NETCONF
Internet-Draft
Intended status: Standards Track
Expires: 29 July 2022

B. Claise
Huawei
M. Nayyar
A. Reddy Sesani
Cisco Systems, Inc.
25 January 2022

Per-Node Capabilities for Optimum Operational Data Collection
draft-claise-netconf-metadata-for-collection-03

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.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 July 2022.

Copyright Notice

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:

* 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.

* 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:

* 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.

* 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.

* 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-cap
abilities/sysc:node-selection/sysc:node-selector:
    +--ro minimum-observable-period?    uint64
    +--ro suggested-observable-period?   uint64
    +--ro optimized-measurement-point?   empty {optimized-measurement-point-featu
re}?
    +--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 {opt
imized-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-po
int-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 {opt
imized-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-po
int-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-inst
ance-identifier
                 +--ro sys-metadata:minimum-observable-period?    uint64
                 +--ro sys-metadata:suggested-observable-period?   uint64
                 +--ro sys-metadata:optimized-measurement-point?   empty {optimiz
ed-measurement-point-feature}?
                 +--ro sys-metadata:corresponding-mib-oid?         yang:object-id
entifier-128
                 +--ro sys-metadata:related-node?                  yang:node-inst
ance-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-capabil
ities">
      </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-capabilitie
s" 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-syst
em-node-metadata"/>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-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-nod
e-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-syst
em-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-nod
e-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-syst
em-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-nod
e-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-syst
```

em-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-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:lower-layer-if</node-select
or>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:higher-layer-if</node-selec
tor>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-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-nod
e-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-syst
em-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-syst
em-node-metadata"/>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-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-syst
em-node-metadata"/>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-octets</no
de-selector>
```

```
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-unicast-pk
ts</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-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-nod
e-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-syst
em-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-nod
e-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-syst
em-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-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-errors</no
de-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:in-unknown-pr
otos</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-octets</n
ode-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
```

```
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-unicast-p
kts</node-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-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-nod
e-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-syst
em-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-nod
e-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-syst
em-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-nod
e-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-syst
em-node-metadata">1000</suggested-observable-period>
      </per-node-capabilities>
      <per-node-capabilities>
        <node-selector>/if:interfaces/if:interface/if:statistics/if:out-errors</n
ode-selector>
        <corresponding-mib-oid xmlns="urn:ietf:params:xml:ns:yang:ietf-system-nod
e-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-syst
em-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-capabili
ties"></system-capabilities>
      <datastore-capabilities>
        <datastore>ds:operational</datastore>
        <per-node-capabilities>
        <optimized-measurement-point xmlns="urn:ietf:params:xml:ns:yang:ietf-syst
em-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-capabilitie
s" 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-syst
em-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-syst
em-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-m
etadata">
    <node-selector>/if:interfaces/if:interface/if:statistics/if:in-errors</node-s
elector>
  </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</optimiz
ed-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", Work in Progress, Internet-Draft, draft-ietf-netconf-notification-capabilities-21, 15 October 2021, <https://www.ietf.org/archive/id/draft-ietf-netconf-notification-capabilities-21.txt>.

[I-D.ietf-netmod-rfc6991-bis]
Schoenwaelder, J., "Common YANG Data Types", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc6991-bis-10, 14 January 2022, <https://www.ietf.org/archive/id/draft-ietf-netmod-rfc6991-bis-10.txt>.

[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", Work in Progress, Internet-Draft, draft-claise-opsawg-service-assurance-architecture-05, 23 April 2021, <https://www.ietf.org/archive/id/draft-claise-opsawg-service-assurance-architecture-05.txt>.

[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

NETCONF
Internet-Draft
Intended status: Standards Track
Expires: 4 August 2024

M. Jethanandani
Kloud Services
K. Watsen
Watsen Networks
1 February 2024

An HTTPS-based Transport for YANG Notifications
draft-ietf-netconf-https-notif-15

Abstract

This document defines a protocol for sending asynchronous event notifications similar to notifications defined in RFC 5277, but over HTTPS. YANG modules for configuring publishers are also defined. Examples are provided illustrating how to configure various publishers.

This document requires that the publisher is a "server" (e.g., a NETCONF or RESTCONF server), but does not assume that the receiver is a server.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 4 August 2024.

Copyright Notice

Table of Contents

1. Introduction

This document defines a protocol for sending asynchronous event notifications similar to notifications defined in NETCONF Event Notifications [RFC5277], but over HTTPS. Using HTTPS, which is a secure form of HTTP Semantics [RFC9110], maximizes transport-level interoperability, while allowing for a variety of encoding options. The protocol supports HTTP/1.1: Message Syntax and Routing [RFC9112] and, HTTP/2 [RFC9113]. While the payload does not change between these versions of HTTP and HTTP/3 [RFC9114], the underlying transport does. Since NETCONF does not support QUIC: A UDP-Based Multiplexed and Secure Transport [RFC9000], support for HTTP/3 [RFC9114], is considered out of scope of this document.

This document defines support for JSON and XML; future efforts may define support for other encodings (e.g., binary). This document requires that the publisher is a "server" (e.g., a NETCONF or RESTCONF server), but does not assume that the receiver is a NETCONF or RESTCONF server. It does expect the receiver to be an HTTPS server to receive the notifications.

This document also defines two YANG 1.1 [RFC7950] modules that extend the data model defined in Subscription to YANG Notifications [RFC8639], enabling the configuration of HTTPS-based receivers.

An example module illustrating the configuration of a publisher not using the data model defined in RFC 8639 is also provided.

Configured subscriptions enable a server (e.g., a NETCONF or RESTCONF server), acting as a publisher of notifications, to proactively push notifications to external receivers without the receivers needing to first connect to the server, as is the case with dynamic subscriptions.

1.1. Applicability Statement

While the YANG modules have been defined as an augmentation of Subscription to YANG Notifications [RFC8639], the notification method defined in this document MAY be used outside of Subscription to YANG Notifications [RFC8639] by using some of the definitions from this module along with the grouping defined in Groupings for HTTP Clients and Servers [I-D.ietf-netconf-http-client-server]. For an example on how that can be done, see Section A.2.

1.2. Note to RFC Editor

This document uses several placeholder values throughout the document. Please replace them as follows and remove this section before publication.

RFC XXXX, where XXXX is the number assigned to this document at the time of publication.

RFC YYYY, where YYYY is the number assigned to [I-D.ietf-netconf-http-client-server].

2024-02-01 with the actual date of the publication of this document.

1.3. Abbreviations

Acronym	Expansion
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
SSE	Server-Sent Events
TCP	Transmission Control Protocol
TLS	Transport Layer Security

Table 1

1.4. 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.

1.4.1. Terms Imported from other RFCs

The following terms are defined in Subscription to YANG Notifications [RFC8639].

* Publisher

* Receiver

* Subscribed Notifications

The following term is defined in RESTCONF Protocol [RFC8040].

* target resource

1.5. Tree Diagram

The tree diagram for the YANG modules defined in this document use annotations defined in YANG Tree Diagrams. [RFC8340].

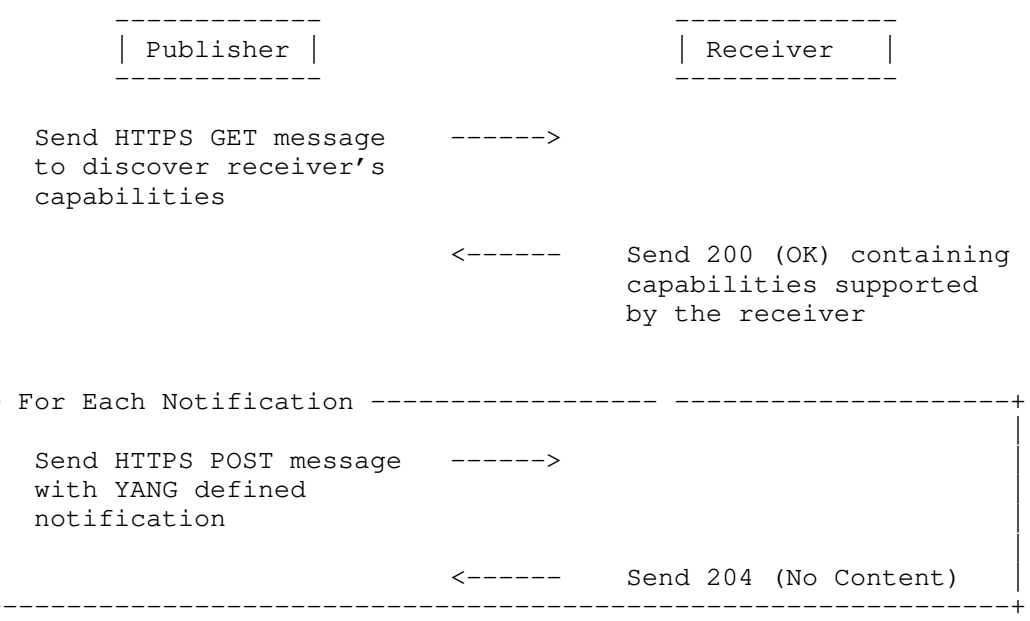
2. Overview of Publisher to Receiver Interaction

The protocol consists of two HTTP-based target resources presented by the receiver. These two resources share a common prefix that the publisher learns from a request it issues, as defined in section 3.2. If the data model in section 6.2 is used, this common prefix is defined by the "path" leaf in the "http-client-parameters" container.

* "capabilities": A target resource enabling the publisher to discover what optional capabilities a receiver supports. Publishers SHOULD query this target before sending any notifications or if ever an error occurs.

* "relay-notification": A target resource enabling the publisher to send one or more notification to a receiver. This document defines support for sending only one notification per message; a future effort MAY extend the protocol to send multiple notifications per message.

The protocol is illustrated in the diagram below:



Note that, for RFC 8639 configured subscriptions, the very first notification must be the "subscription-started" notification.

3. Discovering a Receiver's Capabilities

3.1. Applicability

For publishers using Subscription to YANG Notifications [RFC8639], dynamic discovery of a receiver's supported encoding is necessary only when the "/subscriptions/subscription/encoding" leaf is not configured, per the "encoding" leaf's description statement in the "ietf-subscribed-notification" module.

If the "encoding" leaf is not configured, and the publisher wants to send a notification in a particular format, without going through the setup operation of learning the receiver capabilities, it can do so, but has to be prepared for the case when it receives an error response, because the receiver does not support the format sent by the publisher.

3.2. Request

To learn the capabilities of a receiver, a publisher can issue an HTTPS GET request to the "capabilities" resource (see Section 2) on the receiver with "Accept" header set using the "application/xml" as defined in XML Media Types [RFC7303], and/or "application/json" as defined in JSON [RFC8259] media-types.

3.3. Response

The receiver responds with a "200 (OK)" message, having the "Content-Type" header set to either "application/xml" or "application/json" (which ever was selected), and containing in the response body a list of the receiver's capabilities encoded in the selected format.

Even though a YANG module is not defined for this interaction, the response body MUST conform to the following YANG-modeled format:

```
container receiver-capabilities {
  description
    "A container for a list of capabilities supported by
     the receiver.";
  leaf-list receiver-capability {
    type "inet:uri";
    description
      "A capability supported by the receiver.  A partial list of
       capabilities is defined in the 'Capabilities for HTTPS
       Notification Receivers' registry (see RFC XXXX). Additional
       custom capabilities MAY be defined.";
  }
}
```

As it is possible that the receiver may return custom capability URIs, the publisher MUST ignore any capabilities that it does not recognize.

3.4. Example

The publisher can send the following request to learn the receiver capabilities. In this example, the "Accept" states that the publisher wants to receive the capabilities response in XML but, if not supported, then in JSON.

