

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.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	4
2. Model Overview	5
2.1. Subscription Configuration	6
2.2. YANG RPC	7
2.2.1. "establish-subscription" RPC	7
2.3. Notifications for Adaptive Subscribed Content	8
3. XPath Complexity Evaluation	9
4. Adaptive Subscription YANG Module	10
5. IANA Considerations	14
5.1. Updates to the IETF XML Registry	14
5.2. Updates to the YANG Module Names Registry	15
6. Security Considerations	15
7. Contributors	16
8. Acknowledges	16
9. References	16
9.1. Normative References	16
9.2. Informative References	17
Appendix A. Example YANG Module	18
A.1. "example-wifi-mac" YANG Module	18
Appendix B. Adaptive Subscription and Notification Example	23
B.1. "edit-config" Example	23
B.2. Create Adaptive Subscription Example	24
B.3. "xpath-evaluation-unsupported" error response example	25
B.4. "adaptive-period-update" notification example	26
B.5. Changes between Revisions	27
Authors' Addresses	29

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 0 indicates 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<-65
          </as:xpath-external-eval>
          <as:period>5</as:period>
        </as:adaptive-period>
        <as:adaptive-period>
          <as:xpath-external-eval>
            /wnd:server/wnd:rssi>=-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<-65
        </as:xpath-external-eval>
        <as:period>5</as:period>
      </as:adaptive-period>
      <as:adaptive-period>
        <as:xpath-external-eval>
          wnd:server/wnd:rssi>=-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