CoAP Sensor Streaming Using Buffer Control
draft-jhjung-core-sensor-streaming-00.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on April 22, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Abstract

As the Internet of Things (IoT) technology grows with the development of wireless communication and sensors, many people are interested in Constrained Application Protocol (CoAP), which is the representative protocol of the IoT. In addition, attempts have been made to apply CoAP to sensors that support existing real-time streaming services. However, the CoAP is not suitable for services to support streaming services such as smart band and CCTV. To overcome this drawback, there is an extension called CoAP Observe, but streaming services using CoAP Observe imposes a load on the server, which is not suitable for environments where low power devices act as servers, such as data transfer between sensors and gateways. In this specification, we are considering the situation in which the sensor acts as a server, and in this environment, we define one mechanism to provide efficient streaming service.

Table of Contents

1. Introduction ................................................ 2
   1.1. Background ............................................. 2
   1.2. Terminology ........................................... 2
   1.3. Overview of the proposed scheme .................... 2
2. The Buffer-Control Option ................................... 7
   2.1. Buffer-Control Option meaning in request .......... 7
   2.2. Buffer-Control Option meaning in a notification ..... 8
3. Subject Side Operation ...................................... 8
   3.1. Register ............................................... 8
   3.2. Caching ............................................... 8
   3.3. Change the buffer size ................................ 9
   3.4. Unregister ............................................ 9
4. Observer Side Operation ..................................... 9
   4.1. Register ............................................... 9
   4.2. Buffer Control ....................................... 9
   4.3. Reordering ........................................... 9
5. Security consideration ...................................... 10
6. IANA Considerations ....................................... 10
7. References ................................................ 10
   7.1. Normative References .................................. 10
Authors’ Addresses ........................................... 10
1. Introduction

1.1. Background

The Constrained Application Protocol (CoAP) is the most widely used protocol in constrained environments such as sensor networks. Along with the growth of the Internet of Things services (IoT), a variety of real-time streaming services (e.g., CCTV) are provided by using the CoAP.

However, how to effectively use CoAP for real-time streaming services is still under study. Since the CoAP has been basically designed to support RESTful services, it may not be suitable to use for streaming services. To overcome these drawbacks, the CoAP Observe Extension [RFC7641] has been studied. This extension supports the well-known CoAP observer design pattern. But, In case that a sensor acts as a server, the extension needs to be enhanced, because a lot of loads may be given to the server.

In this document, the CoAP Observe extension [RFC7641] will be extended to support real-time streaming services so as to deal with packet error appropriately and to reduce the load on the sensor. To achieve this goal, a new option "Buffer-Control" of the CoAP Observe extension is additionally defined. This option is used to control the buffer in sensor for real-time streaming service.

1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Overview of the proposed scheme

The main purpose of this document is to propose a new mechanism for real-time streaming. This mechanism will be effective when the sensor device periodically transmits the sensing data to the gateway, such as CCTV or smart band. In this document, a sensor device acts as a CoAP server, and a gateway acts as a client that requests data transmission to the sensor. Our description will be based on the terms defined in RFC 7641. The entity that transmits data periodically is called the subject, and the entity that receives the data is called the observer.
The receiver who wants streaming service MUST go through the registration procedure. The receiver sends a POST message to request the creation of a streaming service to the subject. Various parameters for the streaming service (e.g., authentication information, the interval between messages, or the buffer size) are transmitted with the POST message. The subject receiving the message checks the parameters and returns a 4.xx error code with an error message if it cannot create a resource for streaming. When the resource is successfully created, the subject returns the URL of the generated resource. The observer then issues a GET request with the Observe Option to the received URL, and if the resource representative is normally received, the registration procedure for the streaming service is completed. Figure 1 shows an example flow of registration process.
When the registration process is completed, the subject periodically transmits the sensing information to the observer, until it receives a DELETE message or until the buffer is full. The subject transmits only non-confirmable messages and stores them in the buffer. If an observer receives the message, it performs reordering with Observe option value as a sequence number. Then, it transmits a PUT message to clear the buffer to avoid that the sensor buffer is full. A PUT message is transmitted for each number of messages of ‘sensor buffer
size / 2 - 1’. For instance, if the buffer size is 8, when the observer receives 3 messages from the subject, it transmits a PUT message. Therefore, two PUT messages can be transmitted before the buffer is full, and even if one PUT message is lost, the transmission of the message continues without block time. The PUT message MUST include Buffer-Control option. The last sequence number received as option value until now The PUT message MUST be confirmable. The Subject receiving the PUT message deletes messages from the buffer which have a smaller sequence number than the sequence number included in the message. In other words, the PUT message acts as a cumulative ACK of TCP.