```
GET /some/path/capabilities HTTP/1.1
Host: example.com
Accept: application/xml, application/json;q=0.5
```

If the receiver is able to reply using "application/xml", and assuming it is able to receive JSON and XML encoded notifications, and it is able to process the RFC 8639 state machine, the response might look like this:

```
HTTP/1.1 200 OK
Date: Wed, 26 Feb 2020 20:33:30 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/xml

<receiver-capabilities>
  <receiver-capability>\
    urn:ietf:capability:https-notif-receiver:encoding:json\
  </receiver-capability>
  <receiver-capability>\
    urn:ietf:capability:https-notif-receiver:encoding:xml\
  </receiver-capability>
  <receiver-capability>\
    urn:ietf:capability:https-notif-receiver:sub-notif\
  </receiver-capability>
</receiver-capabilities>
```

If the receiver is unable to reply using "application/xml", the response might look like this:

```
HTTP/1.1 200 OK
Date: Wed, 26 Feb 2020 20:33:30 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/json
Content-Length: nnn

{
   "receiver-capabilities": {
     "receiver-capability": [
       "urn:ietf:capability:https-notif-receiver:encoding:json",
       "urn:ietf:capability:https-notif-receiver:encoding:xml",
       "urn:ietf:capability:https-notif-receiver:sub-notif"
     ]
   }
}
```

4. Sending Event Notifications

4.1. Request

The publisher sends an HTTP POST request to the "relay-notification" resource (see Section 2) on the receiver with the "Content-Type" header set to either "application/json" or "application/xml" and a body containing the notification encoded using the specified format.

XML-encoded notifications are encoded using the format defined by NETCONF Event Notifications [RFC5277] for XML.

JSON-encoded notifications are encoded the same as specified in Section 6.4 in RESTCONF [RFC8040] with the following deviations:

* The notifications do not contain the "data:" prefix used by Server-Sent Events (SSE).

* Instead of saying that, for JSON-encoding purposes, the module name for the "notification" element is "ietf-restconf", the module name will instead be "ietf-https-notif".

4.2. Response

The response on success SHOULD be "204 (No Content)". In case of corrupted or malformed event, the response SHOULD be an appropriate HTTP error response.

4.3. Example

An XML-encoded notification might be sent as follows:

```
POST /some/path/relay-notification HTTP/1.1
Host: example.com
Content-Type: application/xml

<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2019-03-22T12:35:00Z</eventTime>
  <event xmlns="https://example.com/example-mod">
    <event-class>fault</event-class>
    <reporting-entity>
      <card>Ethernet0</card>
    </reporting-entity>
    <severity>major</severity>
  </event>
</notification>
```

A JSON-encoded notification might be sent as follows:

```
POST /some/path/relay-notification HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "ietf-https-notif:notification": {
    "eventTime": "2013-12-21T00:01:00Z",
    "example-mod:event" : {
      "event-class" : "fault",
      "reporting-entity" : { "card" : "Ethernet0" },
      "severity" : "major"
    }
  }
}
```

And, in either case, the response on success might be as follows:

```
HTTP/1.1 204 No Content
Date: Wed, 26 Feb 2020 20:33:30 GMT
Server: example-server
```

5. The "ietf-subscribed-notif-receivers" Module

5.1. Data Model Overview

This YANG module augments the "ietf-subscribed-notifications" module to define a choice of transport types that other modules such as the "ietf-https-notif-transport" module can use to define a transport specific receiver.

module: ietf-subscribed-notif-receivers

```
  augment /sn:subscriptions:
    +--rw receiver-instances
       +--rw receiver-instance* [name]
          +--rw name    string
          +--rw (transport-type)
  augment /sn:subscriptions/sn:subscription/sn:receivers/sn:receiver:
    +--rw receiver-instance-ref?   leafref
```

5.2. YANG Module

The YANG module imports Subscription to YANG Notifications [RFC8639].

```
<CODE BEGINS> file "ietf-subscribed-notif-receivers@2024-02-01.yang"
module ietf-subscribed-notif-receivers {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers";
  prefix "snr";

  import ietf-subscribed-notifications {
    prefix sn;
    reference
      "RFC 8639: Subscription to YANG Notifications";
  }

  organization
    "IETF NETCONF Working Group";

  contact
    "WG Web:   <http://datatracker.ietf.org/wg/netconf>
     WG List:  <netconf@ietf.org>

     Authors: Mahesh Jethanandani (mjethanandani at gmail dot com)
              Kent Watsen (kent plus ietf at watsen dot net)";

  description
    "This YANG module is implemented by Publishers implementing
     the 'ietf-subscribed-notifications' module defined in RFC 8639.

     While this module is defined in RFC XXXX, which primarily
     defines an HTTPS-based transport for notifications, this module
     is not HTTP-specific.  It is a generic extension that can be
     used by any 'notif' transport.

     This module defines two 'augment' statements.  One statement
     augments a 'container' statement called 'receiver-instances'
     into the top-level 'subscriptions' container.  The other
     statement, called 'receiver-instance-ref', augments a 'leaf'
     statement into each 'receiver' that references one of the
     afore mentioned receiver instances.  This indirection enables
     multiple configured subscriptions to send notifications to
     the same receiver instance.

     Copyright (c) 2024 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 Revised 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; see
        the RFC itself for full legal notices.

        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.";

    revision "2024-02-01" {
      description
        "Initial Version.";
      reference
        "RFC XXXX: An HTTPS-based Transport for YANG Notifications.";
    }

    augment "/sn:subscriptions" {
      container receiver-instances {
        description
          "A container for all instances of receivers.";

        list receiver-instance {
          key "name";

          leaf name {
            type string;
            description
              "An arbitrary but unique name for this receiver
               instance.";
          }

          choice transport-type {
            mandatory true;
            description
              "Choice of different types of transports used to
               send notifications.  The 'case' statements must
               be augmented in by other modules.";
          }
          description
            "A list of all receiver instances.";
        }
      }
      description
        "Augment the subscriptions container to define the
         transport type.";
    }
```

```
    augment
      "/sn:subscriptions/sn:subscription/sn:receivers/sn:receiver" {
      leaf receiver-instance-ref {
        type leafref {
          path "/sn:subscriptions/snr:receiver-instances/" +
               "snr:receiver-instance/snr:name";
        }
        description
          "Reference to a receiver instance.";
      }
      description
        "Augment the subscriptions container to define an optional
         reference to a receiver instance.";
    }
  }
  <CODE ENDS>
```

6. The "ietf-https-notif-transport" Module

6.1. Data Model Overview

This YANG module is a definition of a set of receivers that are interested in the notifications published by the publisher. The module contains the TCP, TLS and HTTPS parameters that are needed to communicate with the receiver. The module augments the "ietf-subscribed-notif-receivers" module to define a transport specific receiver.

As mentioned earlier, it uses a POST method to deliver the notification. The "http-receiver/tls/http-client-parameters/path" leaf defines the path for the resource on the receiver, as defined by "path-absolute" in URI Generic Syntax [RFC3986]. The user-id used by Network Configuration Access Control Model [RFC8341], is that of the receiver and is derived from the certificate presented by the receiver as part of "receiver-identity".

An abridged tree diagram representing the module is shown below.

```
   module: ietf-https-notif-transport

     augment /sn:subscriptions/snr:receiver-instances
               /snr:receiver-instance/snr:transport-type:
       +--:(https)
          +--rw https-receiver
             +--rw (transport)
             |  +--:(tls) {tls-supported}?
             |     +--rw tls
             |        +--rw tcp-client-parameters
             |        |  +--rw remote-address    inet:host
             |        |  +--rw remote-port?      inet:port-number
             |        |  +--rw local-address?    inet:ip-address
             |        |  |       {local-binding-supported}?
             |        |  +--rw local-port?       inet:port-number
             |        |  |       {local-binding-supported}?
             |        |  +--rw proxy-server! {proxy-connect}?
             |        |  |     ...
             |        |  +--rw keepalives! {keepalives-supported}?
             |        |        ...
             |        +--rw tls-client-parameters
             |        |  +--rw client-identity!
             |        |  |     ...
             |        |  +--rw server-authentication
             |        |  |     ...
             |        |  +--rw hello-params {tlscmn:hello-params}?
             |        |  |     ...
             |        |  +--rw keepalives {tls-client-keepalives}?
             |        |        ...
             |        +--rw http-client-parameters
             |           +--rw client-identity!
             |           |     ...
             |           +--rw proxy-connect! {proxy-connect}?
             |           |     ...
             |           +--rw path               string
             +--rw receiver-identity {receiver-identity}?
                +--rw cert-maps
                   +--rw cert-to-name* [id]
                      +--rw id             uint32
                      +--rw fingerprint    x509c2n:tls-fingerprint
                      +--rw map-type       identityref
                      +--rw name           string
```

6.2. YANG module

The YANG module imports A YANG Data Model for SNMP Configuration [RFC7407], Subscription to YANG Notifications [RFC8639], and YANG Groupings for HTTP Clients and HTTP Servers [I-D.ietf-netconf-http-client-server].

The YANG module is shown below.

