Department: Head
Editor: Name, xxxx@email

Proxy Experience Replay: Federated Distillation for Distributed Reinforcement Learning

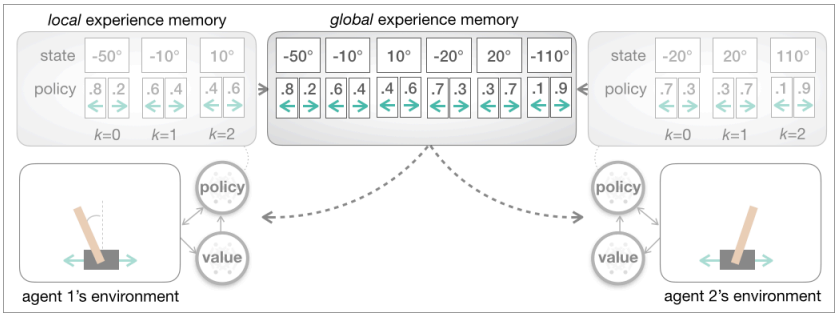
Han Cha*, Jihong Park†, Hyesung Kim‡, Mehdi Bennis†, and Seong-Lyun Kim*§

*Yonsei University
†University of Oulu
‡Samsung Electronics Co., Ltd.
§Corresponding author. Contact him at slkim@yonsei.ac.kr.

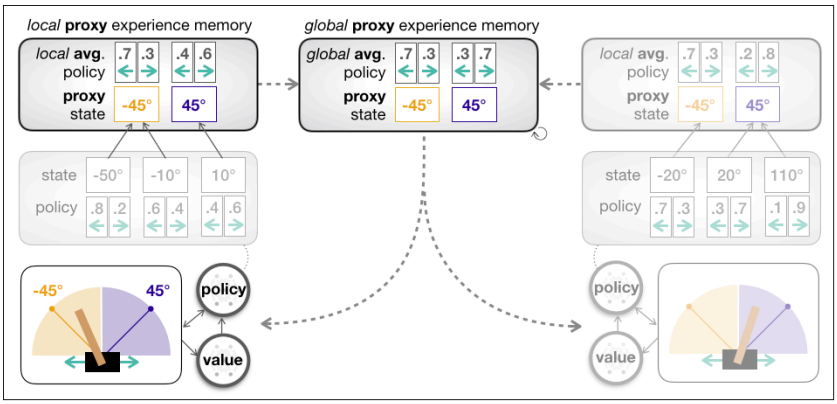
***Abstract*—Traditional distributed deep reinforcement learning (RL) commonly relies on exchanging the experience replay memory (RM) of each agent. Since the RM contains all state observations and action policy history, it may incur huge communication overhead while violating the privacy of each agent. Alternatively, this article presents a communication-efficient and privacy-preserving distributed RL framework, coined federated reinforcement distillation (FRD). In FRD, each agent exchanges its proxy experience replay memory (ProxRM), in which policies are locally averaged with respect to proxy states clustering actual states. To provide FRD design insights, we present ablation studies on the impact of ProxRM structures, neural network architectures, and communication intervals. Furthermore, we propose an improved version of FRD, coined mixup augmented FRD (MixFRD), in which ProxRM is interpolated using the mixup data augmentation algorithm. Simulations validate the effectiveness of MixFRD in reducing the variance of mission completion time and communication cost, compared to the benchmark schemes, vanilla FRD, federated reinforcement learning (FRL), and policy distillation (PD).**

***Keywords*– Machine learning, Distributed Artificial Intelligence, Wireless Communication**

LEARNING from collective experience is a hallmark of intelligent agents powered by distributed deep reinforcement learning (RL), ranging from autonomous drones[1] to self-controlled robots in smart factories.[2] Policy distillation (PD) is one of the most popular distributed RL methods.[3] As shown in Fig. 1(a), in PD, each agent has a neural network (NN) model and its *local replay memory* storing the action policies taken at the agent's observed states. This local replay memory is exchanged across agents. By replaying the globally collected replay memory, every agent



(a) Policy distillation.



(b) Federated reinforcement distillation with proxy replay memory.

Figure 1: A comprehensive structure of distributed deep RL frameworks.

trains its NN model while reflecting the experiences of other agents.

