Deployment of CoAP in Transport Logistics

Markus Becker, Thomas Pötsch, Koojana Kuladinithi and Carmelita Goerg
Communication Networks, TZI
University Bremen, Germany
Email: [mab,thp,koo,cg]@comnets.uni-bremen.de

I. Introduction

Most Internet applications today use web services, which depend on the basic Representational State Transfer (REST) architecture [1]. The REST architecture allows easy integration with web browsers and web-based service providers.

The main objective of the Constrained RESTful Environments (CoRE) working group at the IETF is to see how the REST architecture can be employed in constrained networks. The CoRE working group has already proposed a standard called Constrained Application Protocol (CoAP) [2] which can easily be translated to HTTP to foster integration of constrained networks with the web.

The use of web services for sensor networking applications is seen as an important part in emerging machine to machine (M2M) communications. CoAP is proposed by the IETF to optimize the use of the RESTful web service architecture in constrained nodes and networks, for example Wireless Sensor Networks (WSN).

We demonstrate an IP based solution to integrate sensor networks used in a cargo container, which is termed *Intelligent Container*, with existing logistic processes, highlighting the use of the CoAP protocol for the retrieval of sensor data during land or sea transportation. This solution is implemented and evaluated on TinyOS, widely researched embedded system architecture.

The main objective of this demonstration is to show the use of CoAP for M2M communication in logistic applications for supervision of the environmental conditions during transport. Most of the devices in the WSN deployed in a cargo container or truck trailer have low bandwidth, scarce memory capacity and limited processing capability.

Unlike HTTP based protocols, CoAP operates over UDP and employs a simple retransmission mechanism instead of using complex congestion control as used in standard TCP. It uses a unique Message ID to identify each GET request for retransmission purposes to keep reliability.

Figure 1 highlights the possible applications of deploying CoAP in different components in the logistic process. The *Intelligent Container* architecture consists of a WSN, a freight supervision unit (FSU) and telematic devices. It transmits information such as humidity, temperature of meat, fruits etc. inside a container during land or sea transportation. This information is useful to take logistical process planning decisions to deliver food to markets efficiently and cost effectively.

The FSU is the main computing device which is the border router for the WSN and manages the different kinds of communication with the back-end software. CoAP is mainly used in the WSN part of the *Intelligent Container* to manipulate resources (e.g. temperature, humidity) by using the following methods:

- The GET method is used to retrieve resources from WSN nodes or the FSU. The resource is identified by the requested Uniform Resource Identifier (URI).
- The PUT method is used to modify an existing resource on a sensor node or the FSU.

More details about the operation of the CoAP protocol and the use of CoAP in the logistic process can be found in [2] and [3].

Resource	GET	PUT	Comments
/st	X		Temperature
/sh	X		Humidity
/sv	X		Voltage
/r	X		Temperature, humidity and voltage together
/l	X	X	LEDs
/ck	(X)	X	AES Encryption Key
/rt	X		Routing Table

TABLE I: CoAP Resources on Sensor Nodes

Resource	GET	PUT	Comments
/ni		X	Inform about node integration into 6LoWPAN network
/ri		X	Inform about node's routing table
/warntemplow		X	Below Warning Temperature Low
/warntemphi		X	Above Warning Temperature High

TABLE II: CoAP Resources on the FSU

CoAP client and server can run on the FSU as well as the sensor nodes. Table I and table II highlight resources that can be provided on the sensor nodes and the FSU. For example, for a CoAP GET request issued by the FSU to retrieve humidity from a sensor node, the CoAP server that runs on the sensor node sends the CoAP response with the humidity piggy-backed on an ACK message.

There are three possible options that the FSU uses to send data retrieved from the WSN to the back-end software.

