International Journal of Control and Automation
Vol. 11, No. 7 (2018), pp.129-138
http//dx.doi.org/10.14257/ijca.2018.11.7.11

Efficient Multipath Routing Method for Multilayered oneM2M-based IoT System

Joonsuu Park[1] and KeeHyun Park[2]

Computer Engineering Dept., Keimyung University, Daegu, 42601, Rep. of Korea
[1]parkjoonsuu@stu.kmu.ac.kr, [2]khp@kmu.ac.kr

Abstract

In this study, we propose the Table sHaring multipath Exploration (THE) routing algorithm, which locates multipath between a gateway (MN) and the oneM2M server (IN) in a oneM2M-based IoT system. More than one path between two nodes can be used to distribute traffic loads efficiently and thus improve system performance. The THE routing algorithm locates various multi-paths from an MN to the IN using three processing steps: connection, routing table sharing, and routing table integration.

The THE routing algorithm proposed in this study is evaluated in experiments for three types of network topology: star, linear, and mesh. For performance comparison, times that construct the final routing table for every node are measured. Our experiments show that in most cases the THE routing algorithm performs better than the Percolation-based routing algorithm. In particular, the average number of hops required to travel from an MN to the IN is smaller.

***Keywords:** Routing, IoT system, oneM2M, Multipath, Table sharing multipath exploration routing algorithm*

1. Introduction

The concept of IoT (Internet of Things) [1-6], which enables smart objects to communicate with each other, has become increasingly popular. The concept can be applied to many areas, including metering, traffic control, smart homes, building management, and healthcare [7-14]. As the number of nodes in an IoT system increases, an efficient routing algorithm [15-19] that locates paths from the objects to the IoT server becomes crucial.

In this study, we propose the Table sHaring multipath Exploration (THE) routing algorithm, which locates multipath between a gateway (Middle Node, MN) and the oneM2M server (Infrastructure Node, IN) in an oneM2M-based IoT system. More than one path between two nodes can be used to distribute traffic loads efficiently, thereby improving system performance. The THE routing algorithm finds various multipaths from an MN to the IN by means of three processing steps: connection, routing table sharing and routing table integration. Locating multipath between two nodes is crucial for performance because more than one path between two nodes is used to distribute traffic loads efficiently, thus decreasing message transmission time between nodes in an IoT system.

The THE routing algorithm proposed in this study is evaluated using experiments for three types of network topology: star, linear, and mesh. For performance comparison, times that make up the final routing table for every node are measured. Our experiments reveal that in most cases the THE routing algorithm performs better than the Percolation-based routing algorithm [19]. In particular, the average number of hops required to travel from an MN to the IN is smaller.

Received (March 10, 2018), Review Result (May 28, 2018), Accepted (June 7, 2018)

The remainder of this paper is organized as follows. Section 2 describes related studies, and Section 3 explains the proposed THE routing algorithm. Section 4 discusses the results of our experiments using the proposed algorithm. Finally, Section 6 provides a conclusion and discusses possible directions for future research.

2. Related Studies

2.1. oneM2M-based IoT system

The oneM2M communication protocol is an international standard for IoT systems [7-9]. Figure 1 shows the structure of the oneM2M-based IoT system [7]. In an IoT system, a sensor or device (*i.e.*, an installed program on either) represents an Application Dedicated Node-Application Entity (ADN-AE) that gathers surrounding data and transmits it to the system's Middle Node-Common Service Entity (MN-CSE). An MN-CSE controls or monitors ADN-AEs that belong to the MN-CSE. Moreover, it performs processing that is necessary to achieve efficient communication between ADN-AEs and the Infrastructure Node-Common Service Entity (IN-CSE). A manager or a user can access data stored in the IN-CSE through an ADN-AE.

