Simulation Study of Dynamic Observing Period Adjustment for CoAP-based Monitoring Systems

Sungmo Kim, Jung-Hyok Kwon and Eui-Jik Kim*

Department of Convergence Software, Hallym University, South Korea; imii1992@hallym.ac.kr, jhkwon@hallym.ac.kr, ejkim32@hallym.ac.kr

Abstract

Objectives: This paper is a simulation study of dynamic observing period adjustment (DOPA) to determine the appropriate DOPA stage value that prevents the buffer overflow in various constrained application protocol (CoAP) based monitoring systems. Methods/Statistical Analysis: To determine the appropriate DOPA stage value for various CoAP-based monitoring systems, we set a variety of simulation scenarios that use different values of simulation parameters and investigate the number of successful transmissions and dropped messages of each simulation. Then, we determine the appropriate DOPA stage value of each scenario, which prevents buffer overflow and maximizes the number of successful transmissions simultaneously. Findings: The experimental simulation is conducted under various simulation scenarios. The simulation results show that the appropriate DOPA stage value increases as the number of devices or payload size increases. It also increases when the observing period of servers or clock rate decreases. The correlations between the DOPA stage value and simulation parameters are obtained through these results, which is used for determining the appropriate DOPA stage value. Improvements/Applications: Our study can be used to provide an optimal observing period for buffer overflow prevention in various monitoring systems.

Keywords: Buffer Overflow, CoAP, DOPA, DOPA Stage Value, Monitoring Systems, Observing Period

1. Introduction

Recently, Web-based monitoring systems have received significant attention from various fields to realize Internet of things (IoT) services, such as smart farms and smart buildings. To support such services, the use of lightweight web transfer protocol is necessary, since most IoT services use constrained devices and lossy networks.

The constrained application protocol (CoAP) is a representational state transfer (REST) architecture-based lightweight web transfer protocol developed by the IETF CoRE WG. The CoAP provides asynchronous interactions with a small header over the user datagram protocol (UDP). Since the CoAP operates over the UDP, it has its own congestion control scheme where the devices reduce the number of retransmission messages. However, it does not consider the buffer overflow problem, which causes serious packet loss in constrained environments.

To solve this problem, proposed dynamic observing period adjustment (DOPA) for the CoAP. DOPA adjusts the observing period of servers to prevent buffer overflow of the client when the client monitors servers’ resources using the CoAP observe option. To this end, DOPA sets the buffer threshold of the client. If the buffer of the client reaches the buffer threshold, the client sends a threshold message to the servers to announce the buffer overflow alert. The servers that receive the threshold message increase their observing period using the predefined DOPA stage value. Note that DOPA defines multiple DOPA stages, which have different increasing observing period ratios called DOPA stage values. Each DOPA stage value is obtained based on the maximum DOPA stage number (i.e., MAX DOPA stage) and MAX DOPA stage value. By using DOPA, however, the servers may not appropriately adjust their observing period since the DOPA stage value is only determined by the user or system. For example, if

*Author for correspondence
the DOPA stage value is too small, buffer overflow may occur. On the other hand, when it is too big, the observing period of servers becomes longer than necessary.

In this paper, we focus on a simulation study of DOPA to determine the appropriate DOPA stage value that can prevent buffer overflow in the various system environments. To this end, we set multiple simulation scenarios using different values of simulation parameters (i.e., number of devices, payload size, observing period of servers, and clock rate of client) and then investigate the number of successful transmissions and dropped messages of each simulation. The simulation results show that the appropriate DOPA stage value increases when the number of devices or payload size increases and increases when the observing period of servers or clock rate decreases. By using the results, we define the correlations between parameters and DOPA stage value, which can provide the appropriate DOPA stage value.

The rest of this paper is organized as follows. In Section 2, the system model is described briefly. The simulation setting and results are presented in Section 3. In Section 4, the correlations between each parameter and DOPA stage value is presented. Finally, we conclude this paper in Section 5.

2. System Model

Figure 1 shows the CoAP-based monitoring system architecture. In the figure, the monitoring system consists of a single client and multiple servers. The client manages the associated servers and requests the servers to obtain the sensing data and server status. The servers detect the changes in their surrounding environment using the attached sensors and then transmit the sensing data to the client. All types of devices are interconnected to each other via wireless links, and they communicate with one another using the CoAP. In the system model, we assume that the client and servers are constrained devices with low power and limited ROM, RAM, and CPU.

In order to monitor the servers’ resources periodically, the devices use the CoAP observes option. Specifically, the client transmits the observe command to the target server to inform it that observation will begin. Upon receiving this command, the server periodically detects the changes in its surrounding environment and then transmits the sensing data to the client. Note that the observing period can be predefined by the system depending on the application. In the system, we assume that all associated servers have the same observing period.

