CORECONF: Managing IoT Devices with YANG Models

Abstract

This short paper provides an overview over the CORECONF architecture for employing YANG models in managing IoT devices. CORECONF is based on CoAP as a transfer protocol and YANG-CBOR as its data representation format, analogous to the way the original RESTCONF was defined to use HTTP and YANG-XML or YANG-JSON, and the way NETCONF uses SSH and YANG-XML.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This short paper provides an overview over the CORECONF architecture for employing YANG models in managing IoT devices. CORECONF is based on CoAP [RFC7252] as a transfer protocol and YANG-CBOR [RFC9254] as its data representation format, analogous to the way the original RESTCONF [RFC8040] was defined to use HTTP [STD97] [STD98] and YANG-XML [RFC7950] or YANG-JSON [RFC7951], and the way NETCONF [RFC6241] uses SSH [RFC4251] and YANG-XML [RFC7950]. The mapping of the CORECONF interactions to CoAP is described in [I-D.ietf-core-comi]. In support of CORECONF, early contributions exist for supporting YANG metadata [RFC7952] in YANG-CBOR [I-D.bormann-cbor-yang-metadata] and for a significant performance optimization further shedding the text-based heritage of YANG [I-D.bormann-cbor-yang-standin].

2. Background

2.1. IoT

One definition of the Internet of Things is that it aims to connect devices ("Things") via the Internet that are embedded in everyday physical items, making them "available for interaction over a network, thereby enabling digital interaction with the physical world for humans, services, and/or other Things" [REST-IOT].

The number of Internet nodes that could be called "Things" in this sense has been steadily growing, and is currently estimated to be on the order of 20 billion, possibly doubling in the next eight years [IOT-SCALE].

While Things vary wildly in their technical characteristics, the scale of their deployment creates a practical limit to the power resources available to them: If we were willing to spend 1 % of the global electricity generation just on the connectedness of those Things, we would have a total power budget of about 35 GW for all of them [POWER], leaving less than 2 W on average for each of them. Since there will be many Things that need significantly more than 2 W (and 1 % is maybe too spendy at an energy cost of several dollars per year per Thing), the majority of the Things will need to get by with significantly less than 1 W average sustained power each. If operation from primary batteries is essential to the function of the Thing and battery lifetimes of a few years are desired, this upper boundary shrinks to the order of a few 10s or 100s of μW (for a single CR2032 button cell or 2 AA batteries, respectively).

Together with the need to also limit the non-energy per-device cost of ownership at this scale, this turns a significant part of the Things into devices with low resources available to them, or *constrained* devices [RFC7228] [I-D.ietf-iotops-7228bis].

Architectures that aim to support IoT therefore need to be able to accommodate constrained (low-resource) devices, even if not all IoT devices are subject to the same constraints.

2.2. IETF Network Management

In the late 1980s, at a time when the Internet Protocol Suite was still in active competition with OSI-based proposals, the IETF converged on a *simple* network management solution, SNMP (Simple Network Management Protocol, [RFC1067]).

As with the proposals it initially competed with [RFC1095], SNMP based its data modeling language on ASN.1 (standardized as [X.208] at the time, also known as ISO 8824, previously as the X.409 component of the X.400 messaging standardization project). SNMP employed ASN.1 macros as a textual convention to create the SMIv2 data modeling language as a specialization of ASN.1's eponymous datatype declaration language [STD58]. (In ISO, ASN.1 macros were not much later deprecated in favor of ASN.1 Information Objects, which could not easily be adapted to SNMP's needs; as a result, SNMP continued to stay tied to the 1988 version of ASN.1.) SMIv2 has been criticized for requiring significant, non-trivial transformations of the intended information model [RFC3444] into the limited expressiveness of its generic data model (such as constructing conceptual tables from columnar data objects, see Section 7.1 of [RFC6643] for a description of recovering the structure in an inverse process).