However, in practice, local experience memories can be privacy-sensitive, which prohibits their exchanges. As an example, for autonomous surveillance drones operated by multiple operators, these drones are willing to train their NNs collectively, but without revealing their private observations and policy records. Furthermore, the dimension of possible states can be large, as in high-precision robot manipulation. This leads to huge replay memory sizes, and exchanging them is ill-suited for agents' operations in real-time.

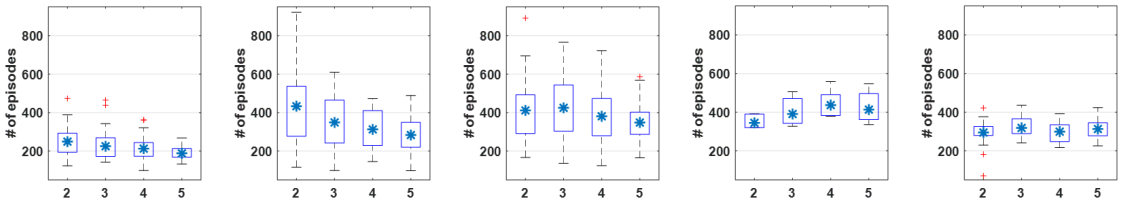
Alternatively, our prior work proposed a privacy-preserving and communication-efficient distributed RL framework, coined *federated reinforcement distillation (FRD)*.[4] As illustrated by Fig. 1(b), in FRD, each agent exchanges its *proxy experience memory replay memory (ProxRM)* consisting of locally averaged policies with respect to proxy states. Under this memory structure, policies are locally averaged over time, while actual states are clustered into the nearest proxy states. Exchanging proxy replay memory can thereby preserve privacy while reducing the communication payload size.

In this article, we aim to provide an FRD design guideline, shedding light on how coarse the proxy states are and how frequent the proxy experience memories are exchanged under which NN architectures. Furthermore, we propose an improved version of FRD, named MixFRD, in which the globally collected proxy replay memory is interpolated using the mixup data augmentation algorithm.[5] Finally, we demonstrate the effectiveness of MixFRD by comparing it with two baseline algorithms, PD and federated reinforcement learning (FRL),[6] in terms of mission completion time and communication payload size.

Federated Reinforcement Distillation (FRD)

In FRD, for a given environment and mission, each agent locally updates its NN model by observing the states while taking actions, and stores

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIS.2020.2994942, IEEE Intelligent Systems



(a) Setting 1. (b) Setting 2. (c) Setting 3. (d) Setting 4. (e) Setting 5.

Figure 2: Cartpole mission completion time under different settings provided in Table 1.

the average action policies per proxy state to construct a ProxRM that is periodically exchanged across agents. For the sake of explanation, hereafter, we focus on the Cartpole environment and describe FRD operations as detailed next.

Environmental Settings

We evaluate the performance of FRD using the *Cartpole-v1* in the OpenAI gym environment.[7] The objective of this game is to make the pole attached to a cart upright as long as possible, by moving the cart left or right. Each agent gets a score of $+1$ for every time slot that the pole remains upright during an episode. Every episode ends if any of the agents has: (i) the pole fallen down; (ii) the cart moved outside a pre-defined range; or (iii) the score reaching 500. Playing the game with multiple episodes, agents complete their mission when any of them first reaches an average score of 490, where the average is taken across 10 latest episodes.

Standalone and Distributed RL Operations

Each agent under study runs the advantage actor-critic (A2C) RL algorithm. In A2C, there is a pair of NNs, in which a policy NN determines the agent's optimal action policies of input states, and the value NN examines the optimality of the policies. To this end, for each action decision, the value NN produces value that is used for calculating an advantage. When the advantage is near zero, the determined action is closer to the optimal action compared to other possible actions. Therefore, the policy NN is trained by minimizing the advantage produced by the value NN. While running the aforementioned standalone RL operations, agents periodically exchange their policy NN outputs and inputs through ProxRMs, whose structures are elaborated in the next subsection. Each agent replays the ProxRM similar to the training procedure of supervised learning.

Table 1: Ablation study settings.

