SACM Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2018

H. Birkholz
Fraunhofer SIT
T. Zhou
Huawei
X. Liu
Jabil
E. Voit
Cisco Systems

October 18, 2017

YANG Push Operations for CoMI
draft-birkholz-yang-push-coap-problemstatement-00

Abstract

This document provides a problem statement, derives an initial gap analysis and illustrates a first set of solution approaches in regard to augmenting YANG data stores based on the CoAP Management Interface with YANG Push capabilities. A binary transfer mechanism for YANG Subscribed Notifications addresses both the requirements of constrained-node networks and the need for semantic interoperability via self-descriptiveness of the corresponding data in motion.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2018.

Table of Contents

1. Context of the Problem

A binary transfer capability for YANG Subscribed Notifications [I-D.ietf-netconf-subscribed-notifications] based on YANG Push [I-D.ietf-netconf-yang-push] can be realized by using existing RFC and I-D work as building blocks. This section is intended to provide a corresponding overview of the existing ecosystem in order to identify gaps and therefore provide a problem statement.

1.1. Binary YANG transfer protocol

The CoAP Management Interface I-D (CoMI [I-D.ietf-core-comi]) defines operations for a YANG data store based on the Constrained Application Protocol (CoAP [RFC7252]). CoAP uses a request/response interaction model that is based on HTTP (similar to RESTCONF [RFC8040]) and allows for multiple transports, including UDP or TCP (see [I-D.ietf-core-coap-tcp-tls]). The Concise Binary Object Representation (CBOR [RFC7049]) is used for the serialization of data in motion in respect to CoAP operations and the data modeled with YANG [I-D.ietf-core-yang-cbor].

1.2. Device-Type Scope

[I-D.ietf-core-comi] states that CoAP "is designed for Machine to Machine (M2M) applications such as smart energy, smart city and building control. Constrained devices need to be managed in an automatic fashion to handle the large quantities of devices that are expected in future installations. Messages between devices need to be as small and infrequent as possible. The implementation complexity and runtime resources need to be as small as possible."

In addition, [I-D.ietf-core-comi] highlights that "CoMI and RESTCONF are intended to work in a stateless client-server fashion. They use a single round-trip to complete a single editing transaction, where NETCONF needs up to 10 round trips. To promote small messages, CoMI uses a YANG to CBOR mapping [I-D.ietf-core-yang-cbor] and numeric identifiers [I-D.ietf-core-sid] to minimize CBOR payloads and URI length."

In essence, via CoMI, a small sensor can emit a set of measurements as binary encoded YANG notifications, which would only add a minimal overhead to the data in motion, but would increase interoperability significantly due to the powerful and widely used semantics enabled by YANG (in contrast to a set of raw values that always require additional context information and imperative guidance to be managed and post-processed appropriately).

1.3. Subscriptions via CoAP

The CoAP pub/sub I-D defines a CoAP Subscribe operation [I-D.ietf-core-coap-pubsub] that is based on observing resources via the Observe option for the GET operation as defined in [RFC7641]. The CoAP pub/sub draft is intended to provide the capabilities and characteristics of MQTT via a CoAP based protocol. The only other CoAP operation that supports the Observe option is the FETCH operation defined in [RFC8132].

The Observe option creates a small corresponding state on the server side that eliminates the need for continuous polling of a resource via subsequent requests. Instead, subsequent responses including both the Observe option and using the token of the request that initiated the observation are returned when the observed resource changes. A subscription (i.e. the observe state retained on the server) can be discarded by the client by sending a correspond CoAP GET with Observe using an Observe parameter of 1 or simply by "forgeting" the observation and return a CoAP Reset after receiving a notification in the context of the subscription. A subscription can also be discarded by the server by sending a corresponding response that does not contain an Observe option.

The subscription used in CoAP pub/sub are used to subscribe to a topic provided by a CoAP broker REST API. YANG Push [I-D.ietf-netconf-yang-push] and corresponding YANG Subscribed Notifications are used to subscribe to data node updates provided by a YANG management interface. YANG subscriptions can include a filter expression (either a subtree expression or an XPATH expression). The encoding rules of XPATH expressions in CBOR are covered by [I-D.ietf-core-yang-cbor].

1.4. Configured Subscriptions and Call-Home

Configured subscriptions are basically static configuration that creates subscription state on the YANG data store when it is started and persists between boot-cycles without the need of a client to create that subscription state. In consequence, a configured subscription can result in unsolicited pushed notifications in respect to a YANG client.