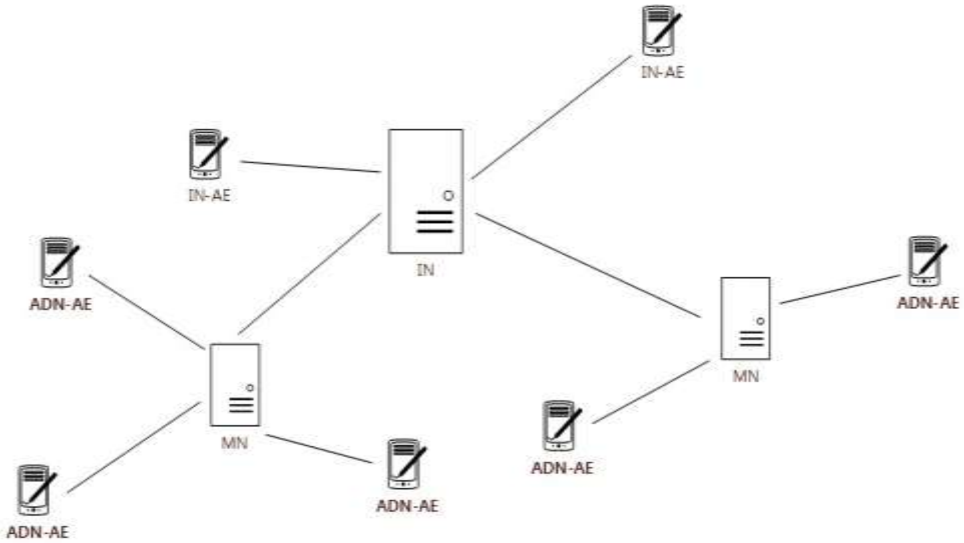

Figure 1. oneM2M-based IoT System Structure

2.2. Routing Algorithms for oneM2M-based IoT Networks

Examining the routing algorithms for wireless sensor networks is worthwhile because wireless sensor networks and IoT systems have many common characteristics. The study in [19] proposed a percolation-based routing algorithm (PR) for oneM2M-based IoT networks to search routes between any two nodes in the networks. A probing packet is sent with its hop count of 1 from the source node. When a node that is not the destination node in the network receives the probing packet, the node increases the packet's hop count by one and sends the packet to its neighboring nodes (except, of course, to the node from which the packet is received). The probing packet records its route whenever the packet visits a node. The packet is destroyed when the hop count exceeds the maximum hop count limit set by the system. When the packet reaches the destination node, the routing record is used to construct a routing table between the two nodes in the network. Several paths are selected via repeating the close-neighbors' chosen procedure until the probe data packets reach the destination node. In such a way, multiple paths between the source and the destination nodes may be built.

3. THE Routing Algorithm

The THE routing algorithm proposed in this study locates various paths between any two nodes in a oneM2M-based IoT system. Finding multipaths between two nodes is crucial for performance because more than one path between two nodes is used to distribute traffic loads efficiently, thereby decreasing message transmission times between the nodes in an IoT system.

3.1. oneM2M-based IoT System Model

Figure 2 shows the oneM2M-based IoT system model used in this study. Surrounding the oneM2M server (IN) are many gateways (MNs) that are directly or indirectly connected to the server. Shadowed MNs are connected directly to the IN and, in this study, are called Direct Connected Nodes (DCNs) of the IN. Other MNs are connected indirectly to the IN via other MNs and are called Indirect Connected Nodes (ICNs). The oneM2M-based IoT system considered in this study is multilayered because gateways exist that are indirectly connected to the server. Figure 2 depicts a three-layered system. An ICN can be either an MN or ADN-AE. Regarding routing problems, it does not need to distinguish an indirectly connected MN from an ADN-AE. This is why no ADN-AEs are shown in the figure.

3.2. THE Routing Algorithm Steps

The THE routing algorithm locates various multipath from an MN to the IN using three processing steps. The first step is a connection step in which DCNs connect to the IN, and then ICNs connects to DCNs and/or other ICNs. The connection information is stored in Local Routing Tables (LRTs) of both sides. The second step involves routing table sharing in which LRTs are shared between the higher layered MNs and the IN by transmitting the LRTs through the higher layered MNs to the IN. When an MN requests a connection to the higher layered MNs or the IN, the MN's LRT is also transmitted to them in order to store the MN's LRT information in their LRTs. Finally, the third step involves routing table integration in which all LRT information received by the IN is integrated into the Global Routing Table (GRT) of the IN. Based on the GRT, the LRT of every node in the system is finalized.