Setting	# of sections for state components (S)	communication period (E)	# of weights per layer (n)	# of hidden layers (ℓ)
1	100	25	**24**	2
2	**100**	25	**100**	2
3	**50**	25	100	2
4	100	**10**	24	1
5	100	**50**	24	1

Proxy Experience Replay Memory Structures

In the Cartpole environment, each state is described by four components: cart location, cart velocity, pole angle, and the angular velocity of the pole. To construct a ProxRM, the state is evenly divided into a number S of clustered values per each component and give an index number to each cluster. Along with the state clusters, we categorize the RM following the nearest distance rule and average out the policies. The proxy states are defined by the middle value of each state cluster. For example, the proxy state is -45° if the state cluster is [-90°, 0°), as illustrated in Fig. 1(b). Finally, we have ProxRM with (proxy state, averaged policies) tuples.

Ablation Study on FRD

To provide FRD design insights, in this section, we study the impact of ProxRM structures, NN architectures, and communication protocols. Specifically, we consider five parameters: the number S of clusters per state component, ProxRM exchanging period E, the number L of a multi-layer perceptron (MLP) NN hidden layers, and the number n of weights per layer.

Under the settings described in Table 1, Fig. 2 shows the following impacts of FRD design parameters.

- Too small number S of clusters makes FRD fail to obtain federation gains. As opposed

to Fig. 2(b), the mission completion time in Fig. 2(d) does not decrease with the number of agents due to too small S. Since S determines the communication payload sizes in FRD, there is a *trade-off between communication efficiency and scalability*.

- Too frequent ProxRM exchanges may be harmful, as observed by Fig. 2(d) compared to (e). Without a sufficient number of local RL iterations, the observed states are hardly correlated across agents, negating the effectiveness of their federation. Since that E determines the number of communication rounds, it is important to *balance the local RL iterations and ProxRM communication interval*.
- A larger NN often has higher sample complexity, and does not always outperform a smaller NN, as shown in Fig. 2(b) compared to (a). Nonetheless, for a larger NN, federation gain is higher, since exchanged ProxRMs more overcomes the lack of training samples. It is therefore crucial to *optimize the number of federating agents subject to their NN architectures.*

Mixup Augmented FRD (MixFRD)

In this section, we propose an enhanced version of FRD coined *MixFRD*, which utilizes the augmentation scheme to enrich ProxRM. With few agents, the exploration of the environment may be insufficient compared to that of many agents. The lack of exploration of ProxRM causes performance instability in the FRD framework. We handle this problem by applying the data augmentation algorithm by generating synthetic ProxRM with downloaded global ProxRM at the local agent.

A *mixup*[5] algorithm is one of the most promising augmentation schemes. The main purpose of mixup is to enhance the generalization capability of NN by generating unseen data samples. The mixup algorithm creates a new data sample and its corresponding label by a linear combination of the randomly picked two existing data samples. In general, but not limited, the portion data samples follows the Beta distribution with coefficient $0 < \alpha = \beta < 1$. The portion is selected near 0 or 1 value in this setting.

To apply mixup to the FRD framework, there is a consideration about selecting what kind of

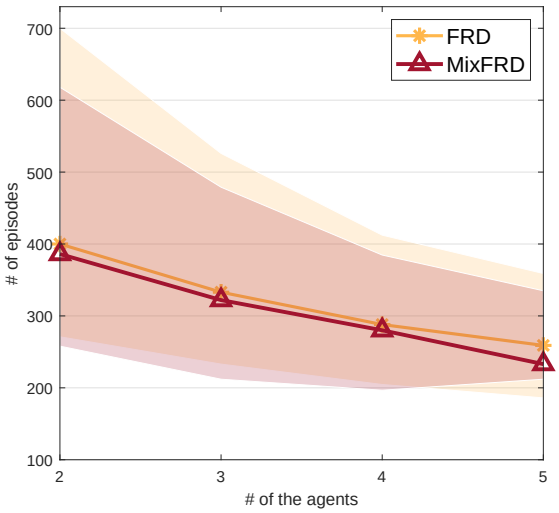


Figure 3: Mission completion time comparison between FRD and MixFRD ($S = 30$, $E = 25$, $n = 50$, $l = 2$).

two samples in ProxRM and the portion. If we pick two memories randomly, the augmented ProxRM lose the reality, in other word, proxy state and averaged action policy are not correlated anymore. This augmented sample may cause severe performance degradation. On the other hand, new ProxRM is similar to either original samples in the general setting of mixup, which is no help for expanding the ProxRM. Therefore, we need to select highly correlated ProxRM samples and the portion reflecting the samples equally.

