Comparison of NMDA datastores
draft-clemm-netmod-nmda-diff-00

Abstract

   This document defines an RPC operation to compare management
   datastores that comply with the NMDA architecture.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   The revised Network Management Datastore Architecture (NMDA)
   [RFC8342] introduces a set of new datastores that each hold YANG-
   defined data [RFC7950] and represent a different "viewpoint" on the
   data that is maintained by a server. New YANG datastores that are
   introduced include <intended>, which contains validated configuration
   data that a client application intends to be in effect, and
   <operational>, which contains at least conceptually operational state
   data (such as statistics) as well as configuration data that is
   actually in effect.

   NMDA introduces in effect a concept of "lifecycle" for management
   data, allowing to clearly distinguish between data that is part of a
   configuration that was supplied by a user, configuration data that
   has actually been successfully applied and that is part of the
   operational state, and overall operational state that includes both
   applied configuration data as well as status and statistics.

   As a result, data from the same management model can be reflected in
   multiple datastores. Clients need to specify the target datastore to
   be specific about which viewpoint of the data they want to access.
   This way, an application can differentiate whether they are (for
   example) interested in the configuration that has been applied and is

actually in effect, or in the configuration that was supplied by a
client and that is supposed to be in effect.

Due to the fact that data can propagate from one datastore to
another, it is possibly for differences between datastores to occur.
Some of this is entirely expected, as there may be a time lag between
when a configuration is given to the device and reflected in
<intended>, until when it actually takes effect and is reflected in
<operational>. However, there may be cases when a configuration item
that was to be applied may not actually take effect at all or needs
an unusually long time to do so. This can be the case due to certain
conditions not being met, resource dependencies not being resolved,
or even implementation errors in corner conditions.

When configuration that is in effect is different from configuration
that was applied, many issues can result. It becomes more difficult
to operate the network properly due to limited visibility of actual
status which makes it more difficult to analyze and understand what
is going on in the network. Services may be negatively affected (for
example, breaking a service instance resulting in service is not
properly delivered to a customer) and network resources be
misallocated.

Applications can potentially analyze any differences between two
datastores by retrieving the contents from both datastores and
comparing them. However, in many cases this will be at the same time
costly and extremely wasteful.

This document introduces a YANG data model which defines RPCs,
intended to be used in conjunction with NETCONF [RFC6241] or RESTCONF
[RFC8040], that allow a client to request a server to compare two
NMDA datastores and report any differences.

2. Key Words

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.

3. Definitions and Acronyms

NMDA: Network Management Datastore Architecture

RPC: Remote Procedure Call

4. **Data Model Overview**

   At the core of the solution is a new management operation, <compare>,
   that allows to compare two datastores for the same data. The
   operation checks whether there are any differences in values or in
   data nodes that are contained in either datastore, and returns any
   differences as output. The output is returned in the format
   specified in YANG-Patch [RFC8072].

   The YANG data model defines the <compare> operation as a new RPC.
   The operation takes the following input parameters:

   o source: The source identifies the datastore that will serve as
      reference for the comparison, for example <intended>.

   o target: The target identifies the datastore to compare against the
      source.

   o filter-spec: This is a choice between different filter constructs
      to identify the portions of the datastore to be retrieved. It
      acts as a node selector that specifies which data nodes are within
      the scope of the comparison and which nodes are outside the scope.
      This allows a comparison operation to be applied only to a
      specific portion of the datastore that is of interest, such as a
      particular subtree. (The filter dow not contain expressions that
      would match values data nodes, as this is not required by most use
      cases and would complicate the scheme, from implementation to
      dealing with race conditions.)

   o all: When set, this parameter indicates that all differences
      should be included, including differences pertaining to schema
      nodes that exist in only one of the datastores. When this
      parameter is not included, a prefiltering step is automatically
      applied to exclude data from the comparison that does not pertain
      to both datastores: if the same schema node is not present in both
      datastores, then all instances of that schema node and all its
      descendants are excluded from the comparison. This allows client
      applications to focus on the differences that constitute true
      mismatches of instance data without needing to specify more
      complex filter constructs.

   The operation provides the following output parameter:

   o differences: This parameter contains the list of differences,
      encoded per RFC8072, i.e. specifying which patches would need to
      be applied to the source to produce the target. When the target
      datastore is <operational>, "origin" metadata is included as part
      of the patch. Including origin metadata can help explain the

cause of a difference, for example when a data node is part of
<intended> but the origin of the same data node in <operational>
is reported as "system".

