Evaluation of Priority Control Mechanism for Remote Monitoring IoT System in Greenhouses

Takuma Tachibana[1(✉)], Eisuke Kasahara[2], Takamasa Yoshida[2], and Hiroshi Mineno[3]

[1] Graduate School of Integrated Science and Technology, Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka 432-8011, Japan
tachibana@minelab.jp

[2] Central Research Laboratory Technology Group, Hamamatsu Photonics K.K., 5000 Hirakuchi, Hamakita-ku, Hamamatsu, Shizuoka 434-8601, Japan
{eisuke.kasahara, takamasa.yoshida}@crl.hpk.co.jp

[3] College of Informatics, Shizuoka University/JST PRESTO, 3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka 432-8011, Japan
mineno@inf.shizuoka.ac.jp

Abstract. The Internet of Things (IoT) has expanded rapidly in recent years. Therefore, there are various types of data available from IoT devices, such as texts, images, and sound. It will become possible to construct a heterogeneous remote monitoring IoT system using a variety of IoT devices. However, a heterogeneous remote monitoring IoT system cannot send complete data because most mobile network services for the IoT system do not guarantee bandwidth. Therefore, we proposed a priority control mechanism for a heterogeneous remote monitoring IoT system that controls the amount of data, and the time it takes to send it for IoT devices, which is decided in accordance with the quality of service. In this paper, we show how we improved the applicability of the operations of a priority control system and evaluated the effectiveness of an actual remote monitoring IoT system by using a greenhouse data-collection system.

Keywords: Internet of Things · Priority control · Mobile networks
Data collection

1 Introduction

All people expect the Internet of Things (IoT) to grow because of downsized sensors and the appearance of various mobile networks [1–3]. The extremely wide variety of IoT devices generate various types of data, such as texts, images, and sounds. Therefore, it has been indicated that the number of devices connected to the IoT and network traffic per device will grow in the next five years [4]. In addition, mobile network service providers have deployed handle services for the IoT system in recent years. However, there is a problem regarding remote monitoring IoT systems [3] that collect multifarious data from remote locations (referred to as heterogeneous remote monitoring IoT systems). The problem is that mobile network services for the IoT

K. Murao et al. (Eds.): MobiCASE 2018, LNICST 240, pp. 258–264, 2018.
https://doi.org/10.1007/978-3-319-90740-6_20

cannot completely sustain the required traffic from a heterogeneous remote monitoring IoT system because almost all mobile network services for the IoT system do not guarantee throughput. To solve this problem, we proposed a priority control mechanism for a heterogeneous remote monitoring IoT system [5]. We proved its effectiveness in enabling a IoT system that guarantees telecommunications at the lowest limit. In this paper, we evaluate the effectiveness of an actual remote monitoring IoT system. In accordance with the availability in actual system operations, we improved the proposed mechanism. The evaluation target in this paper is a greenhouse data-collection system. Through this evaluation, we show the effectiveness of the proposed mechanism for various mobile networks and systems.

The remainder of this paper is organized as follows: Sect. 2 shows related work in terms of methods and protocols for IoT telecommunications. The improved priority mechanism and implementation in the actual system is described in Sect. 3. The experimental results and discussion are shown in Sect. 4, and following that is the conclusion.

2 Related Work

Various methods and protocols for IoT telecommunications have been proposed [6–8]. One protocol for the IoT, MQ Telemetry Transport (MQTT) [9], targets IoT traffic from many devices that send and receive small amounts of data at high frequency. MQTT is a very effective protocol for IoT devices regarding sending lightweight data. In contrast, header size reduction does not affect IoT devices, for example cameras, when they send a large amount of data. In addition, MQTT has a quality of service (QoS) control mechanism for unstable mobile networks. However, this mechanism does not consider data characteristics and does not screen the data to be sent.

Probabilistic prediction-based scheduling [10] is a scheduling model for IoT system traffic. The objective of probabilistic prediction-based scheduling is to reduce the waiting time of high priority packets and keep the waiting time of other priority packet services within tolerable limits. However, probabilistic prediction-based scheduling is unable to meet the IoT application requirements because it does not consider how to determine data that has high priority.

Most existing methods cannot solve the problems in IoT telecommunications in regard to data amount, frequency, and number of devices. Additionally, IoT telecommunications must consider the QoS required by applications. Therefore, it is important to implement a telecommunication mechanism for IoT that considers multiple data characteristics, mobile networks, and QoS required by applications.

3 Priority Control Mechanism

In this section, we explain our mechanism [5]. The mechanism focuses on the priority and characteristics of data to control the data-sending order and data amount by setting the application configuration as a requirement. Therefore, it enables a level of control that satisfies the requirements from the application's QoS.

Figure 1 shows the architecture of the proposed mechanism. The proposed mechanism consists of three elements: IoT devices, a broker server, and an application server. The IoT devices are endpoint devices that are connected to mobile networks.

