RESTful Web Services for the Internet of Things

Markku Laine
Department of Media Technology
Aalto University School of Science
P.O. Box 15400, FI-00076 Aalto, FINLAND
markku.laine@aalto.fi

Abstract—In the coming years, the size and scope of the Internet will greatly increase as physical world devices (e.g., sensors and home appliances) are becoming smart enough to communicate and share data over the Internet—a phenomenon known as the Internet of Things (IoT). The problem of realizing the IoT is, however, how to effectively penetrate the devices into the existing Web. In this paper, we present Constrained Application Protocol (CoAP) and other key enabling technologies together with an end-to-end IP and RESTful Web Services based architecture for integrating physical world devices in constrained environments with the Web.

Keywords-CoAP; HTTP; Internet of Things; REST; Web Services

I. INTRODUCTION

In the vision of the Internet of Things (IoT), the Internet is spreading beyond its core (e.g., routers and servers) and quickly growing fringe (e.g., personal computers and smart phones) to trillions of small embedded devices (e.g., sensors) in the physical world [17]. This huge growth brings both exciting possibilities and challenges to the Internet, such as how to seamlessly integrate constrained devices with the Web.

One way to naturally unify the cyber-world and the physical world is to reuse existing web technologies and standards as much as possible, a research trend treating the IoT as the Web of Things (WoT) [21]. In the WoT, devices not only become IP enabled and interconnected on the Internet but also become enabled to speak the same language, and thus will be able to communicate and interoperate freely on the Web. In order to realize this vision, re-designing and optimizations in payload encoding and application protocols needs to be conducted to meet the special requirements of machine-to-machine (M2M) applications over constrained environments on the IoT.

The rest of the paper is organized as follows. Section II discusses the advantages of using RESTful Web Services over arbitrary Web Services in the context of the IoT. Sections III and IV describe CoAP and how it overcomes the limitations of HTTP in constrained environments. In Section V, we present an end-to-end IP and RESTful Web Services based architecture for extending the Internet to constrained environments. Section VI concludes.

II. RESTFUL WEB SERVICES

Web Services [1] provide a standard and interoperable means for M2M communication on the Internet. The two major classes of Web Services are: (1) Representational State Transfer (REST) [7] compliant Web Services, in which resources are manipulated using a uniform set of “stateless” operations and (2) arbitrary Web Services (WS-*) [17], which expose an arbitrary set of operations and use, for example, SOAP messages [8] [9].

In the context of the IoT, the RESTful Web Services have many advantages over arbitrary Web Services (i.e., SOAP), such as less overhead, less parsing complexity, statelessness, and tighter integration with HTTP [17]. In addition, applications supporting RESTful Web Services perform better on wireless sensor nodes with limited resources [20] as well as are considered easier to learn and implement [10]. For a formal description of RESTful Web Services either Web Services Description Language (WSDL) 2.0 [2] or Web Application Description Language (WADL) [11] can be used.

III. CONSTRAINED APPLICATION PROTOCOL

Although HTTP is widely used with Web Services, it is by no means the only protocol for M2M communication. In June 2010, the Internet Engineering Task Force (IETF) Constrained RESTful Environments (CoRE) working group published the first draft of a RESTful web transfer protocol called Constrained Application Protocol (CoAP) [16]. CoAP includes several HTTP functionalities which have been re-designed for M2M applications over constrained environments on the IoT, meaning it takes into account the low processing power and energy constraints of small embedded devices, such as sensors. In addition, CoAP offers a number of features that HTTP lacks, such as built-in resource discovery, IP multicast support, native push model, and asynchronous message exchange. There are many implementations of CoAP in various languages, such as libcoap1 (an open-source C-implementation) and Sensinode’s NanoService2.

1 libcoap, http://libcoap.sourceforge.net/
IV. THE COAP PROTOCOL STACK

The International Organization for Standardization (ISO) has defined the seven-layer Open Systems Interconnection (OSI) reference model [22]. Next, we will go through the layers from 1 to 4 as well as 7 of the OSI model (cf. Fig. 1), and describe how the CoAP protocol stack overcomes the challenges of the IoT compared to a typical HTTP protocol stack.

