NETCONF B. Claise
Internet-Draft Huawei
Intended status: Standards Track M. Nayyar
Expires: January 12, 2022 A. Reddy Sesani
Cisco Systems, Inc.
July 11, 2021

Per-Node Capabilities for Optimum Operational Data Collection
draft-claise-netconf-metadata-for-collection-02

Abstract

This document proposes a YANG module that provides per-node capabilities for optimum operational data collection. This YANG module augments the YANG Modules for describing System Capabilities and YANG-Push Notification capabilities.

This module defines augmented nodes to publish the metadata information specific to YANG node-identifier as per ietf-system-capabilities datatree.

Complementary RPCs, based on the same node capabilities, simplify the data collection operations.

Table of Contents

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.

The term Client and Server are specified in [RFC8342].

The term Implementation-time and Run-time are specified in [I-D.ietf-netconf-notification-capabilities].

2. Introduction

This document specifies a way to learn from the devices how granular its telemetry and data can be to provide the best post-processing analytics. In the end, the service assurance architecture

[I-D.claise-opsawg-service-assurance-architecture], it's not sufficient to simply stream (or poll) telemetry data, it is equally important to be able to act on the data. As such, a series of extra information about the node capabilities is essential.

The module ietf-system-capabilities [I-D.ietf-netconf-notification-capabilities] provides a structure that can be used to specify YANG related system capabilities for servers. The module can be used in conjunction with YANG Instance Data to make this information available at implementation-time. The module can also be used to report capability information from the server at run-time.

The module ietf-notification-capabilities [I-D.ietf-netconf-notification-capabilities] augments ietf-system-capabilities to specify capabilities related to "Subscription to YANG Datastores" (YANG-Push) [RFC8639]. It provides a starting point by specifying some per-node telemetry-related capabilities. Of particular interest are the following node capabilities:

- minimum-dampening-period

- on-change-supported

- periodic-notifications-supported

- supported-excluded-change-type

Taking the example of on-change-supported and periodic-notifications-supported, it's key to understand whether a publisher is capable of sending on-change notifications versus sending periodic notifications for the selected data store or data nodes. Indeed, not only would the telemetry configuration change depending on the capabilities (on-change versus periodic), but more importantly the client's handling of the telemetry information would change. Upon receipt of an on-change telemetry message, an immediate action could be taken to correct or mitigate the issue, while in case of periodic notification, a comparison with the previous value must first be performed in order to understand if and how the network state has changed.

Exactly like a client that connects to a server is able to discover the capabilities in terms of supported YANG modules, features, deviations, and protocol capabilities; the same client must also be able to discover the required per-node capabilities (also known as metadata) to correctly act on the telemetry information. It forms part of the API contract for managing and monitoring the device. Extending the per-node capabilities specified in

[I-D.ietf-netconf-notification-capabilities], additional per-node capabilities are required.

The YANG module in this document augments the ietf-system-capabilities YANG module in "YANG Modules for describing System Capabilities and Yang-Push Notification Capabilities" [I-D.ietf-netconf-notification-capabilities].

The YANG data model in this document conform to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

3. Concepts

Doing networking data collection for the sake of doing collection is not useful. At the time of network automation, displaying nice graphs from collected data is not useful: the collected data must be acted upon immediately. Some use cases are: network availability, closed loop automation (reconfiguring network based on observed network state changes), service assurance [I-D.claise-opsawg-service-assurance-architecture], etc.

Along with the capabilities specified in ietf-netconf-notification-capabilities [I-D.ietf-netconf-notification-capabilities] YANG model, there is some additional information that can be made available per node-selector to help with this optimum collection of operational data. For example, these additional metadata can help reduce the load on the devices being managed along with the performance improvements because of the way data is subscribed to. Some other metadata can help with the collection automation itself (mapping of config and oper data node, mapping of MIB oid to YANG leaf).

Some metadata are static and can augment the node-capabilities in [I-D.ietf-netconf-notification-capabilities], for both implementation time and run time environments. Other metadata are dynamic and have to be derived during the run-time. They can change based on the role of the device and the scale of the data being observed.

