REST-based Access to SMIv2-structured Information on Constrained Devices

Olaf Bergmann, Carsten Bormann, Stefanie Gerdes
Universität Bremen TZI
Bremen, Germany
Email: {bergmann|cabo|gerdes}@tzi.org

Abstract—**Recently, the IETF has generated a number of standards that are intended to be foundational for large networks of constrained devices, also known as the Internet of Things. Among these are a Web transfer protocol realizing the REST (Representational State Transfer) principles, the Constrained Application Protocol (CoAP), and the Concise Binary Object Representation (CBOR). As with the existing Internet, the Internet of Things will need network management. Management of networks has long been governed by the Structure of Management Information V2 (SMIv2), encoded in ASN.1 Basic Encoding Rules (BER) and the Simple Network Management Protocol (SNMP). Supporting multiple protocols and data encoding formats on a very constrained device is wasteful of its capabilities. We examine how CoAP and CBOR could be used for providing REST-based access to SMIv2-structured information on constrained devices in a way that facilitates interoperation with implementations of the existing protocols.**

I. INTRODUCTION

Successful operation of a network with non-trivial complexity is known to be a difficult management task in the first place. In 2002, the Internet Architecture Board (IAB) held a workshop on Network Management to identify issues with current technologies for network management and shape the future directions for protocol development [1]. At that time, the Internet of Things (IoT) was a faint vision of a few researchers and had not yet come of age. The workshop's participants thus have focused on traditional networks with few restrictions on their available processing power, memory usage, or energy supplies.

The requirements set by the Internet of Things are very different [2]. Many devices are strictly limited in their capabilities to enable battery-powered operation or cost-efficient deployment. Specialized protocols such as Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) [3] and the Constrained Application Protocol (CoAP) [4] have been developed to specifically address these limitations.

Today's predominant protocol for network management, the Simple Network Management Protocol (SNMP) [5], is reasonably light-weight, so it could be an adequate solution. The drawback is that using SNMP would imply adding a second application layer protocol stack besides CoAP which is the designated Web-transfer protocol for most IoT applications. As a consequence, it was suggested to use CoAP instead of SNMP for management operations on constrained devices [6].

We present a management interface for constrained devices based on CoAP that makes use of the Concise Binary Object Representation (CBOR) [7] to limit the bulk and energy usage of the management traffic. CBOR is a binary data format targeted at compact serialization of common data types such as integer or floating point numbers, as well as more complex objects such as arrays or maps. It is specifically designed for constrained environments where efforts to achieve compact message size must be traded against memory usage and processing complexity.

In this paper, we assume that CoAP and CBOR implementations are already present as required for use by the application realized by the constrained node, so from the point of view of the management code, the code size requirements for these two components are effectively zero. If this is not the case for a specific application, an analysis of relative code sizes compared with SNMP message processing and de-/encoding as well as MIB information de-/encoding is needed, which is outside the scope of the present paper; some relevant numbers for the SNMP side can be found in [8].

The paper is structured as follows: Section II gives some additional background on existing models for network management. Sections III and IV describe the use of CBOR and CoAP for light-weight network management. The resulting standardization requirements are discussed in Section V. Other work that is related to our approach is discussed in Section VI, followed by a concluding Section VII.

II. BACKGROUND

Management of networked devices by exposing management information objects to management stations has been an important topic for research and industry for a long time. In a networked world, IT infrastructure obviously constitutes an important asset, and therefore its functioning must be properly managed. In [9], Pavlou gives an overview of existing technical approaches that have evolved over last couple of decades.

Today, one of the most stable data models for managing network entities is SMIv2 [10]. The Internet Engineering Task Force (IETF) has defined SMIv2 modules defining specific elements of the management information base (MIB modules) for essentially every single standards track protocol. Despite of this strong level of coverage, criticism has been passed on SMIv2 for several reasons. In [11], Schönwälder suggests that several implementation errors may be attributed to the use of the outdated 1988 version of ASN.1 (ISO/IEC 8824).

978-1-4799-0913-1/14/$31.00

Closely related to SMIv2 is the use of the Simple Network Management Protocol (SNMP), c.f. RFC 3410 [5] and following. SNMP provides operations for discovery, retrieval, and modification of objects in a MIB, as well as optionally asynchronous change notifications. As SNMP has no atomic operation for retrieval or modification of complex data structures, inconsistent data may occur at the sender or the receiver side [11]. In today's deployments, SNMP is predominantly used for reading information from devices, much less for setting them up (configuration).

