J Inf Process Syst, Vol.16, No.3, pp.733~741, June 2020
https://doi.org/10.3745/JIPS.04.0175

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS

Construction of a Remote Monitoring System in Smart Dust Environment

Joonsuu Park* and KeeHyun Park*

Abstract

A smart dust monitoring system is useful for obtaining information on rough terrain that is difficult for humans to access. One of ways to deploy sensors to gather information in smart dust environment is to use an aircraft in the Amazon rainforest to scatter an enormous amount of small and cheap sensors (or smart dust devices), or to use an unmanned spacecraft to throw the sensors on the moon's surface. However, scattering an enormous amount of smart dust devices creates the difficulty of managing such devices as they can be scattered into inaccessible areas, and also causes problems such as bottlenecks, device failure, and high/low density of devices. Of the various problems that may occur in the smart dust environment, this paper is focused on solving the bottleneck problem. To address this, we propose and construct a three-layered hierarchical smart dust monitoring system that includes relay dust devices (RDDs). An RDD is a smart dust device with relatively higher computing/communicating power than a normal smart dust device. RDDs play a crucial role in reducing traffic load for the system. To validate the proposed system, we use climate data obtained from authorized portals to compare the system with other systems (i.e., non-hierarchical system and simple hierarchical system). Through this comparison, we determined that the transmission processing time is reduced by 49%–50% compared to other systems, and the maximum number of connectable devices can be increased by 16–32 times without compromising the system's operations.

Keywords

Climate Data, Hierarchical System, Relay Dust Device, Rough Terrain, Smart Dust, Traffic Load

1. Introduction

One of the Internet of Things (IoT) application areas [1-6] is the application area in smart dust environment. A smart dust system [7-10] is an IoT system that collects information such as temperature, humidity, and pressure by spreading an enormous amount of tiny/cheap sensors (or smart dust devices) in a wide range using aircraft. It is one of the areas that are receiving a lot of attention because it is possible to collect a large amount of information in areas that are difficult or impossible to access. But scattering thousands of smart dust devices in inaccessible areas makes the devices difficult to be managed. In particular, the remote climate monitoring field, one of the major fields of smart dust applications, has some problems due to the huge amount of devices and different types of devices scattered in a wide area: (1) bottleneck problem, (2) node disconnection problem, and (3) high/low-density areas of devices problem. The bottleneck problem occurs on the IoT server when an enormous

Manuscript received April 17, 2020; first revision May 8, 2020; accepted May 18, 2020.

Corresponding Author: KeeHyun Park (khp@kmu.ac.kr)
* Dept. of Computer Engineering, Keimyung University, Daegu, Korea (parkjoonsuu@gmail.com, khp@kmu.ac.kr)

amount of messages containing sensed data sent from thousands of smart dust devices arrives at the server. When the bottleneck problem is aggravated, the system becomes unstable. This can lead to the system disconnecting connected nodes or refusing node connection requests, which leads to the node disconnection problem.

In a typical network, the nodes that make up the network are installed in a physical location and hence there are few problems with the network configuration. But smart dust devices cannot be deployed as planned, which means the possibility of high/low-density regions of smart dust devices. The presence of high/low-density regions of devices in certain areas of the network creates problems such as bottlenecks and device disconnection.

In this paper, we focus on solving the bottleneck and the device disconnection problems that may occur in a smart dust monitoring system. To address this, we propose and construct a three-layered hierarchical smart dust monitoring system that includes relay dust devices (RDDs). An RDD is a smart dust device with relatively higher computing/communicating power than a normal smart dust device. RDDs play a crucial role in reducing traffic load for the system. To validate the proposed system, we use climate monitoring data obtained from authorized portals to compare the system with other systems (i.e., non-hierarchical system and simple hierarchical system). Through this comparison, we determined that the transmission processing time was reduced by 49%–50% compared to other systems, and the maximum number of connectable devices could be increased by 16–32 times without compromising the system's operation.

The remainder of this paper is organized as follows: previous studies are discussed in Section 2. Section 3 explains the proposed system. In Section 4, we validate the proposed system with other systems. And finally conclusion and future research are discussed in Section 5.