Per-node static metadata includes:

o minimum-observable-period: This is the minimum observable period in nanoseconds for the node-selector. Streaming or polling more frequently then this interval may not fetch useful information as the node could be updated only at this frequency internally. If a close loop automation system would stream or poll more frequently, it could actually draw the wrong conclusions. Let's take the example of interface counters than are updated more frequently than 30 seconds in a distributed system. Streaming interface counters every 30 seconds would see an natural increase in the

interface counters. However, streaming those interface counters every 10 seconds could lead to the wrong conclusion that no packets are received/sent on that specific interface ... triggering an automatic interface troubleshooting action. Hence determining the minimum-observable-period for every monitored leaf is essential for closed loop automation and assurance systems.

- suggested-observable-period: The suggested observable period for this node-selector. This value represents factory default suggested information, only available at implementation time. Let us assume that an assurance system would like to monitor all FIB entries in the router. The router would advertise that the suggested observable period is, let's say, 30 seconds. Those 5 seconds are the factory defaults, provided at the implementation time. Once the router is in production, the observable period would obviously change depending on the environment (as an example, a FIB containing all BGP entries is huge): this dynamic suggest observable period is called the computed-observable-period and is available part of the get-measurement-metadata RPC.

- optimized-measurement-point: In some server design, operational data are usually modeled/structured in a way that the relevant data are grouped together and reside together. In most cases, it is more performant to fetch this data together than as individual leaves: data are structured together internally, grouped together, and therefore fetched together. This feature specifies optimum observable points in the model at which data can be collected and streamed in an efficient way. Depending on the implementation, optimum points can be leaves or a container nodes in the YANG tree. This is a selection node, that means its presence for a node-selector indicates it is the optimized measurement point.

- corresponding-mib-oid: The object identifier (OID) assigned to a SMIv2 definition, corresponding to the node-selector. The object identifier value is written in decimal dotted notation. Existing SNMP MIBs based automations can use this information to migrate to more analytics-ready YANG Modeled data. Working from a single data model system (YANG-based in this case) for data collection simplifies the management, as opposed to use different data models. Therefore, knowing the mapping MIB OID/YANG leaf is important, as transition mechanism towards YANG (for example: moving away from SNMP polling to model-driven telemetry) but also as a way to understand whether the same operational data is metered in both the MIB and YANG worlds, adding to the load on devices. Some IETF RFCs, such as the YANG Interface Management [RFC8343], specify the mapping in the document. However, providing this mapping directly from the server helps automation from a client point of view.

o related-node: Data nodes that are related for closed-loop scenario for data node specified in node-capabilities. In case node-capabilities is an operational node then the associated node-instance-identifier represents config paths directly related to this operational node capabilities. In case node-capabilities is an config node then the associated node-instance-identifier represents operational leaf directly related to this configuration node capabilities. This node is specifically interesting for non NMDA [RFC8342], non openconfig YANG modules. For example, in the initial YANG data model for interface management [RFC7223], which is not NMDA-compliant, advertising the mapping between the admin-status and the oper-status leaves would clearly simplify the closed loop automation. Note that NMDA and the openconfig -state container solved that issue but not all servers are NMDA compliant and openconfig models don't cover all server functions.

A generic RPC, get-system-node-capabilities, provides the capabilities for the nodes in the subtree of the input. If the input node passed is a leaf/leaf-list, then all the metadata for that input node are returned. If the input node is not leaf/leaf-list then the RPC returns the metadata of all of its subtree nodes.

There is some run-time information that is very helpful for the applications to know, to be able to start listening to the device without adding too much additional resource strain on the device. The get-measurement-metadata RPC can be used to fetch this data.

Per-node dynamic metadata includes, part of the get-measurement-metadata RPC:

o optimized-measurement-point: The node-selector is searched up the data tree chain to find the parent node that is the optimized measurement point (if the optimized-measurement-point-feature is supported). If the node-selector itself is the optimized point then same data node is returned in the output. If the node-selector has no optimized measurement point then this optimized-measurement-point leaf is not returned.

o computed-observable-period: the computed observable period for this node-selector (and optimized-measurement-point). The system internally dynamically computes the suggested observable period (relevant for polling or streaming cadence) which can be greater-or-equal to the minimal-observable-period. Since this value is dynamic, this metadata is only available in a run time environment.

o active-measurements - subscribed-measurement-period: List of existing subscriptions for this node-selector. If there are no

active subscriptions then system calculate the measurement-period and this list is not-returned, else, each instance in this list will be pair of active measurement with intended and actual period used by the system.