```
<CODE BEGINS> file "ietf-https-notif-transport@2024-02-01.yang"
module ietf-https-notif-transport {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-https-notif-transport";
  prefix "hnt";

  import ietf-x509-cert-to-name {
    prefix x509c2n;
    reference
      "RFC 7407: YANG Data Model for SNMP Configuration.";
  }

  import ietf-subscribed-notifications {
    prefix sn;
    reference
      "RFC 8639: Subscription to YANG Notifications";
  }

  import ietf-subscribed-notif-receivers {
    prefix snr;
    reference
      "RFC XXXX: An HTTPS-based Transport for YANG Notifications.";
  }

  import ietf-http-client {
    prefix httpc;
    reference
      "RFC YYYY: YANG Groupings for HTTP Clients and HTTP Servers.";
  }

  organization
    "IETF NETCONF Working Group";

  contact
    "WG Web:   <http://datatracker.ietf.org/wg/netconf>
     WG List:  <netconf@ietf.org>

     Authors: Mahesh Jethanandani (mjethanandani at gmail dot com)
              Kent Watsen (kent plus ietf at watsen dot net)";
```

```
     description
       "This YANG module is implemented by Publishers that implement
        the 'ietf-subscribed-notifications' module defined in RFC 8639.

        This module augments a 'case' statement called 'https' into
        the 'choice' statement called 'transport-type' defined
        by the 'ietf-https-notif-transport' module defined in RFC XXXX.

        Copyright (c) 2024 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 Revised 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; see
        the RFC itself for full legal notices.

        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.";

     revision "2024-02-01" {
       description
         "Initial Version.";
       reference
         "RFC XXXX: An HTTPS-based Transport for YANG Notifications.";
     }

     feature receiver-identity {
       description
         "Indicates that the server supports filtering notifications
          based on the receiver's identity derived from its TLS
          certificate.";
     }

     identity https {
       base sn:transport;
       description
         "HTTPS transport for notifications.";
     }

     grouping https-receiver-grouping {
       description
```

```
        "A grouping that may be used by other modules wishing to
         configure HTTPS-based notifications without using RFC 8639.";
      uses httpc:http-client-stack-grouping {
        refine "transport/tcp" {
          // create the logical impossibility of enabling the
          // "tcp" transport (i.e., "HTTP" without the 'S').
          if-feature "not httpc:tcp-supported";
        }
        augment "transport/tls/tls/http-client-parameters" {
          leaf path {
            type string;
            mandatory true;
            description
              "A path to the target resources.  Under this
               path the receiver must support both the 'capabilities'
               and 'relay-notification' resource targets, as described
               in RFC XXXX.";
          }
          description
            "Augmentation to add a receiver-specific path for the
             'capabilities' and 'relay-notification' resources.";
        }
      }
      container receiver-identity {
        if-feature receiver-identity;
        description
          "Maps the receiver's TLS certificate to a local identity
           enabling access control to be applied to filter out
           notifications that the receiver may not be authorized
           to view.";
        container cert-maps {
          uses x509c2n:cert-to-name;
          description
            "The cert-maps container is used by a TLS-based HTTP
             server to map the HTTPS client's presented X.509
             certificate to a 'local' username. Specifically, the
             'name' field within the module is used along with
             'specified' identity to perform the match. If no
             matching and valid cert-to-name list entry is found,
             the publisher MUST close the connection, and MUST
             NOT send any notifications over it.";
          reference
            "RFC 7407: A YANG Data Model for SNMP Configuration.";
        }
      }
    }

    augment "/sn:subscriptions/snr:receiver-instances/" +
```

```
            "snr:receiver-instance/snr:transport-type" {
      case https {
        container https-receiver {
          description
            "The HTTPS receiver to send notifications to.";
          uses https-receiver-grouping;
        }
      }
      description
        "Augment the transport-type choice to include the 'https'
         transport.";
    }
  }
  <CODE ENDS>
```

7. Security Considerations

The YANG modules specified in this document define 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 NETCONF 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.

The YANG modules in this document make use of groupings that are defined in YANG Groupings for HTTP Clients and HTTP Servers [I-D.ietf-netconf-http-client-server], YANG Groupings for TLS Clients and TLS Servers [I-D.ietf-netconf-tls-client-server], and A YANG Data Model for SNMP Configuration [RFC7407]. Please see the Security Considerations section of those documents for considerations related to sensitivity and vulnerability of the data nodes defined in them. Additionally, the parameters defined in the tls-client-grouping in the ietf-tls-client module should follow the recommendations specified in Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). [RFC9325]

There are a number of data nodes defined in the YANG modules that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

* The "path" node in "ietf-subscribed-notif-receivers" module can be modified by a malicious user to point to an invalid URI. Worse still, it could point the URI of their choosing, exploit the vulnerable client, and if redirects are followed to the same URI, track its usage.

  The container "receiver-identity" contains nodes like "cert-maps" that are used by the HTTP server to map to the HTTPS client's certificate to a 'local' username. An unintended modification of these nodes will result in new connection requests be denied.

Some of the readable data nodes in the YANG modules may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. The model does not define any readable subtrees and data nodes that are particularly sensitive or vulnerable.

Some of the RPC operations in the YANG modules may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. The model does not define any RPC operations.

8. IANA Considerations

8.1. The "IETF XML" Registry

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

URI: urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers
Registrant Contact: The IESG
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-https-notif-transport
Registrant Contact: The IESG
XML: N/A, the requested URI is an XML namespace.

8.2. The "YANG Module Names" Registry

This document registers two YANG modules in the "YANG Module Names" registry [RFC6020]. Following the format in [RFC6020], the following registrations are requested:

```
name:      ietf-subscribed-notif-receivers
namespace: urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers
prefix:    snr
reference: RFC XXXX

name:      ietf-https-notif-transport
namespace: urn:ietf:params:xml:ns:yang:ietf-https-notif-transport
prefix:    hnt
reference: RFC XXXX
```

8.3. Registration of 'yang-notif' URN Sub-namespace

This document requests that IANA register a new URN Sub-namespace within the "IETF URN Sub-namespace for Registered Protocol Parameter Identifiers" registry defined in [RFC3553].

Registry Name: yang-notif
Specification: RFC XXXX
Repository: "YANG Notifications" registry

8.4. Registration of 'https' URN Sub-namespace

This document requests that IANA register a new URN Sub-namespace within the "YANG Notifications" registry group defined in [RFC3553].

Registry Name: https-capability
Specification: RFC XXXX
Repository: "Capabilities for HTTPS Notification Receivers" registry

The following note shall be at the top of the registry:

This registry defines capabilities that can be supported by HTTPS-based notification receivers.

The fields for each registry are:

* URN
  - The name of the URN (required).
  - The URN must conform to the syntax described by [RFC8141].
  - The URN must begin with the string "urn:ietf:params:yang-notif:https-capability".
* Reference
  - The RFC that defined the URN.

- The RFC must be in the form "RFC <Number>: <Title>.

* Description

  - An arbitrary description of the capability.

  - The description should be no more than a few sentences.

  - The description is to be in English, but may contain UTF-8 characters as may be needed in some cases.

The update policy is "RFC Required".

Following is the initial assignment for this registry:

```
Record:
   URN:         urn:ietf:params:yang-notif:https-capability:encoding:json
   Reference:   RFC XXXX:An HTTPS-based Transport for YANG Notifications
   Description: Identifies support for JSON-encoded notifications.

Record:
   URN:         urn:ietf:params:yang-notif:https-capability:encoding:xml
   Reference:   RFC XXXX:An HTTPS-based Transport for YANG Notifications
   Description: Identifies support for XML-encoded notifications.

Record:
   URN:         urn:ietf:params:yang-notif:https-capability:sub-notif
   Reference:   RFC XXXX:An HTTPS-based Transport for YANG Notifications
   Description: Identifies support for state machine described in
                RFC 8639, enabling the publisher to send, e.g., the
                "subscription-started" notification.
```

9. References

9.1. 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>.

[RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, DOI 10.17487/RFC3553, June 2003, <https://www.rfc-editor.org/info/rfc3553>.

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

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <https://www.rfc-editor.org/info/rfc5277>.

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

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

[RFC7303] Thompson, H. and C. Lilley, "XML Media Types", RFC 7303, DOI 10.17487/RFC7303, July 2014, <https://www.rfc-editor.org/info/rfc7303>.

[RFC7407] Bjorklund, M. and J. Schoenwaelder, "A YANG Data Model for SNMP Configuration", RFC 7407, DOI 10.17487/RFC7407, December 2014, <https://www.rfc-editor.org/info/rfc7407>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

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

[RFC9112] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/info/rfc9112>.

[RFC9113] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <https://www.rfc-editor.org/info/rfc9113>.

[RFC9114] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/info/rfc9114>.

[RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

[I-D.ietf-netconf-http-client-server] Watsen, K., "YANG Groupings for HTTP 1.1/2.0 Clients and HTTP Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-http-client-server-15, 26 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-http-client-server-15>.

[I-D.ietf-netconf-tls-client-server] Watsen, K., "YANG Groupings for TLS Clients and TLS Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-tls-client-server-36, 29 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-tls-client-server-36>.

9.2. Informative references

[RFC8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017, <https://www.rfc-editor.org/info/rfc8141>.

Appendix A. Configuration Examples

This non-normative section shows two examples for how the "ietf-https-notif-transport" module can be used to configure a publisher to send notifications to a receiver.

In both examples, the publisher, being an HTTPS client, is configured to send notifications to a receiver.

A.1. Using Subscribed Notifications (RFC 8639)

This example shows how an RFC 8639 [RFC8639] based publisher can be configured to send notifications to a receiver.

