

An Efficient Multi-class Message Scheduling Scheme for Healthcare IoT Systems

KeeHyun Park¹, Insung Kim² and Joonsuu Park³

Computer Engineering Dept., Keimyung University,
Daegu, 42601, Rep. of Korea

¹khp@kmu.ac.kr, ²exit135790@gmail.com, ³parkjoonsuu@stu.kmu.ac.kr

Abstract

The Internet of Things (IoT) concept, which involves communication within a network of objects, has become increasingly popular, and development in this area of technology is quite active. IoT environments involve the generation of large amounts of data, and require various levels of Quality of Service. For this reason, message scheduling schemes to deliver the data in IoT environments are considered to be essential.

This paper proposes an efficient multi-class message scheduling algorithm for healthcare IoT environments. In the proposed message algorithm, messages are grouped into three message classes (UNC, RT and DT classes) based on the characteristics of the messages. The proposed message algorithm is resource-efficient because it uses a simpler priority calculation than the Multi-class Q-Learning message scheduling algorithm, which we proposed for IoT systems previously.

Tests were conducted on a part of the oneM2M-based IoT system we have built. These showed that in most cases, the proposed scheduling algorithm performs better than the Multi-class Q-Learning scheduling algorithm.

Keywords: Message scheduling, IoT system, Quality of Service, Multi-class Q-Learning scheduling algorithm

1. Introduction

The concept of Internet of Things (IoT) was introduced in 2005 [1]. Involving communication among a network of objects, IoT systems have become popular [2-4]. Such systems generate large amounts of diverse data, and require various levels of QoS (Quality of Service) [2-5]. Mostly, the data transferred in IoT systems is delivered through generated messages. Therefore, message scheduling schemes in IoT environments should be designed to take this requirement into account [6]. There have been many studies that have proposed good network QoS models in conventional network systems [7-9]. But as unlike conventional network systems, most of the nodes in IoT systems have limited computing resources, it is difficult to achieve a certain level of QoS for given data [10-15].

In this paper, an efficient multi-class message scheduling algorithm for healthcare IoT environments is proposed. The scheduling algorithm takes into account the limited resources and various message characteristics in IoT environments. In the proposed message algorithm, messages are grouped into three message classes based on their characteristics. QoS requirements for the messages are realized by putting different priorities on different message classes. The proposed message algorithm is

Received (January 10, 2018), Review Result (March 19, 2018), Accepted (March 28, 2018)

resource-efficient because the algorithm schedules more messages for a given time than a previous message scheduling algorithm we proposed for IoT systems [16, 18].

The remainder of the paper is composed as follows. In Section 2, some related studies are discussed. An efficient multi-class message scheduling algorithm for healthcare IoT systems is proposed and some experiments on the algorithm are explained in Section 3 and Section 4, respectively. The experiments show that the efficient multi-class message scheduling algorithm proposed in this paper takes a shorter time to complete scheduling work than the Multi-class Q-Learning (MQL) message scheduling algorithm [17-18], which was proposed previously by our research team. Finally, in Section 5, we conclude the paper.

2. Related Studies

2.1. Multi-class Q-Learning Message Scheduling Algorithm

It took our research team several years to construct an IoT system [18]. One of the main goals of the IoT system construction project was to come up with an efficient message scheduling algorithm for the IoT system. In this section, the MQL message scheduling algorithm [17-18], which was previously proposed by our research team for the IoT system, is discussed.

The MQL message scheduling algorithm proposed for the IoT system [18] is based on the Q-learning algorithm [19]. Figure 1 shows the Q-learning agent model for IoT environments. Q-learning is a reinforcement-based learning algorithm that finds an optimal policy by selecting the action with the highest reward in each state [19].