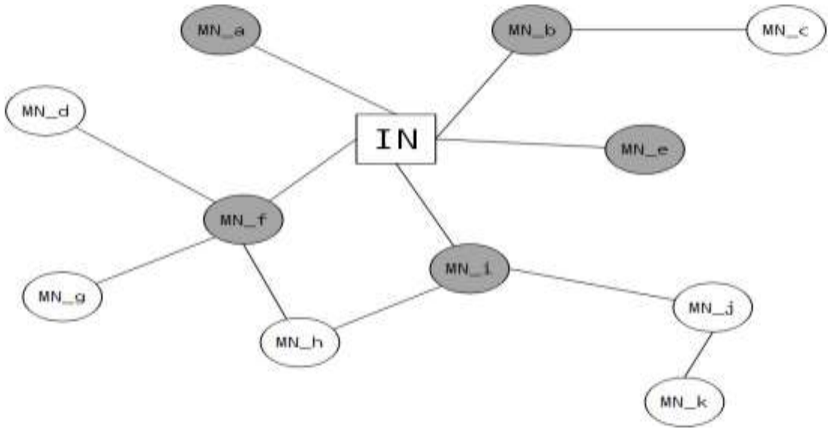

Figure 2. oneM2M-based IoT System Model

3.2.1. Connection Step

Figure 3 shows the connection step of the THE routing algorithm. DCNs are first connected to the IN; then ICNs are connected to DCNs. In addition, LRT information related to the connection is transmitted to the node that receives the connection request.

```
For every node K in the system:
    If an IN exists in K's direct connection range,
          K sends a connection request to the IN
    otherwise
          K sends a connection request to every node in its direct connection range
    If the connection is successful,
          K sends its LRT information to the other end node of the connection
```

Figure 3. THE Routing Algorithm: Connection Step

The left portion of Figure 4 shows direct connection ranges (dotted circles) for all nodes in the system except the IN. For example, MN1, MN4, MN6, and IN exist in the direct connection range of MN3. Because the IN appears in the MN3's direct connection range, MN3 sends the IN a connection request. If the connection is successful, MN3 becomes a DCN of the IN. In the case of MN4, the IN cannot be connected directly. Therefore, MN4 sends a connection request to MN2, MN3, MN5, and MN4 becomes an ICN of the IN.

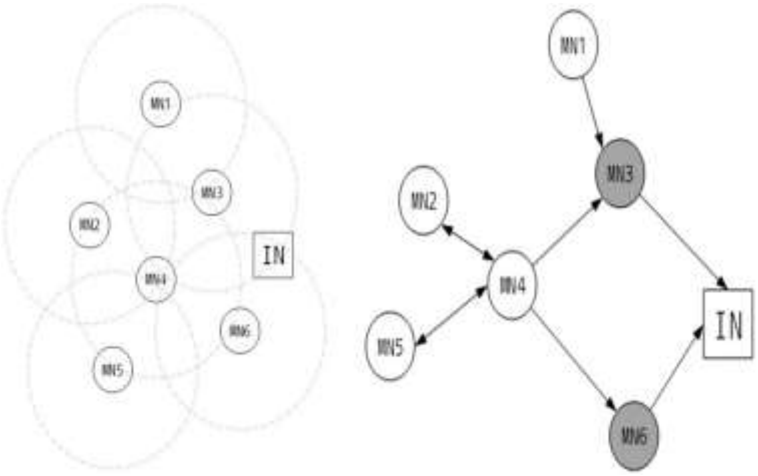

Figure 4. Direct Connection Range of the Nodes (left) and Initial Connection (right)

The right portion of Figure 4 depicts the system after the initial connection process. Arrows between nodes represent the direction of the connection request. For example, an arrow from MN3 to the IN indicates that a connection between them is established by sending a connection request to the IN. A bidirectional arrow between MN4 and MN5 indicates that both nodes send a connection request to each other. Shadowed nodes are DCNs of the IN.