A different approach for managing network devices is followed with the Network Configuration Protocol (NETCONF) [12]. The protocol uses a document-oriented data model, operations are represented by remote procedure calls that are encoded in XML. This allows for more complex operations such as updating several information objects at once, attribute matching, or filtering subtrees.

Closely related to NETCONF is the data modeling language YANG [13]. The YANG vocabulary and grammar are tailored specifically to network device configuration and management, and its XML serialization allows for direct use with NETCONF. SMIv2 MIB modules can be translated automatically into YANG in order to make its adoption easier by leveraging the huge amount of specification work that went into existing MIB modules.

Both SNMP and NETCONF come with a number of security mechanisms each that are specific to these protocols and the transport mechanisms used for them. While the detailed discussion of these security mechanisms is beyond the scope of this document, it should be noted that a translator between SNMP and NETCONF, or a translator from and to a third protocol, will typically need to terminate these security mechanisms: that is it will need to act as a trusted party.

To make management information accessible to REST-based access using Web transfer protocols such as CoAP, two areas need to be addressed: first, a naming scheme for management objects and a data format for these names is required so that the objects can be addressed by a URI, and, second, a data format for encoding the actual information is needed so that they can be transported in the payload data of the Web transfer protocol.

III. DATA FORMAT: URIS

SMIv2 [10] uses a hierarchical name space of Object Identifiers (OIDs), which are structured as sequences of unsigned numbers between 0 and $2^{32} - 1$. As REST-based applications require resources to be named by URIs, these identifiers need to be mapped to CoAP URIs. Based on the URI scheme for SNMP specified in [14], we define a simple URI format for management resources. First, the scheme identifier `snmp` is replaced by `coap`. Instead of `snmp-authority`, only `host` and `port` are supported as the CoAP URI scheme has no `userinfo` component. To make use of the existing path decoder function of the CoAP server, an OID is represented as a sequence of path components that each represent one unsigned number, i.e. the production `oid` uses `%x2F` (solidus) instead of `DOT`.

We further propose restricting the optional `suffix` to `"/*"`. The semantics of `snmp` URIs that end in + (a `GetNext`) does not perfectly map to CoAP: it would require redirecting the sender of that request to the actual resource to prevent a mismatch with the REST principles (CoAP at this time does not support redirects).

Examples for CoAP management URIs are as follows:

```
coap://example.com/ctx1/1/3/6/1/2/1/1/3/0
coap://example.com//1/3/6/1/2/1/1/3/*
```

These two examples show the `sysUpTime.0` object instance in the SNMPv2-MIB (see [14] for the corresponding SNMP URI). The first line indicates that the OID is to be interpreted in `ctx1` at the host `example.com`, while the second line uses the default context. (The latter will be represented in a CoAP request as an empty Uri-Path Option which consumes a single byte.)

An important aspect to note is that in some deployments, the set of context identifiers that need to be supported could possibly conflict with other resources on a device. The base of the management URI therefore should therefore not necessarily be bound to the root of the path space. The actual base should be published in a resource directory [15] with the resource type "mgmt". The following example depicts the description of a management base URI stored in a resource directory:

```
</management//>;rt="mgmt";if="core.b"
```

Here, the management interface is located below the subtree "management" using the default context (more precisely speaking, the URI reference in the link could be considered a URI template). The attribute `rt="mgmt"` indicates that this URI reference denotes the base of the management interface. An additional attribute `if="core.b"` qualifies this resource as a collection of subresources that implements the batch interface (see [16]).

The URI format presented so far is suitable particularly for GET requests to retrieve the current values of a single managed object or an entire subtree of managed objects. Depending on the number of managed objects within the specified subtree, the latter might cause large responses from the server. In this case, the server should use block-wise transfer according to [17] to split the response in smaller parts.

[14] also offers the capability to specify multiple OIDs in one URI (OID groups). We propose not to include this capability for constrained devices. To request multiple OIDs in a single transaction, the OIDs could to be conveyed within the payload of a POST request. The request URI is then set to the root of the management subtree (including the context name and context engine identifier, if present). The OIDs in the POST payload would be encoded similarly to the GET response payload encoding described in the following section.

IV. DATA FORMAT: PAYLOAD

Traditionally, SMIv2 data is encoded in ASN.1 BER (Basic Encoding Rules) — in fact, SMIv2 itself is described in macros as they were offered by a previous version of the ASN.1 language. In a constrained device, code space is available only for a limited number of data format encoders/decoders. One data format that has been specifically designed for the requirements of constrained node applications is CBOR, the Concise Binary Object Representation [7]. Here, we set out to examine whether CBOR is useful also for representing SMIv2-structured management information, or actually beneficially so.