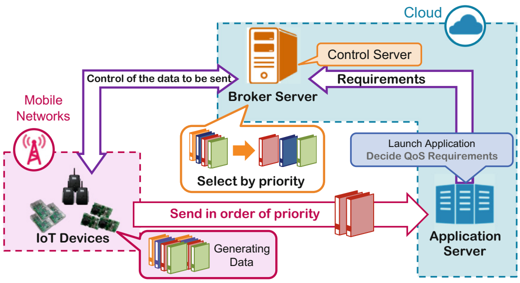


Fig. 1. Architecture of priority control mechanism

The application server is an IoT system server that processes the data from the IoT devices, and the broker server is a priority control server in the mechanism. The application server and broker server are split by function; therefore, they can run on the same physical server. Because of this split, the elements' roles can be clearly defined.

Figure 2 shows the flow of the proposed mechanism. The proposed mechanism consists of three phases; a register phase, a priority telecommunication phase, and a release phase. In the register phase, the application server registers the QoS requirements of its own application. The broker server creates a "priority decision table" and tells the IoT devices the application server address. After that, all the elements establish a TCP connection with each other. In the priority telecommunication phase, the broker server gives priority to the data, which is generated by the IoT devices (referred to as 'content data'), and manages it. Additionally, the broker server permits the content data that has the highest priority to be sent.

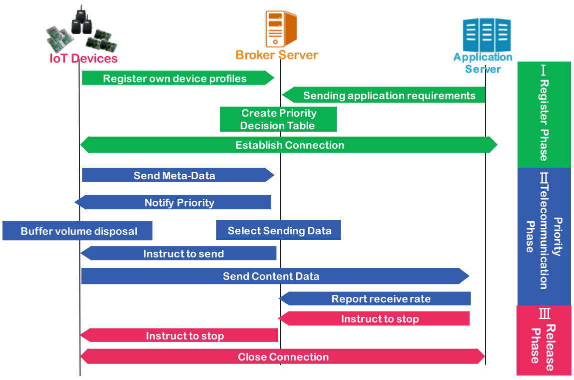


Fig. 2. Flow of priority control mechanism

We improved the priority control mechanism to operate in the actual system. We implemented acknowledgement messages corresponding to each control message in the mechanism to prevent decreasing IoT system availability because, when the system is installed on a remote environment, some connection errors occur due to bad radio wave conditions depending on the location. The sender can determine that the connection was lost and try reconnecting to the receiver if an acknowledgement message is not received within a given time. In this paper, we set the time limit as 20 s.

4 Evaluation and Discussion

In this section, we evaluate the applicability including performance of the priority control mechanism in an actual system that uses different mobile networks to those used in [5]'s evaluation. In this evaluation, the priority control mechanism uploaded data from IoT devices to an application server in accordance with the scenario. Table 1 shows the scenario. This scenario was made by using requirements from the target system. We used two evaluation indicators: the sending ratio of each priority (the ratio of data that could be transmitted within the evaluation times) and the average transfer complete time for each priority (the required time from the generation of data in the IoT devices to complete transition). If all the sending ratios with a higher priority are higher than those with a lower priority and all the average transfer complete times with a higher priority are shorter than those with a lower priority, we can say the proposed mechanism applicable in actual System.

Table 1. Scenario of this evaluation

Data type	Data amount	Data generation interval	Data interval	Priority
Camera (small)	About 220 KB	1 min	5 min	2
			1 min	3
Camera (large)	About 2 MB		10 min	4
			1 min	5

Figure 3 shows the architecture of the greenhouse data-collection system that was used for the evaluation in this paper. The greenhouse data-collection system has 12 camera devices. The priority control mechanism uploads data generated by these camera devices to a cloud server. We installed Raspbian 8 onto Raspberry Pi 2 and used them as IoT devices. Similarly, we used Ubuntu 14.04 LTS as the application and broker servers. The installation site is an agricultural faculty field in Shizuoka University. We used WiMAX2 for the system's mobile network. The average mobile network throughput was 2.30 Mbps (measured by iPerf) at the installation site. The evaluating time was from 3:20 PM July 18 to 10:00 AM July 20 (43 h).

Figure 4 shows the results of the sending ratio and average transfer complete time for each priority. The sending ratios with a higher priority were higher than those with lower priority. In addition, all the average transfer complete times with a higher priority were shorter than those with lower priority. Therefore, proposed mechanism could ensure bandwidth immediately for data that has high priority because it determines which data to send using the application's requirement. These results show that the priority control mechanism could satisfy minimum requirements of the actual IoT system.