4. Base ietf-system-node-metadata YANG module

4.1. Tree View

The following tree diagram [RFC8340] provides an overview of the ietf-system-node-metadata data model.

```
module: ietf-system-node-metadata
  augment /sysc:system-capabilities/sysc:datastore-capabilities/sysc:per-node-
capabilities/sysc:node-selection/sysc:node-selector:
    +--ro minimum-observable-period?    uint64
    +--ro suggested-observable-period?   uint64
    +--ro optimized-measurement-point?   empty {optimized-measurement-point-
feature}?
    +--ro corresponding-mib-oid?         yang:object-identifier-128
    +--ro related-node?                  yang:node-instance-identifier

  rpcs:
    +---x get-measurement-metadata
    |  +---w input
    |  |  +---w node-selector?   yang:node-instance-identifier
    |  +--ro output
    |     +--ro optimized-measurement-point?   yang:node-instance-identifier
{optimized-measurement-point-feature}?
    |     +--ro computed-observable-period?    uint64
    |     +--ro active-measurements* []
    |        +--ro subscribed-measurement-period?   uint64
    +---x get-system-node-capabilities
       +---w input
       |  +---w node-selector?   yang:node-instance-identifier
       +--ro output
          +--ro node-selector-capability* []
             +--ro node?                          yang:node-instance-identifier
             +--ro minimum-observable-period?    uint64
             +--ro suggested-observable-period?   uint64
             +--ro optimized-measurement-point?   empty {optimized-measurement-
point-feature}?
             +--ro corresponding-mib-oid?         yang:object-identifier-128
             +--ro related-node?                  yang:node-instance-identifier
```

4.2. Full Tree View

The following tree diagram [RFC8340] provides an overview of the ietf-system-capabilities and ietf-system-node-metadata data models.

```
module: ietf-system-node-metadata

  rpcs:
    +---x get-measurement-metadata
    |  +---w input
    |  |  +---w node-selector?   yang:node-instance-identifier
    |  +--ro output
    |     +--ro optimized-measurement-point?   yang:node-instance-identifier
{optimized-measurement-point-feature}?
    |     +--ro computed-observable-period?    uint64
    |     +--ro active-measurements* []
    |        +--ro subscribed-measurement-period?   uint64
    +---x get-system-node-capabilities
       +---w input
       |  +---w node-selector?   yang:node-instance-identifier
       +--ro output
          +--ro node-selector-capability* []
             +--ro node?                          yang:node-instance-identifier
             +--ro minimum-observable-period?    uint64
             +--ro suggested-observable-period?   uint64
             +--ro optimized-measurement-point?   empty {optimized-measurement-
point-feature}?
             +--ro corresponding-mib-oid?         yang:object-identifier-128
             +--ro related-node?                  yang:node-instance-identifier
module: ietf-system-capabilities
  +--ro system-capabilities
     +--ro datastore-capabilities* [datastore]
        +--ro datastore                -> /yanglib:yang-library/datastore/name
        +--ro per-node-capabilities* []
           +--ro (node-selection)?
              +--:(node-selector)
                 +--ro node-selector?                             nacm:node-
instance-identifier
                 +--ro sys-metadata:minimum-observable-period?    uint64
                 +--ro sys-metadata:suggested-observable-period?   uint64
                 +--ro sys-metadata:optimized-measurement-point?   empty
{optimized-measurement-point-feature}?
                 +--ro sys-metadata:corresponding-mib-oid?         yang:object-
identifier-128
                 +--ro sys-metadata:related-node?                  yang:node-
instance-identifier
```

