Constrained RESTful Environments
Internet-Draft
Intended status: Informational
Expires: December 19, 2015

T. Carey
Alcatel-Lucent
June 17, 2015

Standard Primitives versus Transport Specific Adaptation
draft-carey-core-std-msg-vs-trans-adapt-00

Abstract

This draft discusses the need for a consistent messaging layer that can be used but the transport protocols as they adapt to the CoAP Request/Response layer. In addition, this draft provides comments to the TCP transport implementaton described by [I-D.tschofenig-core-coap-tcp-tls].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In review of [I-D.tschofenig-core-coap-tcp-tls], we realized that this draft:

- Didn’t support the CoAP message layer’s use of ACK/RST in CON and NON message types or the message-id. In fact, the draft explicityly removed support for the CON message types and didn’t support CoAP ACK mechanisms - relying on the TCP ack/rst/fin messages and timeout mechanisms.

- Didn’t explicitly discuss how piggy backed responses would be handled.

- Made the assumption that the Blockwise protocol was supported but did not describe how Blockwise would be supported within the concep of TCP connections.

- Didn’t explicitly discuss how TCP connections related to the higher layer Request-Response/Observe-Notify and the newer Publish and Subscribe message exchange patterns.

2. Confusion in the CoAP Message Layer

RFC 7252 described a Message Layer to allow for Confirmable/Non-confirmable delivery of Request/Response messages. The unstated purpose of this message layer was that it was to be used for unreliable transports (e.g., UDP, SMS). Several drafts (e.g., Observe [I-D.ietf-core-observe], Block [I-D.ietf-core-block]) and standards groups (e.g., OMA, oneM2M) have referred to the Message Layer (Adaptation Layer) primitives (e.g., CON, NON, ACT, RST, Message id) in their processing. As such the interface between the Application and Request/Response Layer was assumed to extend into the primitives offered by the Adaptation Layer. Subsequent clarifications of the Application Layer interaction was provided that

Applications (e.g., LWM2M Clients and Servers) interact with the CoRE Application Features and/or the underlying Request/Response Layers.

The following Figure depicts the CoAP Layers with the initial set of Transport protocols and CoAP Features.

```
   .----------------------------------------.  -.
   |             Applications               |   |
   |             (LWM2M)                    |   |
   |             .--------------------.     |   | Application
   |             |   Group  |Resource |     |   | Layer
   |             |          |Directory|     |   |
   |   .---------+----------+---------+-----|  -+
   |   |         |  Request/Response        |   | Request/
   |   |  Block  |  Observe/Notify          |   | Response
   |   |         |  Publish/Subscribe       |   | Layer
   +---+---------+--------------------------'  -+
   | Unreliable Transport |                     |
   | Message Layer        |                     | Adaptation
   | (NON-CON)            |                     | Layer
   |                      |                     |
   |      .--------.      |                    -+
   |      |  DTLS  |      |                     |
   +------+--------+------+                     | Transport
   |   UDP    |    SMS    |                     | Layer
   |          |           |                     |
   '----------+-----------'                    -'
```

Figure 1: CoAP Layers

While the clarification of the Application Layer provided an explaination of how Applications should interact with the Request/Response Layer, the discussion highlighted an additional problem. There isn't a single consistent interface between the Adaptation Layer and the Request/Response Layer. This consistency was lost when new Transport protocols did not implement the message primitivies (e.g., CON/NON, ACK, RST) of the UDP/SMS Messaging Layer.

The following Figure depicts the CoAP Layers with the new set of Transport protocols.

```
   .--------------------------------------------------.  -.
   |             Applications (LWM2M)                 |   |
   |             .-------+-----------.                |   | Application
   |             | Group | Resource  |                |   | Layer
   |             |       | Directory |                |   |
   |     .-------+-------+-----------+----------------+  -+
   |     |       |  Request/Response                  |   | Request/
   |     | Block |  Observe/Notify                    |   | Response
   |     |       |  Publish/Subscribe                 |   | Layer
   |     |       |                                    |   |
   '-----+-------+------------------------------------'  -'
   ==================================================== No Standard
   ==================================================== Primitives
   .----------------------.   .---------.   .---------.  -.
   | Unreliable Transport |   | Message |   | Message |   |
   | Message Layer        |   | Layer   |   | Layer   |   | Adaptation
   | (NON-CON)            |   | NON     |   | Web     |   | Layer
   |                      |   |         |   | Socket  |   |
   |      .--------.      |   |-----.   |   +---------+  -+
   |      |  DTLS  |      |   | TLS |   |   | Web     |   |
   +------+--------+------+   |-----+---|   | Socket  |   |
   |   UDP    |    SMS    |   |   TCP   |   |----.    |   | Transport
   |          |           |   |         |   |TLS |    |   | Layer
   '----------------------'   '---------'   |----+----|   |
                                            |   TCP   |   |
                                            '---------'  -+
```

