

NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 17, 2021

Q. Wu
W. Song
Huawei
L. Geng
P. Liu
China Mobile
Q. Ma
Huawei
October 14, 2020

Adaptive Subscription to YANG Notification
draft-wang-netconf-adaptive-subscription-02

Abstract

This document defines a YANG data model and associated mechanism enabling subscriber's adaptive subscriptions to a publisher's event streams with various different period intervals to report updates. Applying these elements allows both subscriber and publisher to automatically adjust the volume of telemetry traffic sent from publisher to the 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 April 17, 2021.

Copyright Notice

Copyright (c) 2020 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Terminology	3
2.	Model Overview	4
2.1.	Subscription Configuration	5
2.2.	YANG RPC	6
2.2.1.	"establish-subscription" RPC	6
2.2.2.	"modify-subscription" RPC	6
2.3.	Notifications for Adaptive Subscribed Content	6
3.	Adaptive Subscription YANG Module	7
4.	IANA Considerations	12
4.1.	Updates to the IETF XML Registry	12
4.2.	Updates to the YANG Module Names Registry	12
5.	Security Considerations	12
6.	Contributors	13
7.	References	13
7.1.	Normative References	13
7.2.	Informative References	14
Appendix A.	Example YANG Module	14
A.1.	"example-wifi-mac" YANG Module	15
Appendix B.	Adaptive Subscription and Notification Example	18
B.1.	"edit-config" Example	18
B.2.	Create Adaptive Subscription Example	19
B.3.	"adaptive-update" notification example	21
	Authors' Addresses	22

[1.](#) Introduction

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

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

- o On-change subscription allows update to be triggered whenever a change in the subscribed information is detected. The periodic interval is set to zero value in the on-change subscription case.

However in some large scale deployments (e.g., wireless network performance monitoring) where an increased data collection rate is being used, it becomes more likely that a burst of streamed data may temporarily overwhelm a receiver and consume expensive network resource (e.g., air interface resource). If the rate at which we can collect a stream of data is set too low, these telemetry data are not sufficient to detect and diagnose problems and verify correct network behavior. There is a need for a service to configure both collectors and publishers with multiple different period intervals and automatically switch to different period intervals according to resource usage change, e.g., when the wireless signal strength falls below a configured low watermark, the subscribed data can be streamed at a higher rate while when the wireless signal strength crosses a configured high watermark, the subscribed data can be streamed at lower rate.

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

This document uses the following terms:

Event: An event is something that happens that may be of interest - a configuration change, a fault, a change in status, crossing a threshold, or an external input to the system, for example. Often, this results in an asynchronous message, sometimes referred to as a notification or event notification, being sent to interested parties to notify them that this event has occurred [[RFC5277](#)].

Client: Defined in [[RFC8342](#)].

Configuration: Defined in [[RFC8342](#)].

Configured subscription: Defined in [[RFC8639](#)]

Configuration datastore: Defined in [[RFC8342](#)].

Notification message: Information intended for a receiver indicating that one or more events have occurred [[RFC8639](#)].

Publisher: An entity responsible for streaming notification messages per the terms of a subscription [[RFC8639](#)].

Receiver: A target to which a publisher pushes subscribed event records. For dynamic subscriptions, the receiver and subscriber are the same entity [[RFC8639](#)].

Subscriber: A client able to request and negotiate a contract for the generation and push of event records from a publisher. For dynamic subscriptions, the receiver and subscriber are the same entity [[RFC8639](#)].

Subscription: A contract with a publisher, stipulating the information that one or more receivers wish to have pushed from the publisher without the need for further solicitation [[RFC8639](#)].

On-change subscription: A datastore subscription with updates that are triggered when changes in subscribed datastore nodes are detected.

Periodic subscription: A datastore subscription with updates that are triggered periodically according to some time interval.

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 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-update" notification to report update interval change.