4.3. YANG Module

```
   <CODE BEGINS> file "ietf-system-node-metadata@2020-03-20.yang"

   module ietf-system-node-metadata {
     yang-version 1.1;
```

```
namespace "urn:ietf:params:xml:ns:yang:ietf-system-node-metadata";
prefix sys-metadata;

import ietf-system-capabilities {
  prefix sysc;
  reference
```

```
        "RFC XXXX: YANG Modules for describing System Capabilities and
         Yang-Push Notification Capabilities";
    }
    import ietf-yang-types {
      prefix yang;
      reference
        "RFC XXXX: Currently draft-ietf-netmod-rfc6991-bis-04, Common
         YANG Data types";
    }

    organization
      "IETF NETCONF (Network Configuration) Working Group";
    contact
      "WG Web:   <https://datatracker.ietf.org/wg/netconf/>
       WG List:  <mailto:netconf@ietf.org>

       Editor:   Benoit Claise
                 <mailto:bclaise@cisco.com>

       Editor:   Munish Nayyar
                 <mailto:mnayyar@cisco.com>

       Editor:   Adithya Reddy Sesani
                 <mailto:adithyas@cisco.com>
      ";
    description
      "This document proposes a YANG module that provides per-node
       capabilities for optimum operational data collection.

       This YANG module augments the YANG Modules for describing
       System Capabilities and Yang-Push Notification capabilities
       [RFC XXXX].

       This module defines augmented nodes to publish the
       metadata information specific to YANG node-identifier as per
       ietf-system-capabilities datatree.

       Complementary RPCs, based on the same node capabilities,
       simplify the data collection operations.

       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 (RFC 2119)
       (RFC 8174) when, and only when, they appear in all
       capitals, as shown here.

       Copyright (c) 2020 IETF Trust and the persons identified as
```

```
         authors of the code.  All rights reserved.

         Redistribution and use in source and binary forms, with or
         without modification, is permitted pursuant to, and subject
         to the license terms contained in, the Simplified BSD License
         set forth in Section 4.c of the IETF Trust's Legal Provisions
         Relating to IETF Documents
         (http://trustee.ietf.org/license-info).

         This version of this YANG module is part of RFC XXXX
         (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
         for full legal notices.";

      revision 2020-03-23 {
        description
          "Initial version";
        reference
          "RFC XXX: Per-Node Capabilities For Closed Loop Automation.";
      }

      feature optimized-measurement-point-feature {
        description
          "Support for optimized measurement point within data tree.";
      }

      grouping system-node-metadata-info {
        description
          "group of metadata properties associated to the
           node-instance.";
        leaf minimum-observable-period {
          type uint64;
          units "nanoseconds";
          description
            "The minimum observable period for this node-selector. Don't
             poll or stream more frequently that minimum observable
             period in nanoseconds as the corresponding counter is not
             updated more frequently.";
        }
        leaf suggested-observable-period {
          type uint64;
          units "nanoseconds";
          description
            "The suggested observable period for this node-selector.
             This value represents factory default suggested
             information, only available at implementation time.";
        }
        leaf optimized-measurement-point {
          if-feature "optimized-measurement-point-feature";
```

```
        type empty;
        description
          "This node-selector is an optimized measurement point.";
      }
      leaf corresponding-mib-oid {
        type yang:object-identifier-128;
        description
          "The object identifier (OID) assigned to a SMIv2 definition,
           corresponding to this node-selector.";
      }
      leaf related-node {
        type yang:node-instance-identifier;
        description
          "In case the node instance is an operational node then the
           associated node-instance-identifier represents the config
           leaf directly related to this operational node. In case the
           node instance is an config node then the associated
           node-instance-identifier represents the operational leaf
           directly related to this configuration node. A typical
           example is the relationship between the admin-status and
           oper-status, which is impossible to detect automatically in
           a non-NMDA environment or for non-openconfig YANG moduels.
           The related-node SHOULD NOT reported for NMDA architectures
           and openconfig YANG modules.";
      }
    }

    augment
      "/sysc:system-capabilities/sysc:datastore-capabilities/"
       + "sysc:per-node-capabilities/"
       + "sysc:node-selection/sysc:node-selector" {
      description
        "Metadata information tied to the per-node-capabilities";
      uses system-node-metadata-info;
    }

    rpc get-measurement-metadata {
      description
        "RPC that returns the optimized measurement per-node
         capabilities and some measurement parameters. This RPC
         is added to allow clients to learn dynamically changing
         metadata for a specific leaf on a server.

         If the server supports the optimized-measurement-point
         feature, then the output data refers to
         optimized-measurement-point. The server will internally
         find the optimized-measurement-point. If it can not find it,
         then no output is returned (for the
```

```
         optimized-measurement-point, computed-observable-period,
         and active-measurements).

         If the server doesn't support the optimized-measurement-point
         feature, then the output data refers to input node selector.";
      input {
        leaf node-selector {
          type yang:node-instance-identifier;
          description
            "node instance for which metadata is requested";
        }
      }
      output {
        leaf optimized-measurement-point {
          if-feature "optimized-measurement-point-feature";
          type yang:node-instance-identifier;
          description
            "The node-selector is searched up the data tree chain to
             find the parent node that is the optimized measurement
             point (if the optimized-measurement-point-feature is
             supported).

             If the node-selector itself is the optimized point then
             same data node is returned in the output.

             If the node-selector has no optimized measurement point
             then this optimized-measurement-point leaf is not
             returned.";
        }
        leaf computed-observable-period {
          type uint64;
          units "nanoseconds";
          description
            "the computed observable period for this node-selector (and
             optimized-measurement-point). The system internally
             dynamically computes the suggested observable period
             (relevant for polling or streaming cadence) which can be
             greater-or-equal to the minimal-observable-period.
             Since this value is dynamic, this metadata is only
             available in a run time environment.";
        }
        list active-measurements {
          description
            "list of existing subscriptions for this node-selector. If
             there are no active subscriptions then system calculate
             the measurement-period and this list is not-returned,
             else, each instance in this list will be pair of active
             measurement with intended and actual period used by the
```

```
              system";
           leaf subscribed-measurement-period {
             type uint64;
             units "nanoseconds";
             description
               "Currently subscribed measurement period for this
                node-selector (and optimized-measurement-point)";
           }
         }
       }
     }

     rpc get-system-node-capabilities {
       description
         "RPC to get the capabilities for the nodes in the subtree of
          the input.
          If the input node passed is a leaf/leaf-list, then
          the same node metadata is returned in the output.
          If the input node is not leaf/leaf-list then metadata of its
          subtree nodes is returned.";
       input {
         leaf node-selector {
           type yang:node-instance-identifier;
           description
             "node instance whose subtree which metadata is requested.";
         }
       }
       output {
         list node-selector-capability {
           description
             "metadata of nodes in the subtree of node-selector.";
           leaf node {
             type yang:node-instance-identifier;
             description
               "instance path of the node inside subtree of
                node-selector.";
           }
           uses system-node-metadata-info;
         }
       }
     }
   }

   <CODE ENDS>
```

