

Demo Abstract: Human–CoAP Interaction with Copper

Matthias Kovatsch
Institute for Pervasive Computing
ETH Zurich, Switzerland
Email: kovatsch@inf.ethz.ch

Abstract—We introduce ‘Copper,’ a generic browser for the Internet of Things based on the Constrained Application Protocol (CoAP). Current estimates foresee that the number of networked embedded devices encompassed by the Internet of Things will be vast. Additionally, most systems will be optimized for the constrained environment with its limited energy and bandwidth. These factors make it difficult for users to observe and control the devices. Thus, a major problem will be node and network management, as experienced before in large wireless sensor network deployments. By adopting well-known patterns from the Web, such as browsing, bookmarking, and linking, we provide a user-friendly management tool for networked embedded devices. By integrating it into the Web browser, we allow for intuitive interaction and a presentation layer that is originally missing in the CoAP protocol suite.

I. INTRODUCTION

The Internet of Things is the vision of interconnecting everyday objects and in a further step making them smart. With the recent advances in the field of wireless sensor networks (WSNs) and the standardization of 6LoWPAN [5] and its relatives, we can now realize this vision using the Internet protocol IPv6. Everyday objects connect directly at the network layer of the Internet. Therefore, current estimates expect the Internet to grow tremendously at its edge. A trillion of networked embedded devices will offer information about themselves as well as their environment. Just like RFID technology is revolutionizing logistics and supply chain management, they will improve business processes such as factory automation, the energy efficiency of buildings and sustainability of entire cities, and our personal daily lives [11].

The vast number, however, as well as the resource-constrained nature of these devices turns node and network management into a challenging task. Although there are mechanisms that aim for zero configuration networking [8], humans will usually be involved to assure correct behavior at the application layer. To this end, we introduce ‘Copper’ (Fig. 1), a generic browser for networked embedded devices running the Constrained Application Protocol (CoAP) [10]. We expect this protocol, or a further development, to be central for the Internet of Things, as it combines well-proven concepts from the Web with efficient mechanism for constrained environments in a long-lived IETF standard. With Copper, we provide a management tool that allows to browse devices in the same fashion users are used to explore the Web.

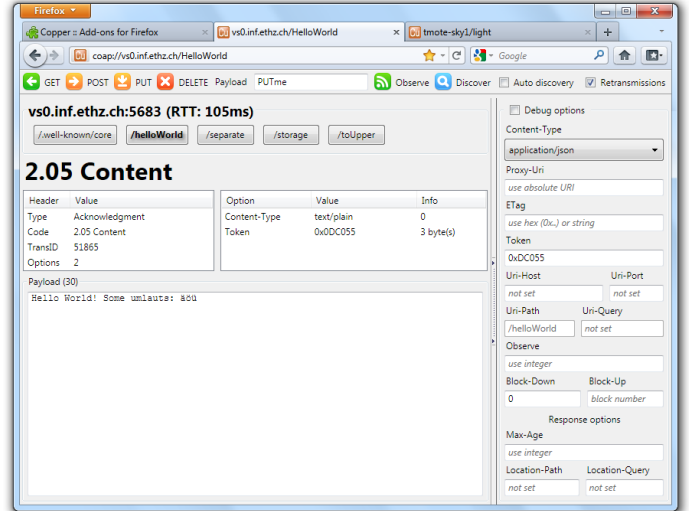


Fig. 1. Internet of Things browser ‘Copper’ implemented as Firefox extension <http://addons.mozilla.org/en-US/firefox/addon/copper-270430/>

II. THE CONSTRAINED APPLICATION PROTOCOL

This section summarizes the features and advantages of CoAP [10]. It is a Web-oriented protocol, i.e., it adopts features from HTTP, the Web protocol. The central ones are the (i) resource abstraction, (ii) RESTful interaction [2], and (iii) extensible header options. These features allowed the Web to evolve from a simple document retrieval mechanism (‘World-Wide Web’) to a thriving application platform (‘Web 2.0’). Being a successful, long-lived IETF standard, HTTP allows us to combine different resources or services with very little scripting effort in so called ‘mashups.’ This interoperability is the key argument for the ‘Web of Things’ initiative [3] to push HTTP down to the device level. HTTP over TCP, however, has a one-to-one communication model, no native push notifications, and is rather heavy-weight for constrained devices (i.e., 8-bit microcontroller class).

To overcome this downside, CoAP combines (i–iii) with size-optimized, reliable datagram communication. On the one hand, it offers URIs (e.g., `coap://vs0.inf.ethz.ch/`), the RESTful methods GET, POST, PUT, and DELETE, and extensions through header options that can be defined independently. On the other hand, CoAP uses UDP, which is lighter than TCP, but moreover allows for efficient IP multicast. Group communication is a significant requirement for the Internet

of Things, for instance for automation applications. To make up for the unreliability of UDP, CoAP defines transactions with retransmissions. Native push notifications for eventing are supported with the publish/subscribe pattern to observe resource changes [4]. Finally, a resource discovery mechanism is provided, which also provides for resource descriptions [9].