2. Previous Works

Ghobakhlou et al. [11] proposes a framework for managing climate data using wired sensor networks (WSN). They divide the layers of the system into three (Mote, Server, and Application) and discuss how to provide a good framework to serve information to the application layer. They also create an environment for providing services by isolating the application layer from the mote layer. The study in [12] also designed their framework nicely. They focused on smoothly servicing by separating the layers, similar to [11]. They created an interface layer between the database layer and the sensing layer. The interface layer communicates with the application data through the web, and with the database layer through a database connection. The feature of both frameworks described above is that connectivity on a scale that is imaginable is assumed. The same is true for [13]. It considers more about the hardware part of the device than [11] and [12]. However, only one device type is considered and the number of devices is very limited in these studies.

Kahn et al. [9] said that the following three things should be considered when determining a network protocol for smart dust networks: (1) the link requires an uninterrupted path; (2) both passive and active device operation must be considered; (3) data transfer requires a compromise in terms of energy. In particular, they mention that bottlenecks can cause serious system problems for a smart dust network.

Buettner et al. [14] discussed the node disconnection problem that is suspected to create a bottleneck as the number of WSN devices increases. Given that their research was conducted using an RFID between a small number of devices in a relative small area, an environment dealing with a large number of devices in a large area could create a more serious bottleneck.

3. A Remote Climate Monitoring System in a Smart Dust Environment

Based on our earlier studies [15-17], we design and implement a three-layered remote monitoring system in a smart dust environment, as shown in Fig. 1. The system can sense and transmit a very large amount of climate data. As shown in Fig. 1, the system consists of three layers: dust devices (DDs) on the first layer, RDDs on the second layer, and finally the IoT server on the third layer. The IoT server consists of Processing Node (PN), Pool Control Node, and Load Balance Node.

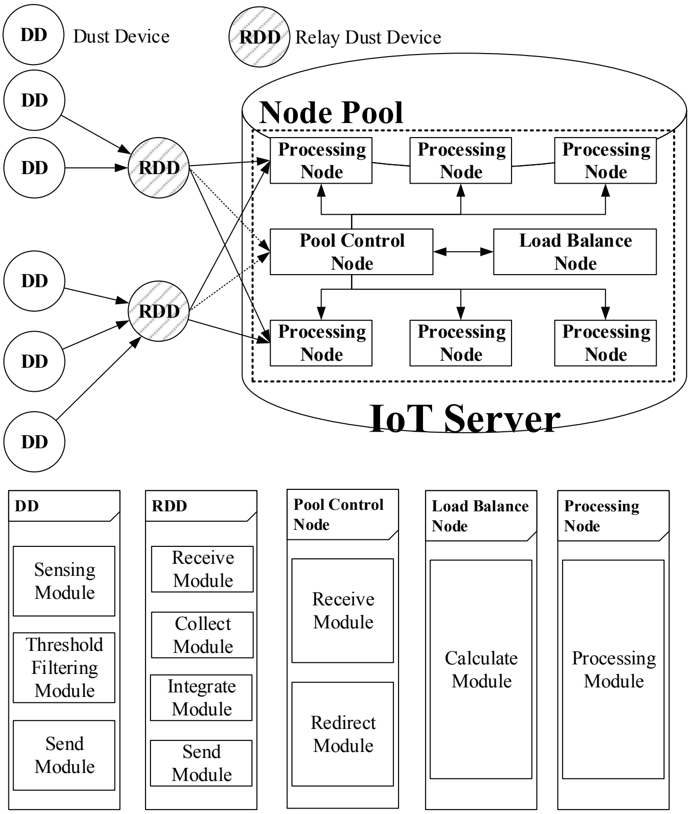

Fig. 1. Overview of the 3-layered hierarchical smart dust system.

The DD is a smart dust device scattered in wide area to sense climate data from adjacent areas. The DD has low computing/communicating power and hence can neither send sensed data in a long distance nor filter/compress data. Therefore, we added the additional (second) layer to the system for the RDDs. The RDD is a special DD with a relatively higher computing/communicating power than normal DD. It is a powerful DD that has a function of processing/transmitting data collected from normal DDs to the IoT server.

