LWIG Working Group
Internet-Draft
Intended status: Informational
Expires: September 27, 2012

A. Castellani
University of Padova

March 26, 2012

Learning CoAP separate responses by examples
draft-castellani-lwig-coap-separate-responses-00

Abstract

This draft aims at providing interesting examples of CoAP separate responses that are useful to aid CoAP implementers on understanding possible rare situation incurring.

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 27, 2012.

Copyright Notice

Table of Contents

1. Introduction

To be done if interest is shown (TBDIIIS).

2. Taxonomy of cases

In the following sections all the possible situations are briefly described.

2.1. Request lost

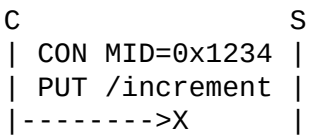

Figure 1: Example of request lost

As shown in Figure 1, this case includes all the situations where the request, including all its retransmissions, has never got through the network up to the server.

2.2. Request ACK lost

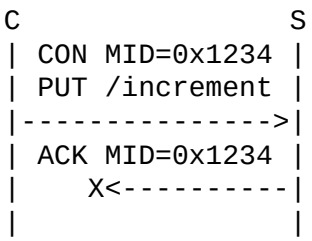

Figure 2: Example of request ACK lost

As shown in Figure 2, this case includes all the situations where the request has got to the server, but the corresponding ACK has never got through the network up to the client.

2.3. Response lost

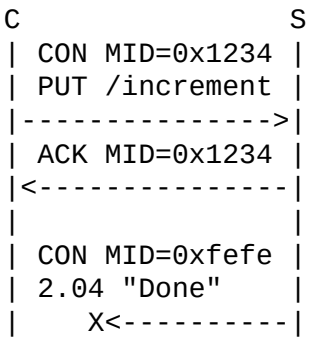

Figure 3: Example of response lost

As shown in Figure 3, this case includes all the situations where the response and its ACK have not been lost, but the corresponding separate response, including all its retransmissions, has never got through the network up to the client.

2.4. Response ACK lost

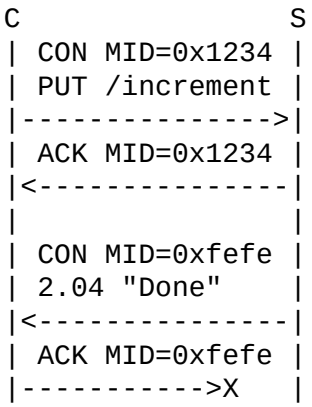

Figure 4: Example of response ACK lost

As shown in Figure 3, this case includes all the situations where the response, its ACK, and the corresponding separate response have not been lost by the network, but the last ACK sent by the client has been lost.

3. Client perspective

In this section interesting situations incurring to a client implementation are discussed.

3.1. No request ACK received

The client cannot distinguish between the first two cases request lost (see Section 2.1) and request ACK lost (see Section 2.1). We call this state C_NO_ACK.

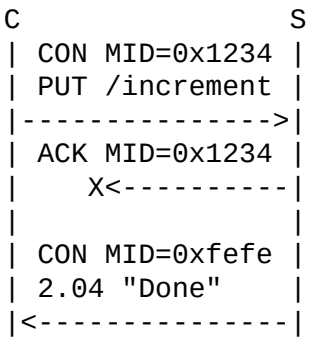

Figure 5: Response without ACK

Figure 5 shows an example where the client is in the C_NO_ACK state and it receives a CON response in the same session (i.e. matching [loc-host, loc-port, rem-host, rem-port, token]). A realistic but optimistic implementation might think that this response is related to the previous not acked request; a pessimistic but paranoid implementation could decide that the response is NOT related to the previous response.

Client implementations supporting only the empty Token (no Token support) are encouraged to randomly select local UDP source port at each new request; this implementation shrewdness smoothly resolves confusion.

Always having the Token Option set to a random value realistically resolves any possible confusion in this case, at the obvious cost of its added complexity in the client implementation and network overhead.

