dx.doi.org/10.7472/jksii.2021.22.2.11>
- [13] 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>

- [14] 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>
- [15] A. Machwe et al., "5G and Vertical Services, use cases and requirements," 5G-PICTURE Project, Telecom Italia, Rome, Italy, Jan. 2018. [Online]. Available: https://www.5g-picture-project.eu/download/5g-picture_d21.pdf
- [16] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapidprototyping for software-defined networks," in *Proceedings of the 9th ACM SIG-COMM Workshop on Hot Topics in Networks*. New York, NY, USA, pp. 19:1 - 19:6, 2010.
<http://doi.acm.org/10.1145/1868447.1868466>
- [17] "Ryu controller," [Online]. Available: <http://osrg.github.com/ryu/>
- [18] "Mini-NFV," [Online]. Available: <https://github.com/josecastillolema/mini-nfv>
- [19] M. Abadi et al., "Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv:1603.04467*, 2016.
<https://arxiv.org/abs/1603.04467>

● Authors ●



담 프로 힘 (Prohim Tam)

2014: Paragon International University (Cambodia), Management of Information System (Bachelor Degree)
 2019 ~ Present: Soonchunhyang University (Korea), Software Convergence (Combined Master and Ph.D. Degree)
 Research Interests: 5G Service-Based Orchestration, Quality of Service, SDN/NFV, AI in Bigdata Transmission, Edge Computing, and IoT.
 E-mail : prohingtam@gmail.com



맛 사 (Sa Math)

2014: Royal University of Phnom Penh (Cambodia), Telecommunication and Electronic Engineering (Bachelor Degree)
 2018 ~ Present: Soonchunhyang University (Korea), Software Convergence (Combined Master and Ph.D. Degree)
 Research Interests: 5G Distributed and Core networking, IoT, Quality of Service, Software Defined Mobile Edge Computing, 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