Fotiou, Nikos; Xylomenos, George; Polyzos, George C.; Islam, Hasan; Lagutin, Dmitrij; Hakala, Teemu; Hakala, Eero

ICN enabling CoAP extensions for IP based IoT devices

Published in:
ICN 2017 - Proceedings of the 4th ACM Conference on Information Centric Networking

DOI:
10.1145/3125719.3132105

Published: 26/09/2017

Document Version
Peer reviewed version

Please cite the original version:
Fotiou, N., Xylomenos, G., Polyzos, G. C., Islam, H., Lagutin, D., Hakala, T., & Hakala, E. (2017). ICN enabling CoAP extensions for IP based IoT devices. In ICN 2017 - Proceedings of the 4th ACM Conference on Information Centric Networking (pp. 218-219) https://doi.org/10.1145/3125719.3132105

ICN enabling CoAP Extensions for IP based IoT devices

Nikos Fotiou, George Xylomenos, George C. Polyzos
Athens Univ. of Econ. & Bus., Greece
{fotiou,xgeorge,polyzos}@aueb.gr

Hasan Islam, Dmitrij Lagutin
Aalto University
Finland
firstname.lastname@aalto.fi

Teemu Hakala, Eero Hakala
Ell-i open source co-operative
Finland
temmi@iki.fi,eero.hakala@ell-i.org

ABSTRACT

The Constrained Application Protocol (CoAP) and its extensions, such as observe and group communication, offer the potential for developing novel IoT applications. However, a full-fledged CoAP-based application requires delay-tolerant communication and support for multicast: since these properties cannot be easily provided by existing IP networks, developers cannot take full advantage of CoAP, preferring to use HTTP instead. In this demo we show how proxying CoAP traffic over an ICN network can unleash the full potential of CoAP, simultaneously shifting overhead and complexity from the (constrained) endpoints to the network.

CCS CONCEPTS

• **Networks** → **Network architectures**; **Middle boxes / network appliances**; *Naming and addressing*;

KEYWORDS

CoAP, Experimentation, Namespaces

1 INTRODUCTION

The Constrained Application Protocol (CoAP) [5] has been called the "HTTP for the Internet of Things" (IoT), as it allows CoAP clients to retrieve or set resources from CoAP servers implemented in constrained devices (the Things). In contrast to HTTP however, it is implemented over UDP and it allows for delayed responses, e.g., a CoAP client may request the value of a resource that is not yet ready; it will initially receive an acknowledgment for its request and when the resource becomes available it will receive the appropriate response. Various CoAP extensions enable novel applications, departing even further from the traditional one request-one response model. For instance, the CoAP observe [2] extension allows CoAP clients to *observe* resources and receive a notification everytime their state changes (i.e., very similar to the publish-subscribe communication paradigm). CoAP group communication is another CoAP extension defined in RFC 7390 [4]. This extension allows clients to retrieve or set resources from a *group* of servers e.g., retrieve the temperature from all sensors of a building, turn on all the lights of a factory, etc. Group names follow the structure of legacy CoAP URIs, nevertheless they can be overloaded with application specific semantics (e.g., a CoAP GET request for the group *coap://floor3.building6/temperature* may result in all temperature sensors in floor 3 of building 6 to send a record).

Despite its potential, CoAP is mostly used in its basic form (one request-one immediate response). The main reason for this is that a full fledged CoAP deployment is cumbersome, mostly due to the IP underlay. For example, in order to support delay-tolerant messaging or publish-subscribe communication (i.e., CoAP observe) CoAP servers should maintain extensive state. Similarly, RFC 7390 suggests that CoAP group communication could be implemented by using IP multicast, with DNS mapping group names (included in the CoAP URI requests) into the appropriate IP multicast address: with this approach CoAP servers should implement IP multicast and, for each group, a specific IP multicast address should be assigned (and configured in CoAP endpoints).

In this demo, we show how the POINT Information-Centric Networking (ICN) architecture can be leveraged, so that IP endpoints that implement only core CoAP, can benefit from CoAP and its extensions.

1.1 The POINT architecture

The POINT architecture allows standard IP traffic to be run over an ICN core network; the ICN core is typically deployed at a single network provider [6]. POINT's core ICN network is implemented using the Publish-Subscribe Internet (PSI) ICN architecture [7], a publish-subscribe architecture, where users interested in receiving specific content subscribe to it, while content owners advertise their content and, if requested, publish it (i.e., they transfer it to the subscribers, by default through multicast). The POINT architecture provides a number of handlers for existing IP-based protocols (e.g., HTTP, CoAP, and basic IP) that map the underlying protocols onto appropriate named objects within the ICN core. Therefore, existing applications can benefit from ICN's features by forwarding their (legacy) traffic through Network Attachment Points (NAPs), where these protocols handlers are implemented.

Figure 1 gives a high level overview of a POINT network with CoAP handlers enabled in its NAPs. NAPs connected to CoAP servers learn the URIs of the available resources and send subscription messages to the ICN core indicating their interest in receiving CoAP requests (encapsulated into POINT objects). Similarly, whenever a NAP connected to CoAP clients (possibly, via the Internet) receives a CoAP request, it encapsulates it into a POINT content item and advertises it in the ICN network. This advertisement will result in the content item being forwarded to an appropriate NAP; this NAP then will decapsulate the CoAP request and will forward it to the appropriate CoAP server. CoAP enabled NAPs can aggregate requests for the same CoAP resource, shifting this way state management from CoAP servers to NAPs.