Figure 2: CoAP Layers (New Transports)

3. Standard Primitives vs Transport Specific Adaptation

If a standard set of primitives were used, each Transport protocol would document how to implement the CON and NON messages with ACK and RST responses. The Request/Response Layer feature would describe how to adapt timeouts and state processing of the Message Layer. This would provide for a clean delineation of responsibility such that developers of new Transport protocols and Request/Response features would know exactly what the behavior is that is provided and consumed by each layer (i.e., Transport, Request/Response).

The following Figure depicts the CoAP Layers with a standard Message Layer.

```
   .--------------------------------------------------.  -.
   |             Applications (LWM2M)                 |   |
   |             .-------+-----------.                |   | Application
   |             | Group | Resource  |                |   | Layer
   |             |       | Directory |                |   |
   |     .-------+-------+-----------+----------------+  -+
   |     |       |  Request/Response                  |   | Request/
   |     | Block |  Observe/Notify                    |   | Response
   |     |       |  Publish/Subscribe                 |   | Layer
   |     |       |                                    |   |
   +-----+-------+------------------------------------+  -+
   |                     Message Layer                |   |
   |                     (NON-CON)                    |   | Adaptation
   |                                                  |   | Layer
   |                                                  |   |
   |      .--------.      .---+----.    .---+---------+  -+
   |      |  DTLS  |      |   |TLS |    |   | Web     |   |
   |------+--------+------|   +----+----+   | Socket  |   |
   |   UDP    |    SMS    |   |   TCP   |   +----+    |   | Transport
   |          |           |   |         |   |TLS |    |   | Layer
   '----------+-----------'   '---------'   +----+----+   |
                                            |   TCP   |   |
                                            '---------'  -'
```

Figure 3: CoAP Layers - Standard Primitives

If transport specific adaptation is used, the transport protocol would specify how the Request/Response layer exchange patterns and features would be adapted by the protocol. This will become very difficult to maintain as each new feature that needs aspects of a transport protocol will have to also include those aspects such as was done in the Observe draft.

The side-effect of losing the standard set of messaging primitives is that each Transport will have to document how that transport adapts to the various elements of the Request/Response Layer (e.g., Block, Observe, Request, Response) rather than document how they would implement the standard set of messaging primitives. In addition each new Request/Response feature will have to document how it will interact with each Transport Layer.

The following Figure depicts the CoAP Layers with Adaptation Layers specific to each Transport Protocol.

```
   .--------------------------------------------------.  -.
   |             Applications                         |   |
   |             (LWM2M)                              |   |
   |             .--------------------.               |   | Application
   |             |   Group  |Resource |               |   | Layer
   |             |          |Directory|               |   |
   |   .---------+----------+---------+---------------+  -+
   |   |         |  Request/Response                  |   | Request/
   |   |   Block |  Observe/Notify                    |   | Response
   |   |         |  Publish/Subscribe                 |   | Layer
   '---+---------+------------------------------------'  -'

   .----------------------.   .---------.   .---------.  -.
   | Unreliable Transport |   | Message |   | Message |   |
   | Block, Req/Resp      |   | Layer   |   | Layer   |   | Adaptation
   | Observe, Pub/Sub     |   | Block...|   | Block...|   | Layer
   |                      |   |         |   |         |   |
   |      .--------.      |   +----.    |   +---------+  -+
   |      |  DTLS  |      |   |TLS |    |   | Web     |   |
   +------+---+----+----- +   +----+----+   | Socket  |   |
   |   UDP    |    SMS    |   |   TCP   |   +----+    |   | Transport
   |          |           |   |         |   |TLS |    |   | Layer
   '----------+---------- '   '---------'   +----+----+   |
                                            |   TCP   |   |
                                            '---------'  -'
```

Figure 4: CoAP Layers - Transport Specific Adaptation

Another side-effect of not using the existing set of message layer primitives is that Applications MUST be aware of the Transport bearer when invoking requests because they have to set the type of message (CON, NON) because a Transport (e.g., TCP) may not support the message (e.g., CON).