The following tree diagrams [[RFC8340](#)] provide an overview of the data model for "ietf-adaptive-subscription.yang" 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 watermark? uint32
          +--rw period 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 watermark? uint32
          +--rw period centiseconds
          +--rw anchor-time? yang:date-and-time
notifications:
  +---n adaptive-period-update
    +--ro id? sn:subscription-id
    +--ro period 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 adaptive periodic parameters:

- o a "period" that defines the new duration between push updates, the period can be changed based on trigger condition.
- o 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.

- o a "watermark" that defines the threshold value of the targeted data object, e.g., it can be lower boundary or upper boundary of targeted data object.
- o a "xpath-external-eval" represents an Evaluation criteria that may be applied against event records in an event stream, which is used to trigger update interval switch. It contains comparisons of datastore node with specific threshold (i.e., watermark) and associated logical operations in the XPath format. Different from stream-xpath-filter defined in [\[RFC8639\]](#), it doesn't influence the event records output generation from a publisher.

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 adaptive subscriptions. These parameters are same as one defined in [Section 2.1](#).

2.2.2. "modify-subscription" RPC

The subscriber MAY invoke the "modify-subscription" RPC for a subscription it previously established. The subscriber will include newly desired values in the "modify-subscription" RPC. Parameters not included MUST remain unmodified. [Section 4.4.2 of \[RFC8641\]](#) provides an example where a subscriber attempts to modify the period and datastore XPath filter of a subscription using NETCONF. The period can be the 'period' parameter defined by ietf-adaptive-subscription.

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 'adpative-update' notification include:

- o a "period" that defines the duration between push updates, the period can be changed based on trigger condition.
- o 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.
- o A selection filter identifying 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. 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 and "watermark" parameter defined by update trigger.

3. 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 2019-12-15 {
  description
    "Initial revision";
  reference
    "RFCxxx Adaptive subscription to YANG notification.";
}

typedef centiseconds {
  type uint32;
  description
    "A period of time, measured in units of 0.01 seconds.";
}

typedef seconds {
  type uint32;
  description
    "A period of time, measured in units of 1 seconds.";
}

typedef operator {
  type enumeration {
    enum unequal {
      description
        "Indicates that the comparision type is unequal to.";
    }
    enum equal {
      description
        "Indicates that the comparision type is equal to.";
    }
    enum less {
      description
        "Indicates that the comparision type is less than.";
    }
    enum less-or-equal {
      description
        "Indicates that the comparision type is less than
        or equal to.";
    }
  }
}
```



```
enum greater {
  description
    "Indicates that the comparision type is greater than.";
}
enum greater-or-equal {
  description
    "Indicates that the comparision type is greater than
    or equal to.";
}
}
description
  "definition of the operator";
}

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 {
        description
          "Defines necessary conditions to switch update interval for
          sending an event record to the subscriber. The event record output
          generation will not be influeced these conditions.";
        key "name";
        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 watermark {
        type uint32;
      }
    }
  }
}
```



```
        description
            "The watermark for targeted data object. The high
            watermark, low watermark can be specified for the
            targeted data object.";
    }
    leaf period {
        type 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 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>

4. IANA Considerations

4.1. Updates to 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 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.  
-----
```

4.2. Updates to the YANG Module Names Registry

This document registers two YANG modules in the YANG Module Names registry [[RFC7950](#)]. . Following the format in [[RFC6020](#)], the following registration has been made:

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

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.

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:

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

6. Contributors

The authors would like to thank Michale Wang for his major contributions to the initial modeling and use cases.

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>>.
- [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>>.

[7.2.](#) Informative References

- [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.](#) Example YANG Module

The example YANG module used in this document represents a simple wifi mac interface.

YANG tree diagram for the "example-wifi-mac" module:


```
module: example-wifi-mac
  +--rw clients
    +--ro client* [mac]
      +--ro mac                yang:mac-address
      +--ro rssi?              int8
      +--ro snr?                uint8
      +--ro ss?                 uint8
      +--ro phy-rate?           uint16
      +--ro channel-support*    uint8
      +--ro neighbors
        | +--ro neighbor-bssid? yang:mac-address
        | +--ro neighbor-channel? uint8
        | +--ro neighbor-rssi?   int8
        | +--ro neighbor-antenna? uint8
        | +--ro channel-load-report? uint8
      +--ro ssid
        +--ro name?             string
        +--ro enabled?          boolean
        +--ro broadcast-filter? boolean
        +--ro multicast-filter? boolean
        +--ro ipv6-ndp-filter?  boolean
        +--ro ipv6-ndp-filter-timer? uint16
        +--ro station-isolation? boolean
```

[A.1.](#) "example-wifi-mac" YANG Module

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

  import ietf-yang-types {
    prefix yang;
  }

  container clients {
    description
      "Top-level container for clients operational state data.";
    list client {
      key "mac";
      config false;
      description
        "List of clients per BSS.";
      leaf mac {
        type yang:mac-address;
        description
          "MAC address of the client.";
      }
    }
  }
}
```



```
leaf rssi {
  type int8;
  description
    "The RSSI of this client in dBm. Expressed as negative
    number";
}
leaf snr {
  type uint8;
  description
    "The SNR of AP to Client, in dB.";
}
leaf ss {
  type uint8;
  description
    "Number of Spatial Streams supported by the client.";
}
leaf phy-rate {
  type uint16;
  description
    "Last used PHY rate of connected client.";
}
leaf-list channel-support {
  type uint8;
  description
    "List of supported channels.";
}
container neighbors {
  description
    "Container for Client beacon reports. Requires 802.11k
    enabled. See Sec. 5.2.7.1 of 802.11k-2008 Standard.";
  leaf neighbor-bssid {
    type yang:mac-address;
    description
      "The BSSID of this neighbor.";
  }
  leaf neighbor-channel {
    type uint8;
    description
      "The channel of this neighbor.";
  }
  leaf neighbor-rssi {
    type int8;
    description
      "The RSSI of this neighbor in dBm, expressed as a negative
      number.";
  }
  leaf neighbor-antenna {
    type uint8;
```



```
    description
      "Antenna details for this neighbor.";
  }
  leaf channel-load-report {
    type uint8;
    description
      "Channel load, as reported by Client to AP
      normalized to 255. See Sec. 10.11.9.3 of 802.11ac-2013
      Spec.";
  }
}
container ssid {
  description
    "Top level container for ssids, including configuration
    and state data.";
  leaf name {
    type string;
    description
      "The name of the SSID.";
  }
  leaf enabled {
    type boolean;
    default "true";
    description
      "The desired operational state (up/down) of this SSID.";
  }
  leaf broadcast-filter {
    type boolean;
    description
      "Convert all downstream broadcast ARP to unicast
      only if Station is associated to the AP. Drop packet
      if Station is not associated to the AP. All other
      broadcast, except DHCP, is dropped by the AP.

      DHCP Offers/ACKs are converted to Unicast, over-the-air.";
  }
  leaf multicast-filter {
    type boolean;
    description
      "Drop all downstream Multicast packets.";
  }
  leaf ipv6-ndp-filter {
    type boolean;
    description
      "Neighbor Advertisements will be cached at the AP (or WLC)
      and unicast in response to Neighbor Solicitations.

      Router Advertisements, in response to a Router Solicitation
```



```
        are converted to Unicast for over-the-air transmission.";
    }
    leaf ipv6-ndp-filter-timer {
        type uint16;
        units "seconds";
        description
            "Time, in seconds, the ndp-filter will cache
             Neighbor Advertisements (NA).";
    }
    leaf station-isolation {
        type boolean;
        description
            "Block Station peer to peer communication.";
    }
}
}
```

[Appendix B.](#) Adaptive Subscription and Notification Example

The examples within this document use the normative YANG module "ietf-adaptive-subscription" as defined in [Section 3](#) and the non-normative example YANG module "example-wifi-mac" as defined in [Appendix A.1](#).