```
============== NOTE: '\' line wrapping per RFC 8792 ================

<?xml version="1.0" encoding="UTF-8"?>
<subscriptions
    xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications\
">
  <receiver-instances
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-recei\
```

```
   vers">
       <receiver-instance>
         <name>global-receiver-def</name>
         <https-receiver
             xmlns="urn:ietf:params:xml:ns:yang:ietf-https-notif-transp\
   ort"
             xmlns:x509c2n="urn:ietf:params:xml:ns:yang:ietf-x509-cert-\
   to-name">
           <tls>
             <tcp-client-parameters>
               <remote-address>receiver.example.com</remote-address>
               <remote-port>443</remote-port>
             </tcp-client-parameters>
             <tls-client-parameters>
               <server-authentication>
                 <ca-certs>
                   <local-definition>
                     <certificate>
                       <name>Server Cert Issuer #1</name>
                       <cert-data>base64encodedvalue==</cert-data>
                     </certificate>
                   </local-definition>
                 </ca-certs>
               </server-authentication>
             </tls-client-parameters>
             <http-client-parameters>
               <client-identity>
                 <basic>
                   <user-id>my-name</user-id>
                   <cleartext-password>my-password</cleartext-password>
                 </basic>
               </client-identity>
               <path
                   xmlns="urn:ietf:params:xml:ns:yang:ietf-https-notif-\
   transport">/some/path</path>
             </http-client-parameters>
           </tls>
           <receiver-identity>
             <cert-maps>
               <cert-to-name>
                 <id>1</id>
                 <fingerprint>11:0A:05:11:00</fingerprint>
                 <map-type>x509c2n:san-any</map-type>
               </cert-to-name>
             </cert-maps>
           </receiver-identity>
         </https-receiver>
       </receiver-instance>
```

```
    </receiver-instances>
    <subscription>
      <id>6666</id>
      <transport xmlns:ph="urn:ietf:params:xml:ns:yang:ietf-https-noti\
  f-transport">ph:https</transport>
      <stream-subtree-filter>
        <some-subtree-filter/>
      </stream-subtree-filter>
      <stream>some-stream</stream>
      <receivers>
        <receiver>
          <name>subscription-specific-receiver-def</name>
          <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ie\
  tf-subscribed-notif-receivers">global-receiver-def</receiver-instanc\
  e-ref>
        </receiver>
      </receivers>
    </subscription>
  </subscriptions>
  <truststore xmlns="urn:ietf:params:xml:ns:yang:ietf-truststore">
    <certificate-bags>
      <certificate-bag>
        <name>explicitly-trusted-server-ca-certs</name>
        <description>
          Trust anchors (i.e. CA certs) that are used to
          authenticate connections to receivers.  Receivers
          are authenticated if their certificate has a chain
          of trust to one of these CA certificates.
          certificates.
        </description>
        <certificate>
          <name>ca.example.com</name>
          <cert-data>base64encodedvalue==</cert-data>
        </certificate>
        <certificate>
          <name>Fred Flintstone</name>
          <cert-data>base64encodedvalue==</cert-data>
        </certificate>
      </certificate-bag>
    </certificate-bags>
  </truststore>
```

A.2. Not Using Subscribed Notifications

In the case that it is desired to use HTTPS-based notifications outside of Subscribed Notifications, an application-specific module would need to define the configuration for sending the notification.

Following is an example module. Note that the module "uses" the "https-receiver-grouping" grouping from the "ietf-https-notif-transport" module.

```
module example-custom-module {
  yang-version 1.1;
  namespace "http://example.com/example-custom-module";
  prefix "custom";

  import ietf-https-notif-transport {
    prefix "hnt";
    reference
      "RFC XXXX:
        An HTTPS-based Transport for Configured Subscriptions";
  }

  organization
    "Example, Inc.";

  contact
    "Support at example.com";

  description
    "Example of module not using Subscribed Notifications module.";

  revision "2024-02-01" {
    description
      "Initial Version.";
    reference
      "RFC XXXX: An HTTPS-based Transport for YANG Notifications.";
  }

  container example-module {
    description
      "Example of using HTTPS notif without having to
       implement Subscribed Notifications.";

    container https-receivers {
      description
        "A container of all HTTPS notif receivers.";
      list https-receiver {
       key "name";
        description
          "A list of HTTPS notif receivers.";
       leaf name {
         type string;
         description
           "A unique name for the https notif receiver.";
```

```
          }
           uses hnt:https-receiver-grouping;
         }
       }
     }
   }
```

Following is what the corresponding configuration looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<example-module xmlns="http://example.com/example-custom-module">
  <https-receivers>
    <https-receiver>
      <name>foo</name>
      <tls>
        <tcp-client-parameters>
          <remote-address>receiver.example.com</remote-address>
          <remote-port>443</remote-port>
        </tcp-client-parameters>
        <tls-client-parameters>
          <server-authentication>
            <ca-certs>
              <local-definition>
                <certificate>
                  <name>Server Cert Issuer #1</name>
                  <cert-data>base64encodedvalue==</cert-data>
                </certificate>
              </local-definition>
            </ca-certs>
          </server-authentication>
        </tls-client-parameters>
        <http-client-parameters>
          <client-identity>
            <basic>
              <user-id>my-name</user-id>
              <cleartext-password>my-password</cleartext-password>
            </basic>
          </client-identity>
          <path>/some/path</path>
        </http-client-parameters>
      </tls>
    </https-receiver>
  </https-receivers>
</example-module>
```

Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by first name): Eric Voit, Henning Rogge, Martin Bjorklund, Reshad Rahman, and Rob Wilton.

In addition, the authors would also like to thank Quifang Ma for providing thoughtful comments as part of shepherd writeup.

Authors' Addresses

Mahesh Jethanandani
Kloud Services
Email: mjethanandani@gmail.com

Kent Watsen
Watsen Networks
Email: kent+ietf@watsen.net

NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 August 2022

Q. Wu
Q. Ma
Huawei
P. Liu
China Mobile
W. Wang
China Telecom

24 February 2022

Data Export Notification Capability
draft-tao-netconf-data-export-capabilities-07

Abstract

This document proposes a YANG module for data export notification capabilities which augments "ietf-system-capabilities" YANG module defined in [RFC9196] and provides additional data export attributes associated with system capabilities for transport specific Notification. This YANG module can be used by the client to learn capability information from the server at runtime or at implementation time, by making use of the YANG instance data file format.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 August 2022.

Copyright Notice

Table of Contents

1. Introduction

Notification capabilities model defined in [RFC9196] allows a client to discover a set of capabilities supported by the server (e.g., basic system capabilities and YANG-Push related capabilities) both at implementation time and at runtime. These capabilities allow the client to adjust its behavior to take advantage of the features exposed by the server.

However the client and the server may still support various different transport specific parameters (e.g., transport protocol, encoding format, encryption). As described in section 3.1 of [RFC8641], a simple negotiation (i.e., inserting hints into error responses to a failed RPC request) between subscribers and publishers for subscription parameters increases the likelihood of success for subsequent RPC requests, but not guaranteed, which may cause unexpected failure or additional message exchange between client and server.

This document defines a corresponding solution that is built on top of [RFC9196]. Supplementing that work are YANG data model augmentations for transport specific notification. The module can be used by the client to discover capability information from the server at runtime or at implementation time, by making use of the YANG instance data file format.

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. Data Export Capability

The YANG module "ietf-notification-capabilities" defined in [RFC9196] specifies the following server capabilities related to YANG Push:

* Supported (reporting) periods for "periodic" subscriptions

* Maximum number of objects that can be sent in an update

* The set of datastores or data nodes for which "periodic" or "on-change" notification is supported

* Supported dampening periods for "on-change" subscriptions

These server capabilities are transport independent, session level capabilities. They can be provided either at the implementation time or reported at runtime.

This document augments System Capabilities model and provides additional data export attributes associated with system capabilities:

* Specification of transport protocols the client can use to establish a transport connection;

* Specification of the encoding selection used (e.g., XML or JSON, Binary) for Data modeled with YANG;

* Specification of secure transport mechanisms that are needed by the client to communicate with the server;

* Specification of the notification message encapsulation type, either one notification per message or multiple notifications per message [I-D. ietf-netconf-notification-messages].

2.1. Tree Diagram

The following tree diagram [RFC8340] provides an overview of the data model.

```
module: ietf-data-export-capabilities
  augment /sysc:system-capabilities:
    +--ro data-export-capabilities
       +--ro data-export-capability* []
          +--ro transport-protocol?            identityref
          +--ro encoding-format*               identityref
          +--ro security-protocol?             identityref
          +--ro message-bundling-support?      empty
```

3. YANG Module

```
<CODE BEGINS> file "ietf-data-export-capabilities.yang"
module ietf-data-export-capabilities {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-data-export-capabilities";
  prefix dec;

  import ietf-system-capabilities {
    prefix sysc;
  }
  import ietf-notification-capabilities {
    prefix inc;
  }
  organization
    "IETF NETCONF (Network Configuration) Working Group";
  contact
    "WG Web:   <https://tools.ietf.org/wg/netconf/>
     WG List:  <mailto:netconf@ietf.org>
     Editor:   Qin Wu <mailto:bill.wu@huawei.com>
     Editor:   Qiufang Ma <mailto: maqiufang1@huawei.com>
     Editor:   Peng Liu <mailto: liupengyjy@chinamobile.com>
     Editor:   Wei Wang <mailto: wangw36@chinatelecom.cn>";
  description
    "This module defines an extension to System Capability and YANG Push
     Notification Capabilities model and provides additional data export
     attributes for transport specific notification.

     Copyright (c) 2019 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;
        see the RFC itself for full legal notices.";

    revision 2020-07-03 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: Data Export Notification Capability";
    }

    identity transport-protocol {
      description
        "Base identity for transport protocol type.";
    }

    identity tcp {
      base transport-protocol;
      description
        "Identity for tcp as transport protocol.";
    }

    identity udp-notif {
      base transport-protocol;
      description
        "Identity for udp notif as transport protocol.";
      reference
        "draft-ietf-netconf-udp-notif:UDP-based Transport
         for Configured Subscriptions";
    }

    identity http-notif {
      base transport-protocol;
      description
        "Identity for http notif as transport protocol.";
      reference
        "draft-ietf-netconf-https-notif: An HTTPS-based
         Transport for Configured Subscriptions";
    }

    identity grpc {
      base transport-protocol;
```

```
    description
      "Identity for grpc as transport protocol.";
  }

  identity security-protocol {
    description
      "Base identity for security protocol type.";
  }

  identity tls {
    base security-protocol;
    description
      "Identity for tls security protocol.";
  }

  identity dtls {
    base security-protocol;
    description
      "Identity for dtls security protocol.";
  }

  identity ssh {
    base security-protocol;
    description
      "Identity for ssh transport protocol.";
  }

  identity encoding-format {
    description
      "Base identity for encoding format type.";
  }

  identity xml {
    base encoding-format;
    description
      "Identity for xml encoding format.";
  }

  identity json {
    base encoding-format;
    description
      "Identity for json encoding format.";
  }

  identity binary {
    base encoding-format;
    description
      "Identity for binary encoding format.";
```

```
     }

     identity cbor {
       base binary;
       description
         "Identity for cbor encoding format.";
     }

     augment "/sysc:system-capabilities" {
       description
         "Add system level capability.";
       container data-export-capabilities {
         description
           "Capabilities related to data export notification capabilities negotia
tion.";
         list data-export-capability {
           description
             "Capability list related to data export notification capabilities ne
gotiation.";
           leaf transport-protocol {
             type identityref {
               base transport-protocol;
             }
             description
               "Type of transport protocol.";
           }
           leaf-list encoding-format {
             type identityref {
               base encoding-format;
             }
             description
               "Type of encoding format.";
           }
           leaf security-protocol {
             type identityref {
               base security-protocol;
             }
             description
               "Type of secure transport.";
           }
           leaf message-bundling-support {
             type empty;
             description
               "Enables message bundling support.";
           }
         }
       }
     }
   }
   <CODE ENDS>
```