After connections are established, LRT is set for every node in the system. For example, the LRTs of the IN and MN3 are depicted in the upper and lower portions of Figure 5, respectively.

Connection permitting node	Connection requesting node
IN	MN3
IN	MN6

Connection permitting node	Connection requesting node
MN3	MN1
MN3	MN4

Figure 5. LRTs of the IN (upper) and MN3 (lower) in the Connection Step

3.2.2. Routing Table Sharing Step

Figure 6 shows the routing table sharing step of THE routing algorithm. In this step, LRT information is propagated to the IN via DCNs. Whenever LRT information is received by a node, it is merged into the node's LRT and the updated LRT information is again sent to the higher layered MNs or the IN. The finalized LRT of the IN is obtained when no further updates occur.

```
For every node K in the system,
    Merge received LRT information with K's LRT
    Send K's updated LRT information to the higher layered
       MNs or the IN
until no additional changes occur in LRTs
```

Figure 6. THE Routing Algorithm: Routing Table Sharing Step

For example, after one instance in which LRT information is received, the LRTs of the IN (upper) and MN3 (lower) are updated as shown in Figure 7.

Connection permitting node	Connection requesting node
IN	MN3
IN	MN6
MN3	MN4
MN6	MN4

Connection permitting node	Connection requesting node
MN3	MN1
MN3	MN4
MN4	MN2
MN4	MN5

Figure 7. LRTs of the IN (upper) and MN3 (lower) After One Instance in which LRT Information is Received during the Routing Table Sharing Step

After three instances in which LRT information is received, the LRT of the IN is finalized as shown in Figure 8.

Connection permitting node	Connection requesting node
IN	MN3
IN	MN6
MN3	MN1
MN3	MN4
MN4	MN2
MN4	MN5
MN5	MN4
MN2	MN4
MN6	MN4
MN4	MN2
MN4	MN5
MN5	MN4
MN2	MN4

Figure 8. Finalized LRT of the IN in the Routing Table Sharing Step

3.2.3. Routing Table Integration Step

The final step of the THE routing algorithm is the integration of IN's LRT in order to convert the IN's to GRT. The routing table integration step involves eliminating redundant routing and circulation information in the LRT. Figure 9 shows the routing table integration step of THE routing algorithm.

```
For every entry (i, j) in IN’s LRT,
    Add new entries (i, j) and (j, i) to the initial GRT of  the IN
    Remove the redundant entries from the GRT
    Remove the circulation routing information from the GRT
Finalize the LRT for every node from the GRT and send the
    LRT to the related node
```

Figure 9. THE Routing Algorithm: Routing Table Integration Step

After three receptions of LRT information, the LRT of the IN is finalized as shown in Figure 10.

Connection permitting node	Connection requesting node	Connection permitting node	Connection requesting node
IN	MN3	MN3	IN
IN	MN6	MN6	IN
MN3	MN1	MN1	MN3
MN3	MN4	MN4	MN3
MN4	MN2	MN2	MN4
MN4	MN5	MN5	MN4
MN5	MN4	MN4	MN5
MN2	MN4	MN4	MN2
MN6	MN4	MN4	MN6
MN4	MN2	MN2	MN4
MN4	MN5	MN5	MN4
MN5	MN4	MN4	MN5
MN2	MN4	MN4	MN2

Figure 10. Initial GRT of the IN in the Routing Table Integration Step

After eliminating redundant and circulation information, the finalized GRT of the IN is obtained, as shown in Figure 11. The first entry of the GRT indicates that a path MN1 -> MN3 -> IN exists from MN1 to the IN. Another path MN1 -> MN3 -> MN4 -> MN6 -> IN exists from MN1 to the IN, as shown in the eighth entry. Therefore, the THE routing algorithm finds multipaths from MN1 to the IN.