The data model is defined in the ietf-nmda-compare YANG module. Its
structure is shown in the following figure. The notation syntax
follows [RFC8340].

```
module: ietf-nmda-compare

  rpcs:
    +---x compare
       +---w input
       |  +---w source          identityref
       |  +---w target          identityref
       |  +---w all?            empty
       |  +---w (filter-spec)?
       |     +--:(subtree-filter)
       |     |  +---w subtree-filter?   <anydata>
       |     +--:(xpath-filter)
       |        +---w xpath-filter?     yang:xpath1.0 {nc:xpath}?
       +--ro output
          +--ro (compare-response)?
             +--:(no-matches)
             |  +--ro no-matches?    empty
             +--:(differences)
                +--ro differences
                   +--ro yang-patch
                      +--ro patch-id    string
                      +--ro comment?    string
                      +--ro edit* [edit-id]
                         +--ro edit-id      string
                         +--ro operation    enumeration
                         +--ro target       target-resource-offset
                         +--ro point?       target-resource-offset
                         +--ro where?       enumeration
                         +--ro value?       <anydata>
```

Structure of ietf-nmda-compare

5. YANG Data Model

<CODE BEGINS> file "ietf-nmda-compare@2018-06-12.yang"
module ietf-nmda-compare {

  yang-version 1.1;

```
namespace "urn:ietf:params:xml:ns:yang:ietf-nmda-compare";

prefix cp;

import ietf-yang-types {
  prefix yang;
}
import ietf-datastores {
  prefix ds;
}
import ietf-yang-patch {
  prefix ypatch;
}
import ietf-netconf {
  prefix nc;
}

organization "IETF";
contact
  "WG Web:   <http://tools.ietf.org/wg/netconf/>
   WG List:  <mailto:netconf@ietf.org>

   Author: Alexander Clemm
           <mailto:ludwig@clemm.org>

   Author: Yingzhen Qu
           <mailto:yingzhen.qu@huawei.com>

   Author: Jeff Tantsura
           <mailto:jefftant.ietf@gmail.com>

   Author: Andy Bierman
           <mailto:andy@yumaworks.com>";

description
  "The YANG data model defines a new operation, <compare>, that
   can be used to compare NMDA datastores.";

revision 2018-06-12 {
  description
    "Initial revision";
  reference
    "RFC XXXX: Comparison of NMDA datastores";
}

/* RPC */
rpc compare {
  description
```

```
        "NMDA compare operation.";
      input {
        leaf source {
          type identityref {
            base ds:datastore;
          }
          mandatory true;
          description
            "The source datastore to be compared.";
        }
        leaf target {
          type identityref {
            base ds:datastore;
          }
          mandatory true;
          description
            "The target datastore to be compared.";
        }
        leaf all {
              type empty;
          description
            "When this leaf is provided, all data nodes are compared,
                     whether their schema node pertains to both datastores or
             not. When this leaf is omitted, a prefiltering step is
             automatically applied that excludes data nodes from the
             comparison that can occur in only one datastore but not
             the other.  Specifically, if one of the datastores
             (source or target) contains only configuration data and
             the other datastore is <operational>, data nodes for
             which config is false are excluded from the comparison.";
        }
        choice filter-spec {
          description
            "Identifies the portions of the datastores to be
                 compared.";
          anydata subtree-filter {
            description
              "This parameter identifies the portions of the
               target datastore to retrieve.";
            reference "RFC 6241, Section 6.";
          }
          leaf xpath-filter {
            if-feature nc:xpath;
            type yang:xpath1.0;
            description
              "This parameter contains an XPath expression
               identifying the portions of the target
               datastore to retrieve.";
```

```
          }
        }
      }
      output {
        choice compare-response {
          leaf no-matches {
            type empty;
            description
              "This leaf indicates that the filter did not match anything
               and nothing was compared.";
          }
          container differences {
            uses ypatch:yang-patch;
            description
              "The list of differences, encoded per RFC8072.";
          }
          description
            "Comparision results.";
        }
      }
    }
  }
}
<CODE ENDS>
```

6. Example

   The following example compares the difference between <operational>
   and <intended> for object "explicit-router-id", as defined in data
   module [I-D.ietf-ospf-yang].

   RPC request:

```
      <rpc message-id="101"
          xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <compare xmlns="urn:ietf:params:xml:ns:yang:ietf-nmda-compare"
            xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
          <source>ds:operational</source>
          <target>ds:intended</target>
          <xpath-filter
            xmlns:rt="urn:ietf:params:xml:ns:yang:ietf-routing"
            xmlns:ospf="urn:ietf:params:xml:ns:yang:ietf-ospf">\
              /rt:routing/rt:control-plane-protocols\
              /rt:control-plane-protocol/ospf:ospf\
          </xpath-filter>
        </compare>
      </rpc>
```

RPC reply, when a difference is detected:

```
<rpc-reply
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="101">
  <differences
      xmlns="urn:ietf:params:xml:ns:yang:ietf-nmda-compare"
      xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin">
    <yang-patch>
      <patch-id>ospf router-id</patch-id>
      <comment>diff between operational and intended</comment>
      <edit>
        <edit-id>1</edit-id>
        <operation>replace</operation>
        <target>/ietf-ospf:explicit-router-id</target>
        <value>
          <ospf:explicit-router-id
             or:origin="or:system">1.1.1.1<explicit-router-id>
        </value>
      </edit>
    </yang-patch>
  </differences>
</rpc-reply>
```

RPC reply when no difference is detected:

```
<rpc-reply
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  message-id="101">
  <differences
    xmlns="urn:ietf:params:xml:ns:yang:ietf-nmda-compare"/>
</rpc-reply>
```