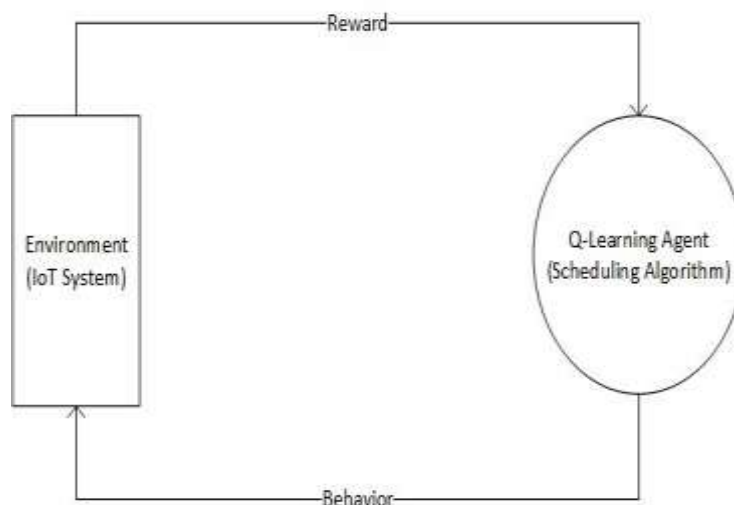


Figure 1. Q-Learning Agent Model for IoT Environment

In the MQL message scheduling algorithm, messages to be transmitted are classified into real-time (or urgent) messages and delay-tolerant messages. Real-time messages should be scheduled to transmit on time, while delay-tolerant messages can be delayed to allow the transmission of real-time messages. There are two message queues in the algorithm – real-time message priority queue and delay-tolerant message priority queue. Table 1 shows the message classifications in the algorithm.

Table 1. Message Classifications in the MQL Algorithm

Priority	Message Class	Message Priority
High	Real-time message	Real number
Low	Delay tolerant message	Real number

The MQL message scheduling algorithm provides QoS by determining the transmission times of delay-tolerant messages without interfering in real-time message transmission. The MQL algorithm evaluates to see if previous decisions made on the transmission times for delay-tolerant messages are appropriate or not. Then the evaluation result is applied to the next scheduling decision. The algorithm is designed to find optimal scheduling decisions through a successive Q-learning process (*i.e.*, scheduling decisions, evaluations and applications).

The MQL algorithm was compared with the Multiclass Based Dynamic Priority (MBDP) algorithm [20], one of the well-known multiclass message scheduling algorithms. It was found that the throughputs of the MQL algorithm increase almost linearly as the measurement time increases, whereas the throughputs of the MBDP algorithm increase with decreases in the increasing ratio [18].

2.2. Weighted Fair Queue Scheduling Algorithm

The Weighted Fair Queue (WFQ) algorithm is one of the queuing algorithms that gives narrowband signals an advantage over broadband signals in packet-switched communication networks. The higher priority is given to narrowband signals, while a lower priority is given to broadband signals, so that fair allocation of network bandwidth to all the data flows in the network can be achieved.

In this algorithm, broadband signals share the resources that remain after low-bandwidth signals have been transmitted. The WFQ algorithm can prevent high-bandwidth traffic from overwhelming the resources of a network, to minimize the average latency and avoid partial or complete failure of low-bandwidth communications during periods of peak network traffic.

3. Efficient Multi-class Message Scheduling Algorithm for Healthcare IoT Systems

In this section, the efficient multi-class message scheduling algorithm for healthcare IoT systems proposed in this paper is explained. Because the proposed algorithm uses a simpler priority calculation than the MQL algorithm, it requires a shorter decision time for message scheduling, making it more appropriate for IoT systems, where computing resources are very limited.

3.1. Message Classification in the Proposed Scheduling Algorithm

In the proposed algorithm, messages are classified according to their characteristics into unconditional (UNC), real-time (RT) and delay-tolerant (DT) messages. UNC messages should be transmitted within their deadlines. RT messages need to be transmitted to meet their deadlines when possible, while DT messages can be transmitted when their transmission does not interfere with the transmission of UNC messages, and when the priority of the DT message queue is higher than that of the RT message queue.