A popular variant of the configured subscription as defined in [I-D.ietf-netconf-yang-push] is the Call Home procedure defined in [RFC8071]. In this approach, a Transport Layer application association with the YANG client is initiated by the YANG data store. After this "initial phase, in which the YANG server is acting like a client", the existing Transport Layer connection (or session, in case of, for example, TLS) is then used to the YANG client to initiate a subscription (i.e. the YANG client is initiating a dynamic subscription based on a pre-configured request retained and issued by the YANG data store).

1.5. Bootstrapping of Drop-Shipped Pledges

[I-D.ietf-anima-bootstrapping-keyinfra] highlights that effectively "to literally 'pull yourself up by the bootstraps' is an impossible action. Similarly, the secure establishment of a key infrastructure without external help is also an impossibility."

According to [I-D.ietf-anima-bootstrapping-keyinfra] the bootstrapping approach Call-Home has problems and limitations, which (amongst others) the draft itself is trying to address:

- the pledge requires realtime connectivity to the vendor service
- the domain identity is exposed to the vendor service (this is a privacy concern)
- the vendor is responsible for making the authorization decisions (this is a liability concern)

A Pledge in the context of [I-D.ietf-anima-bootstrapping-keyinfra] is "the prospective device, which has an identity installed by a third-party (e.g., vendor, manufacturer or integrator)."

A Pledge can be "drop-shipped", which refers to "the physical distribution of equipment containing the 'factory default' configuration to a final destination. In zero-touch scenarios there is no staging or pre-configuration during drop-ship."

In the scope of Call-Home as a part of YANG Push, either the factory default configuration of a drop-shipped Pledge that is a YANG data store would require to include the "home to Call Home" configuration or it has to be configured locally.

[I-D.ietf-netconf-zerotouch] is intended to provide more flexibility to the Call-Home procedure already - by allowing to stage connection attempts to a locally administered network and if that fails fall back to connecting to a remotely administered network. Alas, [I-D.ietf-netconf-zerotouch] is either prone to the same limitations as cited above or requires local configuration in order to find the home to Call-Home.

The "Join Registrar" defined by [I-D.ietf-anima-bootstrapping-keyinfra] mitigates the cited problems and limitation by introducing "a representative of the domain that is configured, perhaps autonomically, to decide whether a new device is allowed to join the domain. The administrator of the domain interfaces with a Join Registrar (and Coordinator) to control this process. Typically a Join Registrar is "inside" its domain."

2. Summary of the Problem Statement

Currently, the following gaps are identified:

- no CoAP Subscribe procedure for dynamic YANG subscriptions is standardized that is able to convey a filter expression and

potentially other metadata required in the context of a YANG Subscribed Notifications application association. Analogously, new payload types (e.g. a FETCH payload media-type) have to be defined.

- no CoAP Call Home feature is standardized to support a popular variant of configured YANG subscriptions.

- no general Call Home mechanism is standardized that enables the discovery of "a home to Call Home" or that would be able to deal with "changing homes" in a dynamic but secure manner.

In addition to the identified gaps, the semantics of metadata - if there are any - that have to be conveyed to or from a YANG data store in order to subscribe to a (filtered) YANG module or data node are not identified.

The problem statement could be summarized as follows:

"There is no complete solution based on CoAP to enable a freshly unpacked YANG data store ("drop-shipped pledge", e.g. the cliche light bulb) to discover an appropriate home it can than Call-Home to in a secure and trusted manner in order to push (un-)solicited subscribed notifications."

3. Potential Approaches and Solutions

There are multiple approaches that could lead to viable solutions that address the identified gaps. The following sections illustrate the general solution context and some of the most promising approaches.

3.1. YANG subscription variants

A YANG Push update subscription service both provides support for dynamic subscription (i.e. subscription state created by a client request, allowing for solicited push notifications in the context of an up-time cycle of the server) and configured subscription (i.e. subscription configuration retained on the server, allowing for unsolicited push notifications across up-time cycles of the server).

3.2. YANG Push via CoAP

The two CoAP operations that enable a subscription mechanism are GET and FETCH (i.e. by supporting the Observe option). Both operations are viable candidates for creating a CoAP-based YANG Push mechanism for CoMI.

3.3. Dynamic Subscriptions

Using CoAP, the client issuing the initial subscription request creates the subscription state. Examples are the GET or FETCH operation including an Observe option using an Observe parameter of 0 (zero).

3.3.1. YANG Push via GET

This usage scenario requires two consecutive operations. It is not possible to transfer a filter expression included in a GET operation. In consequence, a POST operation on a collection resource has to be conducted in order to convey a filter expression to the YANG data store, allowing it to return an URI that contains the data node information filtered in respect to the posted filter expression (encoded in CBOR).

This variant allows for multiple clients to observe a specific filtered data node without conducting a POST operation, if the corresponding URI is made known to other clients that did not conduct the POST operation or, for example, is canonically linked to/ derivable from a filter expression.

