NETCONF WG
Internet-Draft
Updates: 6241 (if approved)
Intended status: Standards Track
Expires: January 1, 2019

M. Jethanandani
J. Lam
A. Leung
Cisco Systems, Inc.
A. Bierman
YumaWorks, Inc.

June 30, 2018

Binary Encoding for NETCONF
draft-mahesh-netconf-binary-encoding-01

Abstract

This document describes a method by which a NETCONF [RFC6241] client and server can negotiate an alternate form of encoding.

This document updates RFC 6241.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

NETCONF [RFC6241], by default, supports XML encoding for its payload. However, XML can be very verbose, specially for operational data. This document proposes a mechanism by which clients and servers can negotiate an alternate form of encoding, e.g. binary encoding, and use that to exchange data. Binary encoding can reduce the physical size of the data exchanged, in some cases by as much as 66%, while preserving the original data.

Several encoding mechanisms have been proposed, including CBOR [RFC7049] and JSON [RFC8259]. This document does not advocate any particular encoding format. Instead, it leaves it up to the negotiation between client and server to decide the form of encoding. For an example of how to encode YANG in CBOR format, see CBOR Encoding of Data Modeled with YANG [I-D.ietf-core-yang-cbor] and JSON Encoding of Data Modeled with YANG [RFC7951].

This document updates NETCONF [RFC6241].

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Protocol Negotiation

NETCONF clients and servers exchange a hello as part of establishing a connection. As part of the hello exchange, the client advertises an ordered list of encoding it would like to use, while the server advertises an unordered list of encoding that it is capable of supporting. If no match of encoding is found, the session is dropped. If a match is found the client issues a request that is encoded with first match found. Thereafter, both the Message layer in Figure 1 of NETCONF [RFC6241] and the YANG data within the Message Layer, are in the form of new encoding. This includes RPC, Actions and Notifications.

This draft suggests advertisement of the following additional capability.

2.1. Encoding

2.1.1. Overview

The :encoding capability indicates what alternate encoding format each side is willing to support. The client and server send a comma separated list (with no white spaces) of encoding formats they are willing to support. The client sends a list of encoding ordered by preference, while the server includes an unordered list of encoding.

Both the client and server examine each others <hello> message for this capability. If not present, the default encoding is used, which is XML. The client examines its list against the server list, checked in the order of preference it sent do the server. If a matching encoding format is found, the client picks that encoding for the remainder of the session, starting with the first <rpc> request. All <rpc>, <rpc-reply>, <action> and <notification> messages MUST be encoded in this negotiated encoding.

Both the client and the server MUST support the "application/xml" media type to be backward compatible with NETCONF [RFC6241].

The base:1.1 negotiation defined in NETCONF [RFC6241] determines the message framing that is used for the entire session.

2.1.2. Example

In this example, the client supports the following encoding formats shown in a preferred order.

- Concise Binary Object Representation (CBOR) with YANG Schema Item iDentifier (SID) - cbor+sid

- Google Protocol Buffers (GPB) - gpb

- Thrift - thrift

In this example, the client advertises its (abbreviated) <hello> as follows. Some extra white spaces have been added for display purposes only.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
    <capability>urn:ietf:params:netconf:base:1.1</capability>
    <capability>
      urn:ietf:params:netconf:capability:encoding:1.0?format=
      application/cbor+sid,application/gpb,application/thrift
    </capability>
  </capabilities>
</hello>
```

The server supports the following encoding formats shown in no particular order of preference.

- cbor+sid

- gpb

In this example, the server advertises its (abbreviated) <hello> as follows. Some white spaces have been added for display purposes only.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
    <capability>urn:ietf:params:netconf:base:1.1</capability>
    <capability>
      urn:ietf:params:netconf:capability:encoding:1.0?
      format=application/cbor+sid,application/gpb
    </capability>
    <capability>
      urn:ietf:params:netconf:capability:config-id?id=2130
    </capability>
    <!--  rest of URIs ... -->
 </capabilities>
 <session-id>4</session-id>
</hello>
```

The common encoding formats in both the list are "application/cbor+sid", and "application/gpb", but since cbor+sid appear first on the client list, "application/cbor+sid" is selected as the form of encoding for the remainder of the session.

2.1.3. Dependencies

None.

2.1.4. Capability Identifier

The :encoding capability is identified by the following capability string:

```
urn:ietf:params:netconf:capability:encoding:1.0?format={name, ...}
```

The :encoding capability MUST be advertised in every server <hello> message and the URI MUST contain a "format" argument assigned a comma-separated list (with no white spaces) of formats supported by the device. For the list of supported formats, this document requests the creation of a new registry. See IANA Considerations for details.

The client on the other hand SHOULD advertise this capability in its <hello> message, but it MAY omit it if XML encoding is desired.

For example (line wrapped for display purposes only)

```
urn:ietf:params:netconf:capability:encoding:1.0?format=
application/cbor+sid,application/gpb,application/thrift
```

2.1.5. New Operations

The :encoding capability does not introduce any new protocol operation.

2.1.6. Modification to Existing Operations

The :encoding capability does not modify any existing protocol operations.

2.1.7. Interactions with Other Capabilities

The :encoding capability does not interact with any other capabilities.

2.2. CBOR and SID

One of the encoding formats that can be advertised by the client or the server is CBOR [RFC7049]. The payload consists of YANG [RFC7950] data, and YANG requires the use of unique identifiers, implemented in NETCONF [RFC6241] using names. To allow for encoding of YANG data models, a more compact method has been identified, called YANG Schema Item iDentifier (SID) [I-D.ietf-core-yang-cbor]. Clients and servers can advertise their capability for this form of encoding using "application/cbor+sid".

SID does not define encoding for NETCONF operations today. It is expected that a new SID range would have to be identified for NETCONF protocol operations.

3. Security Considerations

4. IANA Considerations

This document registers a URI and requests the creation of a new registry.

4.1. NETCONF Capability URNs

This document requests the registry of an URI in the IETF XML registry [RFC3688]. The IANA registry "Network Configuration Protocol (NETCONF) Capability URNs" needs to be updated to include the following capability.

Index	Capability Identifier
:encoding	urn:ietf:params:netconf:capability:encoding:1.0

4.2. New Registry

The document also requests the creation of a new registry, called "Network Configuration Protocol (NETCONF) Encoding formats", that should be populated with the following entries.

Encoding Formats
cbor+sid
gpb
thrift

5. Acknowledgements

The authors would like to thank Juergen Schoenwaeider for his comments on the draft.

6. References

6.1. Normative References

[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-06 (work in progress), February 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[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/info/rfc6241>.

[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>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8259] 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>.

6.2. Informative References

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

Authors’ Addresses

Mahesh Jethanandani

Email: mjethanandani@gmail.com

Jason Lam
Cisco Systems, Inc.

Email: lamj@cisco.com

Alfred Leung
Cisco Systems, Inc.

Email: alfleung@cisco.com

Andy Bierman
YumaWorks, Inc.

Email: andy@yumaworks.com