With its data model specification language, SNMP imported the concept of ASN.1 Object Identifiers (OIDs), identifiers represented as sequences of unsigned integers, each of which addresses one level of a hierarchy of potentially delegated registries. The representation of OIDs in ASN.1's Basic Encoding Rules (BER, [X.209]) can be reasonably efficient, and the more human-readable presentation is a simple sequence of decimal unsigned integers separated by dots. OIDs' delegation technique allows anyone to coin OIDs, once they have received the delegation of an

OID prefix ("arc") somewhere in the OID hierarchy. This keeps OIDs popular in many environments, such as in the IETF Security area, but also makes it hard to enforce any structure or semantics on namespaces represented by OIDs.

When IETF standardization of network management for IoT began in earnest in the early 2010s, SNMP was already in the process of being obsoleted by NETCONF [RFC6241] for many applications, in particular those involving remote configuration and not just monitoring. However, the simplicity and low resource requirements of the SNMP protocol compared to the XML-based NETCONF [SNMP-NETCONF] were a good match for constrained nodes, and SNMP MIBs (SMIv2 modules) were widely available for many areas of application relevant to IoT. This caused initial efforts in the IoT network management space to be based on SNMP (for example, the 6LoWPAN MIB [RFC7388]).

At the same time, the NETCONF protocol ([RFC6241], originally [RFC4741]) was getting established in the wider Internet, and with it the modeling language YANG [RFC6020]. Both efforts were closely tied to the idiosyncrasies of XML, which was popular at the time as a framework for text-based data representation formats, even though its predecessor SGML was originally created for document interchange [ISO 8879:1986].

In addition to allowing the monitoring of state data, NETCONF was specifically addressing the configuration of network devices, which was supported by SNMP implementations only in a limited way. NETCONF was designed to allow "the functionality of the management protocol to closely mirror the native functionality of the device" (Section 1 of [RFC4741]).

"The YANG data modeling language [RFC6020] has been developed for specifying NETCONF data models [...]" (Section 1.2 of [RFC6241]). Instead of employing the ASN.1 macro language used for SMIv2, YANG has been designed as a new (directly human-readable) computer language, initially with an XML-based representation format and thus generic data model (which we call YANG-XML).

Since YANG was created, the popularity of XML has faded away towards an increased popularity of JSON [STD90]. A JSON-based representation (which we call YANG-JSON, [RFC7951]) has been added to YANG with YANG 1.1 [RFC7950], for use as an alternative format that can be used with the RESTCONF protocol [RFC8040]. This has to work around both JSON's own interoperability problems [RFC7493] and the need to be bug-compatible with XML idiosyncrasies (for instance, see Section 6.10 of [RFC7951]).

Both YANG-XML and YANG-JSON build an identifier from names that resemble XML QNames (Qualified Names, i.e., colon-separated pairs of an optional prefix and a name; the prefix is usually registered). Instance-identifiers, i.e., pointers into a data tree, are constructed by separating a sequence of (qualified) names and index values with slash characters and square brackets, respectively. This is similar in concept as, but dramatically less efficient (both in space and processing time) than, the ASN.1 BER representation of OIDs known from SNMP.

2.3. CoAP

The Constrained Application Protocol (CoAP, [RFC7252]) was designed as an application protocol for constrained nodes that is based on the Representational State Transfer [REST] architecture it has in common with HTTP. Initially, CoAP provided the usual basic complement of methods known from HTTP (GET, PUT, POST, DELETE). This was later extended by three additional methods: FETCH to perform the equivalent of a GET with additional parameters sent in the body of the request, and PATCH/iPATCH (idempotent PATCH) for updating a resource based on data in the body of the request [RFC8132].

CoAP's "observe" extension can be used to automatically obtain updates to a previously obtained representation of a resource [RFC7641]. The "block-wise" extension enables the transfer of data items that would not fit into (or are otherwise considered too large for) a single packet [RFC7959].