ICN '17, September 26–28, 2017, Berlin, Germany

ACM ISBN 978-1-4503-5122-5/17/09…$15.00
https://doi.org/10.1145/3125719.3132105

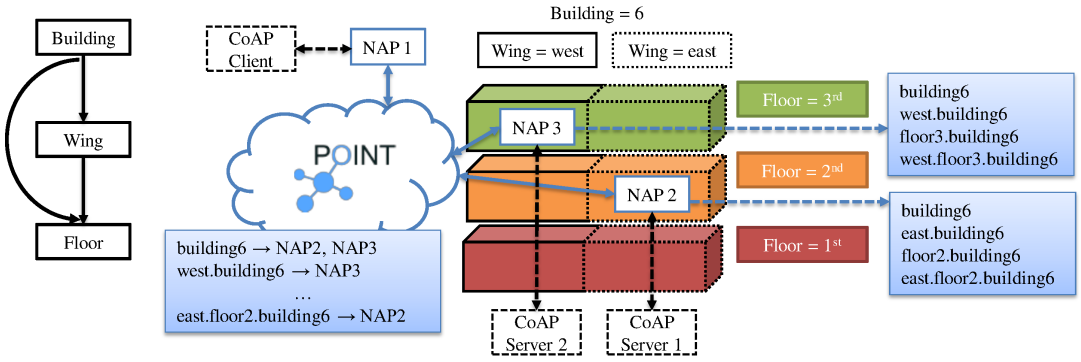

Figure 1: High level overview of a POINT-enabled building management system.

2 ENABLING COAP EXTENSIONS OVER POINT

2.1 Approach

The POINT approach for supporting the CoAP observe extension is detailed in [3]. In a nutshell, the POINT NAPs aggregate requests for the same resource, hence from the CoAP server's perspective only a single CoAP client is visible. Moreover, update notifications are transmitted using multicast, thus conserving network resources. POINT leverages its information-centrism and its inherent support for multicast to support seamless and hassle-free group communication among CoAP endpoints. In particular, it takes advantage of PSI's name structure in order to organise group "attributes"; then it assigns values to these attributes to construct group names, and maps these names into the appropriate PSI *scopes*.

In order to illustrate this concept we consider the case of a building management system (depicted in Figure 1). In this case there are CoAP servers located inside buildings and each server is attached to a NAP. Buildings are numbered with a *building* number, and then subdivided in *wings* and *floors*; these are the possible group attributes, which are hierarchically organised as shown in the left part of Figure 1. A CoAP client may send a request to a group of CoAP servers; the group name is created by assigning "values" to (some of) the specified attributes, e.g., by setting $building = building6$, $wing = west$, and $floor = floor3$ the group name *floor3.west.building6* is constructed. POINT NAPs are configured with values for the specified attributes, so that by using these values they can construct all possible group names e.g., a NAP located in building6, west, 3^{rd} floor, creates the names *building6*, *building6.west*, *floor3.building6*, and *floor3.west.building6*. Then, each NAP subscribes to the ICN content identifier that corresponds to each name.

Using this scheme, a CoAP request for a group (for example, *coap://floor2.building6/temperature*), is encapsulated into a POINT content item and is advertised in the ICN network using as an identifier the FQDN of the CoAP server as specified in the request's URL (i.e., in our example 'floor2.building6'); all NAPs that have subscribed to this identifier will receive that item, will decapsulate the CoAP request and will forward it to the corresponding CoAP servers. Following a similar approach, CoAP responses are encapsulated into a POINT content item and are forwarded to the appropriate NAPs and eventually to interested CoAP clients.

2.2 Key contributions

The benefits of POINT to CoAP in general, are highlighted in [1]. Similarly, the benefits of POINT to CoAP observe are highlighted in [3]. When it comes to CoAP group communication, our approach enables issuing requests to groups of CoAP servers that implement the standard version of the CoAP protocol (i.e., they do not support RFC 7390). As a result, CoAP servers do not have to implement IP multicast. Moreover, there is no need for modifications to DNS. CoAP servers are oblivious to group names, since names are handled by the NAPs, thus Things management becomes much easier. The ICN core makes group name administration easier: new attributes can be easily added to the namespace without affecting already deployed NAPs. Moreover, group names do not have to be mapped a priori to a lower layer network address.

ACKNOWLEDGMENTS

This research was supported by the EU funded H2020 ICT project POINT, under contract 643990.

REFERENCES

[1] N. Fotiou, H. Islam, D. Lagutin, T. Hakala, and G.C. Polyzos. 2016. CoAP over ICN. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. 1–4. https://doi.org/10.1109/NTMS.2016.7792438
[2] K. Hartke. 2015. *Observing Resources in the Constrained Application Protocol (CoAP)*. RFC 7641. IETF.
[3] Hasan Islam, Dmitrij Lagutin, and Nikos Fotiou. 2017. Observing IoT Resources over ICN. In *2017 1st IFIP Netowrking Workshop on Information-Centric Fog Computing (ICFC)*.
[4] A. Rahman and E. Dijk. 2014. *Group Communication for the Constrained Application Protocol (CoAP)*. RFC 7390. IETF.
[5] Z. Shelby, K. Hartke, and C. Bormann. 2014. *The Constrained Application Protocol (CoAP)*. RFC 7252. IETF.
[6] D. Trossen, M. J. Reed, J. Riihijarvi, M. Georgiades, N. Fotiou, and G. Xylomenos. 2015. IP over ICN - The better IP?. In *European Conference on Networks and Communications (EuCNC)*,. 413–417.
[7] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V.A. Siris, and G.C. Polyzos. 2012. Caching and mobility support in a publish-subscribe internet architecture. *IEEE Communications Magazine* 50, 7 (July 2012), 52–58.