5. Examples

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

XML data tree for the ietf-interface YANG module [RFC8343]:

```
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
  <interface>
    <name/>
    <description/>
    <type/>
    <link-up-down-trap-enable/>
    <admin-status/>
    <oper-status/>
    <last-change/>
    <if-index/>
    <phys-address/>
    <higher-layer-if>
      <!-- # entries: 0.. -->
    </higher-layer-if>
    <lower-layer-if>
      <!-- # entries: 0.. -->
    </lower-layer-if>
    <speed/>
    <statistics>
      <discontinuity-time/>
      <in-octets/>
      <in-unicast-pkts/>
      <in-broadcast-pkts/>
      <in-multicast-pkts/>
      <in-discards/>
      <in-errors/>
      <in-unknown-protos/>
      <out-octets/>
      <out-unicast-pkts/>
      <out-broadcast-pkts/>
      <out-multicast-pkts/>
      <out-discards/>
      <out-errors/>
    </statistics>
  </interface>
</interfaces>
```

Example1: Demonstrating the querying metadata for all system schema nodes for the ietf-interfaces [RFC8343].

```
  <!-- Request -->
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <system-capabilities xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
capabilities">
      </system-capabilities>
    </filter>
  </get>
</rpc>

  <!-- Response -->
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
  <system-capabilities xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
capabilities" xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
    <datastore-capabilities>
      <datastore>ds:operational</datastore>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface</node-selector>
        <optimized-measurement-point xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata"/>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata"></corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:admin-status</node-
selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.7</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:oper-status</node-
selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.8</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
```

```
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:if-index</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.1</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:phys-address</node-
selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.6</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
```

```
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:lower-layer-if</node-
selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.31.1.2.1.2</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:higher-layer-if</node-
selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.31.1.2.1.1</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:speed</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.5</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics</node-
selector>
        <optimized-measurement-point xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata"/>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.31.1.1</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/
if:discontinuity-time</node-selector>
        <optimized-measurement-point xmlns="urn:ietf:params:xml:ns:yang:ietf-
```

```
system-node-metadata"/>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.31.1.1.1.19</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-octets</
node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.10</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-unicast-
pkts</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.11</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-
multicast-pkts</node-selector>
```

```
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.31.1.1.1.2</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-
broadcast-pkts</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.31.1.1.1.3</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-
discards</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.13</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-errors</
node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.14</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-unknown-
protos</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.15</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
```

```
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-
octets</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.16</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-
unicast-pkts</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.17</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-
multicast-pkts</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.31.1.1.1.4</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-
broadcast-pkts</node-selector>
```

```
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.31.1.1.1.5</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-
discards</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.19</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-
errors</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">1.3.6.1.2.1.2.2.1.20</corresponding-mib-oid>
        <minimum-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</minimum-observable-period>
        <suggested-observable-period xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
    </datastore-capabilities>
  </system-capabilities>
</data>
</rpc-reply>
```