Independent of the discussion here, CoAP is already used by ecosystem-specific IoT management protocols such as OMA's Lightweight M2M (LwM2M) [LWM2M].

3. CORECONF: Approach

In 2012, Ersue, Romascanu and Schönwälder described eight IoT use cases and derived requirements for "Management of Networks of Constrained Devices" [COMAN].

The CORECONF effort was started in 2013 by van der Stok as "CoAP Management Interfaces" (COMI) [I-D.draft-vanderstok-core-comi-00], initially focused on the simpler SNMP ecosystem, replacing just the SNMP protocol (and its security solution) by CoAP, but keeping the data model. This enabled a device to share code and complexity between the network management and the device's application protocol.

Veillette and Pelov, among others, joined the effort soon with the "Constrained Objects Language" (CoOL) proposal [I-D.draft-veillette-core-cool-00]. At the time, the management community had started work on RESTCONF [I-D.draft-bierman-netconf-restconf-00], replacing the custom NETCONF protocol by a simplified REST-based architecture (i.e., employing HTTP), which lent itself to realizing a further simplified variant over CoAP as well. CoOL provided a first representation of YANG data structures in the Concise Binary Object Representation (CBOR) format ([RFC7049], now [STD94]). This allowed leaving behind the inefficient text-based representation formats on which YANG-XML (and later YANG-JSON) were built.

CoOL also pioneered the idea of a Fully-qualified data node ID (FQDNID), which was a 31-bit integer standing in for the (text-based) YANG data name. The intention was to annotate YANG models by information that would allow deriving FQDNIDs and related numbers. The CoAP protocol would be extended by a `Fields` Option, which would extend the URI by descending into the YANG tree based on a list of FQDNIDs. (This was later replaced by a special syntax that could be used within the request URI, and ultimately by switching to CoAP's FETCH and iPATCH methods [RFC8132] instead of GET/PUT, supplying the identifying information in the CoAP request payload.)

Discussion about the best way to derive an efficient form of the YANG data name dominated the early discussion about CORECONF. Briefly, a hash-based approach was favored [I-D.draft-vanderstok-core-comi-05], but it turned out to be too hard to manage the collisions during model evolution, and hash values also didn't provide a main benefit of a more structured allocation of integer IDs: Since related IDs usually occur in a cluster, a simple delta encoding of the IDs can provide excellent encoding efficiency. A hierarchical delta encoding of IDs is the basis of today's YANG-CBOR representation [RFC9254].

To obtain an integer ID that exhibits good locality, boosting the benefits from delta-encoding, the community arrived at large (63-bit) unsigned integers. These integers are called YANG SIDs, YANG Schema Item iDentifiers, and can identify YANG schema nodes, YANG identities, YANG modules, or YANG features. The huge linear SID space is managed by employing a hierarchy of IANA registries [RFC9595], first carving up the large space into million-plus "mega-ranges", then, for an individual YANG model, allocating out of a chosen mega-range a SID range of, say, 100 integers. Each YANG model can in turn assign its individual SIDs from its SID range automatically (or even have them assigned automatically after the fact via a designated expert) or make use of an optimized assignment method tailored to the individual model [I-D.toutain-lpwan-sid-allocation].

4. Discussion

The complete CORECONF architecture is built from four specifications:

1. YANG-CBOR [RFC9254], published July 2022. This maps YANG data into CBOR as a representation format, similar to how YANG-XML [RFC7950] maps into XML and YANG-JSON [RFC7951] maps into JSON. It is itself quite efficient, but carries data in text form that YANG modules (such as those exported from [RFC6991]) define as text-based (such as date/time or IP/MAC addresses). [I-D.bormann-cbor-yang-standin] proposes a remedy. Discussion is ongoing on whether this should be added in the transparent way the current document chooses (so it can be seamlessly applied to existing modules such as those from [RFC6991]) and/or by adding information to the YANG modules in a future YANG extension. [I-D.bormann-cbor-yang-metadata] proposes a way to represent YANG metadata [RFC7952], which essentially offer compatibility to the use of XML attributes in YANG-XML, efficiently and in a much less contorted way than that defined for JSON.