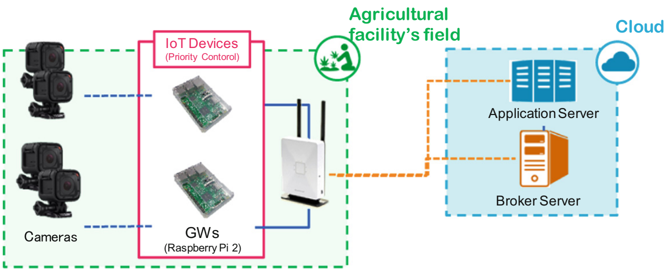
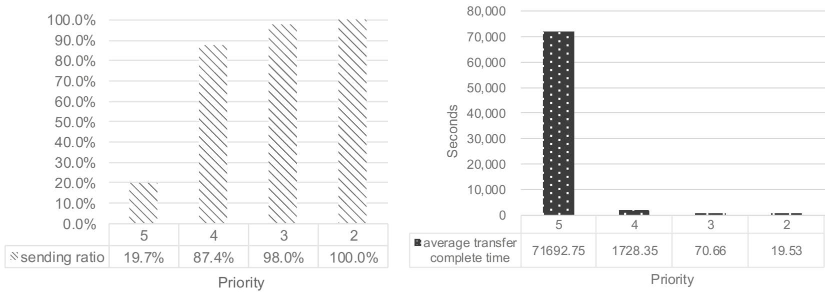


Fig. 3. Architecture of greenhouse data-collection system



Priority	5	4	3	2
sending ratio	19.7%	87.4%	98.0%	100.0%

Priority	5	4	3	2
average transfer complete time	71692.75	1728.35	70.66	19.53

Fig. 4. Sending ratio and average transfer complete time for each priority

However, the average transfer complete time for the highest priority was too long (19.53 s) because the calculated transfer time was 0.75 s if the bitrate was 2.30 Mbps and the data size was 220 kBs. Therefore, the average transfer complete time was too long regarding network throughput. Figure 5 shows the amount of received bitrates that were calculated per 20 s. As shown in Fig. 5, there was dispersion over the course of time in the received bitrates, especially during the daytime. We consider that it was caused by a long round-trip time in this environment. When mobile networks that have long round-trip times are used, the priority control mechanism spends time sending and receiving control messages between each element. Furthermore, the average round-trip time of the mobile network was 702 ms (measured by ping) at the installation site. Therefore, the performance of the current priority control mechanism is affected by the round-trip time. We should solve the problems of the long average transfer complete time and uneven bitrate through a mobile network that has a long round-trip time.

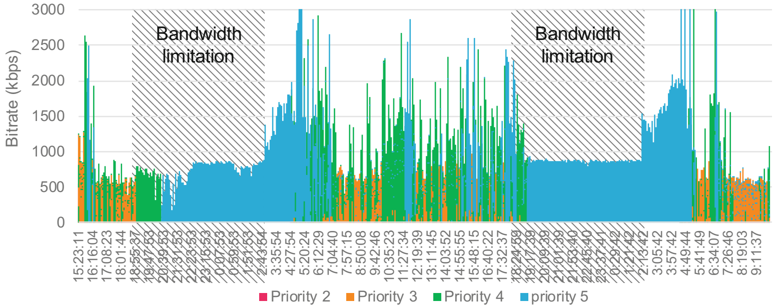


Fig. 5. Received bitrates calculated every 20 s

5 Conclusion

In this paper, we evaluated the applicability of a priority control mechanism in an actual remote monitoring IoT system. We improved the mechanism in accordance with the system operations. We evaluated proposed mechanism in an actual remote monitoring IoT system, a greenhouse data-collection system as actual remote monitoring IoT system. We showed that the priority control mechanism satisfied the minimum requirements of the actual IoT system. On the other hand, the proposed mechanism's performance deteriorated when it was used over high round-trip time networks. In future work, we will overcome this performance problem in high delay network. Furthermore, we will evaluate our mechanism in various network environments.

Acknowledgements. This work was partially supported by the JSPS (KAKENHI JP26280028) and the Cooperative Research Project Program of the RIEC, Tohoku University.

References

1. Smith, T.F.: Towards a definition of the Internet of Things (IoT). IEEE (2016). http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf
2. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
3. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. **29**(7), 1645–1660 (2013)
4. Cisco Visual Networking Index: Global mobile data traffic forecast update, 2014–2019. Technical report, Cisco (2015)
5. Tachibana, T., Furuichi, T., Mineno, T.: Implementing and evaluating priority control mechanism for heterogeneous remote monitoring IoT system. In: Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services, pp. 239–244. ACM (2016)

6. Sheng, Z., Mahapatra, C., Zhu, C., Leung, V.C.: Recent advances in industrial wireless sensor networks toward efficient management in IoT. IEEE Access **3**, 622–637 (2015)
7. Palattella, M.R., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, L.A., Boggia, G., Dohler, M.: Standardized protocol stack for the internet of (important) things. IEEE Commun. Surv. Tutor. **15**(3), 1389–1406 (2013)
8. Sheng, Z., Yang, S., Yu, Y., Vasilakos, A.V., McCann, J.A., Leung, K.K.: A survey on the IETF protocol suite for the Internet of Things: standards, challenges, and opportunities. IEEE Wirel. Commun. **20**(6), 91–98 (2013)
9. MQ Telemetry Transport. http://mqtt.org/
10. Sharma, R., Kumar, N., Gowda, N.B., Srinivas, T.: Probabilistic prediction based scheduling for delay sensitive traffic in Internet of Things. Procedia Comput. Sci. **52**, 90–97 (2015)