Each message class has its own message queue, and there are three messages queues for the three message classes, with the UNC messages queue having the

highest priority. A UNC message is scheduled to transmit as soon as it enters the UNC queue only if no other higher priority UNC messages are already present in the queue. An RT message is scheduled to transmit when the UNC queue is empty and when the priority of the RT message queue is higher than that of the DT message queue and no other higher priority RT messages are already present in the RT queue. DT messages can be scheduled when the UNC queue is empty, but DT messages are transmitted at the very lowest priority.

Each message is classified according to how much delay in the message's transmission is tolerable. UNC messages should be sent immediately, otherwise a patient's life would be in danger. Examples of UNC messages are abnormal ECG, abnormal oxygen saturation, patient falls, etc. RT messages such as Normal ECG, Normal Oxygen saturation are messages that should be transmitted as soon as possible. A delay in the transmission would cause inconvenience to patients or medical staff. Messages related to weight, basic metabolism or bone mass can be classified as DT messages because their transmission can be delayed without severe inconvenience to patients or medical staff. Table 2 shows the message classification used in the proposed algorithm.

Table 2. Message Classification

Message Class	Message	Matching Queue	Priority
Unconditional(UNC)	Abnormal ECG, Abnormal Oxygen saturation, Patient falls, etc.	Unconditional Queue	High
Real-Time(RT)	Normal ECG, Normal Oxygen saturation, etc.	Middle Weighted Queue	Medium
Delay Tolerant(DT)	Weight, Basic metabolism, bone mass, etc.	Low Weighted Queue	Low

3.2. Scheduling Algorithm

Figure 2 shows the message classification module. Once a message is received, the message enters one of three message queues according to its state.

```

FUNCTION dist(msg)
  IF msg.state EQUAL unc
    ENQUEUE(uncq, msg)
  ELSE IF msg.state EQUAL rt
    ENQUEUE (rtq, msg)
  ELSE IF msg.state EQUAL dt
    ENQUEUE (dtq, msg)
  ELSE
    RETURN -1
  END IF
END FUNCTION
    
```

Figure 2. Message Classification Module

Once a message enters into a queue, the proposed scheduling algorithm is invoked. Figure 3 shows the scheduling algorithm. If the UNC message queue ('uncq') is not empty, the first UNC message in the queue is sent immediately. If the queue is empty, the algorithm checks whether the RT message queue ('rtq') is not empty. Then, the first RT message in the RT message queue is sent and the priority of the RT (DT) message queue is decreased (increased). In other words, the priority of the selected (unselected) queue for scheduling decreases (increases).

Whenever an RT message is sent, the priority of the RT message queue is decreased by 10% and the priority of the DT message queue is increased by 10%. In addition, whenever a DT message is sent, the priority of the DT message queue is decreased by 30% and the priority of the RT message queue is increased by 30%. The algorithm ensures that the transmission of DT messages is not blocked indefinitely by setting the increment of the priority of the DT message queue (10%) as less than that of the priority of the RT message queue (30%).

When the priority of the RT message queue is higher than that of the DT message queue but the RT message queue is empty, then the first DT message in the DT message queue is sent and the priorities of the queues remain unchanged.

```
WHILE
  IF NOT IS-EMPTY(uncq)
    DEQUEUE(uncq)
  ELSE IF NOT IS-EMPTY(rtq)
    dtq.priority := dtq. priority + rtq. priority * 0.1
    rtq. priority := rtq. priority - rtq. priority * 0.1
    DEQUEUE(rtq)
  ELSE IF NOT IS-EMPTY(dtq) AND dtq. priority >= rtq. priority
    rtq. priority = rtq. priority + dtq. priority * 0.3;
    dtq. priority = dtq. priority - dtq. priority * 0.3;
    DEQUEUE(dtq);
  ELSE IF IS-EMPTY(rtq) AND NOT IS-EMPTY(dtq) AND rtq. priority >=
dtq. priority
    DEQUEUE(dtq)
  ELSE IF IS-EMPTY(dtq) AND NOT IS-EMPTY(dtp) AND dtq. priority >=
rtq. priority
    DEQUEUE(rtq)
  END IF
END WHILE
```

Figure 3. Proposed Message Scheduling Algorithm

Figure 4 shows how the priority of a message queue changes as a message in the queue is transmitted. Figure 5 shows how the algorithm determines the transmission times of delay-tolerant messages.