- Option 1: The communication is a proprietary protocol as currently used in M2M telematic systems.
- Option 2: The communication between the FSU and the back-end software can also use CoAP. This helps

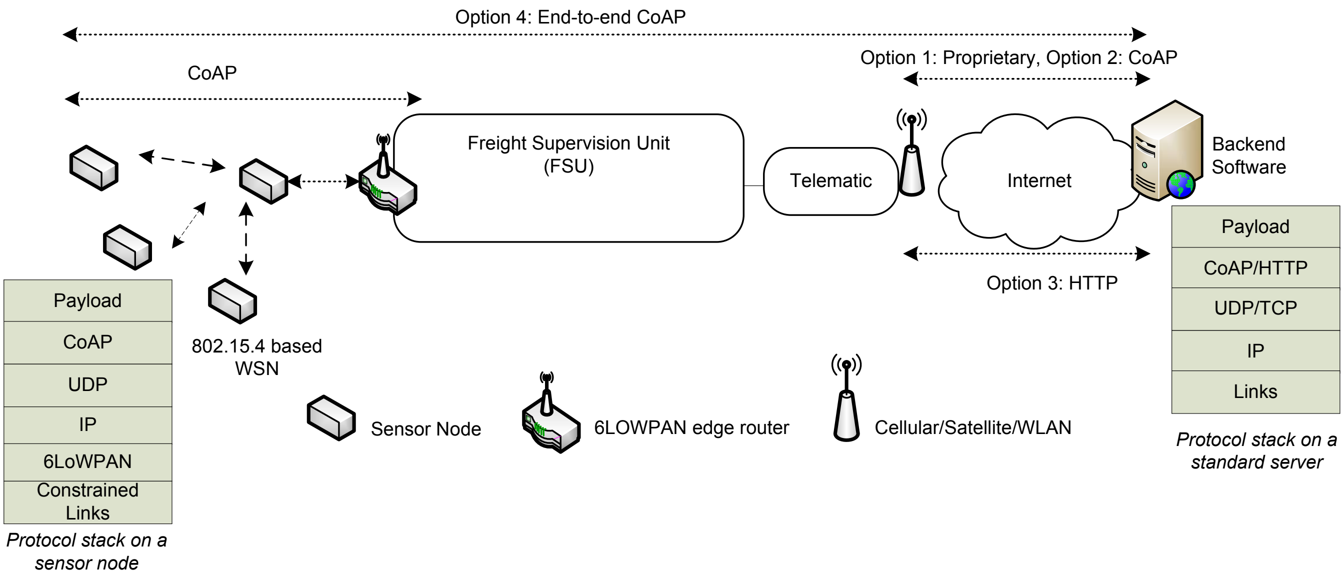

Fig. 1: Deployment of CoAP in the Intelligent Container

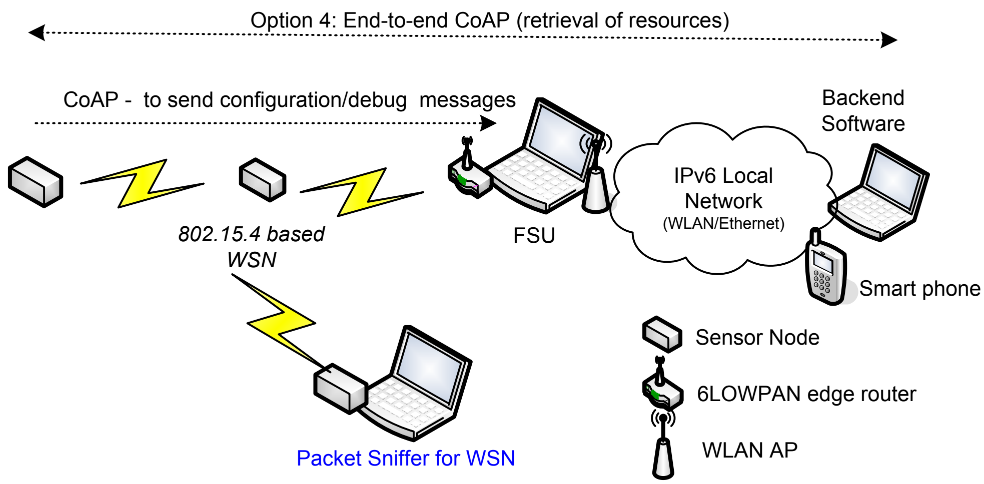

Fig. 2: Demo setup to show the use of CoAP in the Intelligent Container

in transmitting a low number of bytes especially via telematic devices with cellular/satellite connections using an open protocol.

- Option 3: The communication between the FSU and the back-end software can use a standard IP protocol such as HTTP. In this case, the FSU can be used as a proxy between CoAP and HTTP based networks.