The IoT server is designed to process information received from RDD more efficiently through Node group called Node Pool. The Node Pool is composed of PNs that process actual data, Load Balance Node that can distribute data processing in consideration of processing status of each PN, and Pool Control Node that manages the entire Pool Node. After receiving the data from the RDDs, the Pool Control Node finds a suitable PN with the help of the Load Balance Node and requests data processing to the PN.

Some important software modules are depicted in Fig. 1. The Threshold Filtering module in a DD is a module that performs the threshold filtering work to reduce the number of sensed data to send. That is, this module only sends sensed data to the Send module when the difference between the current sensed data value and the previous sensed data value is higher than a specific threshold value. The Integrate module in RDD is a module that performs an integration filtering phase to further reduce network traffic. That is, this module checks to find the common parts of the data collected from DDs in order to merge the parts.

The Calculate module in the Load Balance module finds an appropriate PN for processing data received by the Receive module. The Redirect module in the Pool Control Node leads the connection to the selected PN. The Processing module in the PN performs data-specific processing.

Fig. 2 illustrates the flow of data processing in the entire system.

As shown in Fig. 2, the proposed system attempts to deal with a massive amount of monitoring data. Therefore, there are two phases to reduce network traffic in order to alleviate the bottleneck problem as follows: (1) the threshold filtering phase and (2) the integration filtering phase.

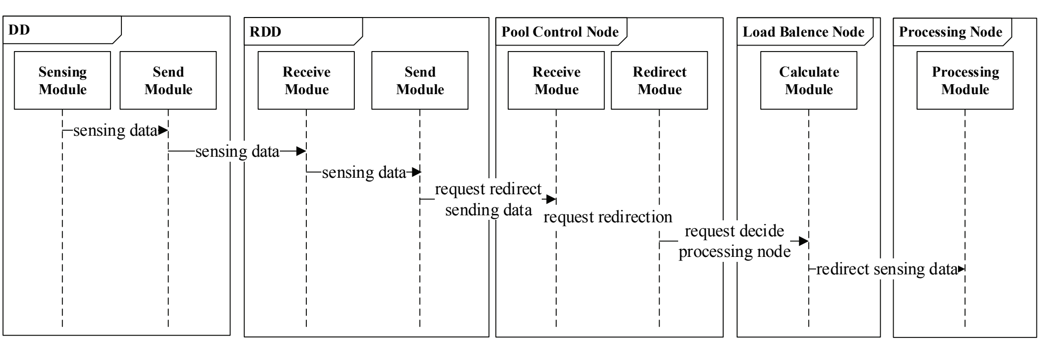

Fig. 2. Flow of data processing in the entire system.

In the threshold filtering phase, a DD does not transmit sensed data when the difference between the current sensing data value and the previous data value is below a certain threshold. A DD cannot send sensed data when the current data is very similar to the previous sensed data. The threshold values are set differently according to the type of the devices. This phase is very concise and requires very little computing power, so it is appropriate for use in a DD.

In the integration filtering phase, an RDD integrates several data into one message. For example, if five types of devices have transmitted twenty data per device, one hundred data are integrated into one message (not hundred messages) to send to the IoT server, which means the number of messages to send to the IoT server is greatly reduced. Fig. 3 is the flowchart of filtering processing work explained above.

If the threshold is set to a large value, a very large compression ratio can be expected, but the values of the sensing data can be very discrete. In other hands, if the threshold is set to a very small value, data similar to the actual measured value can be obtained, but the compression rate is lowered. In other words, determining the appropriate threshold is one of the most important points that can increase performance. One way to determine the optimal threshold is to use the method of determining the Nyquist Rate [18]. The integration filtering phase is very different depending on the type of device but can be performed by integrating protocol-specific headers.