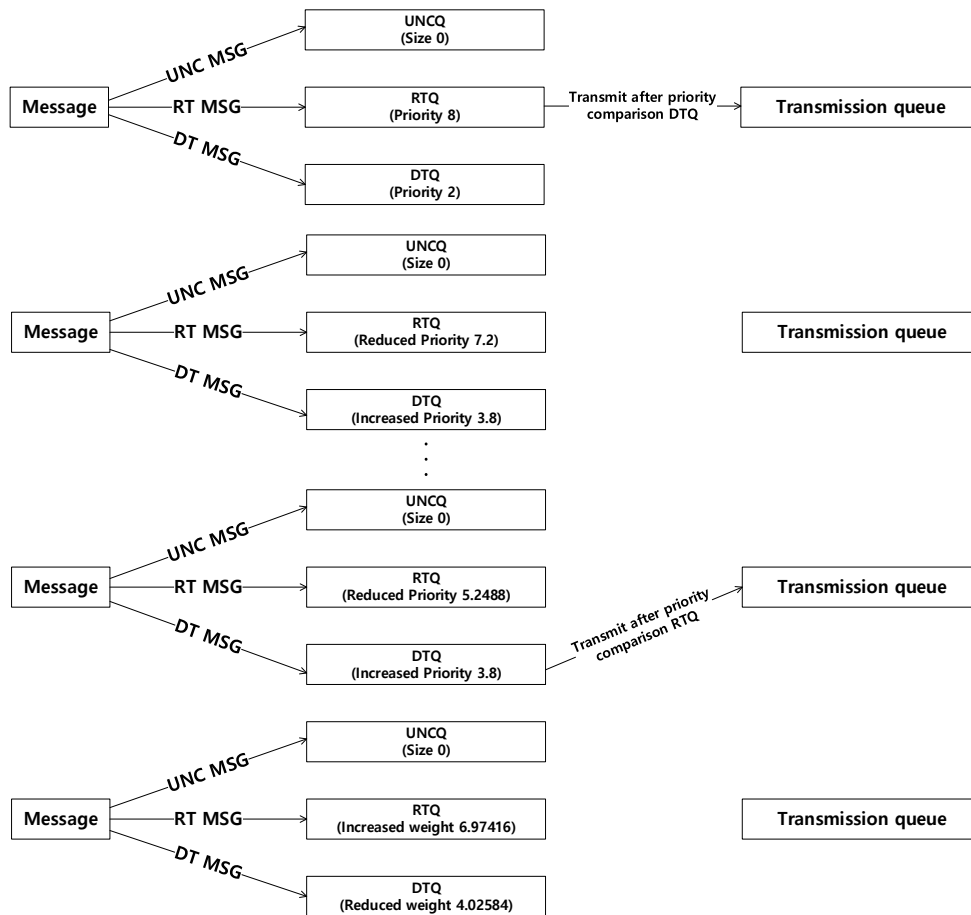


Figure 4. Changes of Queues' Priorities

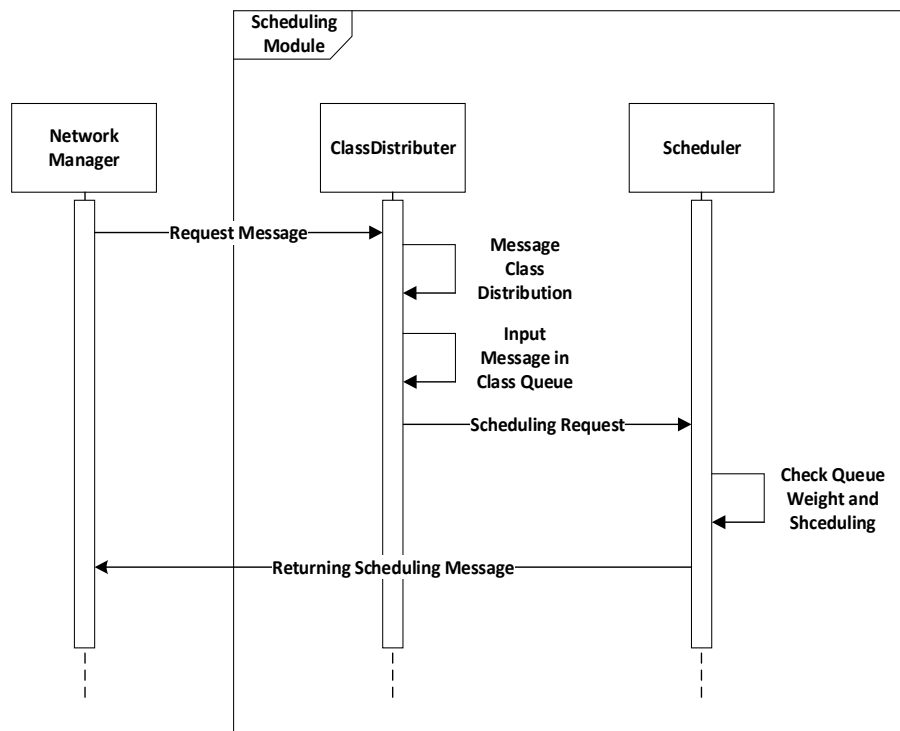


Figure 5. Process of the Proposed Scheduling Algorithm

4. Experiments

4.1. Experiment Environment

The healthcare IoT system used in this study was constructed based on the oneM2M communication protocol [21-22], one of the IoT communication standards. Figure 6 shows a part of the oneM2M-based IoT system used in this experiment. In the figure, MN and IN represent Middle Node and Infrastructure Node, respectively. AE and CSE represent Application Entity and Common Service Entity, respectively. MN and IN are oneM2M terminologies representing a gateway and the IoT server, respectively.

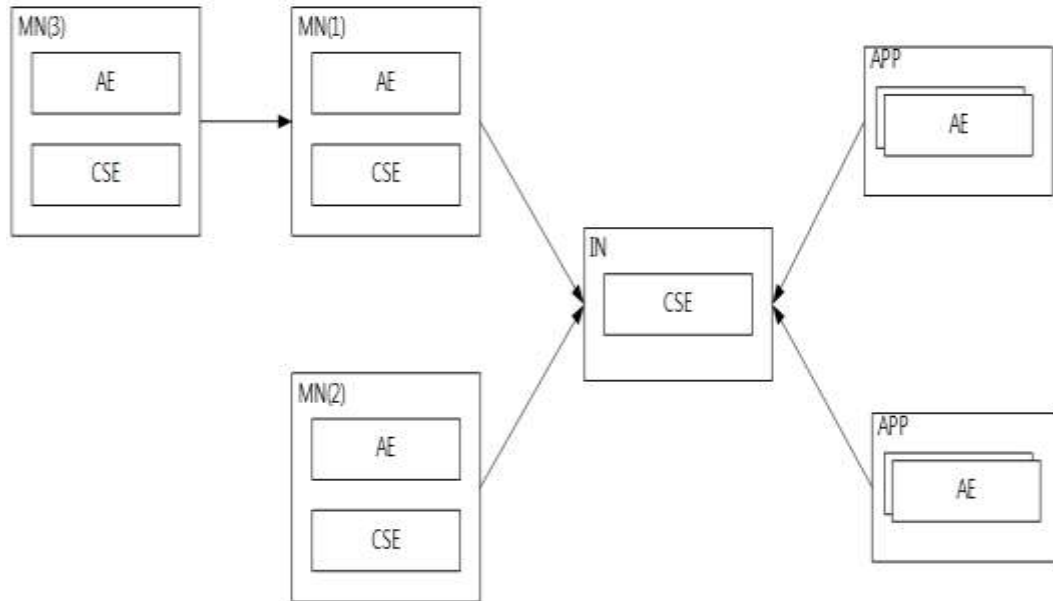


Figure 6. A Part of the oneM2M-based IoT System used in the Experiments

Table 3 shows the specifications of the system used in the experiments

Table 3. Specifications of System used in the Experiments

Classification	Message sending node	Message receiving and scheduling node
CPU	Intel Core I7-3770(3.4GHz)	Intel Core I7-3770(3.4GHz)
Main Memory	8GB	8GB
HDD	SSD	SSD
OS	Window 7	Window 7

4.2. Performance Measurements

In this experiment, MN(3) sends messages and MN(1) receives the messages to schedule them. Message scheduling time is measured until a certain number of messages has been scheduled completely. Table 4 shows the scheduling times of the proposed algorithm and the MQL algorithm for ten thousand and twenty thousand messages.