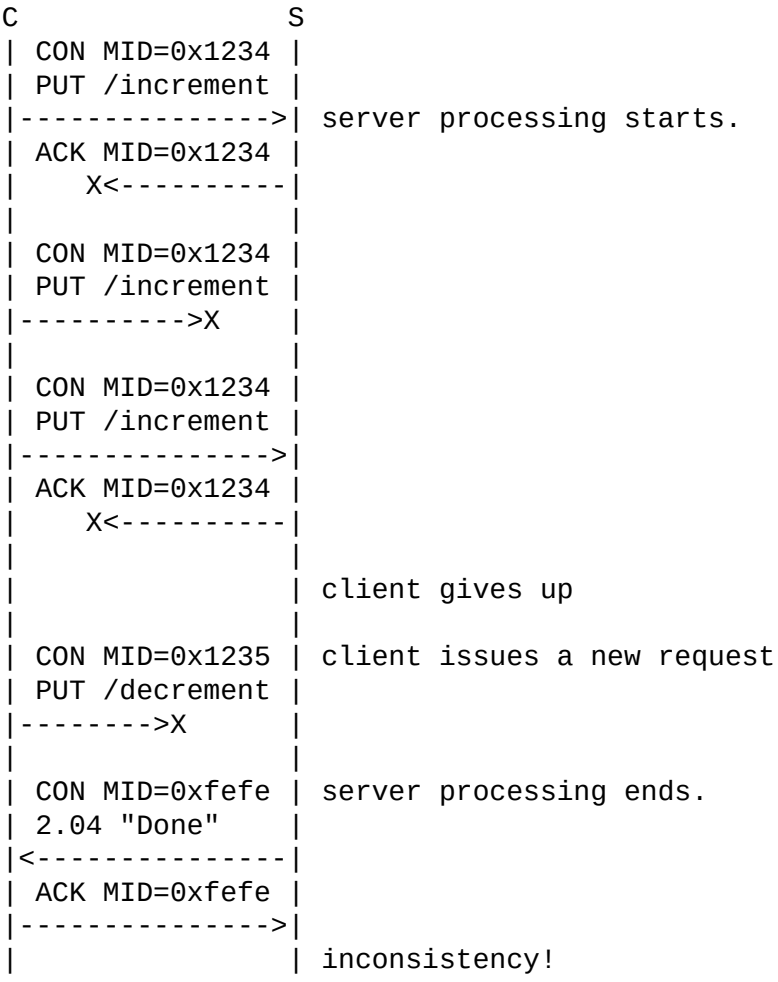

Figure 6: Naive client

Figure 6 shows an incident occurring to a naive client implementation using the empty Token and a static local UDP port. This leads to the indication that a client should in general avoid reusing the same session , i.e., [loc-host, loc-port, rem-host, rem-port, token], even if it has failed.

3.2. Response ACK lost

After sending the ACK to the response, a provident client implementation keeps the request session up for some time. This avoids issuing new requests in the same session (i.e. [loc-host, loc-port, rem-host, rem-port, token]), and allows smoothly responding with an empty ACK to response retransmissions received by the server.