In the Cartpole environment, the angle of the pole is the most relevant information to accomplish the group mission among the four components. To ensure the correlation between two memories, we sort the ProxRM along with the angle of the pole and select two adjacent replay memories. With this memory, we conduct the mixup with a portion of 0.5. We can say that the newly constructed ProxRM is an interpolated version of the original one. In Fig. 3, the MixFRD shows some gain in terms of variance as expected.

We can measure the correlation among the ProxRM by measuring the distance between each state clusters, e.g., Euclidean distance, cosine distance, Jaccard distance, etc.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MIS.2020.2994942, IEEE Intelligent Systems

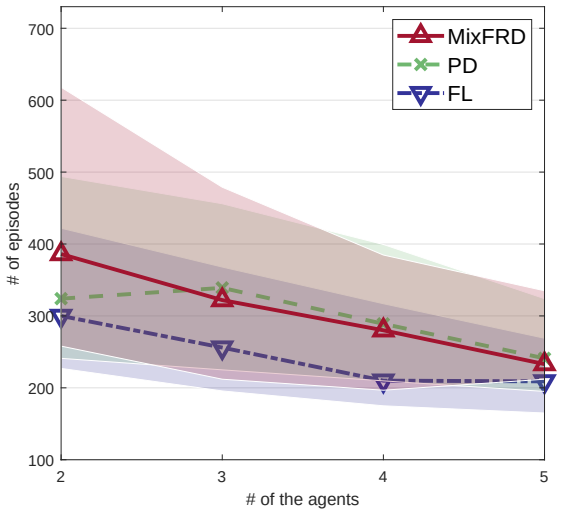


Figure 4: Mission completion time comparisons among MixFRD, PD, and FRL.

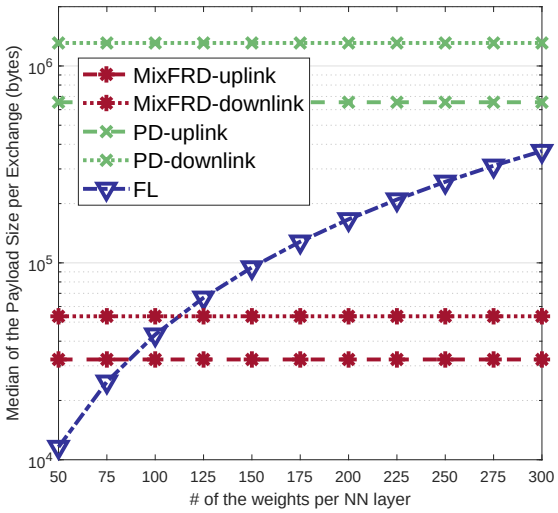


Figure 5: Uplink and downlink payload size comparison for two agents with server.

Performance Evaluation

We numerically compare the performance of the proposed MixFRD framework with two baseline distributed deep RL frameworks: policy distillation (PD)[3] and federated reinforcement learning (FRL).[6] Each framework uses a different form of knowledge: each framework exchanges ProxRM, RM, weight parameters of local NN and the sizes of each knowledge are denoted by M^p, M, $W(n,l)$, respectively. The subscripts L, G denote the local and global knowledge, respectively. Note that agents using FRL reflect global knowledge by exchanging the local weights to the global weight parameters.

We evaluate the performance with two metrics: (i) mission completion time, (ii) payload size per knowledge exchange. We conduct the simulation under the following settings: $S = 30$, $E = 25$, $n = 50$, $l = 2$. The lines represent the median, and the colored area is between the top-25 percentile and the top-75 percentile of each case. In the latter evaluation, we conduct the simulation for two agents and adjust n of policy network only.

The ProxRM size M^p is determined by the number of distinct state observations, and upper bounded by the entire cluster dimension S^4. As S goes to infinity, ProxRM converges to RM, i.e., MixFRD is identical with PD. Note that ProxRM size M^p increases with S only if the preceding state observations are sufficiently diverse.