4. IANA Considerations

4.1. Updates to the IETF XML Registry

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in [RFC3688], the following registration has been made:

```
URI:
   urn:ietf:params:xml:ns:yang:ietf-data-export-capabilities
Registrant Contact:
   The IESG.
XML:
   N/A; the requested URI is an XML namespace.
```

4.2. Updates to the YANG Module Names Registry

This document registers one YANG module in the "YANG Module Names" registry [RFC6020]. Following the format in [RFC6020], the following registration has been made:

```
name:
   ietf-data-export-capabilities
namespace:
   urn:ietf:params:xml:ns:yang:ietf-data-export-capabilities
prefix:
   dec
reference:
   RFC XXXX (RFC Ed.: replace XXX with actual RFC number and remove
   this note.)
```

5. 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 NETCONF 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.

All protocol-accessible data nodes are read-only and cannot be modified. The data in these modules is not security sensitive. Access control may be configured, to avoid exposing the read-only data.

When this data is in file format, data should be protected against modification or unauthorized access using normal file handling mechanisms.

6. Contributors

Ran Tao
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China
Email: taoran20@huawei.com

Liang Geng
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing 10053

Email: gengliang@chinamobile.com

Thomas Graf
Swisscom
Binzring 17
Zuerich 8045
Switzerland

Email: thomas.graf@swisscom.com

7. References

7.1. 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>.

[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>.

[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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[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>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

[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>.

[RFC9196] Lengyel, B., Clemm, A., and B. Claise, "YANG Modules Describing Capabilities for Systems and Datastore Update Notifications", RFC 9196, DOI 10.17487/RFC9196, February 2022, <https://www.rfc-editor.org/info/rfc9196>.

7.2. Informative References

[I-D.ietf-netconf-https-notif]
Jethanandani, M. and K. Watsen, "An HTTPS-based Transport for YANG Notifications", Work in Progress, Internet-Draft, draft-ietf-netconf-https-notif-09, 24 October 2021, <https://www.ietf.org/archive/id/draft-ietf-netconf-https-notif-09.txt>.

[I-D.ietf-netconf-notification-messages]
Voit, E., Jenkins, T., Birkholz, H., Bierman, A., and A. Clemm, "Notification Message Headers and Bundles", Work in Progress, Internet-Draft, draft-ietf-netconf-notification-messages-08, 17 November 2019, <https://www.ietf.org/archive/id/draft-ietf-netconf-notification-messages-08.txt>.

[I-D.ietf-netconf-udp-notif]
Zheng, G., Zhou, T., Graf, T., Francois, P., Feng, A. H., and P. Lucente, "UDP-based Transport for Configured Subscriptions", Work in Progress, Internet-Draft, draft-ietf-netconf-udp-notif-04, 21 October 2021, <https://www.ietf.org/archive/id/draft-ietf-netconf-udp-notif-04.txt>.

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

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

[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>.

Appendix A. Usage Example of interaction with UDP Notif and HTTP Notif for Configured Subscription

The following instance-data example describes the Data Export Notification capabilities of a hypothetical "acme-router".

```
<?xml version="1.0" encoding="UTF-8"?>
<instance-data-set xmlns=
"urn:ietf:params:xml:ns:yang:ietf-yang-instance-data">
  <name>acme-router-notification-capabilities</name>
  <content-schema>
    <module>ietf-system-capabilities@2020-03-23</module>
    <module>ietf-notification-capabilities@2020-03-23</module>
    <module>ietf-data-export-capabilities@2020-03-23</module>
  </content-schema>
  <!-- revision date, contact, etc. -->
  <description>Server Capability Discovery</description>
  <content-data>
    <system-capabilities
      xmlns="urn:ietf:params:xml:ns:yang:ietf-system-capabilities"
      xmlns:inc="urn:ietf:params:xml:ns:yang:ietf-notification-capabilities"
      xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
      <data-export-capabilities>
        <data-export-capability>
          <transport-protocol>http-notif</transport-protocol>
          <encoding-format>json</encoding-format>
          <encoding-format>xml</encoding-format>
        </data-export-capability>
        <data-export-capability>
          <transport-protocol>udp-notif</transport-protocol>
          <encoding-format>binary</encoding-format>
        </data-export-capability>
      </data-export-capabilities>
    </system-capabilities>
  </content-data>
</instance-data-set>
```

In addition, the client could also query data export capability from the server. For example, the client sends <get> request message to the the server to query data export capability from the server.

```
 <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-capabili
ties">
       <data-export-capabilities/>
     </system-capabilities>
    </filter>
   </get>
 </rpc>
```

The server returns server data export capability using <rpc-reply> as follows:

```
<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:dec="urn:ietf:params:xml:ns:yang:ietf-data-export-capabilities">
    <data-export-capabilities>
      <data-export-capability>
        <transport-protocol>http-notif</transport-protocol>
        <encoding-format>json</encoding-format>
      </data-export-capability>
      <data-export-capability>
        <transport-protocol>udp-notif</transport-protocol>
        <encoding-format>binary</encoding-format>
      </data-export-capability>
    </data-export-capabilities>
  </system-capabilities>
</data>
</rpc-reply>
```

Appendix B. Changes between Revisions

v06-v07

* Delete the per-node related capability parameters from the Appendix.

v05-v06

* Revise abstract and introduction sessions so that the scope of this draft is not limited to telemetry but other notification.

* Revise the description of module ietf-system-capabilities to align with the latest version of draft-ietf-netconf-notification-capabilities.

* Remove compression-mode, timer-event-support and suppress-redundant parameters in the model.

* Move per-node related capability parameters to appendix section.

* Add a container to wrap data export capabilities to cleanly separate different groups of capabilities.

v04 - v05

* Change per-node-capabilities related parameters into empty type.

* Revise abstract and introduction section to only focus on capability fetching mechanism from the client to the server.

* Update Usage Example of interaction with HTTP-Notif and UDP-Notif for Configured Subscription.

v03 - v04

* Add interface namespace in the Adaptive Subscription usage example.

* subtrees and data nodes changes in the security section.

* Two compression mode related identities change.

* Move message-bundling-support parameer to system capabilities level.

* Add an example to discuss report reciever capability from the client per yang instance file format.

* Change encoding format from leaf to leaf-list and support multiple encoding formats for the same transport specific notif.

v02 - v03

* Change 'data-export-capabilities' into list type to support multiple transport protocol, encoding on the server.

* Add Usage Example of interaction with UDP based Transport for Configured Subscription.

* Add Thomas Graf as a contributor;

* Update motivation in the introduction to clarify why this work is needed.

* Support udp notif and http notif as two optional transport in the YANG data model.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China

Email: bill.wu@huawei.com

Qiufang Ma
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China
Email: maqiufang1@huawei.com

Peng Liu
China Mobile
Beiqijia Town, Changping District
Beijing
Email: liupengyjy@chinamobile.com

Wei Wang
China Telecom
32 Xuanwumen West St, Xicheng District
Beijing
Email: wangw36@chinatelecom.cn

NETCONF Working Group
Internet-Draft
Intended status: Experimental
Expires: 3 November 2022

Q. Wu
W. Song
Huawei
P. Liu
China Mobile
Q. Ma
Huawei
W. Wang
China Telecom
Z. Niu
Microsoft
2 May 2022

Adaptive Subscription to YANG Notification
draft-wang-netconf-adaptive-subscription-10

Abstract

This document defines a YANG data model and associated mechanism enabling the subscriber's adaptive subscriptions to a publisher's event streams with various different period intervals to report updates. Applying these elements allows servers automatically adjust the rate and volume of telemetry traffic sent from a publisher to receivers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 November 2022.

Table of Contents

1. Introduction

YANG-Push subscriptions [RFC8641] allow subscriber applications to request a continuous customized stream of updates from a YANG datastore without needing to poll. It defines a mechanism (i.e., update trigger) to determine when an update record needs to be generated. Two types of subscriptions are introduced in [RFC8641], distinguished by how updates are triggered: periodic and on-change.

* Periodic subscription allows subscribed data to be streamed to the destination at a configured fixed periodic interval;

* On-change subscription allows update to be triggered whenever a change in the subscribed information is detected.

However in some large scale deployments (e.g., massive data collection for wireless network performance monitoring) where an increased data collection rate is used, it becomes more likely that both clients and servers are temporarily overwhelmed with a burst of streamed data and consumes expensive network resource (e.g., bandwidth resource, radio resource) and computation resource, therefore hard to continuously monitor operational data, especially values that fall outside normal operational ranges. If the rate at which we can collect a stream of data is set too low or chosen to get low priority telemetry data dropped, these telemetry data are not sufficient to detect and diagnose problems and verify correct network behavior.

A client might choose to monitor the operational state and send a request to modify the data collection rate on the server. But how often the client evaluates if the modification of the data collection rate is required highly depends on the current collection rate, collecting a stream of data at a low rate prevents the subscriber from capturing sufficient data for timely decision-making, which may result in service discontinuity. In addition, when tens of thousands of network devices need to be managed, frequent follow-up modification requests are prone to errors.