III. THE HUMAN IN THE LOOP

In the following, we argue why in typical situations humans must remain in the loop. At the network layer, devices can automatically join a network, retrieve an address, and start communicating with other nodes without user input. At the application layer, however, this is hard because bindings (i.e., the configuration of endpoints devices should communicate with) decide about the semantics of an application. With device profiles, it is possible to automatically bind matching endpoints (e.g., a switch with a lamp or a dimmer with a dimmable light). This, however, fails if multiple endpoints with the same profile are available and an endpoint shall only control a specific endpoint or a subset.

More extensive descriptions and software agents could help making such decisions automatically, but first users or programmers will have to define the rules to be applied. This also only works when all devices observe the same standard. That is, however, unlikely with the plethora of slightly different device types from many different vendors, which might be unrelated to networking such as home appliances companies or sensor equipment specialists. Just like Web applications require some amount of glue code to mash up services, programmers will have to ensure device interoperability.

Therefore, users require an appropriate tool for the Internet of Things: to explore devices, test applications, and manage them both. It would help making bindings, defining correct rules, and developing glue code. Such a tool is particularly interesting for CoAP, which is designed for machine-to-machine communication and does not provide a presentation layer.

IV. COPPER (CU): COAP FOR THE USER

We designed Copper, a generic browser for the Internet of Things. The idea of such a browser has already been around. The BIT (Browser for the Internet of Things) framework provides a “single point of interaction for users when accessing the services of a variety of tagged objects.” [7] In opposition to Copper, it is meant for physical artifacts that are linked to their virtual presence through tags such as RFID or barcodes and do not provide direct interaction. A comparable solution to Copper was presented by Schor et al. [8]. Their work allowed to browse and modify the properties of WSN nodes. They used, however, an intermediate JavaEE server that rendered the information in HTML and used corresponding forms to interact via application-level gateway. This approach is also used in [3]. Closer to our project, the DPWS Explorer¹ is a standalone application that allows exploring DPWS-enabled devices [6] and provides a UI for interaction.

We developed Copper as an extension for the Mozilla Firefox Web browser, so it integrates well into the users’ way of interacting with the Web. The extension registers a protocol handler for the scheme `coap`. Thus, CoAP URIs can simply be entered into the address bar or followed from links on Web pages. Thereupon, the user can use (automatic) resource discovery, browse the resources of the host, and interact through buttons corresponding to the RESTful methods. Firefox can also bookmark CoAP resources in the usual way, for instance by clicking on the star in the address bar. Copper is designed to support multiple versions of CoAP, as implementations usually focus on a specific version to be consistent while the draft keeps changing. We currently support two draft versions: 03, which is supported by the C implementation `libcoap`² and the current Contiki-2.x release, and 06, which is the latest draft. Additionally, we implemented blockwise transfers [1], observing [4], and a minimal version of the CoRE link format [9]. The presentation of resources is currently static and basically a pretty-print for the binary protocol.

In future work, we will make use of Mozilla’s powerful XUL/XHTML rendering engine to provide more expedient and intuitive presentations for different types of resources. We will also investigate the CoRE link format to provide semantic descriptions for an automatic selection or generation of presentations.

V. DEMONSTRATION SETUP

Finally, we give an overview of the demonstration. Our browser will interact with different CoAP implementations (`libcoap`², `jCoAP`³, `Californium`⁴, and CoAP for Contiki) that run on different systems (the local computer, a server in the cloud, simulated motes in Cooja, and real motes deployed on-site). We demonstrate how Copper helps exploring and managing the nodes and discuss ideas for our future work.

REFERENCES

- [1] C. Bormann and Z. Shelby. Blockwise transfers in CoAP. draft-ietf-core-block-03, 2011.
- [2] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Phd thesis, UC Irvine, 2000.
- [3] D. Guinard, V. Trifa, and E. Wilde. A Resource Oriented Architecture for the Web of Things. In *Proc. IoT*, Tokyo, Japan, 2010.
- [4] K. Hartke and Z. Shelby. Observing Resources in CoAP. draft-ietf-core-observe-02, 2011.
- [5] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC4944, 2007.
- [6] G. Moritz, E. Zeeb, S. Pruter, F. Golatowski, D. Timmermann, and R. Stoll. Devices Profile for Web Services in Wireless Sensor Networks: Adaptations and Enhancements. In *Proc. ETFA*, Mallorca, Spain, 2009.
- [7] C. Roduner and M. Langheinrich. BIT A Framework and Architecture for Providing Digital Services for Physical Products. In *Proc. IoT*, Tokyo, Japan, 2010.
- [8] L. Schor, P. Sommer, and R. Wattenhofer. Towards a Zero-Configuration Wireless Sensor Network Architecture for Smart Buildings. In *Proc. BuildSys*, Berkeley, CA, USA, 2009.
- [9] Z. Shelby. CoRE Link Format. draft-ietf-core-link-format-05, 2011.
- [10] Z. Shelby, K. Hartke, C. Bormann, and B. Frank. Constrained Application Protocol (CoAP). draft-ietf-core-coap-06, 2011.
- [11] J.-P. Vasseur and A. Dunkels. *Interconnecting Smart Objects with IP: The Next Internet*. Morgan Kaufmann, 2010.

¹<http://ws4d.e-technik.uni-rostock.de/dpws-explorer/>

²<http://sourceforge.net/projects/libcoap/>

³<http://code.google.com/p/jcoap/>

⁴<https://github.com/mkovatsc/Californium>