To be able to represent SMIv2 data in CBOR, the SMIv2 data model needs to be mapped into the CBOR data model, which is an extension of JSON's data model[18]. One way to do so is to convert SMIv2 into YANG-structured data [13], using the SMIv2 identifiers as YANG identifiers. This can then in turn be translated into JSON using data-model driven XML to JSON translation [19]. The JSON-modeled data can then be trivially represented in CBOR [7]. While this approach does appear to work [6], it requires the transfer of those identifiers from the constrained nodes, increasing the bulk of the packets and requiring storage of all these identifiers on the constrained nodes.

One way to avoid at least some of the overhead of transporting these identifiers is to devise a compression scheme, However, compression is expensive in CPU power (and thus energy), code space and state. To achieve good compression, the redundancy between packets needs to be exploited, which doesn't fit well with the stateless approach common between SNMP and REST.

One other way is to use the numerically structured ASN.1 Object ID (OID) data already defined in the SMIv2-structured MIB modules. Using the IP MIB [20] as an example, instead of using the JSON-style structure:

```
{"ipForwarding": 1,
 "ipDefaultTTL": 255,
 "ipInReceives": 4208076309,
 "ipForwDatagrams": 2923451775}
```

(69 bytes when represented in CBOR), we might use:

```
{ [1, 3, 6, 1, 2, 1, 4, 1, 0]: 1,
  [1, 3, 6, 1, 2, 1, 4, 2, 0]: 255,
  [1, 3, 6, 1, 2, 1, 4, 3, 0]: 4208076309,
  [1, 3, 6, 1, 2, 1, 4, 6, 0]: 2923451775}
```

(54 bytes, shown in CBOR diagnostic notation). While the gain from this first transformation is limited, significant savings, as well as simplification, is achieved by then factoring out common prefixes:

```
{ [1, 3, 6, 1, 2, 1, 4]:
  { [1, 0]: 1,
    [2, 0]: 255,
    [3, 0]: 4208076309,
    [6, 0]: 2923451775}}
```

(35 bytes). This structure not only mirrors likely implementations on constrained devices. It also can be easily translated back into SNMP without any knowledge of the MIB modules involved, facilitating the insertion of application-agnostic gateways between traditional SNMP management stations and CoAP-controlled constrained devices.

Limited additional savings could then be achieved by:

- replicating the ASN.1 BER trick of transforming the first two numbers of the OID into one,
- translating sequences of numbers in OIDs that happen to be below 256 into byte strings instead of arrays of unsigned numbers.

TABLE I. REPRESENTATION SIZE IN AN IP-MIB TRACE

Form	Size (Bytes)
ASN.1 BER	2,804,755
CBOR unfactored	2,677,461
CBOR factored	786,912

Table I shows the sizes of a number of representation forms for a larger trace of IP-MIB SNMP responses. As in the simple example above, the unfactored CBOR form is only a small improvement over the equivalent ASN.1 BER form, while the factored form (without any of the further optimizations) yields a representation that is about 70 % smaller than the unfactored one.

V. STANDARDIZATION REQUIREMENTS

To enable the use of SMIv2-structured information in a REST ecosystem, a number of specifications are required. The translation algorithm outlined in Section IV needs to be formalized, also itemizing how SMIv2's data types (*textual conventions*) are represented in the CBOR data model. The semantics of CoAP GET and PUT methods need to be defined, probably in terms of the equivalent SNMP operations.

Where SNMP provides traps or INFORMs, CoAP offers Observation Relationships [21]; the utility of these, for management requirements that are traditionally covered by traps and INFORMs, needs to be examined more closely. In its most basic form, a client indicates interest in observing a management resource by sending a non-confirmable GET request with the resource's URI and an Observe option included. The server then can send update notifications on this particular resource any time its has significantly changed. The matching between the initial request that has established the observation relationship and the update notifications is based on the observer's transport address and the token used in the CoAP message. As a consequence, there is no need to repeat the OID of the subscribed resource in the notification, potentially saving some bandwidth as OIDs usually are significantly longer than even the maximum CoAP token length.

Observation notifications can be sent in confirmed or non-confirmed messages. If reliable delivery of notifications is desired, a confirmed message will be used that can be retransmitted a number of times until an acknowledgement is received. Even when non-confirmed messages are used, the server (the device that offers a management resource) can detect if the observing client no longer exists by sending confirmable notifications once in a while and eventually remove a non-acknowledging client from its internal list of observers.