There is a need for a service to balance between data management cost and real time streaming telemetry. To achieve this, servers can be configured with multiple different period intervals and corresponding subscription update policy which allows servers/publishers automatically switch to different period intervals according to the network condition change without the interaction with the client for policy update instruction, e.g., when the wireless signal strength falls below a configured threshold, the subscribed data can be streamed at a higher rate to capture potentially important data and events (e.g., continuous service degeneration); while when the wireless signal strength crosses a configured threshold, the subscribed data can be streamed at a lower rate.

This document defines a YANG data model and associated mechanism enabling the subscriber’s adaptive subscriptions to a publisher’s event streams. Applying these elements allows servers to automatically adjust the rate and volume of telemetry traffic sent from a publisher to receivers.

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.

The following terms are defined in [RFC5277] [RFC7950] [RFC3198] [RFC8342] [RFC8639] [RFC8641] and are not redefined here:

* Event
* Client
* Configuration
* Configured subscription
* Configuration datastore
* Notification message
* Publisher
* Receiver
* Subscriber
* Subscription
* On-change subscription
* Periodic subscription
* Selection filter

This document defines the following term:

Adaptive Subscription: Apply subscription update policy on the servers and allow servers/publishers automatically switch to different period intervals according to the network condition change without the interaction with the client for update policy instruction.

2. Model Overview

This document defines a YANG module "ietf-adaptive-subscription", which augments the "update-trigger" choice defined in the "ietf-yang-push" module [RFC8641] with subscription configuration parameters that are specific to a subscriber's adaptive subscription.

In addition to subscription state notifications defined in [RFC8639] and notifications for subscribed content defined in [RFC8641], "ietf-adaptive-subscription" YANG module also defines "adaptive-period-update" notification to report the update interval change.

The following tree diagrams [RFC8340] provide an overview of the data model for "ietf-adaptive-subscription" module.

```
module: ietf-adaptive-subscription
  augment /sn:subscriptions/sn:subscription/yp:update-trigger:
    +--rw (adaptive-subscription)?
       +--:(adaptive-subscriptions)
          +--rw adaptive-subscriptions
              +--rw adaptive-period* [name]
                 +--rw name                      string
                 +--rw xpath-external-eval       string
                 +--rw period                    yp:centiseconds
                 +--rw anchor-time?              yang:date-and-time
  augment /sn:establish-subscription/sn:input/yp:update-trigger:
    +-- (adaptive-subscription)?
       +--:(adaptive-subscriptions)
          +--rw adaptive-subscriptions
             +--rw adaptive-period* [name]
                +--rw name                      string
                +--rw xpath-external-eval       string
                +--rw period                    yp:centiseconds
                +--rw anchor-time?              yang:date-and-time
  notifications:
    +---n adaptive-period-update
       +--ro id?                                      sn:subscription-id
       +--ro period                                   yp:centiseconds
       +--ro anchor-time?                             yang:date-and-time
       +--ro (selection-filter)?
          +--:(by-reference)
          |  +--ro selection-filter-ref               selection-filter-ref
          +--:(within-subscription)
             +--ro (filter-spec)?
                +--:(datastore-subtree-filter)
                |  +--ro datastore-subtree-filter?   <anydata> {sn:subtree}?
                +--:(datastore-xpath-filter)
                   +--ro datastore-xpath-filter?     yang:xpath1.0 {sn:xpath}?
```

2.1. Subscription Configuration

For adaptive subscriptions, triggered updates will occur at the boundaries of specified time intervals when a trigger condition is satisfied. These boundaries can be calculated from the following adaptive periodic parameters:

* a "name" represents the name of each adaptive period;

* a "period" that defines the new duration between push updates. The period can be switched based on trigger conditions;

* an "anchor-time"; update intervals fall on the points in time that are a multiple of a "period" from an "anchor-time". If an "anchor-time" is not provided, then the "anchor-time" MUST be set with the creation time of the initial update record.

* an "xpath-external-eval" represents a standard XPath evaluation expression (See section 6.4 of [RFC7950]) that is applied against the targeted data object, which is used to trigger/control the update interval switching within the server. It follows the rules defined in section 3.4 of [XPATH1.0] and contains comparisons of the targeted datastore node with its value to the specific threshold in the XPath format. Different from selection filter defined in [RFC8641],

  - it is applied against a single targeted object rather than a set of target objects.

  - it monitors a specific data object change and evaluates the trigger condition associated with the targeted object to be true or false using XPATH rules and does not influence the event records output generation from a publisher.

  How often the XPath expression criterion is evaluated is up to the publisher's implementation. With minimal delay, the expression can be evaluated whenever changes to targeted object occur, or at the end of each high-frequency streaming update period. To reduce the frequency of evaluation, the server can choose to check targeted object change at every multiple (e.g.,2 or 3) high-frequency streaming update periods.

  The represented expression defined in "xpath-external-eval" is evaluated in the following XPath context:

- The set of namespace declarations is the set of prefix and namespace pairs for all YANG modules implemented by the server, where the prefix is the YANG module name and the namespace is as defined by the "namespace" statement in the YANG module.

- If the leaf is encoded in XML, all namespace declarations in scope on the "xpath-external-eval" leaf element are added to the set of namespace declarations. If a prefix found in the XML is already present in the set of namespace declarations, the namespace in the XML is used.

- The set of variable bindings is empty.

- The function library is the core function library defined in [XPATH1.0] and the function defined in Section 10 in RFC 7950.

- The context node is the root node.

For the cases where multiple list instances are needed to handle in "xpath-external-eval", XPath abbreviated syntax can be used to identify a particular instance, e.g., to represent a comparison for a leaf in a list entry:

```
/if:interfaces/if:interface[if:name="eth0"]/if:in-errors>1000.
```

The server MUST convert the XPath expression defined in "xpath-external-eval" to a boolean value and internally apply the "boolean" function defined in Section 4.3 in [XPATH1.0] if the evaluated result is not a boolean.

Note that the adaptive subscription may not be supported by every YANG datastore nodes. A publisher MAY decide to simply reject an adaptive subscription with "adaptive-unsupported" (defined in Section 2.2.1.1) if the scope of the subscription contains selected data nodes for which adaptive subscription is not supported.

2.2. YANG RPC

2.2.1. "establish-subscription" RPC

The augmentation of YANG module "ietf-yang-push" made to RPCs specified in YANG module "ietf-subscribed-notifications" [RFC8639] is introduced. This augmentation concerns the "establish-subscription" RPC, which is augmented with parameters that are needed to specify a subscriber's adaptive subscriptions. These parameters are the same as ones defined in Section 2.1.

2.2.1.1. RPC Failures

As specified in [RFC8639] and [RFC8641], RPC error responses from the publisher are used to indicate a rejection of an RPC for any reason. This document introduces three new RPC errors for "establish-subscription" RPC.

establish-subscription
adaptive-unsupported
xpath-evaluation-unsupported
multi-xpath-criteria-conflict

Adaptive-unsupported is used to indicate that the adaptive subscription is not supported for any objects that are selectable by the filter.

Xpath-evaluation-unsupported is used to indicate that a server fails to parse syntax defined in "xpath-external-eval". The failure can be caused by either a syntax error or some XPath 1.0 syntax not supported against the specific object.

Multi-xpath-criteria-conflict is used to indicate that the multiple Xpath evaluation criteria represented by "xpath-external-eval" is evaluated as conflict, i.e., more than one condition expressions are evaluated to "true". Such a conflict may also cause an ongoing adaptive-subscription terminated.

For an example of how above RPC errors can be returned, see the "xpath-evaluation-unsupported" error response illustrated in Appendix B.3.

2.3. Notifications for Adaptive Subscribed Content

The adaptive update notification is similar to subscription state change notifications defined in [RFC8639]. It is inserted into the sequence of notification messages sent to a particular receiver. The adaptive update notification cannot be dropped or filtered out, it cannot be stored in replay buffers, and it is delivered only to impacted receivers of a subscription. The identification of adaptive update notification is easy to separate from other notification messages through the use of the YANG extension "subscription-state-notif". This extension tags a notification as a subscription state change notification.

The objects in the ’adaptive-period-update’ notification include:

* a "period" that defines the duration between push updates, the period can be changed based on trigger conditions.

* an "anchor-time"; update intervals fall on the points in time that are a multiple of a "period" from an "anchor-time". If an "anchor-time" is not provided, then the "anchor-time" MUST be set with the creation time of the initial update record.

* A selection filter is to identify YANG nodes of interest in a datastore. Filter contents are specified via a reference to an existing filter or via an in-line definition for only that subscription based on XPath Evaluation criteria defined in section 6.4 of [RFC7950]. Referenced filters allow an implementation to avoid evaluating filter acceptability during a dynamic subscription request. The "case" statement differentiates the options. Note that filter contents are not affected by "xpath-external-eval" parameter defined by the update trigger.

3. XPath Complexity Evaluation

YANG-Push subscriptions [RFC8641] specify selection filters to identify targeted YANG datastore nodes and/or datastore subtrees for which updates are to be pushed. In addition, it specifies update policies which contain conditions that trigger generation and pushing of new update records. To support a subscriber's adaptive subscription defined in this document, the trigger condition can also use similar selection filters to express a standard XPath Evaluation criterion (section 6.4 of [RFC7950]) against targeted data objects.

Similar to on-change subscriptions, adaptive subscriptions are particularly effective for data that changes infrequently, the following complex design choices need to be cautious, although these designs have already been well supported by the section 3.4 of [XPATH1.0]:

* Support XPath Evaluation criteria against every data object;

* Support more than one target object selection and operation(e.g., addition, subtraction, division and multiplication) in the XPath evaluation criterion;

* Support any type of node set in the XPath evaluation criterion, e.g., string,int64, uint64, and decimal64 types;

* Both objects in the XPath Evaluation criterion to be compared are node-sets;

* Two objects to be compared are in different data types, e.g., one is an integer, the other is a string

As described in section 6.4 of RFC7950, Numbers in XPath 1.0 are IEEE 754 [IEEE754-2008] double-precision floating-point values; some values of int64, uint64, and decimal64 types cannot be exactly represented in XPath expressions.

If two objects to be compared are in different data types, conversion function is needed to convert different data types into numbers.

If both objects in XPath Evaluation criteria to be compared are node-sets, more computation resources are required which add complexity.

To reduce these complexities, the following design principles are recommended:

* XPath Evaluation criteria against a minimal set of data objects in the data model, the minimal set of data objects can be advertised using Notification capabilities model defined in [RFC9196].

