CoAP improvements to meet embedded device hardware constraints
Workshop Prague, 25th March 2011

Davide Barbieri
Concept Reply
via Cardinale Guglielmo Massaia 83
10147 – Torino, Italy
+39 011 2911 582
d.barbieri@reply.it

INTRODUCTION

In the scope of a research project conducted in the laboratories of Concept Reply (member of the IPSO Alliance) it is being developed a prototype, based on Contiki[1], with the purpose of creating a wireless sensor network supporting embedded web services.

We are testing different protocols such as uDPWS[2] (a subset of the DPWS standard) and the emerging CoAP/CoRE[3], which has recently gained a lot of attention by researchers in the field of IoT, since it is specifically designed to work with few resources in terms of memory and processing power.

Our focus moved on a recent IETF draft regarding SOAP-over-COAP[4] because of the lightness of CoAP with the complete customization of the SOAP protocol to describe web service applications.

Besides web services, we are developing different logical models in SOAP, depending on the resource and device type, which can be used even with DPWS.

This position paper describes two possible approaches to solve two problems we are currently tackling with CoAP protocol experiments: maximum message length notification and subscription option for energy saving.

MAXIMUM MESSAGE LENGTH NOTIFICATION

In the last release of Contiki (v. 2.5) it has been implemented the CoAP protocol as specified in the draft-ietf-core-coap-03 [5]. It is also present a demo application both for the server side and the client one.

In this demo it is estimated a very small amount of information (at maximum 120 bytes) to transmit to the client about the hosted services. Our logical model is considerably larger (about 1 kB) and for this reason buffer limitations could be a significant problem to exchange data.

The IPv6 minimum MTU is 1280 bytes, whilst the maximum MTU for IEEE 802.15.4 networks is 127 bytes. The 6LoWPAN adapting layer, natively supported in Contiki, provides an efficient fragmentation method to allow the transmission of larger packets, as we need in our applications.

During our deployment experience with TelosB devices, we realized that packet size is sometimes a big problem because if the received message does not fit in the input buffer it could cause unpredictable effects on the receiving side, e.g. operating system restarting because of buffer overflow.

For this reason, it can be useful to specify during the resource discovery sequence (client sends a GET *.well-known/core*) the maximum accepted length of the response message with the resource description (in our case the SOAP logical model). The maximum packet size can be written in a new *option header* of the CoAP message. In this way continuous request transmissions (because of lost response messages due to the excessive length) can be avoided.

Moreover, if the receiving device simply discards input message because it exceeds the maximum accepted length, it can send an ACK message in response with a specific bit flag enabled. This flag means that the request message was received but the receiving device was not able to correctly process it and reply because of some buffer constraints.

The receiver can also append its MTU in the ACK message, so that the other side could send a subset of the previous data. In this way our logical models could be built dynamically by varying the message dimension, depending on the maximum allowed MTU specified in the *option header*. So we have obtained a basic flow control.

SUBSCRIPTION OPTION FOR ENERGY SAVING

Another improvement for the CoAP protocol applies to the subscription procedure. In the last draft it is specified that the client sets a *subscription lifetime* (e.g. 60 s) while requesting a resource. Once it expires, the server must refresh the resource attribute with an ACK response. Moreover, it must also refresh this value every time it changes (without ACK).

This approach can be excessively energy consuming in some applications (e.g. temperature sensor) if this attributes modifies rapidly and if the device is power-limited (e.g. through a battery).

For this reason we propose to add a specific bit flag which can be set by the client if it requires a “real-time” notification of the resource value. Instead, if this flag is disabled, response message occurs only when the *subscription lifetime* expires, reducing therefore energy requirements.

REFERENCES

[1] http://www.sics.se/contiki/

[2] http://code.google.com/p/udpws/

[3] http://tools.ietf.org/wg/core/

[4] http://tools.ietf.org/html/draft-moritz-core-soap-over-coap-00

[5] http://tools.ietf.org/html/draft-ietf-core-coap-03