1	2	3	4	5
IN	MN3	MN1		
IN	MN3	MN4	MN2	
IN	MN3	MN4		
IN	MN3	MN4	MN5	
IN	MN3	MN4	MN6	
IN	MN3			
IN	MN6	MN4	MN2	
IN	MN6	MN4	MN3	MN1
IN	MN6	MN4	MN3	
IN	MN6	MN4	MN5	
IN	MN6	MN4		
IN	MN6			

Figure 11. Finalized GRT of the IN

The finalized LRT of every node in the system is then generated from the GRT and transmitted to the related node. For example, as shown in Figure 12, the finalized LRT of MN1 is produced based on the GRT, and the LRT is transmitted to MN1. Figure 13 shows that two paths exist from MN1 to the IN.

1	2	3	4	5
IN	MN3	MN1		
IN	MN6	MN4	MN3	MN1

Figure 12. Finalized LRT of the MN1

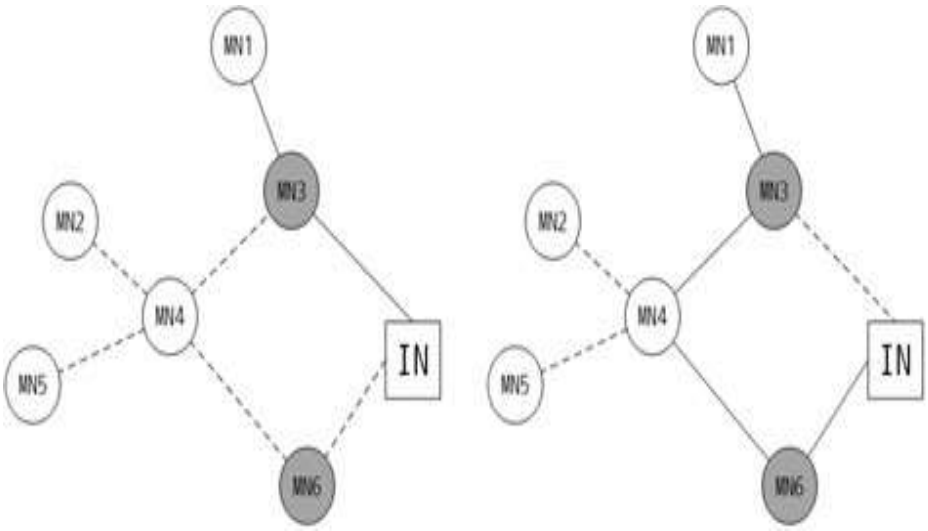

Figure 13. Multipath from MN1 to the IN

4. Experiments

In addition to the PR algorithm described in Section 2, the THE routing algorithm proposed in this study was evaluated in experiments for three types of network topology. For performance comparison, times to construct the final routing table for every node were measured. Figure 14 (a)–(c) show the three types of network topology used in the experiments, where each topology consists of ten nodes.

Table 1 shows the average times used to construct the final routing table for every node measured in the experiments. For a star network depicted in Figure 14 (a), the THE routing algorithm outperforms the PR algorithm considerably by as much as 137%. The THE routing algorithm proved to be the best performer with this network topology because every MN can reach the IN in a single hop, and the routing information needs to be delivered only once. By contrast, with the PR algorithm, the IN must send a probing packet to every MN and then determine retransmission of each probing packet. For a linear network depicted in Figure 14 (b), the PR algorithm outperforms the THE routing algorithm by 61.3%. The THE algorithm proved to be the worst performer with this network topology because the amount of routing information that accumulates increases considerably every time information is delivered between nodes. By contrast, the PR algorithm transmits a relatively small-sized packet that contains information only about the routing nodes the packet visits. However, the linear network topology is rarely employed in the real IoT world because of poor performance.

The network topology employed most often in the real IoT world is the mesh network topology, which is shown in Figure 14(c). For this topology, the THE routing algorithm outperforms the PR algorithm by 30.0%. The experiments revealed that the THE routing algorithm performs better, as the average number of hops required to travel from a MN to the IN is smaller. The average hop counts in Figure 14(a)–(c) are 1, 5, and 1.6, respectively.