2. YANG-SID [RFC9595], published July 2024. This defines the registration and assignment processes for YANG-SIDs, the efficient binary unsigned integer representation for the text-based names found in YANG-XML and YANG-JSON, as well as a representation format ("SID file") for recording the mapping between YANG names and YANG SIDs that applies to each YANG module in use. These detailed processes need to take into account that YANG SIDs will be needed for many existing YANG modules, as well as YANG modules that are newly being developed and stay in this stage for a while.

Both YANG-CBOR and YANG-SID are designed so they can be used in any YANG environment, including outside CORECONF, e.g., in a RESTCONF environment to provide an efficient binary representation there as well.

3. COMI (CoAP Management Interface for CORECONF, [I-D.ietf-core-comi]). COMI employs YANG-CBOR, YANG-SID, and CoAP to provide simplified network management operations for constrained devices. The datastore model supported today is that of a single unified datastore (with potential later extensions possible; e.g., for obtaining some of the benefits discussed in Section 2 of [RFC8342]). As a replacement for SNMP's traps and special YANG notification protocols, COMI employs CoAP's "observe" mechanism, applied to "event stream resources". A potential extension to be added in a separate document could handle the remaining problem of how to control the amount of data that may be in a response to a request operating on a non-trivial data tree, potentially including, but not limited to, pagination [I-D.ietf-netconf-list-pagination].

   Revision -16 has passed Working Group Last Call (WGLC) on 2023-09-15. This (and previous revisions) has been implemented in several research environments and, during discussions in conjunction with the May 2024 T2TRG interim meeting in Paris, was considered to meet the goals these environments planned to achieve. Based on implementation experience, the WG has since explored a number of further rounds of simplification, one of which is still outstanding; completion is considered imminent.

4. Constrained YANG Module Library [I-D.ietf-core-yang-library]. This document describes a simplified constrained version of the YANG library [RFC8525] that provides information about the YANG modules, datastores, and datastore schemas used by a constrained network management server (e.g., a CORECONF server). This document has passed Working Group Last Call (WGLC) already in 2020, but has been in hibernation since, while the other parts of CORECONF were completed. With renewed discussion about the "big" YANG library [I-D.ietf-netconf-yang-library-augmentedby], some fallout can be expected for the constrained version as well. (Note that basic resource discovery is already provided in the CoRE ecosystem by `/.well-known/core`, the Link format employed there [RFC6690], as well as the CoRE Resource Directory [RFC9176].)

In summary, CORECONF makes the use of YANG models accessible for managing low-resource devices. Beyond this immediate objective, there is a more general discussion about the use of data/interaction modeling languages for IoT in general, not just for network management.

For instance, IETF's ASDF WG has specified the Semantic Definition Format (SDF, [I-D.ietf-asdf-sdf]). SDF describes Things in terms of their interaction opportunities ("affordances"), structured into properties, actions, and events, as well as the data models behind these.

It is just a convention that YANG is used for modeling network management affordances and SDF for application-level affordances. To potentially cross-pollinate between these techniques, Kiesewalter explored bidirectional automatic translation between SDF and YANG based forms of a data/interaction model [I-D.kiesewalter-asdf-yang-sdf]. To obtain productive interworking between YANG, other relevant modeling techniques, and different fields of application, one

discussion that will become necessary is how to consolidate support for the rather different evolution patterns used by network management specifications and application interactions (APIs).

5. Security Considerations

CORECONF uses CoAP, which can be protected using DTLS [RFC6347] [RFC9147] or OSCORE [RFC8613] with EDHOC [RFC9528]. While both SNMP [STD62] [STD78] and NETCONF/RESTCONF [STD91] came with elaborate network-management-focused security data models, CORECONF acknowledges that the security models for network management of constrained devices will often need to align with application security models, such as [RFC9200] or the specific security model used in an IoT ecosystem and its approach to commissioning [NORDIC-COMMISSIONING].