* XPath Evaluation criteria only support condition expressions that filter updates based on numbers.

* One object to be compared in the XPath Evaluation criteria MUST be a leaf data node.

* The other object to be compared in the XPath Evaluation criteria MUST be a number data type.

If a server receives an XPath Evaluation criterion with some XPath syntax unsupported against the specific object, an RPC error with "xpath-evaluation-unsupported" should be returned.

4. Adaptive Subscription YANG Module

<CODE BEGINS> file "ietf-adaptive-subscription@2020-02-14.yang"

```
module ietf-adaptive-subscription {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-adaptive-subscription";
  prefix as;
  import ietf-subscribed-notifications {
    prefix sn;
  }
  import ietf-yang-push {
    prefix yp;
  }
  import ietf-yang-types {
```

```
      prefix yang;
    }

    organization
      "IETF NETCONF (Network Configuration) Working Group";
    contact
      "";
    description
      "NETCONF Protocol Data Types and Protocol Operations.
       Copyright (c) 2020 IETF Trust and the persons identified as
       the document authors.  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; see
       the RFC itself for full legal notices.";

    revision 2020-02-14 {
      description
        "Initial revision";
      reference
        "RFCxxx Adaptive subscription to YANG notification.";
    }

    identity adaptive-unsupported {
      base sn:establish-subscription-error;
      description
        "Adaptive-subscription is not supported for any objects
         that are selectable by the filter.";
    }

    identity xpath-evaluation-unsupported {
      base sn:establish-subscription-error;
      description
        "Unable to parse the xpath evaluation criteria defined in
        xpath-external-eval because of a syntax error or some
         XPath 1.0 syntax not supported against the specific object.";
    }

    identity multi-xpath-criteria-conflict {
      base sn:establish-subscription-error;
      base sn:subscription-terminated-reason;
      description
```

```
        "Multiple Xpath evaluation criteria represented by
         'xpath-external-eval' is evaluated as conflict, i.e.,
         more than one condition expressions are evaluated to
        'true'.";
   }

     grouping adaptive-subscription-modifiable {
       description
         "This grouping describes the datastore-specific adaptive subscription
          conditions that can be changed during the lifetime of the
          subscription.";
       choice adaptive-subscription {
         description
           "Defines necessary conditions for sending an event record to
                  the subscriber.";
         container adaptive-subscriptions {
           list adaptive-period {
             key "name";
             description
               "Defines necessary conditions to switch update interval for
               sending an event record to the subscriber. The event record output
               generation will not be influenced these conditions.";
             leaf name {
               type string {
             length "1..64";
              }
           description
             "The name of the condition to be matched.  A device MAY further
              restrict the length of this name; space and special
              characters are not allowed.";
            }
             leaf xpath-external-eval {
               type string;
               description
                 "A XPath string, representing a logical expression,
                  which can contain comparisons of datastore values
                  and logical operations in the XPath format.";
             }
             leaf period {
               type yp:centiseconds;
               mandatory true;
               description
                 "Duration of time that should occur between periodic
                  push updates, in units of 0.01 seconds.";
             }
             leaf anchor-time {
               type yang:date-and-time;
               description
```

```
                "Designates a timestamp before or after which a series
                 of periodic push updates are determined.  The next
                 update will take place at a point in time that is a
                 multiple of a period from the 'anchor-time'.
                 For example, for an 'anchor-time' that is set for the
                 top of a particular minute and a period interval of a
                 minute, updates will be sent at the top of every
                 minute that this subscription is active.";
            }
          }
          description
            "Container for adaptive subscription.";
        }
      }
    }

    augment "/sn:subscriptions/sn:subscription/yp:update-trigger" {
      description
        "This augmentation adds additional subscription parameters
         that apply specifically to adaptive subscription.";
      uses adaptive-subscription-modifiable;
    }
    augment "/sn:establish-subscription/sn:input/yp:update-trigger" {
      description
        "This augmentation adds additional subscription parameters
           that apply specifically to datastore updates to RPC input.";
      uses adaptive-subscription-modifiable;
    }

    notification adaptive-period-update {
      sn:subscription-state-notification;
      description
        "This notification contains a push update that in turn contains
         data subscribed to via a subscription.  In the case of a
         periodic subscription, this notification is sent for periodic
         updates.  It can also be used for synchronization updates of
         an on-change subscription.  This notification shall only be
         sent to receivers of a subscription.  It does not constitute
         a general-purpose notification that would be subscribable as
         part of the NETCONF event stream by any receiver.";
      leaf id {
        type sn:subscription-id;
        description
          "This references the subscription that drove the
           notification to be sent.";
      }
      leaf period {
        type yp:centiseconds;
```

```
        mandatory true;
        description
          "New duration of time that should occur between periodic
           push updates, in units of 0.01 seconds.";
      }
      leaf anchor-time {
        type yang:date-and-time;
        description
          "Designates a timestamp before or after which a series
           of periodic push updates are determined.  The next
           update will take place at a point in time that is a
           multiple of a period from the 'anchor-time'.
           For example, for an 'anchor-time' that is set for the
           top of a particular minute and a period interval of a
           minute, updates will be sent at the top of every
           minute that this subscription is active.";
      }
      uses yp:datastore-criteria {
        refine "selection-filter/within-subscription" {
          description
            "Specifies the selection filter and where it originated
             from.  If the 'selection-filter-ref' is populated, the
             filter in the subscription came from the 'filters'
             container.  Otherwise, it is populated in-line as part
             of the subscription itself.";
        }
      }
    }
  }
  <CODE ENDS>
```

5. IANA Considerations

5.1. 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 to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-adaptive-subscription
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

5.2. Updates to the YANG Module Names Registry

This document registers one YANG module in the YANG Module Names registry [RFC7950]. Following the format in [RFC6020], the following registration is requested to be made:

```
---------------------------------------------------------------------
   Name:         ietf-adaptive-subscription
   Namespace:    urn:ietf:params:xml:ns:yang:ietf-adaptive-subscription
   Prefix:       as
   Reference:    RFC xxxx
---------------------------------------------------------------------
```

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 NETCONF 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.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

* /sn:subscriptions/sn:subscription/yp:update-trigger/as:adaptive-subscriptions/as:adaptive-period/as:period

* /sn:subscriptions/sn:subscription/yp:update-trigger/as:adaptive-subscriptions/as:adaptive-period/as:anchor-time

* /sn:establish-subscription/sn:input/yp:update-trigger/as:adaptive-subscriptions/as:adaptive-period/as:period

* /sn:establish-subscription/sn:input/yp:update-trigger/as:adaptive-subscriptions/as:adaptive-period/as:anchor-time

7. Contributors

Thanks Michael Wang, Liang Geng for their major contributions to the initial modeling and use cases.

Michael Wang
Email: wangzitao@huawei.com

Liang Geng
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing 10053

Email: gengliang@chinamobile.com

8. Acknowledges

We would like to thank Rob Wilton, Thomas Graf, Andy Bierman, Michael Richardson, Henk Birkholz for valuable review on this document, special thanks to Thomas and Michael for organizing the discussion on several relevant drafts and reach the common understanding on the concept and ideas. Thanks Michael for providing CHIP/Matter WIFI statistics reference.

9. References

9.1. 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>.

[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>.

[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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[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>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

[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>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

[RFC9196] Lengyel, B., Clemm, A., and B. Claise, "YANG Modules Describing Capabilities for Systems and Datastore Update Notifications", RFC 9196, DOI 10.17487/RFC9196, February 2022, <https://www.rfc-editor.org/info/rfc9196>.

9.2. Informative References

[CHIP] CSA, "Connected Home over IP Specification", 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>.

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

[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>.

[XPATH1.0] W3C, "https://www.w3.org/TR/1999/REC-xpath-19991116/", 11 November 1999.

Appendix A. Example YANG Module

The example YANG module used in this document represents a Wi-Fi Network Diagnostics data specified in [CHIP] which can be used by a Node to assist a user or Administrative Node in diagnosing potential problems.

YANG tree diagram for the "example-wifi-network-diagnostic" module:

```
module: example-wifi-network-diagnostic
  +--rw server
  |  +--rw bssid?                        yang:mac-address
  |  +--rw security-type?                enumeration
  |  +--rw wifi-version?                 enumeration
  |  +--rw channel-num?                  int8
  |  +--rw rssi?                         int8
  |  +--rw beacon-lost-count?            int8
  |  +--rw beacon-rx-count?              int8
  |  +--rw packet-multicast-rx-count?    int8
  |  +--rw packet-multicast-tx-count?    int8
  |  +--rw packet-unicast-rx-count?      int8
  |  +--rw packet-unicast-tx-count?      int8
  |  +--rw current-max-rate?             int8
  |  +--rw overrun-count?                int8
  +--rw events
     +--rw event* [name]
        +--rw name                   string
        +--rw disconnection?         enumeration
        +--rw association-failure?   enumeration
        +--rw connection-status?     enumeration
```