During the observe, the client and servers use DOPA to prevent buffer overflow. When the client transmits the threshold message to the servers, the servers increase their observing period using the DOPA stage value, while the servers’ observing period becomes shorter up to the predefined observing period if they do not receive the threshold message for a period of time. DOPA can prevent buffer overflow by adjusting the observing period of servers. However, the existing DOPA may degrade network performance in terms of the number of dropped messages and successful transmissions, since it just uses the DOPA stage value determined by the user without any consideration of system environments.

3. Simulation Study for Appropriate DOPA Stage Value

In this section, we conduct experimental simulations under various system environments to determine the appropriate DOPA stage value. To consider the various system environments, we set multiple simulation scenarios where different values of system parameters (i.e., number of devices, observing period of servers, clock rate of client, payload size) are applied. Then, we determine the appropriate DOPA stage values of each simulation by investigating the number of successful transmissions and dropped messages. The detailed simulation setting and results are described in the following subsections.

3.1 Simulation Setting

We conduct the simulation by using MATLAB to determine the appropriate DOPA stage value. In the simulation, we
assume that a single client and multiple servers are placed in a $10 \times 10$ square room. The client has a 1Mbyte buffer, and the buffer threshold is set to 0.9Mbytes (i.e., 90% of buffer size). In addition, the clock rate is set to 10MHz; thus, the client processes 10Kbits of data per second. Each server periodically transmits 125bytes of sensing data to the client. The observing period and the number of devices are set to 10ms and 150, respectively. We set the simulation time to 100 seconds. The detailed initial simulation parameters are listed in Table 1.

In the simulation, we use the fixed MAX DOPA stage (i.e., stage 5), while the various MAX DOPA stage values are applied to each simulation for performance comparison. In each simulation, the different DOPA stage values can be obtained through the MAX DOPA stage and MAX DOPA stage value. Table 2 shows an example of DOPA stage values when the MAX DOPA stage value is set to 1.5.

### 3.2 Simulation Result

Figures 2 and 3 show the number of successful transmissions and the number of dropped messages when the initial simulation parameters are used. In the simulation, the MAX DOPA stage value is changed from 1.45 to 1.55 in increments of 0.05. In Figure 2, the number of successful transmissions increases when the MAX DOPA stage value is 1.5, as the servers’ use the shorter observing period when the smaller MAX DOPA stage value is applied. However, when the MAX DOPA stage value is 1.45, the number of successful transmissions sharply decreases, as the number of dropped messages is increased, as shown in Figure 3. As a result, the appropriate MAX DOPA stage is 1.5 in the case of this simulation.

To investigate the impact on the number of devices, we change the number of devices from 150 to 120 in this simulation. As shown in Figure 4, the maximum number of successful transmissions is obtained when the MAX DOPA stage value is set to 1.2. In addition, in Figure 5, the number of dropped messages sharply increases when the MAX DOPA stage is set to 1.1. Compared to the previous simulation, the appropriate MAX DOPA stage value decreases, as the decreased number of devices reduces the total number of transmissions. From these results, we can see that the decreased number of devices results in the decreased appropriate MAX DOPA stage value.
From the results of Figures 6 and 7, we can obtain the appropriate MAX DOPA stage value when the observing period decreases from 10ms to 5ms. In this scenario, the appropriate MAX DOPA stage value is 3.0, since the short observing period generates significant traffic in the network, which causes buffer overflow of the client. To investigate the impact of the clock rate, we change the clock rate from 10MHz to 12MHZ. Figures 8 and 9 show that the high processing speed decreases the appropriate MAX DOPA stage value. Specifically, the appropriate MAX DOPA stage value decreases from 1.5 to 1.25 compared to the first simulation.

Figures 10 and 11 shows the number of successful transmissions and the number of dropped messages when the payload size increases from 125bytes to 200bytes. Compared to the other simulations, this simulation result exhibits the minimum number of successful transmissions, as the observing period increases as the payload size increases. In this scenario, the appropriate MAX DOPA stage value is 2.4.
where \( N \) is the number of devices, \( T \) is the expected number of transmissions per second when the initial observing period is used, \( P_S \) is the payload size per transmission by one CoAP server, and \( CR \) is the clock rate of the client.

\[ \frac{N \cdot T \cdot P_S}{CR} = \text{Appropriate MAX DOPA stage value} \] (1)

5. Conclusion

In this paper, we conducted experimental simulations to find the appropriate DOPA stage value for various system environments. We changed the various parameters in each simulation to consider the various system environments. The simulation results showed that the appropriate DOPA stage value increases when the number of devices or payload size increases. On the other hand, it decreases when the observing period or the clock rate increases. By using these results, we defined the correlations between parameters and appropriate DOPA stage value to find the appropriate DOPA stage value easily.

6. Acknowledgements

This work was supported by NRF (National Research Foundation of Korea) Grant funded by the Korean Government (NRF-2016-Fostering Core Leaders of the Future Basic Science Program/Global Ph.D. Fellowship Program) (2016H1A2A1908620).

7. References


8. Hartke K. Observing resources in the constrained application protocol (CoAP), RFC 7641. 2015 September.