Example2: Demonstrating the querying metadata of all optimized-measurement-point(s). Use containment and selection nodes filtering criteria to express which all metadata you want. In this example: get query filter only to "select" the node-instance-identifier, optimized-measurement-point nodes, for the ietf-interfaces [RFC8343]. There are two optimized-measurement-points: interface and statistics.

```
   <!-- Request -->
 <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get>
     <filter type="subtree">
     <system-capabilities xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
capabilities"></system-capabilities>
      <datastore-capabilities>
        <datastore>ds:operational</datastore>
        <per-node-capabilities>
        <optimized-measurement-point xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata"/>
        </per-node-capabilities>
      </datastore-capabilities>
    </filter>
   </get>
 </rpc>

  <!-- Response -->

<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
  <system-capabilities xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
capabilities" xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
    <datastore-capabilities>
      <datastore>ds:operational</datastore>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface</node-selector>
        <optimized-measurement-point xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata"/>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics</node-
selector>
        <optimized-measurement-point xmlns="urn:ietf:params:xml:ns:yang:ietf-
system-node-metadata"/>
      </per-node-capabilities>
    </datastore-capabilities>
  </system-capabilities>
</data>
</rpc-reply>
```

Example3: Demonstrating the usage of RPC to query the device for computed-measurement-period and the subscribed-measurement-period(s) for the in-errors YANG leaf.

```
<!-- Request -->

<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get-measurement-metadata xmlns="urn:ietf:params:xml:ns:yang:ietf-system-
node-metadata">
    <node-selector>/if:interfaces/if:interface/if:statistics/if:in-errors</
node-selector>
  </get-measurement-metadata>
</rpc>

<!-- Response -->
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <optimized-measurement-point>/if:interfaces/if:interface/if:statistics</
optimized-measurement-point>
  <computed-measurement-period>3000</computed-measurement-period>
  <active-measurements>
    <subscribed-measurement-period>1000</subscribed-measurement-period>
  </active-measurements>
  <active-measurements>
    <subscribed-measurement-period>1000</subscribed-measurement-period>
  </active-measurements>
  <active-measurements>
    <subscribed-measurement-period>1000</subscribed-measurement-period>
  </active-measurements>
</rpc-reply>
```

6. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

7. IANA Considerations

7.1. The IETF XML Registry

This document registers two URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested:

URI: urn:ietf:params:xml:ns:yang:ietf-system-node-metadata
Registrant Contact: The NETCONF WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

8. Open Issues

"related-node" should be split into two: "related-config-node" and "related-state-node"?

Explain how to use the RPC from the client side, along with the different options.

Expand on the active measurement use case

nanosecond: an overkill?

security considerations: see https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines

9. References

9.1. Normative References

[I-D.ietf-netconf-notification-capabilities]
Lengyel, B., Clemm, A., and B. Claise, "YANG Modules describing Capabilities for Systems and Datastore Update Notifications", draft-ietf-netconf-notification-capabilities-16 (work in progress), April 2021.

[I-D.ietf-netmod-rfc6991-bis]
Schoenwaelder, J., "Common YANG Data Types", draft-ietf-netmod-rfc6991-bis-06 (work in progress), April 2021.

[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>.

[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>.

[RFC8341] 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>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

9.2. Informative References

[I-D.claise-opsawg-service-assurance-architecture] Claise, B., Quilbeuf, J., Lopez, D. R., Voyer, D., and T. Arumugam, "Service Assurance for Intent-based Networking Architecture", draft-claise-opsawg-service-assurance-architecture-05 (work in progress), April 2021.

[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>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <https://www.rfc-editor.org/info/rfc7223>.

[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>.

[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>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

Acknowledgements

The authors would like to thank ... for their reviews and feedback.

Authors' Addresses

Benoit Claise
Huawei

Email: benoit.claise@huawei.com

Munish Nayyar
Cisco Systems, Inc.
Milpitas
California
United States

Email: mnayyar@cisco.com

Adithya Reddy Sesani
Cisco Systems, Inc.
Milpitas
California
United States

Email: adithyas@cisco.com