Table 2: Communication payload sizes.

Cases	Payload size (byte)
MixFRD-uplink	$b_p \times M_L^p$
MixFRD-downlink	$b_p \times M_G^p$
PD-uplink	$b_{rm} \times M_L$
PD-downlink	$b_{rm} \times M_G$
FRL	$b_w \times W(n,l)$

The communication payload sizes of each framework are defined in Table 2. We define the unit size of ProxRM, RM, and weights of FRL as $b_p = 12$, $b_{rm} = 24$, and $b_w = 4$, respectively. RM comprise of six float components, which are four state components and two action policies, respectively. ProxRM comprises of one signed integer and two float components, which are index of state cluster and two action policies, respectively. The weight of FRL is float integer, which is the weight parameter. In the case of MixFRD, the server defines the ProxRM structure in advance, in which all the agents and server have information about the entire state cluster index and corresponding proxy state. Therefore, the agents in MixFRD share only the state cluster index.

Comparison with PD

In Fig. 4, the proposed MixFRD has a compatible performance with the PD when the number of cooperating agents are above two. Furthermore, the MixFRD has far less payload size compared to that of the PD, as in Fig. 5. Although the results are captured in the two-agents case, the gap between the two frameworks cannot be narrowed as the number of cooperating agents increases. The payload sizes of the MixFRD and PD have a positive correlation with the exploration duration of learning agents. As the agents hold the pole straight, the number of the state observations increases within similar bounds. By utilizing MixFRD, the agents make these RMs with few ProxRMs, which has far fewer number compared to the RM. Furthermore, as the cooperating agents increase, the number of global RM increases exponentially. Therefore, MixFRD can replace PD by exchanging the RM payload size.

Comparison with FRL

In Fig. 4, the performance of FRL overwhelms that of the MixFRD. However, as the size of local NN increases, the payload size of FRL increases with the squared scale of the size, as presented in Fig. 5. The MixFRD framework has an advantage in payload size constrained scenarios, e.g., cooperating through wireless channels like swarm of drones, automated process of smart factories, etc. Furthermore, all the agents require homogeneous local NN structure to use FRL, which is a strong assumption due to the diversity of computational capability of . The MixFRD can provide compatible distributed RL framework among agents which have heterogeneous local NN structure.

Conclusions

In this article, we proposed a communication-efficient and privacy-preserving distributed RL framework, FRD, in which agents exchange ProxRMs without revealing raw RMs. Furthermore, we proposed an improved version of FRD, MixFRD, by leveraging the mixup data augmentation algorithm to interpolate ProxRMs locally. Our ablation studies showed that improving the FRD performance hinges on ProxRM structures, NN architectures, and communication protocols. Co-designing these system parameters could, therefore, be an interesting topic for future research.

REFERENCES

1. J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proceedings of the IEEE,* vol. 107, no. 11, pp. 2204-2239, November 2019.
2. H. Shiri, J. Park, and M. Bennis, "Massive autonomous UAV path planning: A neural network based mean-field game theoretic approach," in Proc. of the *Global Communications Conference (Globecom)*, 2019.
3. A. Rusu, S. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," in Proc. of the *International Conference on Learning Representations (ICLR),* San Juan, Puerto Rico, May 2016.
4. H. Cha, J. Park, H. Kim, S.-L. Kim, and M. Bennis, "Federated reinforcement distillation with proxy experience memory," presented at *28th International Joint Conference on Artificial Intelligence (IJCAI-19) 1st Wksp. Federated Machine Learning for User Privacy and Data Confidentiality (FML'19)*, Macau, August 2019.
5. H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: beyond empirical risk minimization," in Proc. of the *International Conference on Learning Representations (ICLR),* Vancouver, BC, Canada, April-May 2018.
6. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. of the *the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 54, Fort Lauderdale, FL, USA, April 2017.
7. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint: https://arxiv.org/abs/1606.01540*, 2016.