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

[B.1.](#) "edit-config" Example

The client configure adaptive subscription parameters on the server. The adaptive subscription configuration parameters require the server to scan all clients every 5 seconds if the ssid value of client is greater than -65dB; If the ssid value of client is less than -65dB, switch to 60 seconds period value, and then scan all clients every 60 seconds.


```
<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">
        <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-mac
        </yp:datastore-xpath-filter>
        <as:adaptive-subscriptions
          xmlns:as="urn:ietf:params:xml:ns:yang:ietf-adaptive-subscription">
          <as:adaptive-period>
            <as:condition-expression>as:clients/as:client[ssid > -65]</as:
              condition-expression>
            <as:watermark>-65</as:watermark>
            <as:period>5</as:period>
          </as:adaptive-period>
          <as:adaptive-period>
            <as:condition-expression>as:clients/as:client[ssid < -65]</as:
              condition-expressioni>
            <as:watermark>-65</as:watermark>
            <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 a adaptive subscription. The adaptive subscription configuration parameters require the server to scan all clients every 5 seconds if the ssid value of client is greater than -65dB; If the ssid value of client is less than -65dB, switch to 60 seconds period value, and then scan all clients every 60 seconds. ([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:ex="https://example.com/sample-data/1.0">
      /ex:example-wifi-mac
    </yp:datastore-xpath-filter>
    <as:adaptive-subscriptions
      xmlns="urn:ietf:params:xml:ns:yang:ietf-adaptive-subscription">
      <as:adaptive-period>
        <as:condition-expression>as:clients/as:client[ssid > -65]
        </as:condition-expressioni>
        <as:watermark>-65</as:watermark>
        <as:period>5</as:period>
      </as:adaptive-period>
      <as:adaptive-period>
        <as:condition-expression>as:clients/as:client[ssid < -65]
        </as:condition-expressioni>
        <as:watermark>-65</as:watermark>
        <as:period>60</as:period>
      </as:adaptive-period>
    </as:adaptive-subscriptions>
  </establish-subscription>
</netconf:rpc>
```

In another example, the adaptive subscription configuration parameters could also require the server to scan all clients every 5 seconds if the difference between maximum value of client ssid and minimum value of client ssid is greater than 0.20dB; If the difference between maximum value of client ssid and minimum value of client ssid is less than 20dB, switch to 60 seconds period value and then scan all clients every 60 seconds.


```
<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:ex="https://example.com/sample-data/1.0">
      /ex:example-wifi-mac
    </yp:datastore-xpath-filter>
    <as:adaptive-subscriptions>
      <as:data-path>as:clients/as:client</as:data-path>
      <as:target>ssid</as:target>
      <as:adaptive-period>
        <as:condition-expression>as:clients/as:client[max(ssid)-min(ssid) >20]
        </as:condition-expressioni>
        <as:watermark>20</as:watermark>
        <as:period>5</as:period>
      </as:adaptive-period>
      <as:adaptive-period>
        <as:condition-expression>as:clients/as:client[max(ssid)-min(ssid) < 20]
        </as:condition-expressioni>
        <as:watermark>20</as:watermark>
        <as:period>60</as:period>
      </as:adaptive-period>
    </as:adaptive-subscriptions>
  </establish-subscription>
</netconf:rpc>
```

B.3. "adaptive-update" notification example

Upon the server switches to from the update interval 5 seconds to the new update interval 60 seconds, Before sending event records to receivers, the "adaptive-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">
  <eventTime>2016-11-21T13:51:00Z</eventTime>
  <adaptive-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-mac
    </yp:datastore-xpath-filter>
  </adaptive-update>
</notification>
```

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

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

Email: gengliang@chinamobile.com

Peng Liu
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing 10053

Email: liupengyjy@chinamobile.com

Qiufang Ma
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: maqiufang1@huawei.com