6. IANA Considerations

This document has no IANA actions.

7. Informative References

[COMAN]   Ersue, M., Romascanu, D., and J. Schönwälder, "Management of Networks of Constrained Devices: Use Cases and Requirements", Work in Progress, Internet-Draft, draft-ersue-constrained-mgmt-00, 9 July 2012, <https://datatracker.ietf.org/doc/html/draft-ersue-constrained-mgmt-00>.

[I-D.bormann-cbor-yang-metadata]   Bormann, C., "Representing metadata annotations in YANG-CBOR", Work in Progress, Internet-Draft, draft-bormann-cbor-yang-metadata-00, 18 April 2024, <https://datatracker.ietf.org/doc/html/draft-bormann-cbor-yang-metadata-00>.

[I-D.bormann-cbor-yang-standin]   Bormann, C. and M. Matějka, "Stand-in Tags for YANG-CBOR", Work in Progress, Internet-Draft, draft-bormann-cbor-yang-standin-00, 21 February 2024, <https://datatracker.ietf.org/doc/html/draft-bormann-cbor-yang-standin-00>.

[I-D.draft-bierman-netconf-restconf-00]   Bierman, A., Björklund, M., Watsen, K., and R. Fernando, "RESTCONF Protocol", Work in Progress, Internet-Draft, draft-bierman-netconf-restconf-00, 4 September 2013, <https://datatracker.ietf.org/doc/html/draft-bierman-netconf-restconf-00>.

[I-D.draft-vanderstok-core-comi-00]   Van der Stok, P., "CoAp Management Interfaces", Work in Progress, Internet-Draft, draft-vanderstok-core-comi-00, 28 June 2013, <https://datatracker.ietf.org/doc/html/draft-vanderstok-core-comi-00>.

[I-D.draft-vanderstok-core-comi-05]   Van der Stok, P., Greevenbosch, B., Bierman, A., Schönwälder, J., and A. Sehgal, "CoAP Management Interface", Work in Progress, Internet-Draft, draft-vanderstok-core-comi-05, 27 October 2014, <https://datatracker.ietf.org/doc/html/draft-vanderstok-core-comi-05>.

[I-D.draft-veillette-core-cool-00]   Veillette, M. and A. Pelov, "Constrained Objects Language", Work in Progress, Internet-Draft, draft-veillette-core-cool-00, 1 November 2015, <https://datatracker.ietf.org/doc/html/draft-veillette-core-cool-00>.

[I-D.ietf-asdf-sdf]   Koster, M., Bormann, C., and A. Keränen, "Semantic Definition Format (SDF) for Data and Interactions of Things", Work in Progress, Internet-Draft, draft-ietf-asdf-sdf-18, 28 February 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-asdf-sdf-18>.

[I-D.ietf-core-comi]   Veillette, M., Van der Stok, P., Pelov, A., Bierman, A., and C. Bormann, "CoAP Management Interface (CORECONF)", Work in Progress, Internet-Draft, draft-ietf-core-comi-19, 3 November 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-core-comi-19>.

[I-D.ietf-core-yang-library]   Veillette, M. and I. Petrov, "Constrained YANG Module Library", Work in Progress, Internet-Draft, draft-ietf-core-yang-library-03, 11 January 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-core-yang-library-03>.

[I-D.ietf-iotops-7228bis]   Bormann, C., Ersue, M., Keränen, A., and C. Gomez, "Terminology for Constrained-Node Networks", Work in Progress, Internet-Draft, draft-ietf-iotops-7228bis-00, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-iotops-7228bis-00>.