Observation replaces the process of polling by a management station with active sending of updates. As the latter can fit into the sleep schedule of the constrained device emitting them, this is immediately useful. A translating gateway could create Observation Relationships to the CoAP servers it serves and store data for independent retrieval by the SNMP monitoring stations using it. To do this properly in a REST framework, CoAP servers would need to provide lifetime data (CoAP Option "MaxAge").

As CoAP transfers representations with defined Internet media types ("MIME types"), the payload of the CoAP response needs to be tagged with such a media type. We suggest registering `application/smiv2+cbor` for this purpose, along with registering a numeric Content-Format option value for CoAP.

VI. RELATED WORK

The use of SNMP or NETCONF for managing devices in constrained environments has been investigated by Sehgal et al. in [22]. The comparison shows that SNMPv1 can be implemented in less than 9 KB Code and uses ∼700 B RAM size (stack and data). To use SNMPv3 with security, more than 30 KB Code and ∼1.5 KB RAM for stack and data are required. While the NETCONF protocol implementation uses less code, additional code and dynamic memory for Transport Layer Security (TLS) is required. Moreover, the handling of XML documents according to the NETCONF specification is done in flash memory using Contiki's Coffee File System. As stated in [22], generating the output document tag-by-tag requires up to ten file write-cycles. As a result, using NETCONF on constrained devices for more than occasional retrieval or update of configuration data, the wear of the flash memory chips can significantly impact a device's lifetime, depending on the chips' erase block size and endurance cycles.

Given the success of REST-based design especially regarding scalability of applications that use HTTP for conveying data, it is a natural next step to apply these principles to other transfer protocols as well. In [23], Bierman et al. present a profile for NETCONF over HTTP that adheres to this architectural principle. Strictly following the REST-based design, the authors have opted for hypermedia-driven application state (c.f. [24]) so that the client does not have to know the data model (MIB) at all. Constrained devices benefit from this approach as it saves both, code and memory. On the other hand, responses from the server grow bigger as they need to convey the next possible transitions from the current state. Depending on the client's limitations, this could be even worse due to the additional energy and dynamic memory required for receiving and processing packets.

A straight-forward approach for managing constrained devices using CoAP and CBOR is described in [6]. A device that supports this protocol offers two specific resources for accessing the device's MIB and a translation table that maps descriptors to integer values, respectively. Single MIB variables are represented as a CoAP URI with a common prefix followed by either its descriptor or the corresponding OID, multiple values are addressed via the CoAP payload. As mentioned in Section IV the translation table provides the mapping between MIB descriptors and numeric constants but also comes with significant overhead for a client. The approach proposed in the present paper enables clients to directly use the numeric object identifiers from MIB modules without any translation step.

VII. CONCLUSIONS

This paper sketches an approach for translating SNMP-based management into the REST-based primitives that have been standardized by the IETF for use in constrained node networks. Some of the devices that SNMP was originally developed to work on would be considered somewhat constrained devices today, so it is not surprising that applying some of SNMP's original principles, such as the identification of managed objects by sequences of small numbers, provides benefits today. Replacing the SNMP protocol by CoAP not only saves the constrained device from needing one additional protocol implementation, it also helps to reduce the number of data formats that need to be supported in many cases. It may also allow focusing on a single security environment for authorization of both application and management operations. Including a number of data items in one response and making use of CBOR's structuring capabilities for factoring out some identification data may allow significant savings, such as 70 % less data volume in one trace. The CoAP protocol itself is a relatively well-fitting replacement for the SNMP request-response protocol. Other SNMP principles such as SNMP URIs can be adapted to CoAP with little loss in commonality.

REFERENCES

[1] J. Schoenwaelder, "Overview of the 2002 IAB Network Management Workshop," Internet Requests for Comments, RFC Editor, RFC 3535, May 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3535.txt

[2] M. Ersue, D. Romascanu, and J. Schoenwaelder (Eds.), "Management of Networks with Constrained Devices: Problem Statement, Use Cases and Requirements," Internet Engineering Task Force, Internet-Draft draft-ersue-constrained-mgmt-03, February 2013, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ersue-constrained-mgmt-03.txt

[3] Z. Shelby (Ed.), S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," Internet Requests for Comments, RFC Editor, RFC 6775, November 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6775.txt

[4] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained Application Protocol (CoAP)," Internet Engineering Task Force, Internet-Draft draft-ietf-core-coap-18, June 2013, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-core-coap-18.txt