ACKNOWLEDGMENT

Han Cha is currently pursuing a Ph.D. degree at the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea. He received a B.S. degree at Yonsei University in 2015. His current research includes resource management, interference-limited networks, radio hardware implementation, AI-assisted wireless communication, IoT communications, dynamic spectrum access, cognitive radio. He was awarded the IEEE DySPAN Best Demo Paper in 2018, titled opportunistic map based flexible hybrid duplex systems in dynamic spectrum access as a coauthor. Contact him at chan@ramo.yonsei.ac.kr.

Jihong Park is currently a Post-Doctoral Researcher with the University of Oulu, Finland. His research interests include distributed machine learning and ultra-reliable designs in 5G communication systems and beyond. He received the B.S. and Ph.D. degrees from Yonsei University, Seoul, Korea. From 2016 to 2017, he was a Post-Doctoral Researcher with Aalborg University, Denmark. He was a Visiting Researcher with Hong Kong Polytechnic University; KTH, Sweden; Aalborg University, Denmark; and New Jersey Institute of Technology, USA, in 2013, 2015, 2016, and 2017, respectively. From 2020, He will be a Lecturer at the School of Information Technology, Deakin University, Australia. He received the 2014 IEEE GLOBECOM Student Travel Grant, the 2014 IEEE Seoul Section Student Paper Contest Bronze Prize, and the 6th IDIS-ETNEWS (The Electronic Times) Paper Contest Award sponsored by the Ministry of Science, ICT, and Future Planning of Korea. Contact him at jihong.park@oulu.fi

Hyesung Kim is currently an engineer with the Samsung Electronics Co. Ltd., Seoul, Korea. His research interests include edge computing/caching, 5G communications system, distributed machine learning, and blockchain. He received the B.S. & Ph.D. in Electrical & Electronic Engineering from Yonsei University, Seoul, South Korea. He was awarded in the IEEE Seoul Section Student Paper Contest Bronze Prize in 2017, titled mean-field game-theoretic edge caching for ultra dense networks as a first author. Contact him at hye1207@gmail.com.

Mehdi Bennis is currently an Associate Professor with the Centre for Wireless Communications, University of Oulu, Oulu, Finland, where he is also an Academy of Finland Research Fellow and the Head of the Intelligent Connectivity and Networks/Systems Group (ICON). He has coauthored one book and published more than 200 research articles in international conferences, journals, and book chapters. His current research interests include radio resource management, heterogeneous networks, game theory, and machine learning in 5G networks and beyond. Dr. Bennis was a recipient of several prestigious awards, including the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best Paper Award for the Journal on Wireless Communications and Networks, the All-University of Oulu Award for research, and the 2019 IEEE ComSoc Radio Communications Committee Early Achievement Award. He is an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS. Contact him at mehdi.bennis@oulu.fi

Seong-Lyun Kim is a Professor of wireless networks at the School of Electrical & Electronic Engineering, Yonsei University, Seoul, Korea, heading the Robotic & Mobile Networks Laboratory (RAMO) and the Center for Flexible Radio (CFR+). He is co-directing H2020 EUK PriMO-5G project, and leading Smart Factory TF of 5G Forum, Korea. His research interest includes radio resource management, information theory in wireless networks, collective intelligence, and robotic networks. He received B.S. in Economics from Seoul National University, and M.S. & Ph.D. in Operations Research with Application to Wireless Networks from Korea Advanced Institute of Science & Technology (KAIST). He was an Assistant Professor of Radio Communication Systems at the Department of Signals, Sensors & Systems, Royal Institute of Technology (KTH), Stockholm, Sweden. He was a Visiting Professor at the Control Engineering Group, Helsinki University of Technology (now Aalto), Finland, the KTH Center for Wireless Systems, and the Graduate School of Informatics, Kyoto University, Japan. He served as a technical committee member or a chair for various conferences, and an editorial board member of IEEE Transactions on Vehicular Technology, IEEE Communications Letters, Elsevier Control Engineering Practice, Elsevier ICT Express, and Journal of Communications and Network. He served as the leading guest editor of IEEE Wireless Communications and IEEE Network for wireless communications in networked robotics, and IEEE Journal on Selected Areas in Communications. He also consulted various companies in the area of wireless systems both in Korea and abroad. He published numerous papers, including the coauthored book (with Prof. Jens Zander), Radio Resource Management for Wireless Networks. He is the corresponding author for this article and contact him at slkim@yonsei.ac.kr.