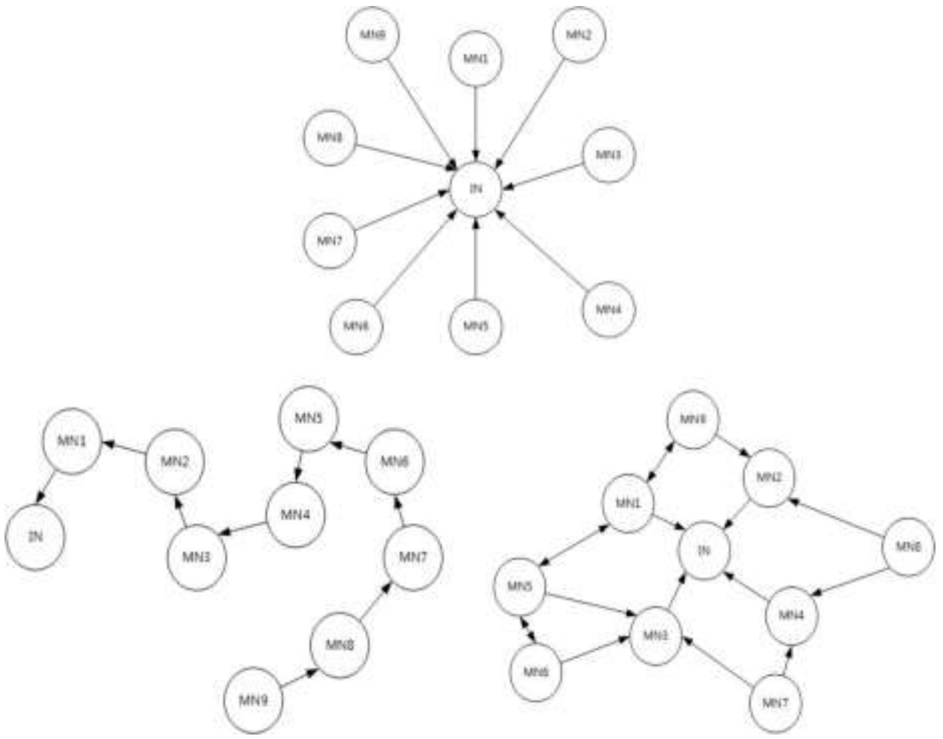

Figure 14. (a) Star, (b) Linear, and (c) Mesh Network Topologies

Table 1. Average Times to Construct the Final Routing Tables for Every Node

	Star Network	Linear network	Mesh network
THE routing algorithm	207.71ms	2121.53ms	2364.54ms
PR routing algorithm	491.91ms	1315.61ms	3071.94ms

5. Conclusion

In this study, we proposed the THE routing algorithm, which locates multipath between MNs and the IN in a oneM2M-based IoT system. More than one path between two nodes can be used to distribute traffic loads efficiently, thereby decreasing message transmission times between the nodes in an IoT system. THE routing algorithm finds various multipaths from an MN to the IN using three processing steps: connection, routing table sharing, and routing table integration.

The proposed THE routing algorithm was evaluated in experiments for three types of network topology: star, linear, and mesh. For performance comparison, times that construct the final routing table for every node were measured. The experiments revealed that in most cases the THE routing algorithm performs better than the PR algorithm, in particular the average number of hops required to travel from an MN to the IN is smaller.

Acknowledgements

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

References

[1] ITU. The Internet of Things. ITU Internet reports. 2005. URL:https://www.itu.int/net/wsis/tunis/newsroom/stats/The-Internet-of-Things-2005.pdf(03.24.2017).

[2] European Commission, Internet of Things-An action plan for Europe. Commission of the European Communities. 2009. URL:http://www.eesc.europa.eu/?i=portal.en.ten-opinions.18007(03.24.2017).

[3] D. Evans, "How the Internet of Everything Will Change the World. CISCO Blog", 2012 URL:https://blogs.cisco.com/digital/how-the-internet-of-everything-will-change-the-worldfor-the-better-infographic (03.24.2017).

