Abstract

This document defines shim header for the transfer of CoAP messages over non-TCP/IP constrained networks. In this environment, IP and UDP or TCP are not used, so that additional shim header as a container for addresses of sender/receiver and the length of CoAP header and its payload is required.

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). 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 April 14, 2014.

Copyright Notice

Copyright (c) 2013 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 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.
# Table of Contents

1. Introduction .................................................. 4  
2. Terminology .................................................. 4  
3. Shim Header .................................................. 5  
   3.1. Header Format ........................................... 5  
   3.2. Distinguishing between CoAP Header and Shim Header..... 6  
   3.3. Example ................................................ 7  
4. Security Considerations ...................................... 9  
5. IANA Considerations ........................................ 9  
6. Acknowledgments ............................................ 10  
7. References ................................................ 10  
   7.1. Normative References ..................................... 10  
   7.2. Informative References .................................. 10
1. Introduction

CoAP [CoAP] is a data transfer protocol over constrained nodes and networks, e.g., 6LoWPAN. This protocol uses TCP/IP suit and some other additional protocols designed for low-power and lossy networks (LLN) like Routing Protocol for LLN (RPL) [RPL].

Nowadays many kinds of sensor are applied in variable areas like industry, building management, security service, environmental monitoring and etc. Many of sensor manufactures are still reluctant to use TCP/IP stack. TCP/IP still seems to be heavy to be applied to the constrained node. Instead, they use various vendor-specific transfer protocols over various media like IEEE 802.15.4, RS485, CAN, and RS232 (or UART). This is a big hurdle for CoAP to be prevalent.

In order to make it easy to apply CoAP to these non-TCP/IP constrained node, we need to define new very simple header which mimic IP header. But there are two restrictions on applying CoAP to the non-TCP/IP nodes. The first is the lack of address. Peer-to-peer-type media like UART and RS232 has no address to identify nodes. RS485 has an integer value as an identifier, but there is no standardized way to present or carry this value over networks. IEEE 802.15.4 has network address in their standard, but this address is allocated by a coordinator dynamically. 64-bit-long extended address is too long and sometimes not used.

The second restriction is that CoAP does not have PDU size information in its header. CoAP calculates the PDU size with the information from underlying protocol layer like the payload size in IP header.

Therefore, in order to apply CoAP to non-TCP/IP nodes, new header for CoAP must be defined to complement these two restrictions. In this document, new Shim Header is specifies as a underlying protocol for CoAP, and an example is added to explain this shim header.

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 [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.
Shim Header

In front of a CoAP header, a slim header is inserted instead of IP header and this contains addresses of CoAP sender and receiver and the length of CoAP header and its payload.

Local ID

A local ID is an address of CoAP sender or receiver, and it is uniqueness in a vendor-specific non-TCP/IP constrained network. This ID has a 16-bit-long hexadecimal value, that is IEEE 802.15.4 network address compatible, and it can be extensible later.

CoAP Proxy

A CoAP proxy in this document is a CoAP-to-CoAP proxy from [CoAP]. In additional, this CoAP proxy can read and insert a shim header from/to the front of CoAP header.

Additional terminology for CoAP can be found in [CoAP].

3. Shim Header

3.1. Header Format

<table>
<thead>
<tr>
<th>Pre</th>
<th>Ver</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoAP Src Local ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoAP Dst Local ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1 Shim Header Format

Fig. 1 shows a basic format of shim header. Next is the explanation of each field in the header.

* Pre: it is abbreviation of Preamble, and a 4-bit-long field, and has a meaning of the start of shim header. ’0xa’ is used currently.
Ver: it is abbreviation of Version, and also a 4-bit-long field. Currently the version number is '1'.

Length: it is the length of CoAP header and its payload followed by the shim header. This has a value between 1 and 255.

CoAP Source Local ID, CoAP Destination Local ID: are the local IDs of the sender and receiver of its CoAP message. Each field has 16-bit field.

Fig. 2 shows the extended format of shim header. This format is used when the length of CoAP header and its payload is larger than 255. In order to have an extended length field, the value of Length field is set to '0', which means the next field is not CoAP source ID, but 16-bit extended length field. This field can have a range of value from 0 to 65535.

```
+----+----+---------+
|Pre |Ver | 0x00    |
+----+----+---------+
    | Ext. Length |
+-------------------+
    | CoAP Dst Local ID |
+-------------------+
    | CoAP Dst Local ID |
+-------------------+
Figure 2 Shim Header Format with Extended Length
```

3.2. Distinguishing between CoAP Header and Shim Header

According to the type of constrained nodes, Shim header can be hardcoded, or not be involved in the CoAP node implementation. But at the following cases, CoAP and shim header must be distinguished when a packet receiver of a node meets a start point of some header.