3.3.2. YANG Push via FETCH

This usage scenario requires only one operation. A FETCH operation can include a body that is capable to contain a filter expression and potentially other metadata that might be required to establish a suitable subscription state on the YANG data store.

It might be possible that this variant could introduce a slight delay in respect to response time if providing a filtered resource requires a lot of computation time on a constrained device. I.e. the resource cannot be prepared "beforehand".

3.4. Configured Subscriptions

Using CoAP, the server retains configuration that creates subscription state when the YANG data store is started. The client has to have or gain knowledge of the CoAP tokens that are included in the responses created in the context of the subscription state create from server configuration.

3.4.1. Retaining the Content of a GET Operation as Configuration

This usage scenario "mimics" the receiving of a subscription request by storing the corresponding information that are relevant for creating a subscription state as configuration on the YANG data

store. I.e. the configuration would be including the YANG client IP address and the CoAP token to be used in the responses that convey the subscribed notifications.

This variant requires that the client also knows or gains knowledge of the corresponding CoAP token in order to not discard the incoming responses.

3.4.2. Call Home via CoAP

This usage scenario defines the Call Home procedure standardized in [RFC8071] as an additional capability of CoAP. DTLS or TLS state is initiated by the YANG data store and triggers a dynamic subscription procedure of the YANG client using the session initiated by the YANG data store.

3.4.3. Dynamic Home Discovery

This usage scenario is based on the Bootstrapping Remote Secure Key Infrastructures I-D [I-D.ietf-anima-bootstrapping-keyinfra] and EST over secure CoAP I-D [I-D.vanderstok-ace-coap-est] and requires the standardization of a general use of Join Registrars in the context of YANG data stores that support YANG Push via static subscriptions.

4. IANA considerations

This document includes no requests to IANA, but solutions drafts incubated via this document might.

5. Security Considerations

This document includes no security considerations, but solution drafts incubated via this document will.

6. Acknowledgements

Carsten Bormann, Klaus Hartke, Michel Veillette

7. Change Log

First version -00

8. Normative References

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-08 (work in progress), October 2017.

[I-D.ietf-core-coap-pubsub]
Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", draft-ietf-core-coap-pubsub-02 (work in progress), July 2017.

[I-D.ietf-core-coap-tcp-tls]
Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", draft-ietf-core-coap-tcp-tls-09 (work in progress), May 2017.

[I-D.ietf-core-comi]
Veillette, M., Stok, P., Pelov, A., and A. Bierman, "CoAP Management Interface", draft-ietf-core-comi-01 (work in progress), July 2017.

[I-D.ietf-core-sid]
Veillette, M., Pelov, A., Turner, R., Minaburo, A., and A. Somaraju, "YANG Schema Item iDentifier (SID)", draft-ietf-core-sid-01 (work in progress), May 2017.

[I-D.ietf-core-yang-cbor]
Veillette, M., Pelov, A., Somaraju, A., Turner, R., and A. Minaburo, "CBOR Encoding of Data Modeled with YANG", draft-ietf-core-yang-cbor-05 (work in progress), August 2017.

[I-D.ietf-netconf-subscribed-notifications]
Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Custom Subscription to Event Notifications", draft-ietf-netconf-subscribed-notifications-05 (work in progress), October 2017.

[I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", draft-ietf-netconf-yang-push-10 (work in progress), October 2017.

[I-D.ietf-netconf-zerotouch]
Watsen, K., Abrahamsson, M., and I. Farrer, "Zero Touch Provisioning for NETCONF or RESTCONF based Management", draft-ietf-netconf-zerotouch-17 (work in progress), September 2017.

[I-D.vanderstok-ace-coap-est]
Kumar, S., Stok, P., Kampanakis, P., Furuhed, M., and S. Raza, "EST over secure CoAP (EST-coaps)", draft-vanderstok-ace-coap-est-02 (work in progress), June 2017.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <https://www.rfc-editor.org/info/rfc7641>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8071] Watsen, K., "NETCONF Call Home and RESTCONF Call Home", RFC 8071, DOI 10.17487/RFC8071, February 2017, <https://www.rfc-editor.org/info/rfc8071>.

[RFC8132] van der Stok, P., Bormann, C., and A. Sehgal, "PATCH and FETCH Methods for the Constrained Application Protocol (CoAP)", RFC 8132, DOI 10.17487/RFC8132, April 2017, <https://www.rfc-editor.org/info/rfc8132>.

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing, Haidian District
China

Email: zhoutianran@huawei.com

Xufeng Liu
Jabil
8281 Greensboro Drive, Suite 200
McLean VA 22102
USA

Email: Xufeng_Liu@jabil.com

Eric Voit
Cisco Systems

Email: evoit@cisco.com