Table 4. Scheduling Times of the Proposed Algorithm and the MQL Algorithm (in seconds)

Number of messages	10000		20000	
Scheduling algorithm	Proposed algorithm	MQL algorithm	Proposed algorithm	MQL algorithm
Unconditional	1.85910634	2.151123	3.72721318	4.2772447
Real-Time	2.14892292	2.1351221	4.87487882	4.1992401
Delay Tolerant	3.2103836	4.4092522	7.31081816	8.689497

When one thousand UNC messages are scheduled, the proposed algorithm and the MQL algorithm require 1.86 seconds and 2.15 seconds, respectively. This means that the proposed algorithm performs 13.5% better in terms of the scheduling time. For RT messages, the two algorithms perform similarly. When one thousand DT messages are scheduled, the proposed algorithm and the MQL algorithm require 3.2 seconds and 4.4 seconds, respectively. This means that the proposed algorithm performs 27.2% better.

Figure 7 and Figure 8 compare the two algorithms' performance in terms of scheduling time, and show the scheduling times when ten thousand and twenty thousand messages are scheduled, respectively. The comparisons show that in most cases, the proposed scheduling algorithm performs better than the MQL scheduling algorithm. This is mainly because the proposed algorithm uses simpler priority calculation than the MQL algorithm.

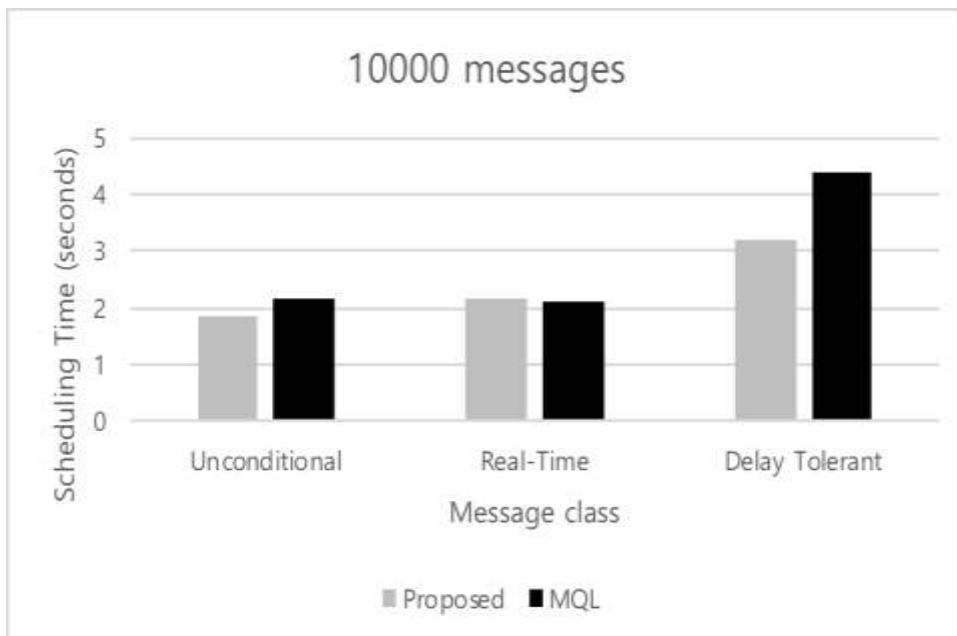


Figure 7. Scheduling Time Comparison (10000 messages)