[I-D.ietf-netconf-list-pagination]   Watsen, K., Wu, Q., Andersson, P., Hagsand, O., and H. Li, "List Pagination for YANG-driven Protocols", Work in Progress, Internet-Draft, draft-ietf-netconf-list-pagination-05, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-list-pagination-05>.

[I-D.ietf-netconf-yang-library-augmentedby]   Lin, Z., Claise, B., and I. D. Martinez-Casanueva, "Augmented-by Addition into the IETF-YANG-Library", Work in Progress, Internet-Draft, draft-ietf-netconf-yang-library-augmentedby-01, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-yang-library-augmentedby-01>.

[I-D.kiesewalter-asdf-yang-sdf]   Kiesewalter, J. and C. Bormann, "Mapping between YANG and SDF", Work in Progress, Internet-Draft, draft-kiesewalter-asdf-yang-sdf-01, 6 November 2021, <https://datatracker.ietf.org/doc/html/draft-kiesewalter-asdf-yang-sdf-01>.

[I-D.toutain-lpwan-sid-allocation]   Minaburo, A. and L. Toutain, "SCHC Rule Access Control", Work in Progress, Internet-Draft, draft-toutain-lpwan-sid-allocation-02, 23 February 2023, <https://datatracker.ietf.org/doc/html/draft-toutain-lpwan-sid-allocation-02>.

[IOT-SCALE]   Statista, "Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2033", June 2024, <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>.

[ISO 8879:1986]   International Organization for Standardization, "Information processing - Text and office systems - Standard generalized markup language (SGML)", ISO Standard 8879, 1986.
A scan of the paper form of ISO 8879:1986 is available at: https://nvlpubs.nist.gov/nistpubs/Legacy/FIPS/fipspub152.pdf

[LWM2M]   OMA SpecWorks, "About LwM2M", n.d., <https://www.openmobilealliance.org/lwm2m>.

[NORDIC-COMMISSIONING]   Nordic Semiconductor, "OpenThread Commissioning", n.d., <https://docs.nordicsemi.com/bundle/ncs-latest/page/nrf/protocols/thread/overview/commissioning.html>.

[POWER]   Statista, "Electricity generation worldwide from 1990 to 2023", June 2024, <https://www.statista.com/statistics/270281/electricity-generation-worldwide/>.

[REST]   Fielding, R., "Architectural Styles and the Design of Network-based Software Architectures", Ph.D. Dissertation, University of California, Irvine, ISBN 0-599-87118-0, 2000, <http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf>.

[REST-IOT]   Keränen, A., Kovatsch, M., and K. Hartke, "Guidance on RESTful Design for Internet of Things Systems", Work in Progress, Internet-Draft, draft-irtf-t2trg-rest-iot-15, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-irtf-t2trg-rest-iot-15>.

[RFC1067]   Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1067, DOI 10.17487/RFC1067, August 1988, <https://www.rfc-editor.org/rfc/rfc1067>.

[RFC1095]   Warrier, U. and L. Besaw, "Common Management Information Services and Protocol over TCP/IP (CMOT)", RFC 1095, DOI 10.17487/RFC1095, April 1989, <https://www.rfc-editor.org/rfc/rfc1095>.

[RFC3444]   Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <https://www.rfc-editor.org/rfc/rfc3444>.

[RFC4251]   Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/rfc/rfc4251>.

[RFC4741]   Enns, R., Ed., "NETCONF Configuration Protocol", RFC 4741, DOI 10.17487/RFC4741, December 2006, <https://www.rfc-editor.org/rfc/rfc4741>.

[RFC6020]   Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/rfc/rfc6020>.

[RFC6241]   Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed.,
            "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241,
            June 2011, <https://www.rfc-editor.org/rfc/rfc6241>.

[RFC6347]   Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2",
            RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/rfc/
            rfc6347>.

[RFC6643]   Schoenwaelder, J., "Translation of Structure of Management Information
            Version 2 (SMIv2) MIB Modules to YANG Modules", RFC 6643, DOI 10.17487/
            RFC6643, July 2012, <https://www.rfc-editor.org/rfc/rfc6643>.