A.1. "example-wifi-mac" YANG Module

```
module example-wifi-network-diagnostic {
  yang-version 1;
  namespace "http://example.com/yang/wifi-network-diagnostic";
  prefix wnd;

  import ietf-yang-types {
    prefix yang;
  }

  container server {
    description
      "Configuration of the WiFi Server logical entity.";
    leaf bssid {
      type yang:mac-address;
      description
        "The MAC address of a wireless access point.";
    }
    leaf security-type {
      type enumeration {
        enum unspecified {
          value 0;
        }
        enum none {
          value 1;
        }
        enum wep {
          value 2;
        }
        enum wpa {
          value 3;
        }
        enum wpa2 {
          value 4;
        }
        enum wpa3 {
          value 5;
        }
      }
      description
        "The type of Wi-Fi security used. A value of 0
         indicate that the interface is not currently
         configured or operational.";
    }
    leaf wifi-version {
      type enumeration {
        enum 80211a {
          value 0;
        }
```

```
          enum 80211b {
            value 1;
          }
          enum 80211g {
            value 2;
          }
          enum 80211n {
            value 3;
          }
          enum 80211ac {
            value 4;
          }
          enum 80211ax {
            value 5;
          }
        }
        description
          "The highest 802.11 standard version usable
           by the Node.";
      }
      leaf channel-num {
        type int8;
        description
          "The channel that Wi-Fi communication is currently
           operating on. A value of 0indicates that the interface
           is not currently configured or operational.";
      }
      leaf rssi {
        type int8;
        description
          "The RSSI of the Nodes Wi-Fi radio in dBm.";
      }
      leaf beacon-lost-count {
        type int8;
        description
          "The count of the number of missed beacons the
           Node has detected.";
      }
      leaf beacon-rx-count {
        type int8;
        description
          "The count of the number of received beacons. The
           total number of expected beacons that could have been
           received during the interval since association SHOULD
           match the sum of BeaconRxCount and BeaconLostCount. ";
      }
      leaf packet-multicast-rx-count {
        type int8;
```

```
        description
          "The number of multicast packets received by
           the Node.";
        }
        leaf packet-multicast-tx-count {
          type int8;
          description
            "The number of multicast packets transmitted by
             the Node.";
        }
        leaf packet-unicast-rx-count {
          type int8;
          description
            "The number of multicast packets received by
             the Node.";
        }
        leaf packet-unicast-tx-count {
          type int8;
          description
            "The number of multicast packets transmitted by
             the Node.";
        }
        leaf current-max-rate {
          type int8;
          description
            "The current maximum PHY rate of transfer of
             data in bytes-per-second.";
        }
        leaf overrun-count {
          type int8;
          description
            "The number of packets dropped either at ingress or
             egress, due to lack of buffer memory to retain all
             packets on the ethernet network interface. The
             OverrunCount attribute SHALL be reset to 0 upon a
             reboot of the Node..";
        }
      }
      container events {
        description
          "Configuration of WIFI Network Diagnostic events.";
        list event {
          key "name";
          description
            "The list of event sources configured on the
             server.";
          leaf name {
            type string;
```

```
          description
            "The unique name of an event source.";
        }
        leaf disconnection {
          type enumeration {
            enum de-authenticated {
              value 1;
            }
            enum dis-association {
              value 2;
            }
          }
          description
            "A Nodes Wi-Fi connection has been disconnected as a
             result of de-authenticated or dis-association and
             indicates the reason.";
        }
        leaf association-failure {
          type enumeration {
            enum unknown {
              value 0;
            }
            enum association-failed {
              value 1;
            }
            enum authentication-failed {
              value 2;
            }
            enum ssid-not-found {
              value 3;
            }
          }
          description
            "A Node has attempted to connect, or reconnect, to
             a Wi-Fi access point, but is unable to successfully
             associate or authenticate, after exhausting all
             internal retries of its supplicant.";
        }
        leaf connection-status {
          type enumeration {
            enum connected {
              value 1;
            }
            enum notconnected {
              value 2;
            }
          }
          description
```

```
              "A Node's connection status to a Wi-Fi network has
               changed. Connected, in this context, SHALL mean that
               a Node acting as a Wi-Fi station is successfully
               associated to a Wi-Fi Access Point.";
          }
        }
      }
    }
```

Appendix B. Adaptive Subscription and Notification Example

The examples within this document use the normative YANG module "ietf-adaptive-subscription" defined in Section 4 and the non-normative example YANG module "example-wifi-network-diagnostic" defined in Appendix A.1.

This section shows some typical adaptive subscription and notification message exchanges.

B.1. "edit-config" Example

The client configures adaptive subscription policy parameters on the server. The adaptive subscription configuration parameters require the server to support two update intervals (i.e., 5 seconds, 60 seconds) and report updates every 60 seconds if the rssi value is greater than or equal to -65dB; If the rssi value is less than -65dB, switch to 5 seconds period value to report updates.

```
 <rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
   <target>
    <running/>
   </target>
   <config
    xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <top
     xmlns="http://example.com/schema/1.2/config"
     xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
     <yp:datastore
      xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
              ds:running
     </yp:datastore>
     <yp:datastore-xpath-filter
      xmlns:wnd="https://example.com/sample-data/1.0">
             /wnd:example-wifi-network-diagnostic
     </yp:datastore-xpath-filter>
     <as:adaptive-subscriptions
      xmlns:as="urn:ietf:params:xml:ns:yang:ietf-adaptive-subscription">
      <as:adaptive-period>
       <as:xpath-external-eval>
        /wnd:server/wnd:rssi&lt;-65
       </as:xpath-external-eval>
       <as:period>5</as:period>
      </as:adaptive-period>
      <as:adaptive-period>
       <as:xpath-external-eval>
        /wnd:server/wnd:rssi&gt;=-65
       </as:xpath-external-eval>
       <as:period>60</as:period>
     </as:adaptive-period>
     </as:adaptive-subscriptions>
    </top>
   </config>
  </edit-config>
 </rpc>
```

B.2. Create Adaptive Subscription Example

The subscriber sends an "establish-subscription" RPC with the parameters listed in to request the creation of an adaptive subscription. The adaptive subscription configuration parameters require the server to report updates every 5 seconds if the rssi value is less than -65dB; If the rssi value is greater than or equal to -65dB, switch to 60 seconds period value. (Section 2)

```
<netconf:rpc message-id="101"
 xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
 <establish-subscription
  xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"
  xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <yp:datastore
   xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
          ds:running
  </yp:datastore>
  <yp:datastore-xpath-filter
   xmlns:wnd="https://example.com/sample-data/1.0">
       /wnd:example-wifi-network-diagnostic
  </yp:datastore-xpath-filter>
  <as:adaptive-subscriptions
   xmlns:as="urn:ietf:params:xml:ns:yang:ietf-adaptive-subscription">
   <as:adaptive-period>
    <as:xpath-external-eval>
     wnd:server/wnd:rssi&lt;-65
    </as:xpath-external-eval>
    <as:period>5</as:period>
   </as:adaptive-period>
   <as:adaptive-period>
    <as:xpath-external-eval>
     wnd:server/wnd:rssi&gt;=-65
    </as:xpath-external-eval>
    <as:period>60</as:period>
   </as:adaptive-period>
  </as:adaptive-subscriptions>
 </establish-subscription>
</netconf:rpc>
```

B.3. "xpath-evaluation-unsupported" error response example

If the subscriber has authorization to establish the subscription with a server, but the server had not been able to fully satisfy the request from the subscriber, the server should send an RPC error response.

For instance, if the XPATH 1.0 syntax against the targeted data object defined in "xpath-external-eval" is not supported by the server' s implementation, the server returns a reply indicating a failure. The following <rpc-reply> illustrates an example:

```
<?xml version="1.0" encoding="utf-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <rpc-error>
    <error-type>application</error-type>
    <error-tag>invalid-value</error-tag>
    <error-severity>error</error-severity>
    <error-app-tag>
      ietf-adaptive-subscription:xpath-evaluation-unsupported
    </error-app-tag>
    <error-path xmlns:wnd="https://example.com/sample-data/1.0">
      /wnd:server/wnd:rssi
    </error-path>
  </rpc-error>
</rpc-reply>
```

Since adaptive subscription allows a server to be configured with multiple different period intervals and corresponding XPath evaluation criteria to trigger update interval switch in the server, it may be possible for the server to return multiple <rpc-error> elements with "xpath-evaluation-unsupported" failure specified by different error paths. The subscriber can use this information in future attempts to establish a subscription.

B.4. "adaptive-period-update" notification example

Upon the server switches from the update interval 5 seconds to the new update interval 60 seconds, before sending event records to receivers, the "adaptive-period-update" notification should be generated and sent to the receivers to inform the receivers that the update interval value is switched to the new value.

```
<notification
 xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
 xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
 <eventTime>2016-11-21T13:51:00Z</eventTime>
 <adaptive-period-update
  xmlns="http://example.com/ietf-adaptive-subscription">
  <id>0</id>
  <period>60</period>
  <yp:datastore
   xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
       ds:running
  </yp:datastore>
  <yp:datastore-xpath-filter
   xmlns:ex="https://example.com/sample-data/1.0">
       /ex:example-wifi-network-diagnostic
  </yp:datastore-xpath-filter>
 </adaptive-period-update>
</notification>
```

B.5. Changes between Revisions

v09 -v10

* Change the draft intended status to "experimental"

* Problem statement refinement

v08 -v09

* Define two new RPC errors to report when adaptive subscription unsupported or multiple XPath criteria conflict.

* Remove the "watermark" parameter.

* Add clarification about how to evaluate the XPath expression defined in "xpath-external-eval".

* Add clarification about how to compare a targeted data object in a specific list entry.

v07 -v08

* Define a new RPC error to report when an XPath syntax defined in "xpath-external-eval" is unsupported by a server.

* Add a new example showing how the RPC error being returned by a publisher.

* The usage examples fixed in the Appendix.

* Grammatical errors correction(missing articles, plurality mismatches, etc).

v06 -v07

* The usage examples typo fixed in the Appendix.

* Add reference to RFC7950 XPATH Evaluation section and XPATH 1.0

* Clarify the definitions of 'xpath-external-eval' and 'selection-filter' by reusing XPATH Evaluation rules in RFC7950.

* Add a new terminology "adaptive subscription".

* Add one section to discuss Arbitrary XPath Complexity.

v05 -v06

* Replace example-wifi-mac module with example-wifi-network-diagnostic using WIFI statistics specified in CHIP specification.

* Update adaptive subscription Example to align with WIFI example module change.

* Add one more reference to CHIP Specification.

v04 -v05

* Remove "modify-subscption" RPC usage.

* Module update to fix the nits.

* Update adaptive subscription Example.

* Other Editorial changes.

v03 - v04

* Add missing subtrees and data nodes in the security section;

* Change "adaptive-update" notification into "adaptive-period-update" notification;

* Other Editorial changes.

v02 - v03

* Clarify the difference between low priority telemetry data dropping and collection rate switching in the introduction section;

* Update the abstract and introduction section to focus on collection rate switching in the server without interaction with the remote client;

* Format usage example and change ssid into rssi in the appendix;

* Use boilerplate and reuse the terms in the terminology section.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China
Email: bill.wu@huawei.com

Wei Song
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China
Email: songwei80@huawei.com

Peng Liu
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing
Email: liupengyjy@chinamobile.com

Qiufang Ma
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China
Email: maqiufang1@huawei.com

Wei Wang
China Telecom
32 Xuanwumen West St, Xicheng District
Beijing
Email: wangw36@chinatelecom.cn

Zhixiong Niu
Microsoft
Email: Zhixiong.Niu@microsoft.com