There also exists the 4th option that the back-end software can directly retrieve sensor data from the WSN. In this case, the back-end software should use the CoAP protocol.

The initial evaluation of CoAP compared to HTTP based resource retrieval and also the evaluation of our own CoAP implementation on the TinyOS platform has been published in [3].

II. Demo Setup

The focus of this demonstration is to show the use of the CoAP protocol, developed by the authors [4] to manipulate resources in the WSN part of the *Intelligent Container*. Compared to figure 1, this demonstration uses the following functionalities to highlight the use of CoAP in different parts of the *Intelligent Container* as follows.

- Retrieval of sensor data (temperature, humidity, etc) from the *Intelligent Container* to the back-end software (use of option 4 in figure 1) or to the FSU. Here, the sensor node acts as a CoAP server.
- Sending of configuration information to the FSU. A newly joined sensor node sends */ni PUT* to inform the FSU about joining the WSN. Here, the sensor node acts as a CoAP client.
- Sending of debugging messages to the FSU. For example,

Hardware	Quantity	Functions
Laptops	3	1 to setup FSU and act as a 6LoWPAN edge router 1 to setup back-end software (to use as a CoAP client and to show retrieval of sensor data) 1 to show packet details (CoAP, UDP and 6LoWPAN headers, RPL messages) in WSNs
Sensors (e.g. TelosB)	4	1 to use as a 6LOWPAN border router, connected to the FSU 2 to create a 2 hop WSN 1 to capture packets in the WSN, connected to the Packet Sniffer laptop
Smart phone	1	to use as a CoAP client and to display retrieval of sensor data
Access point (optional)	1	to provide WLAN connectivity between the FSU and the back-end software

TABLE III: Hardware used

a sensor node uses */ri PUT* to log the routing table information to the FSU.

- The Use of the CoAP protocol over single hop and multiple hop WSNs. The connectivity is made using RPL routing protocol.
- Display of packets formats in the WSN. This shows the details of CoAP messages, RPL messages (to create routes) and the 6LoWPAN header compression used in the WSNs. The packet sniffer in figure 2 will be used to show these details.

The resources implemented in the sensor nodes and the FSU are shown in table I and table II.

All the hardware components required and the details of the test-bed setup are shown in table III and figure 2. This demonstration can be set up in 30 minutes and it requires a couple of power sockets and one or two tables to keep the hardware mentioned in table III. Further, the authors would like to use a 802.15.4 free channel, which will not be affected from channels used by the other demonstrations. The WSN part of this demo consists of the following software modules.

- CoAP protocol implemented by the authors [4].
- blip 2.0 [5] for TinyOS platform. This is the Berkley 6LoWPAN implementation together with the RPL routing protocol [6].
- TelosB sensor nodes with TinyOS and 802.15.4 radio stack.

III. Acknowledgments

This research project ('The Intelligent Container') is supported by the Federal Ministry of Education and Research, Germany, under reference number 01IA10001.

References

[1] R. Fielding, "Representational State Transfer (REST)," *Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine*, 2000.

[2] Z. Shelby, B. Frank, and D. Sturek, "Constrained Application Protocol (CoAP)," Internet-Draft (work in progress), Available at: http://tools.ietf.org/html/draft-ietf-core-coap-07, Jul. 2011. [Online]. Available: http://tools.ietf.org/html/draft-ietf-core-coap-07

[3] K. Kuladinithi, O. Bergmann, T. Pötsch, M. Becker, and C. Görg, "Implementation of CoAP and its Application in Transport Logistics," in *proceedings of Extending the Internet to Low power and Lossy Networks(IP+SN 2011)*, 2011.

[4] Th. Pötsch and M. Becker, "TinyOS CoAP installation instructions," Available at: http://docs.tinyos.net/index.php/CoAP, last accessed 2011-02-11. [Online]. Available: http://docs.tinyos.net/index.php/CoAP

[5] S. Dawson-Haggerty, "Design, implementation, and evaluation of an embedded IPv6 stack. Master's thesis," 2010.

[6] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks," Internet-Draft (work in progress), draft-ietf-roll-rpl-19.txt. http://tools.ietf.org/html/draft-ietf-roll-rpl-19, Dec. 2010.