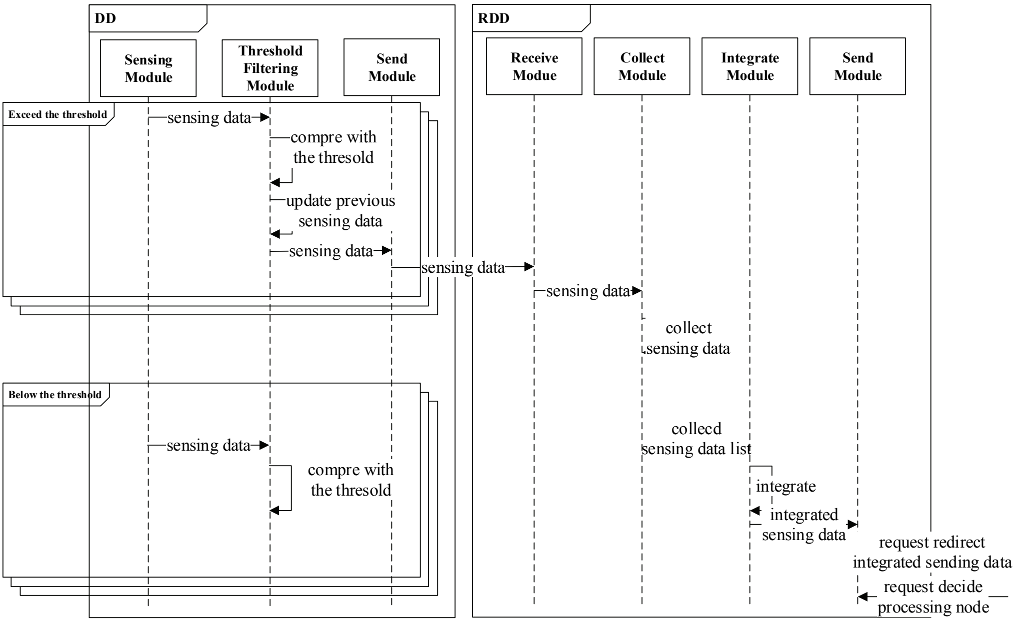

Fig. 3. Flowchart of the filtering processing work.

4. Performance Analysis

The smart dust monitoring system discussed above is constructed for this study. We use official climate data obtained from the Korea Meteorological Administration (KMA, https://data.kma.go.kr) and the National Information Society Agency (NIA, https://www.data.go.kr) to validate the system. Table 1 shows some of the actual data collected from the KMA and NIA. DDs and RDDs run as processes on virtual machine concurrently.

Table 1. Sensing data sample

Time	Temperature	Humidity	Sunshine	Ground temperature
2019-05-01 14:00	22.7	34	1	28.2
2019-05-01 15:00	23.4	30	1	29.1
2019-05-01 16:00	23.4	29	0.9	25.9
2019-05-01 17:00	23.2	29	0.9	23.6
2019-05-01 18:00	22.2	30	1	20.1
2019-05-01 19:00	21.5	26	0.7	17.2
2019-05-01 20:00	20.3	26	0	15.2

Table 2 shows the experimental results with various numbers of RDDs. Ten percent of DDs are designated as RDDs. Thresholds are set to 1.0, 0.5. 0.1, 1.0 for temperature, humidity, sunshine, and ground temperature data, respectively.

In Table 2, the "non-hierarchical system" is a general system that is not hierarchical. The "simple hierarchical system" is a hierarchical system with DDs and the IoT server (i.e., systems without any RDDs). Finally, the "proposed system" is the system proposed in this paper, as shown in Fig. 1.

In Table 2, there are "-" lines between 3,200 (6,400) DDs through 51,200 DDs for the non-hierarchical system (simple hierarchical system). These are the cases when the messages containing sensed data that are sent to the Pool Control Node cause bottlenecks or the system hangs because the number of devices exceeds the system can manage.

Table 2. Comparison of processing times by the number of connected RDDs

Number of RDDs	Non-hierarchical system	Simple hierarchical system	Proposed system
100	9,502	9,588	4,290
200	19,000	19,220	8,486
400	38,000	37,596	18,821
800	76,000	76,123	43,727
1,600	152,003	152,673	73,832
3,200	-	300,096	160,690
6,400	-	-	280,962
12,800	-	-	630,554
25,600	-	-	1,008,551
51,200	-	-	2,762,345