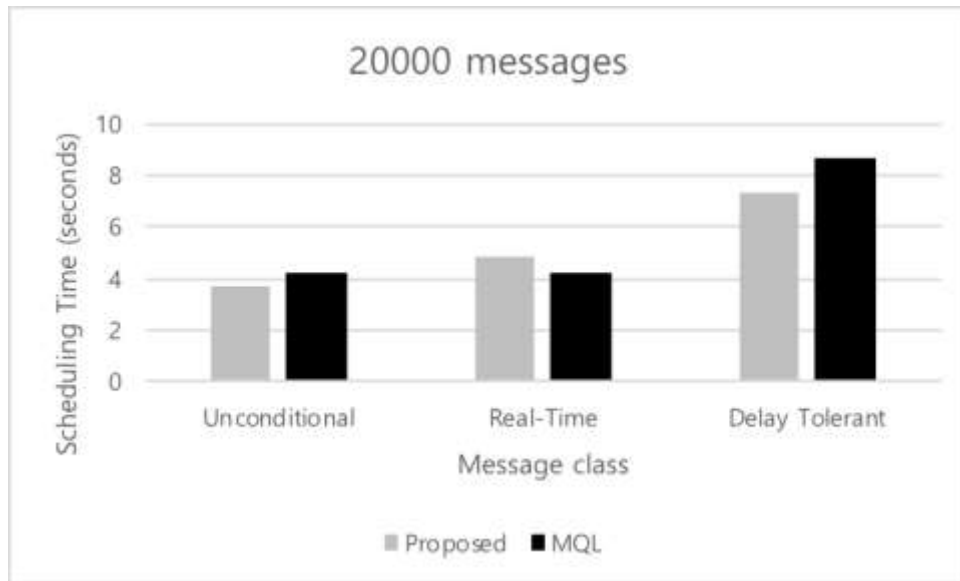


Figure 8. Scheduling Time Comparison (20000 messages)

5. Conclusion

In this paper, an efficient multi-class message scheduling algorithm for healthcare IoT environments is proposed. In the proposed message algorithm, messages are grouped based on their characteristics into three message classes (UNC, RT and DT classes). The proposed algorithm uses simpler priority calculation than the MQL algorithm, which was proposed previously by our research team.

Experiments are conducted on a part of the oneM2M-based IoT system which we have built. The experiments show that the proposed scheduling algorithm performs better than the MQL scheduling algorithm in most cases. The proposed message scheduling algorithm is resource-efficient because the algorithm schedules more messages for a given time than the previous message scheduling algorithm, which we proposed for IoT systems.

Acknowledgments

This paper is a revised and expanded version of a paper entitled “Weighted Unfair Queue Scheduling Algorithm” presented in the Proceedings of the 5th International Conference on Interdisciplinary Research Theory and Technology, Daejeon, Dec. 21-23, 2017.

References

- [1] I. Strategy, “Policy Unit (SPU) ITU Internet Reports 2005: TheInternet of Things”, Geneva: International Telecommunication Union (ITU), (2005).
- [2] M. B. Tamboli and D. Dambawade, “Secure and efficient CoAP based authentication and access control for Internet of Things (IoT)”, 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), (2016), pp. 1245-1250.
- [3] X. Li, “A queue scheduling approach to QoS support in terminal communication access network”, 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), (2016), pp. 1974-1979.
- [4] R. Duan, X. Chen, and T. Xing, “A QoS Architecture for IOT”, in 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, (2011), pp. 717-720.
- [5] Z. Mahmood, H. Ning and A. Ghafoor, “Lightweight Two-Level Session Key Management for End User Authentication in Internet of Things”, in 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and

- IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), (2016), pp. 323-327.
- [6] M. Carlson, W. Weiss, S. Blake, Z. Wang, D. Black, and E. Davies, "An architecture for differentiated services", RFC 2475, (1998) December.
- [7] M. Liu, Z. Li, X. Guo and E. Dutkiewicz, "Performance Analysis and Optimization of Handoff Algorithms in Heterogeneous Wireless Networks", IEEE Transactions on Mobile Computing, vol. 7, no. 7, (2008), pp. 846-857.
- [8] S. Li, L. Xu, X. Wang and J. Wang, "Integration of hybrid wireless networks in cloud services oriented enterprise information systems", Enterprise Information Systems, vol. 6, no. 2, (2012), pp. 165-187.
- [9] A. E. Al-Fagih, F. M. Al-Turjman, W. M. Alsalihi and H. S. Hassanein, "A Priced Public Sensing Framework for Heterogeneous IoT Architectures", IEEE Transactions on Emerging Topics in Computing, vol. 1, no. 1, (2013), pp. 133-147.
- [10] R. Callon, A. Viswanathan and E. Rosen, "Multiprotocol label switching architecture", IETF Request for Comments (RFC), vol. 3031, (2001).
- [11] I. F. Akyildiz, S. Weilian, Y. Sankarasubramanian and E. Cayirci, "A survey on sensor networks", IEEE Communications Magazine, vol. 40, no. 8, (2002), pp. 102-114.
- [12] J. Guo, L. D. Xu, G. Xiao and Z. Gong, "Improving Multilingual Semantic Interoperation in Cross-Organizational Enterprise Systems Through Concept Disambiguation", IEEE Transactions on Industrial Informatics, vol. 8, no. 3, (2012), pp. 647-658.
- [13] S. Hachani, L. Gzara and H. Verjus, "A service-oriented approach for flexible process support within enterprises: application on PLM systems", Enterprise Information Systems, vol. 7, no. 1, (2013), pp. 79-99.
- [14] T. Melodia and I. F. Akyildiz, "Cross-layer QoS-aware communication for ultra wide band wireless multimedia sensor networks", IEEE Journal on Selected Areas in Communications, vol. 28, no. 5, (2010), pp. 653-663.
- [15] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks", IEEE Communications Surveys & Tutorials, vol. 14, no. 2, (2012), pp. 265-278.
- [16] J. Park, "A study on scheduling algorithm and multi-path exploration algorithm for PHD Networks", M. Sc., Keimyung, Daegu, (2016).
- [17] M. Rouse, "What is weighted fair queueing (WFQ)", Available: <http://searchnetworking.techtarget.com/definition/weighted-fair-queueing>, (2006).
- [18] K. H. Park, J. Park and J. Whi Lee, "An IoT System for Remote monitoring of Patients at Home", Applied Sciences, doi:10.3390/app7030260, vol. 7, no. 3, (2017).
- [19] C. J. Watkins and P. Dayan, "Q-learning", Machine learning, vol. 8, no. 3-4, (1992), pp. 279-292.
- [20] M. K. Giluka, N. Rajoria, A. C. Kulkarni, V. Sathya and B. R. Tamma, "Class based dynamic priority scheduling for uplink to support M2M communications in LTE", Proc. of IEEE World Forum on Internet of Things, (2014), pp. 313-317.
- [21] oneM2M. (2016, 30, August). Requirements. Available: http://www.onem2m.org/images/files/deliverables/Release2/TS-0002-Requirements-V2_7_1.pdf
- [22] oneM2M. (2016, 30, August). Functional Architecture. Available: http://www.onem2m.org/images/files/deliverables/Release2/TS-0001-%20Functional_Architecture-V2_10_0.pdf.

Authors



Keehyun Park received his B.Sc. in Computer Science from Kyungpook National University (Korea) in 1979, his M.Sc. in Computer Science from KAIST (Korea) in 1981, and his Ph.D. degree in Computer Science from Vanderbilt University (USA) in 1990. He has been a professor in the Computer Science and Engineering Department at Keimyung University (Korea) since March 1981. His research interests include Mobile/Network Communication Systems, Device Management for u-healthcare Systems, Embedded Systems and Parallel Processing Systems.



Insung Kim is a Computer Science Ph.D. candidate at Keimyung University in Daegu (Korea). His interests include IoT Systems and Communications Protocols.



Joonsuu Park received his M.Sc. in Computer Science from Keimyung University in Daegu (Korea) in 2016. His research interests include intelligent communication networks, IoT Systems and IoT communication security.