Figure 2 shows a simple example of transferring data. If a message sent by a subject is lost, the observer can send a GET request to request a message of a specific sequence number. The GET request MUST include Buffer-Control and the positive integer number meaning the specific sequence number as. The observer can also stop the streaming service by sending a DELETE message, and the subject deletes the URL when it receives a delete.
Figure 2: Transferring Data Example
2. The Buffer-Control Option

The Buffer-Control option has different meanings, depending on the request’s method, Option Value, whether it is included in the request for the message, or included in the response.

2.1. Buffer-Control Option meaning in request

Table 1 shows the meaning of the options depending on the method.

<table>
<thead>
<tr>
<th></th>
<th>Option Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Positive Integer</td>
<td>Sequence number of message to request</td>
</tr>
<tr>
<td>PUT</td>
<td>0</td>
<td>Buffer size to change</td>
</tr>
<tr>
<td>PUT</td>
<td>Positive Integer</td>
<td>Sequence number of the last message</td>
</tr>
<tr>
<td>POST</td>
<td>0</td>
<td>Buffer size to change</td>
</tr>
</tbody>
</table>

TABLE 1: Meaning of Buffer-Control Option

When included in a GET request, the Buffer-Control Option identifies the specific sequence number of the message. A GET request with the Buffer-Control option is a message that the observer transmits to the subject to request a message that was lost during transmission. In this case, the Buffer-Control option indicates the sequence number of the message to be re-requested, and the option value is a positive integer.

When the option is included in a PUT request, the Buffer-Control option has a different meaning depending on the Option value as follows:

If the option value is 0, the Buffer-Control option is used to change the buffer size. The buffer size to be changed is included in the payload.

If the option value is a positive integer number, the Buffer-Control option is used to empty the subject’s buffer. In this
case, the option value includes the sequence number of the recently received message, and the Buffer-Control option is used like TCP's cumulative ack. The subject that receives the message containing this option deletes messages whose sequence number is less than or equal to the sequence number contained in the option value of the Buffer-Control option in its buffer.

The Buffer-Control option is included in the PUT message and has the same meaning as when the option value is 0. In POST messages, the Buffer-Control option has an option value of only 0.

2.2. Buffer-Control Option meaning in a notification

When the Buffer-Control option is included in a notification sent from the subject to the observer, that option indicates the size of the buffer that is left over.

3. Subject Side Operation

3.1. Register

The subject who supports streaming service MUST have the resource to make a resource for streaming service. In this specification, the URL of the resource is "/streaming". Also subject MUST be able to handle CoAP Observe Option.

3.2. Caching

The Subject MUST have a buffer to support Error Control. The buffer size is determined according to the parameters included in the POST message when the POST message for creating the resource for the streaming service is received.

3.3. Change the buffer size

The Subject MUST have a buffer for error control and be able to change its buffer size. The buffer size is very important in this mechanism. If the size of the buffer is large, the number of PUT messages issued to empty the buffer can be reduced. In addition, the subject maintains the message transmission until the buffer is exhausted without considering other factors, so the observer can perform congestion control by changing the buffer size of the subject. The buffer size is transmitted until the buffer is full and the transmitted message is stored in the buffer. Through the PUT message containing the Buffer-Control option received from the Observer, the messages confirmed to arrive well are deleted from the buffer.
3.4. Unregister

Since this mechanism basically applies between the sensor and the gateway, the resources created for the streaming service can be observed by only one Observer. Therefore, when a DELETE message is received from the Observer, the corresponding resource should be deleted and the buffer allocated for the streaming service should be released.

4. Observer Side Operation

4.1. Register

An observer who wants streaming service MUST request the creation of a resource for receiving the streaming service. In the extension defined in this specification, a resource for streaming is generated and serviced. Therefore, the observer MUST generate a streaming resource by transmitting a POST message to the subject containing various parameters such as the data resource to be streamed, the buffer size, and the interval between messages. When the resource is successfully created, it receives the resource URL to be streamed from the subject.

4.2. Buffer Control

For the stability of streaming services, the observer should remove the received message from the subject’s buffer. This mechanism minimizes the burden on the subject as much as possible, so the observer MUST be able to manage both the error control and the state of the subject buffer. In addition, it MUST be able to change the subject’s buffer size based on network conditions. For example, if the error rate is high, it is possible to control the error by transmitting the PUT message more frequently by reducing the buffer size. If the error rate is low, the buffer size can be increased to reduce the number of control messages issued.

4.3. Reordering

Since the CoAP message is basically UDP-based, the transmission order and reception order of messages may be different. Therefore, the observer MUST be able to re-order the message using the sequence number included in the CoAP Observe Option.
5. Security consideration

The security consideration will apply to Section 11 of [RFC7252], the CoAP specification, and Section 7 of [RFC7641], Observing resources in the CoAP.

6. IANA Considerations

TBD

7. References

7.1. Normative References


Authors’ Addresses

Joong-Hwa Jung
Kyungpook National University
Daehakro 80, Bukgu, Daegu, South Korea 41566

Email: godopu16@gmail.com