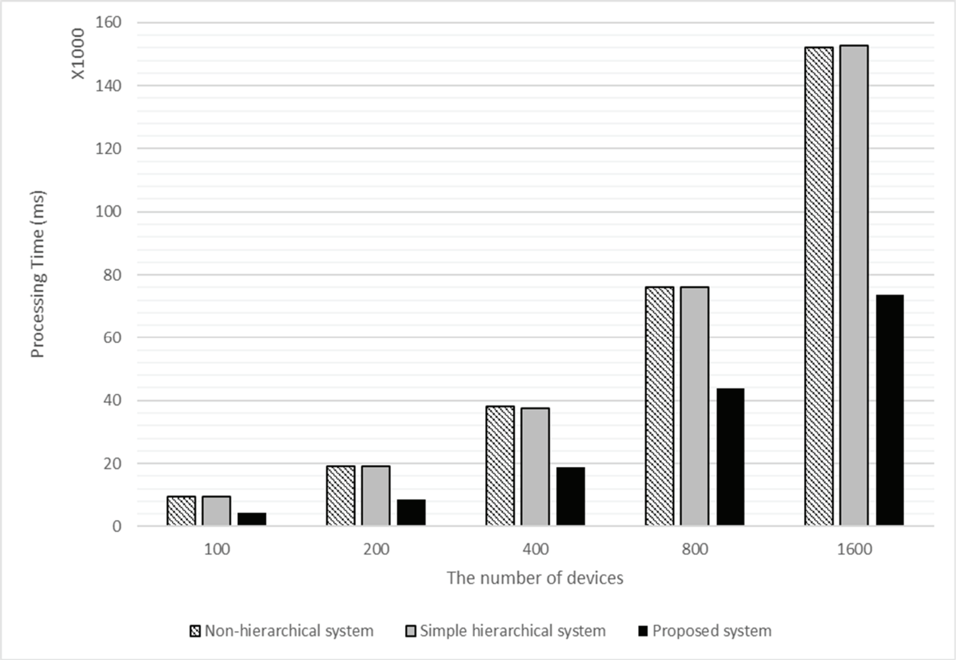

Fig. 4. Comparison of processing times by the number of connected RDDs.

Fig. 4 is a performance comparison graph by the various number of devices, from 100 to 1,600 RDDs. As can be seen in Fig. 4, the proposed system spends as less as about 49% (50%) of the processing time compared to the non-hierarchical system (the simple hierarchical system). And, as shown in Table 2, the non-hierarchical system (the simple hierarchical system) stops its operations when more than 3,200 (6,400) devices are connected, while the proposed system is still stable even when 51,200 devices are connected. This means that the proposed system can connect 16 (8) times as many devices as the non-hierarchical system (the simple hierarchical system) does. Therefore, the proposed system can deal with the node disconnection problem more efficiently than the other system does.

5. Conclusions

We pointed out the three problems that can occur in a smart dust environment where a large number of devices are connected to collect a huge amount of sensed data: (1) the bottleneck problem, (2) the node disconnection problem, and (3) the high/low-density area of devices problem. In this paper, we propose and construct a three-layered hierarchical smart dust monitoring system in order to solve the bottleneck and the device disconnection problems that may occur in a smart dust monitoring system. To validate the proposed system, we use the climate monitoring data obtained from authorized portals to compare the system with other systems (i.e., non-hierarchical system and simple hierarchical system).

The comparison experiments show that transmission processing times are reduced by 49%–50% compared to other systems, which means that the bottleneck problem is alleviated by the proposed system. The experiments also show that the maximum number of connectable devices could be increased by 16–32 times without compromising the system's operation. This means that the proposed system is more stable with a large number of devices connected, and the node disconnection problem is solved to some degree.

We are currently engaged in research that aims to solve the remaining problem (i.e., the high/low-density area of devices problem). In other words, we are trying to locate smart dust devices using techniques such as triangulation or true range multilateration [19-21] to achieve a further reduction in traffic load.

Acknowledgement

This research was funded by the Basic Science Research Programs through the National Research Foundation of Korea (NRF), grant number funded by the Ministry of Education, Science and Technology (No. NRF-2018R1D1A1B07043982).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, 2010.