[RFC6690]   Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690,
            DOI 10.17487/RFC6690, August 2012, <https://www.rfc-editor.org/rfc/rfc6690>.

[RFC6991]   Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/
            RFC6991, July 2013, <https://www.rfc-editor.org/rfc/rfc6991>.

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

[RFC7228]   Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node
            Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-
            editor.org/rfc/rfc7228>.

[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/rfc/rfc7252>.

[RFC7388]   Schoenwaelder, J., Sehgal, A., Tsou, T., and C. Zhou, "Definition of Managed
            Objects for IPv6 over Low-Power Wireless Personal Area Networks
            (6LoWPANs)", RFC 7388, DOI 10.17487/RFC7388, October 2014, <https://www.rfc-
            editor.org/rfc/rfc7388>.

[RFC7493]   Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493,
            March 2015, <https://www.rfc-editor.org/rfc/rfc7493>.

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

[RFC7950]   Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI
            10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/rfc/rfc7950>.

[RFC7951]   Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI
            10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/rfc/rfc7951>.

[RFC7952]   Lhotka, L., "Defining and Using Metadata with YANG", RFC 7952, DOI 10.17487/
            RFC7952, August 2016, <https://www.rfc-editor.org/rfc/rfc7952>.

[RFC7959]   Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <https://www.rfc-editor.org/rfc/rfc7959>.

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

[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/rfc/rfc8132>.

[RFC8342]   Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/rfc/rfc8342>.

[RFC8525]   Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/rfc/rfc8525>.

[RFC8613]   Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <https://www.rfc-editor.org/rfc/rfc8613>.

[RFC9147]   Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/rfc/rfc9147>.

[RFC9176]   Amsüss, C., Ed., Shelby, Z., Koster, M., Bormann, C., and P. van der Stok, "Constrained RESTful Environments (CoRE) Resource Directory", RFC 9176, DOI 10.17487/RFC9176, April 2022, <https://www.rfc-editor.org/rfc/rfc9176>.

[RFC9200]   Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments Using the OAuth 2.0 Framework (ACE-OAuth)", RFC 9200, DOI 10.17487/RFC9200, August 2022, <https://www.rfc-editor.org/rfc/rfc9200>.

[RFC9254]   Veillette, M., Ed., Petrov, I., Ed., Pelov, A., Bormann, C., and M. Richardson, "Encoding of Data Modeled with YANG in the Concise Binary Object Representation (CBOR)", RFC 9254, DOI 10.17487/RFC9254, July 2022, <https://www.rfc-editor.org/rfc/rfc9254>.

[RFC9528]   Selander, G., Preuß Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", RFC 9528, DOI 10.17487/RFC9528, March 2024, <https://www.rfc-editor.org/rfc/rfc9528>.

[RFC9595]   Veillette, M., Ed., Pelov, A., Ed., Petrov, I., Ed., Bormann, C., and M. Richardson, "YANG Schema Item iDentifier (YANG SID)", RFC 9595, DOI 10.17487/RFC9595, July 2024, <https://www.rfc-editor.org/rfc/rfc9595>.

[SNMP-NETCONF]   Sehgal, A., Perelman, V., Kuryla, S., and J. Schonwalder, "Management of
resource constrained devices in the internet of things", Institute of Electrical and
Electronics Engineers (IEEE), IEEE Communications Magazine vol. 50, no. 12, pp.
144-149, DOI 10.1109/mcom.2012.6384464, December 2012, <https://doi.org/
10.1109/mcom.2012.6384464>.

[STD58]   Internet Standard 58, <https://www.rfc-editor.org/info/std58>.
At the time of writing, this STD comprises the following:

McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of
Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/
RFC2578, April 1999, <https://www.rfc-editor.org/info/rfc2578>.

McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual
Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999,
<https://www.rfc-editor.org/info/rfc2579>.

McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance
Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999,
<https://www.rfc-editor.org/info/rfc2580>.

[STD62]   Internet Standard 62, <https://www.rfc-editor.org/info/std62>.
At the time of writing, this STD comprises the following:

Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing
Simple Network Management Protocol (SNMP) Management Frameworks", STD
62, RFC 3411, DOI 10.17487/RFC3411, December 2002, <https://www.rfc-
editor.org/info/rfc3411>.

Case, J., Harrington, D., Presuhn, R., and B. Wijnen, "Message Processing and
Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC
3412, DOI 10.17487/RFC3412, December 2002, <https://www.rfc-editor.org/info/
rfc3412>.

Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol
(SNMP) Applications", STD 62, RFC 3413, DOI 10.17487/RFC3413, December 2002,
<https://www.rfc-editor.org/info/rfc3413>.

Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of
the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, DOI
10.17487/RFC3414, December 2002, <https://www.rfc-editor.org/info/rfc3414>.

Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model
(VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC
3415, DOI 10.17487/RFC3415, December 2002, <https://www.rfc-editor.org/info/
rfc3415>.

Presuhn, R., Ed., "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, DOI 10.17487/RFC3416, December 2002, <https://www.rfc-editor.org/info/rfc3416>.

Presuhn, R., Ed., "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3417, DOI 10.17487/RFC3417, December 2002, <https://www.rfc-editor.org/info/rfc3417>.

Presuhn, R., Ed., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, DOI 10.17487/RFC3418, December 2002, <https://www.rfc-editor.org/info/rfc3418>.

[STD78]   Internet Standard 78, <https://www.rfc-editor.org/info/std78>.
            At the time of writing, this STD comprises the following:

            Schoenwaelder, J., "Simple Network Management Protocol (SNMP) Context EngineID Discovery", STD 78, RFC 5343, DOI 10.17487/RFC5343, September 2008, <https://www.rfc-editor.org/info/rfc5343>.

            Harrington, D. and J. Schoenwaelder, "Transport Subsystem for the Simple Network Management Protocol (SNMP)", STD 78, RFC 5590, DOI 10.17487/RFC5590, June 2009, <https://www.rfc-editor.org/info/rfc5590>.

            Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 5591, DOI 10.17487/RFC5591, June 2009, <https://www.rfc-editor.org/info/rfc5591>.

            Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 6353, DOI 10.17487/RFC6353, July 2011, <https://www.rfc-editor.org/info/rfc6353>.

[STD90]   Internet Standard 90, <https://www.rfc-editor.org/info/std90>.
            At the time of writing, this STD comprises the following:

            Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[STD91]   Internet Standard 91, <https://www.rfc-editor.org/info/std91>.
            At the time of writing, this STD comprises the following:

            Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[STD94]   Internet Standard 94, <https://www.rfc-editor.org/info/std94>.
          At the time of writing, this STD comprises the following:

          Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)",
          STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-
          editor.org/info/rfc8949>.

[STD97]   Internet Standard 97, <https://www.rfc-editor.org/info/std97>.
          At the time of writing, this STD comprises the following:

          Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD
          97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/
          rfc9110>.

[STD98]   Internet Standard 98, <https://www.rfc-editor.org/info/std98>.
          At the time of writing, this STD comprises the following:

          Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD
          98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <https://www.rfc-editor.org/info/
          rfc9111>.

[X.208]   International Telephone and Telegraph Consultative Committee, "Specification
          of Abstract Syntax Notation One (ASN.1)", CCITT Recommendation X.208,
          November 1988. Technically identical with ISO 8824:1988.

[X.209]   International Telephone and Telegraph Consultative Committee, "Specification
          of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)", CCITT
          Recommendation X.209, 1988. Technically identical with ISO 8825:1988.

Author's Address

Carsten Bormann
Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany
Phone: +49-421-218-63921
Email: cabo@tzi.org