A. Physical Layer and Data Link Layer

Wireless networks are essential for the IoT because wiring up sensors is too expensive. The IEEE 802.15.4 standard [12] specifies the physical layer (PHY) and media access control (MAC) for low-rate wireless personal networks (LR-WPANs), which focuses on low-power consumption, low cost, and low data transfer rate communication between constrained devices.

B. Network Layer

IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) standard [13] [14], defined by the IETF, brings the Internet Protocol (IP) to small embedded devices (e.g., sensors) in even the most constrained networks, such as IEEE 802.15.4. In addition to 6LoWPAN, the IETF Routing over Low Power and Lossy Networks (ROLL) working group has defined IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) [19] for smart object internetworking. Together these networking technologies provide means for small embedded devices to integrate into the Internet.

C. Transport Layer

HTTP typically relies on the Transmission Control Protocol (TCP), which has performance problems over Low-Power and Lossy Networks (LLNs), is sensitive to mobility, does not provide multicast support, and has high overhead for short-lived transactions [17]. CoAP, on the other hand, is built on top of the User Datagram Protocol (UDP), which provides significantly lower overhead and supports multicast.

D. Application Layer

As described earlier, CoAP provides RESTful Web Services optimized for resource-constrained networks and devices, and thus makes the protocol suitable to the IoT and M2M applications. In addition, it provides reliability (with default timeout and exponential back-off retransmission) mechanisms even without the use of TCP as transport layer. One of the most important design goals of CoAP has also been to keep the message overhead as small as possible.

HTTP can also be used over 6LoWPAN. However, the results show that the power consumption and bytes transferred per transaction are drastically lower when using CoAP over 6LoWPAN compared to HTTP over 6LoWPAN, and thus increasing the battery lifetime of constrained devices [5].

E. Payload

The W3C Efficient XML Interchange (EXI) [15] format is a very compact, high performance Extensible Markup Language (XML) representation, which significantly reduces bandwidth requirements without compromising efficient use of other resources, such as battery life, code size, processing power, and memory.

In [17], four encoding techniques (XML, EXI\textsuperscript{3}, BXML, and Fast Infoset) were compared using typical sensor markup languages (RDF, ZigBee Smart Energy, and OGC SensorML). The results show that EXI encoding is superior, having up to 97 percent smaller size than the equivalent XML, making it a reasonable option to be used over 6LoWPAN and similar constrained networks. According to [4], the pure compression efficiency for EXI is extremely good compared to other compression schemes (Gzip, enhanced Xmill (Xwrt), and XMLPPM).

V. EXTENDING THE INTERNET TO CONSTRAINED ENVIRONMENTS

Fig. 2 illustrates an end-to-end IP and RESTful Web Services based architecture for integrating constrained devices with the Web. The architecture allows realizing the same interface design over HTTP and CoAP because both protocols implement the REST paradigm.

---

\textsuperscript{3} Strict schema-informed mode with bit alignment
In order to provide a client of a protocol to transparently access resources of a server of another protocol, a device providing cross-protocol HTTP-CoAP mapping [16]—HTTP-CoAP cross-protocol proxy (HC proxy) [3]—is needed. In their paper [6], Colitti et al. describe the design and implementation of such a system, which integrates a REST/CoAP wireless sensor network (WSN) with a REST/HTTP web application to allow a visualization of WSN measurements in a web browser.

VI. CONCLUSION

In this paper, we discussed on the utilization of RESTful Web Services in the context of the Internet of Things (IoT). We introduced a promising, RESTful web server protocol called CoAP, which is similar to HTTP but has been re-designed and optimized especially for machine-to-machine (M2M) applications over constrained environments on the IoT. According to recent studies, it seems that the use of standards (e.g., CoAP and EXI) and web paradigms are the key success factors for extending the Internet to constrained environments and making the vision of the IoT become reality.

REFERENCES