CoAP Streaming
draft-loreto-core-coap-streaming-00

Abstract

This specification defines a simple mechanism for streaming media data from a server to a client in a constrained network over CoAP. The mechanism take advantage of the observer design pattern described in CoAP, extending it, to support streaming media transfer between nodes.

Note

Discussion and suggestions for improvement are requested, and should be sent to core@ietf.org.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 28, 2012.

Copyright Notice

Copyright (c) 2012 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
Table of Contents

1. Introduction .................................................. 3
   1.1. Terminology ............................................... 3
2. COAP Streaming Extension ..................................... 3
   2.1. Definition ............................................... 3
   2.2. CoAP Streaming negotiation .............................. 6
3. Security Considerations ....................................... 7
4. IANA Considerations ........................................... 8
5. Acknowledgements ............................................. 8
6. References .................................................... 8
   6.1. Normative References .................................... 8
   6.2. Informative References .................................. 8
Authors' Addresses ............................................... 9
1. Introduction

The Constrained Application Protocol (CoAP)[I-D.ietf-core-coap] is a specialized web transfer protocol used with constrained networks and nodes for machine-to-machine applications.

CoAP has a number of extensions. One of its extensions is the "Observing Resource in CoAP" draft [I-D.ietf-core-observe]. The "Observing Resource in CoAP" draft defines a mechanism to push resource representations from servers to interested clients.

However, the CoAP specification or any of its extensions do not define any mechanism to transfer media information between the nodes. If a node in a constrained network wants to transfer some streaming media information with any other node, it does not have any specific mechanism to do so.

The main purpose of this document is to extend the "Observing Resource in CoAP" draft [I-D.ietf-core-observe] to support streaming media transfer between the nodes.

In addition, this specification adds a new option Streaming to the Constrained Application Protocol (CoAP). The main purpose of this option is to indicate when a message will break into chunks of known size.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. COAP Streaming Extension

2.1. Definition

The CoAP Streaming mechanism extends the "Observing Resource in CoAP" draft [I-D.ietf-core-observe]. The basic life cycle of an application using CoAP streaming is as follows:

- The client registers itself with a resource by performing a GET request that includes an Observe Option. The registration process is defined in the "Observing Resource in CoAP" draft [I-D.ietf-core-observe].
If the observation relationship is established between the server and the client, the server sends a CoAP streaming response to the client, including the Observe Option, whenever some new media chunk is available.

A CoAP streaming response includes the following options:

- The Observe and Token defined in the "Observing Resource in CoAP" draft [I-D.ietf-core-observe].
- A New Option header called 'streaming' that indicates the message will break into chunks of known size.
- End of File (EOF) that indicates the end of the transmission of the media. This value will help to differentiate between a connection terminated by a fault and one that is correctly terminated.

The following example includes a media chunk in the observation message:

```
2.05 Content
  Observe: 12
  Token: 0x4a
  Streaming: chunked
  Payload:
    This is the data in the first chunk
```

Figure 1: Transfer response

The CoAP streaming mechanism is based on the capability of the server to send several pieces of information in different responses.

Figure 2 illustrates the media communication between client and server.
A notification containing a chunk can be confirmable or non-confirmable (i.e. sent in a confirmable or non-confirmable message). However, all the notifications using the mechanism defined in this draft MUST use non-confirmable messages.

The server MUST set the value of the Observe Option in each notification to the 16 least-significant bits of a strictly increasing sequence number that MUST contain no gap. If there is no gap in the Observe Option between two sequence notification a client is receiving, no chunk has been lost or delayed by the network.

Note that, differently from what specified in Section 4.4 of [I-D.ietf-core-observe], a server can send more than one notification
per second per client, token and resource. The server can then happen to reuse the same option value with the same client, token and resource within approximately $2^{16}$ seconds (roughly 18.2 hours).

Since CoAP runs over UDP, chunks (i.e. notifications) can arrive to the client in a different order than they were sent or a chunk can be lost. The ordering of the chunks is determined by the Observe option that, in this case, can be seen as similar to the "sequence number" of the RTP [RFC3550]. It is left to the application to choose the appropriate action, if any, when it detects that a chunk is missing.

A notification containing a chunk is made up of the bytes after the options, if any; its maximum length SHOULD be 1280 bytes.

The chunks are encoded as "text/plain; charset=utf-8" setting the Content-Type option to 0. How the media is translated to text format is out of the scope of this document.

2.2. CoAP Streaming negotiation

Figure 3 illustrates a client discovering the different codecs a resource (i.e. videocamera) on a server supports, using the Core Resource Discovery mechanism [I-D.ietf-core-link-format].

The client then chooses a proper code to observe selecting the right URI.
3. Security Considerations

This presents no security considerations beyond those in section 8 of the Observing Resources in CoAP specification [I-D.ietf-core-observe]
4. IANA Considerations

The IANA is requested to add the following "CoAP Option Numbers" entry as per Section 11.2 of [I-D.ietf-core-coap].

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>Streaming</td>
<td>Section 2</td>
</tr>
</tbody>
</table>

5. Acknowledgements

The authors of this draft would like to thank Heidi-Maria Rissanen.

6. References

6.1. Normative References

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)",
draft-ietf-core-coap-09 (work in progress), March 2012.

[I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP",
draft-ietf-core-observe-05 (work in progress), March 2012.


6.2. Informative References

[I-D.ietf-core-link-format]
Shelby, Z., "CoRE Link Format",
draft-ietf-core-link-format-11 (work in progress), January 2012.
Authors' Addresses

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: salvatore.loreto@ericsson.com

Oscar Novo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: oscar.novo@ericsson.com