* The CoAP receiver has multiple heterogeneous media, and is not able to identify the received interface among them. And some media use shim header on their interfaces and the others are not.

* The CoAP receiver receives CoAP messages from two or more other CoAP nodes.
As a result, the start point of shim header must be differentiated from it of CoAP header. According to the CoAP draft, CoAP header starts with version and message type, which has a range of values from 0x04 to 0x07. But in the case of shim header, it starts with preamble field, and its value is '0xa' to identify shim header against COAP header.

3.3. Example

As shown in [CoAP], the first example in Appendix A illustrates a basic confirmable CoAP request and response in piggy-backed manner. In this section, we add shim header to this example. In this case, the client and the server are connected via non-TCP/IP. The client can be a data collector node or a proxy node. The server may be a temperature sensor. In this example, we assume that IDs of the client and the server are 0x0000, and 0x0001 respectively.
Client   Server
(0x0000) (0x0001)

Shim: Length=16, Src=0x0000, Dst=0x0001
Header: GET (T=CON, Code=0.01, MID=0x7d34)
Uri-Path: "temperature"

Shim: Length=11, Src=0x0001, Dst=0x0000
Header: 2.05 Content (T=ACK, Code=2.05, MID=0x7d34)

Payload: "22.3 C"

Figure 3 Message flow and Shim header
4. Security Considerations

The shim header for the transfer of CoAP messages is not intended to use TCP/IP protocol suits. Fundamental motivation of the approach is that lots of things (i.e. sensors and actuators) used in current operation for several service domain as well as to be used as nodes connected with Internet to build an IoT network frequently utilize vendor-specific protocols instead of TCP/IP protocols. Therefore, several TCP/IP based security mechanisms cannot be used directly for the approach in this draft. These are IKEv2, IPSec, DTLS, HIP and others considered in [SecCon]. In particular, security binding to DTLS, which is one of main features of CoAP, cannot work in a service domain using shim header.

Instead, vendor specific secure protocols can be applied. Several security protocols that can be standard solutions (e.g. security architecture for ZigBee or EAP for IEEE 802.15.4) or proprietary solutions have been proposed for several transmission media. Also, development of secure scheme running in application layer can be an alternative solution.

To use shim header in IoT, this draft considers heterogeneous networks using different protocols suited for their transmission media. A proxy box is required for the scenario. A CoAP-Shim proxy supports protocol translation for allowing a node used shim header to communicate with CoAP enabled nodes.

In case of using a proxy, end to end security and privacy are major concerns from the aspect of security. If the proxy is infected or spoofed, all messages translated by the proxy are simply eavesdropped, modified, replayed, selectively forwarded and/or falsely delivered. These attacks can subsequently lead to more serious threats. Several vulnerabilities and possible threats to the CoAP have been well described in [CoAP] and [SecCon]. Especially, section 11.2 of [CoAP] presents security issues introduced by using proxy. In addition, availability of proxy must be guaranteed even though computing power and memory space are better than those of resource constrained nodes in IoT. It is naturally obvious strategy that an attacker sends flooded false packets to a proxy, thereby launching denial-of-service attacks since a death of proxy results in death of network used shim header.

5. IANA Considerations

(TBD)
6. Acknowledgments

(TBD)

7. References

7.1. Normative References


7.2. Informative References


Author’s Addresses

Seung-Hun Oh
ETRI
1000-6 Oryong-dong, Buk-gu, Gwangju, 500-480, Korea
Phone: +82-62-970-6655
Email: osh93@etri.re.kr

Shimkwon Yoon
ETRI
1000-6 Oryong-dong, Buk-gu, Gwangju, 500-480, Korea
Phone: +82-62-970-6969
Email: yoonsk@etri.re.kr

Byung-Tak Lee
ETRI
1000-6 Oryong-dong, Buk-gu, Gwangju, 500-480, Korea
Phone: +82-62-970-6624
Email: bytelee@etri.re.kr

Ilkyun Park
M2Soft
27 Seongsuiro7gil, Seongdong-gu, Seoul, 133-827, Korea
Phone: +82-2-2188-8558
Email: ikpark@m2soft.co.kr

Namhi Kang
Duksung Women’s University
419 Sangmoon-dong, Dobong-gu, Seoul, 132-714 Korea
Email: kang@duksung.ac.kr
URI: http://www.duksung.ac.kr