The same request in RESTCONF (using JSON format):

```
POST /restconf/operations/ietf-nmda-compare:compare HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json
Accept: application/yang-data+json

{ "ietf-nmda-compare:input" {
   "source" : "ietf-datastores:operational",
   "target" : "ietf-datastores:intended".
   "xpath-filter" : \
       "/ietf-routing:routing/control-plane-protocols\
       /control-plane-protocol/ietf-ospf:ospf"
   }
  }
```

The same response in RESTCONF (using JSON format):

```
HTTP/1.1 200 OK
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
Content-Type: application/yang-data+json

{ "ietf-nmda-compare:output" : {
    "differences" : {
      "ietf-yang-patch:yang-patch" : {
        "patch-id" : "ospf router-id",
        "comment" : "diff between operational and intended",
        "edit" : [
          {
            "edit-id" : "1",
            "operation" : "replace",
            "target" : "/ietf-ospf:explicit-router-id",
            "value" : {
                "ietf-ospf:explicit-router-id" : "1.1.1.1"
                "@ietf-ospf:explicit-router-id" : {
                "ietf-origin:origin" : "ietf-origin:system"
                }
            }
          }
        ]
      }
    }
  }
}
```

7. Open Issues

Currently, origin metadata is included in RPC output per default in
comparisons that involve <operational>. It is conceivable to
introduce an input parameter that controls whether origin metadata
should in fact be included.

Currently the comparison filter is defined using subtree and XPath as
in NETCONF[RFC6241]. It is not clear whether there is a requirement
to allow for the definition of filters that relate instead to target
resources per RESTCONF [RFC7950].

8. Possible Future Extensions

It is conceivable to extend the compare operation with a number of
possible additional features in the future.

Specifically, it is possible to define an extension with an optional
feature for dampening. This will allow clients to specify a minimum
time period for which a difference must persist for it to be
reported. This will enable clients to distinguish between
differences that are only fleeting from ones that are not and that
may represent a real operational issue and inconsistency within the
device.

For this purpose, an additional input parameter can be added to
specify the dampening period. Only differences that pertain for at
least the dampening time are reported. A value of 0 or omission of
the parameter indicates no dampening. Reporting of differences MAY
correspondingly be delayed by the dampening period from the time the
request is received.

To implement this feature, a server implementation might run a
comparison when the RPC is first invoked and temporarily store the
result. Subsequently, it could wait until after the end of the
dampening period to check whether the same differences are still
observed. The differences that still persist are then returned.

9. IANA Considerations

9.1. Updates to the IETF XML Registry

This document registers one URI in the IETF XML registry [RFC3688].
Following the format in [RFC3688], the following registration is
requested:

   URI: urn:ietf:params:xml:ns:yang:ietf-nmda-compare

   Registrant Contact: The IESG.

   XML: N/A, the requested URI is an XML namespace.

9.2. Updates to the YANG Module Names Registry

This document registers a YANG module in the YANG Module Names
registry [RFC7950]. Following the format in [RFC7950], the following
registration is requested:

   name: ietf-nmda-compare

   namespace: urn:ietf:params:xml:ns:yang:ietf-nmda-compare

   prefix: cp

   reference: RFC XXXX

10. Security Considerations

   Comparing discrepancies between datastores requires a certain amount
   of processing resources at the server. An attacker could attempt to
   attack a server by making a high volume of comparison requests.
   Server implementations can guard against such scenarios in several
   ways. For one, they can implement NACM in order to require proper
   authorization for requests to be made. Second, server
   implementations can limit the number of requests that they serve in
   any one time interval, potentially rejecting requests made at a
   higher frequency than the implementation can reasonably sustain.

11. Acknowledgments

   We thank Rob Wilton, Martin Bjorklund, Mahesh Jethanandani, Lou
   Berger, and Kent Watsen for valuable feedback and suggestions on an
   earlier revision of this document.

12. References

12.1. Normative References

   [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>.

   [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>.

   [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>.

   [RFC8072] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Patch
              Media Type", RFC 8072, DOI 10.17487/RFC8072, February
              2017, <https://www.rfc-editor.org/info/rfc8072>.

   [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>.

   [RFC8340]   Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams",
               BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018,
               <https://www.rfc-editor.org/info/rfc8340>.

   [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/info/rfc8342>.

12.2. Informative References

   [I-D.ietf-ospf-yang]
               Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem,
               "Yang Data Model for OSPF Protocol", draft-ietf-ospf-
               yang-11 (work in progress), April 2018.

Authors' Addresses

   Alexander Clemm
   Huawei
   2330 Central Expressway
   Santa Clara, CA 95050
   USA

   Email: ludwig@clemm.org

   Yingzhen Qu
   Huawei
   2330 Central Expressway
   Santa Clara, CA 95050
   USA

   Email: yingzhen.qu@huawei.com

   Jeff Tantsura
   Nuage Networks

   Email: jefftant.ietf@gmail.com

   Andy Bierman
   YumaWorks

      Email: andy@yumaworks.com