[4] S. Li, L. Xu and S. Zhao, "The Internet of things: A Survey. // Information Systems Frontiers", DOI: 10.1007/s10796-014-9492-7, vol. 17, no. 2, **(2015)**, pp. 243-259.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", // IEEE Communications and Tutorials, DOI: 10.1109/COMST.2015.2444095, vol. 17, no. 4, **(2015)**, pp. 2347-2376.

[6] L. A. Grieco, M. Alaya, T. Montei and K. Drira, "Architecting Information Centric ETSI-M2M systems", // Proceedings of Pervasive Computing and Communications Workshops / Budapest, DOI: 10.1109/PerComW.2014.6815203, **(2014)**, pp. 211-214.

[7] oneM2M, Functional Architecture // OneM2M TS-0001-V1.6.1. 2015. URL:http://www.onem2m.org, (03.24.2017).

[8] oneM2M, Requirements // OneM2M TS-0002-V1.0.1. 2015. URL:http://www.onem2m.org, (04.01.2017).

[9] oneM2M, Published Specifications // oneM2M. 2015. URL:http://www.onem2m.org/technical/published-documents, (04.01.2017).

[10] K. H. Park and J. G. Pak, "An Integrated gateway for various PHDs in u-Healthcare Environments", // BioMed Research International, DOI: 10.1155/2012/954603, vol. 2012, article ID. 9546033, **(2012)**, pp. 1-7.

[11] K. H. Park and S. H. Lim, "Message processing at integrated PHD gateways for servicing various PHDs", // International Journal of Multimedia and Ubiquitous Engineering, DOI: 10.14257/ijmue.2014.9.3.35, vol. 9, no. 3, **(2014)**, pp. 367-374.

[12] K. H. Park and S. H. Lim, "A secure bio-medical data management system for a very large number of various PHDs", // BioMed Research International, DOI:10.1155/2015/941053, vol. 2015, article ID. 941053, **(2015)**, pp. 1-17.

[13] J. G. Pak and K. H. Park, "Advanced Pulse Oximetry System for Remote Monitoring and Management", // BioMed Research International, DOI:10.1155/2012/930582, **(2012)**, article ID. 930582, **(2012)**, pp. 1-8.

[14] W. Zhang, X. Wang, B. Lu and T. H. Kim, "Secure Encapsulation and Publication of Biological Services in the Cloud Computing Environment", // BioMed Research International, DOI: 10.1155/2013/170580, vol. 2013, article ID, 170580, **(2013)**, pp. 1-8.

[15] W. Stallings, "Data and Computer Communications", 10th ed. Prentice Hall Publishing Company, New Jersey, **(2014)**.

[16] J. Abazeed, N. Faisal, S. Zubair and A. Ali, "Routing protocols for wireless multimedia sensor network: A survey", // Journal of Sensors, DOI: 10.1155/2013/469824, vol. 2013, article ID. 469824, **(2013)**, pp. 1-11.

[17] Z. Li, R. Li, T. Pei, Z. Xiao and X. Chen, "Survey of geographical routing in multimedia wireless sensor networks", // Information Technology Journal, DOI: 10.3923/itj.2011.11.15, vol. 10, no. 1, **(2011)**, pp. 11-15.

[18] M. Saleem, A. Gianni, D. Caro and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions", // Information Sciences, DOI: 10.1016/j.ins.2010.07.005, vol. 181, no. 20, **(2011)**, pp. 4597-4624.

[19] J. Lu, J. An, X. Li, J. Yang and L. Yang, "A Percolation based M2M Networking Architecture for Data Transmission and Routing", // KSII Transactions on Internet and Information Systems, DOI: 10.3837/tiis.2012.02.012, vol. 6, no. 2, **(2012)**, pp. 649-663.

Authors

Joonsuu Park, received his M.Sc degree in Computer Science from Keimyung University, Daegu, Korea in 2016. His research interests include intelligent communication networks, IoT System, IoT communication security, and so on.

Keehyun Park, 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.