4. Standardize Message Layer

Using the existing CoAP Message Layer as the standard set of primitives allows IETF Drafts that focus on features in the Request/ Response Layer to know what is provided by any Transport protocol. Likewise IETF Drafts for CoRE elements will know th e messages that are needed to be either implemented or provided.

Note: The draft does not suffest Message Layer mechanisms like transport specific timeout processing will be exposed, just the messaging.

The following Figure depicts the message interactions between the CoAP Layers using a standardized Message Layer.

```
   Application Layer

                   REQ (CON, NON)       ^
                       |                |
                       |                |
   |-------------------+----------------+----------------------|
                       |                |
                       |                |
                       v               RSP


   Request/Response Layer
    Request-Response,
    Observe-Notify,
    Block, Pub-Sub

                   Confirmable                 Non-confirmable
                                        |
                                        |
                                        |
           CON      ^       ^       ^   |      NON    ^
            |       |       |       |   |       |      |
            |       |       |       |   |       |      |
   |--------+-------+-------+-------+---|-------+------+-------|
            |       |       |       |   |       |      |
            |       |       |       |   |       |      |
            v    CON-ACK CON-RST CON-TMO        v   NON-RST
          \                             /     \              /
           \        Message-Id         /       \ Message-Id /
            ---------------------------         ------------

   Message Layer
   (Tansport Adaptation)

   |----------------Transport Protocol Specific----------------|

   Transport Layer
   (TCP, UDP, SMS, Websockets)
```

Figure 5: CoAP Layers - Standard Message Layer

5. Issues with the Current TCP Draft

The current draft [I-D.tschofenig-core-coap-tcp-tls], has the following issues:

1. TCP Connections: TCP connections in the current draft are currently limited to a single Request/Response information exchange. This limitation means that multiple TCP connections are needed for parallel information exchanges. For example, an Observe/Notification information exchange would have to be on a different TCP connection as a simple Get request, causing multiple costly TCP connections to be established. In addition, long lived TCP connections could not be supported unless the Application serialized the Request/Response exchange which is difficult with Request/Response features like Observe. As such, modifications to the draft to allow Long TCP connections with multiple Request/Response Information exchanges is needed.

2. Blockwise Transfer: The current draft does not include documentation of how to handle Block transfers especially with the use of the TCP ack and empty messages. Actually the Blockwise transfer draft should be modified to use the Request/ Response terminology instead of the ACK message terms (e.g., Section 3.1 Block2 examples. This is an example of the confusion caused by not having a standardized set of message primitives.

3. Accounting for Request/Response Layer Usage - Observe: The current TCP draft needs to document how to account for:

   * Confirmable messages in the Observe draft (section 1.2, 3.4, 3.6, 4.5, 4.5.1): The use of message ID in non-confirmable messages (section 4.5) and adpatation of congestion control (section 4.5.1). The TCP draft should document how it emulates the behavior of the confirmable messages in each of the sections. For example the use of TCP acks as a replacement for CON message ACKs.

   * Use of Message Id in non-confirmable messages in the Observe draft (section 4.5): Since Message Ids are elided, the draft needs to document how the RST messages for Notifications should be handled unless Message Ids are indeed supported in a future TCP draft.

   * Adaptation of congestion control in the Observe draft (section 4.5.1): The TCP drafts needs to document how congestion control would be done for simultaneous Notifications.

- Use of the Message Id to ensure no duplication through the Request/Response Layer: The TCP protocol will only ensure duplication at the TCP layer. The TCP protocol doesn't prevent an invoking Request/Response layer from sending the message more than once for any reason (good or bad). As such the support of Message ID is still needed as the TCP layer is insufficiant because the solution cannot address possibilities at the Request/Response layer.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

None

8. References

[I-D.ietf-core-block]
Bormann, C. and Z. Shelby, "Block-wise transfers in CoAP", draft-ietf-core-block-17 (work in progress), March 2015.

[I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP", draft-ietf-core-observe-16 (work in progress), December 2014.

[I-D.tschofenig-core-coap-tcp-tls]
Bormann, C., Lemay, S., Technologies, Z., and H. Tschofenig, "A TCP and TLS Transport for the Constrained Application Protocol (CoAP)", draft-tschofenig-core-coap-tcp-tls-03 (work in progress), April 2015.

Author's Address

Timothy Carey
Alcatel-Lucent
TX
US

Phone: +1-972-415-2065
Email: timothy.carey@alcatel-lucent.com