[5] J. Case, R. Mundy, D. Partain, and B. Stewart, "Introduction and Applicability Statements for Internet Standard Management Framework," Internet Requests for Comments, RFC Editor, RFC 3410, December 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3410.txt

[6] P. van der Stok and B. Greevenbosch, "CoAp Management Interfaces," Internet Engineering Task Force, Internet-Draft draft-vanderstok-core-comi-02, January 2014, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-vanderstok-core-comi-02.txt

[7] C. Bormann and P. Hoffman, "Concise Binary Object Representation (CBOR)," Internet Engineering Task Force, RFC 7049, October 2013. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7049.txt

[8] S. Kuryla and J. Schönwälder, "Evaluation of the resource requirements of snmp agents on constrained devices," in *Proceedings of the 5th International Conference on Autonomous Infrastructure, Management, and Security: Managing the Dynamics of Networks and Services*, ser. AIMS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 100–111. [Online]. Available: http://dl.acm.org/citation.cfm?id=2022216.2022233

[9] G. Pavlou, "On the evolution of management approaches, frameworks and protocols: A historical perspective," *Journal of Network and Systems Management*, vol. 15, no. 4, pp. 425–445, Oct. 2007, 00031. [Online]. Available: http://link.springer.com/10.1007/s10922-007-9082-9

[10] K. McCloghrie, D. Perkins, J. Schoenwaelder *et al.*, “Structure of Management Information Version 2 (SMIv2),” Internet Requests for Comments, RFC Editor, RFC 2578, April 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2578.txt

[11] J. Schönwälder, “Protocol-independent data modeling: Lessons learned from the SMIng project,” *Communications Magazine, IEEE*, vol. 46, no. 5, p. 148–153, 2008, 00007. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4511663

[12] R. Enns, M. Bjorklund, J. Schoenwaelder and A. Bierman (Eds.), “Network Configuration Protocol (NETCONF),” Internet Requests for Comments, RFC Editor, RFC 6241, June 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6241.txt

[13] M. Bjorklund (Ed.), “YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF),” Internet Requests for Comments, RFC Editor, RFC 6020, October 2010. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6020.txt

[14] D. Black, K. McCloghrie, and J. Schoenwaelder, “Uniform Resource Identifier (URI) Scheme for the Simple Network Management Protocol (SNMP),” Internet Requests for Comments, RFC Editor, RFC 4088, June 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4088.txt

[15] Z. Shelby, C. Bormann, and S. Krco, “CoRE Resource Directory,” Internet Engineering Task Force, Internet-Draft draft-ietf-core-resource-directory-01, December 2013, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-core-resource-directory-01.txt

[16] Z. Shelby and M. Vial, “CoRE Interfaces,” Internet Engineering Task Force, Internet-Draft draft-ietf-core-interfaces-01, December 2013, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-core-interfaces-01.txt

[17] C. Bormann and Z. Shelby (Ed.), “Blockwise transfers in CoAP,” Internet Engineering Task Force, Internet-Draft draft-ietf-core-block-14, October 2013, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-core-block-14.txt

[18] D. Crockford, “The application/json Media Type for JavaScript Object Notation (JSON),” RFC 4627 (Informational), Internet Engineering Task Force, July 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4627.txt

[19] L. Lhotka, “Modeling JSON Text with YANG,” Internet Engineering Task Force, Internet-Draft draft-lhotka-netmod-yang-json-02, September 2013, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-lhotka-netmod-yang-json-02.txt

[20] S. Routhier (Ed.), “Management Information Base for the Internet Protocol (IP),” Internet Requests for Comments, RFC Editor, RFC 4293, April 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4293.txt

[21] K. Hartke, “Observing Resources in CoAP,” Internet Engineering Task Force, Internet-Draft draft-ietf-core-observe-11, October 2013, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-core-observe-11.txt

[22] A. Sehgal, V. Perelman, S. Kuryla, and J. Schönwälder, “Management of Resource Constrained Devices in the Internet of Things,” *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 144–149, 2012.

[23] A. Bierman, M. Bjorklund, K. Watsen, and R. Fernando, “RESTCONF Protocol,” Internet Engineering Task Force, Internet-Draft draft-bierman-netconf-restconf-03, December 2013, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-bierman-netconf-restconf-03.txt

[24] S. Parastatidis, J. Webber, G. Silveira, and I. S. Robinson, “The role of hypermedia in distributed system development,” in *Proceedings of the First International Workshop on RESTful Design*, ser. WS-REST ’10. New York, NY, USA: ACM, 2010, pp. 16–22. [Online]. Available: http://doi.acm.org/10.1145/1798354.1798379