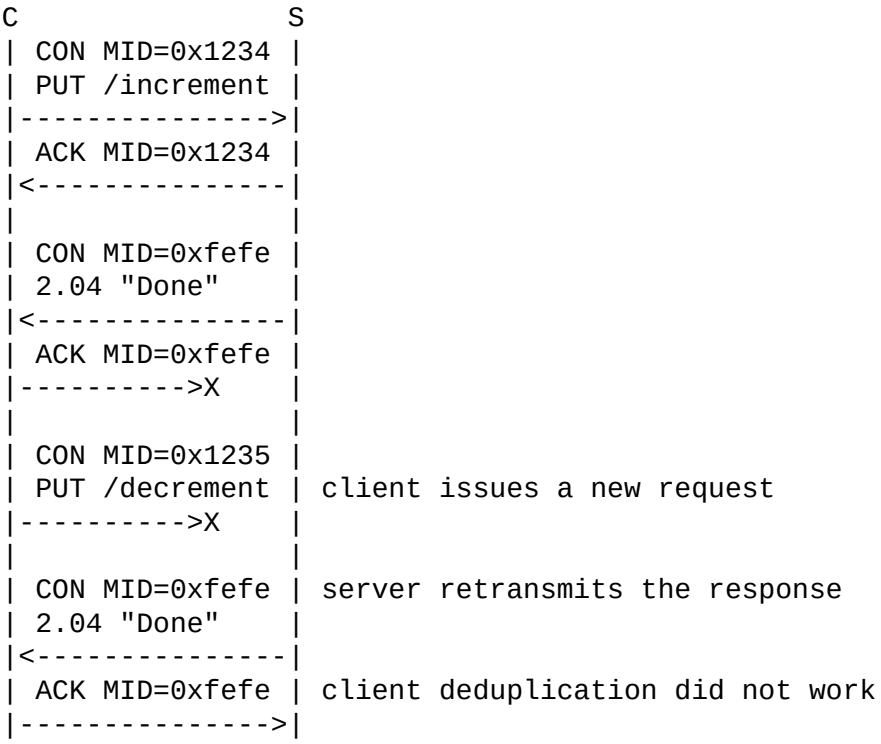

Figure 7: Inexperienced client

Figure 7 shows an incident occurring to an inexperienced client not having robust deduplication in place and reusing the same session.

4. Server perspective

TBDIIIS

4.1. Request ACK lost

TBDIIIS

4.2. Response lost

The server is said to be in S_NO_ACK when no empty ack to the response is received by the server.

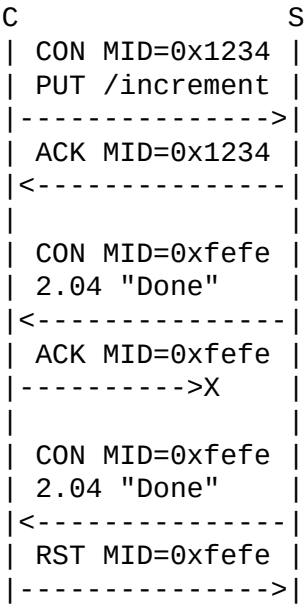

Figure 8: Forgetful client

Figure 8 shows the realistic case of a server in state S_NO_ACK, that deals with a client that do not mantains a successful session open after receiving the response. An optimistic server implementation might think that the client has received the response even if it has replied with a RST.

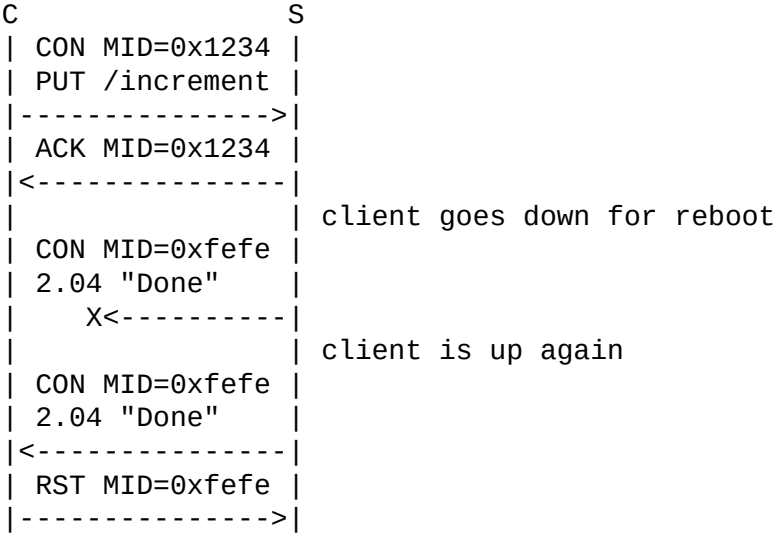

Figure 9: Rebooting client

However as Figure 9 shows that the very same result might be happen if the server is interacting with a rebooting client.

Open issue: Should the server change its behaviour depending on the fact that it received a RST instead of an ACK? (e.g., Should it apply the actual change to the resource only after an ACK to the response has been received?)

5. Acknowledgements

Special thanks to Carsten Bormann for substantial contributions on this topic that significantly aided me in compiling this document, to Mattia Gheda for the long discussion had on this topic, and to Jeroen Hoebeke that publicly started discussion on this topic in the CoRE mailing list.

Thanks to Zach Shelby, Salvatore Loreto and Guido Moritz for helpful comments and discussions that have shaped the document.

6. 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.

Author's Address

Angelo P. Castellani
University of Padova
Via Gradenigo 6/B
Padova 35131
Italy

Email: angelo@castellani.net