[2] G. R. Gonzalez, M. M. Organero, and C. D. Kloos, "Early infrastructure of an internet of things in spaces for learning," in *Proceedings of 2008 8th IEEE International Conference on Advanced Learning Technologies*, Santander, Spain, 2008, pp. 381-383.

[3] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 291-319, 2018.

[4] B. Sterling, *Shaping things–Mediawork pamphlets*. Cambridge, MA: The MIT Press, 2005.

[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, 2013.

[6] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497-1516, 2012.

[7] B. Warneke, M. Last, B. Liebowitz, and K. S. Pister, "Smart dust: communicating with a cubic-millimeter computer," *Computer*, vol. 34, no. 1, pp. 44-51, 2001.

[8] J. M. Kahn, R. H. Katz, and K. S. Pister, "Emerging challenges: mobile networking for "smart dust"," *Journal of Communications and Networks*, vol. 2, no. 3, pp. 188-196, 2000.

[9] J. M. Kahn, R. H. Katz, and K. S. Pister, "Next century challenges: mobile networking for "Smart Dust"," in *Proceedings of the 5th Annual ACM/IEEE international Conference on Mobile Computing and Networking*, Seattle, WA, 1999, pp. 271-278.

[10] Smart Dust [Online]. Available: http://www.itfind.or.kr/WZIN/jugidong/1140/114004.htm.

[11] A. Ghobakhlou, S. Shanmuganthan, and P. Sallis, "Wireless sensor networks for climate data management systems," in *Proceedings of the 18th World IMACS/MODSIM Congress*, Cairns, Australia, 2009, pp. 13-17.

[12] E. Kanagaraj, L. M. Kamarudin, A. Zakaria, R. Gunasagaran, and A. Y. M. Shakaff, "Cloud-based remote environmental monitoring system with distributed WSN weather stations," in *Proceedings of 2015 IEEE SENSORS*, Busan, South Korea, 2015, pp. 1-4.

[13] P. Susmitha and G. S. Bala, "Design and implementation of weather monitoring and controlling system," *International Journal of Computer Applications*, vol. 97, no. 3, pp. 19-22, 2014.

[14] M. Buettner, B. Greenstein, A. Sample, J. R. Smith, and D. Wetherall, "Revisiting smart dust with RFID sensor networks," in *Proceedings of the 7th ACM Workshop on Hot Topics in Networks (HotNets-VII)*, Calgary, Canada, 2008, pp. 37-42.

[15] J. Park and K. Park, "A high-performance implementation of an IoT system using DPDK," *Applied Sciences*, vol. 8, no. 4, article no. 550, 2018.

[16] J. Park and K. Park, "A study on system architecture design for plane dynamic scaling in smart dust environments," in *Proceedings of 2019 the KIPS Spring Conference*, Seoul, South Korea, 2019.

[17] K. Park, I. Kim, and J. Park, "A high speed data transmission method for DPDK-based IoT systems," in *Proceedings of the International Conference on Future Information & Communication Engineering*, 2018, pp. 325-327.

[18] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no 3, pp. 379-423, 1948.

[19] J. Caffery and G. L. Stuber, "Subscriber location in CDMA cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 2, pp. 406-416, 1998.

[20] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, MA, 2000, pp. 32-43.

[21] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, San Francisco, CA, 2003, pp. 1734-1743.

Joonsuu Park https://orcid.org/0000-0003-2572-3208

He received B.S. and M.S. degrees in Computer Engineering from Keimyung University, Daegu, the Republic of Korea, in 2014 and 2016. From 2016 to early 2017, he worked as a back-end developer at EC Company in Daegu, Korea. He worked as a freelance server developer in from 2017 to early 2018. Since late 2018, he began studying PhD in Computer Engineering from Keimyung University, Daegu, Korea.

KeeHyun Park https://orcid.org/0000-0002-3208-4216

He received his B.Sc. and M.Sc. degrees in Computer Science from Kyungpook National University, Korea, and from KAIST, Korea, in 1979 and 1981, respectively, and his Ph.D. degree in Computer Science from Vanderbilt University, USA, 1990. He has been a professor of Computer Science and Engineering Department at Keimyung University, Korea since March 1981. His research interests include mobile/network communication system, device management for u-healthcare systems, embedded system and parallel processing system.