Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

A. Clemm
Huawei - Futurewei Technologies, Inc.
E. Voit
Cisco Systems
X. Liu
Volta Networks
I. Bryskin
T. Zhou
G. Zheng
Huawei
H. Birkholz
Fraunhofer SIT

October 22, 2018

Smart Filters for Push Updates
draft-clemm-netmod-push-smart-filters-01

Abstract

This document defines a YANG model for Smart Filters for push updates. Smart Filters allow to filter push updates based on values of pushed datastore nodes and/or state, such as previous updates. Smart Filters provide an important building block for service assurance and network automation.

This revision of the document is intended as a placeholder, containing the problem statement of draft-clemm-netconf-push-smart-filters-ps-00 that has recently expired. The YANG model itself still needs to be defined.

Table of Contents

1. Introduction

YANG-Push [yang-push] allows client applications to subscribe to continuous datastore updates without needing to poll. YANG-Push subscriptions allow client applications to select which datastore nodes are of interest. For this purpose, filters that act as node selectors are offered. However, what is currently not supported are filters that filter updates based on values, such as sending updates only when the value falls within a certain range. Also not supported are filters that would require additional state, such as sending updates only when the value exceeds a certain threshold for the first time but not again until the threshold is cleared. We refer to such filters as "Smart Filters", with further subcategories of "smart stateless filters" and "smart stateful filters", respectively.

Smart Filters involve more complex subscription and implementation semantics than the simple selection filters that are currently offered as part of YANG-Push. They involve post processing of updates that goes beyond basic update generation for polling avoidance and place additional intelligence at the server. Because of this, Smart Filter functionality was not included in the YANG-Push

specification, although it was recognized that YANG-Push could be extended to include such functionality if needed. This is the purpose of this specification.

Smart Filters facilitate service assurance, because they allow client applications to focus on "outliers" and updates that signify exceptions and conditions of interest have the biggest operational significance. They save network resources by avoiding the need to stream updates that would be discarded anyway, and allow applications to scale better since larger networks imply a larger amount of Smart Filtering operations delegated away from the application to the network. Smart Filters also facilitate network automation as they constitute an important ingredient to specify triggers for automated actions.

2. Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Definitions and Acronyms

Datastore node: An instance of management information in a datastore. Also known as "object".

Smart Filter: A filter that involves some processing, such as comparing values or differentiating behavior depending on state.

TCA: Threshold Crossing Alert.

YANG-Push: A server capability that allows client applications to subscribe to network management datastore updates.

4. Problem Statement

YANG-Push provides client applications with the ability to subscribe to continuous updates from network management datastores, obviating the need to perform polling and resulting in more robust and efficient applications. However, many applications do not require every update, only updates that are of certain interest.

For example, an update concerning interface utilization may be only needed when a certain utilization level is breached. Sending continuous updates when utilization is low might divert processing resources away from updates regarding interfaces whose utilization

level may reach a critical point that requires attention. Doing so will require a filter based on an object value. Even sending continuous updates when utilization is high may be too much and counterproductive. It may be sufficient to send an update when a threshold is breached to raise a flag of attention, but then not to continue sending updates while the condition still persists but simply let the client application know when the threshold is cleared. This behavior cannot be accomplished simply by a value-based filter, but requires additional state to be maintained (so that the server has a memory whether or not the condition of a breached threshold has already been reported in prior update cycles).

What is needed are "Smart Filters" that provide the ability to apply filters based on object values, possibly also state state. Smart Filters are useful for Service Assurance applications that need to monitor operational data for values that fall outside normal operational ranges. They are also useful for network automation, in which automated actions are automatically triggered based on when certain events in the network occur while certain conditions hold. A YANG-Push subscription with a Smart Filter can in effect act as a source for such events. Combined with an optional check for a condition when an event is observed, this can serve as the basis of action triggers.

Smart Filters for Push Updates will provide support for the following features:

- Support for Smart Filter extensions to YANG-Push subscriptions. The targeted model takes a "base" YANG-Push subscription and subjects updates to an additional filtering stage that is based on value.

- Support for selected stateful filters:

  * This includes specifically support for generalized "threshold crossing alert" filters, or filters that provide an update only when a datastore node's value passes a filter for the first time, and not again until the datastore node's value passes a counter filter. In effect, the support involves attaching filter and counter filter to a datastore node, including a switch at the datastore node indicating which filter is in effect, and providing a distinction in the update which filter (e.g. onset of clear) was applied.

  * It may include additional filters, such a "recent high water mark" filters that allow to specify a time horizon until the current high water mark clears. A recent high water mark

filter sends an update to an object only if its new value is greater than the last value that had been previously reported.

- In addition to new filters, support for features to make them easier to use:
  - Support for refined on-change update semantics that allow client to distinguish whether datastore node values were omitted or included because the datastore node was created or deleted, or because the datastore node's value fell outside filter range.
  - Support for a heartbeat that indicates that a filter is still in effect after a longer period of inactivity.

It is easy to conceive of filters that are very smart and powerful yet also very complex. While filters as defined in YANG-Push may be a tad too simple for the applications envisioned here, it is important to keep filters still simple enough to ensure broad implementation and support by networking devices. The purpose of Smart Filters defined in this effort is to address the 90% of cases that can be addressed using 10% of the complexity. Items like the following will therefore be outside the scope:

- Filters that involve freely programmable logic.
- Filters that aggregate or otherwise process information over time. An example would be filters that compute an aggregate over a time series of data (e.g. a datastore node's average or top percentile value)
- Filters that aggregate or compare values of several datastore nodes (e.g. the maximum or average from datastore nodes in a list).

5. Smart Filter Data Model

The following section contains an initial YANG data model for smart filters. The model is at this point still incomplete and included as a starting point only. At this point, the model defines a simple threshold filter. When used with a subscription, objects that meet the filter criterion (i.e. the threshold comparison) are included in the update whereas any other object is filtered.

The model will be extended to define a full "smart threshold" model in a later revision. This will add the feature of a hysteresis threshold, i.e. a counter threshold that allows to define when a crossed threshold should be cleared. The value of the hysteresis

threshold can be set to a lower value than the threshold itself to avoid unnecessary updates in case of oscillations). It will also add a notion of state to remember whether a threshold crossing has already been reported, to avoid repeated inclusion of objects in updates that remain above their threshold. By including metadata, clients will be able to distinguish between the violation and the clearing of thresholds.

The model will furthermore be extended for smart filters that are not threshold-related, such as the previously mentioned recent high water marks.

```
<CODE BEGINS> file "ietf-smart-filter@2018-10-22.yang"
module ietf-smart-filter {
  yang-version "1.1";
  namespace "urn:ietf:params:xml:ns:yang:ietf-smart-filter";
  prefix "sf";

  import ietf-yang-types {
    prefix yang;
        reference
          "RFC 6991: Common YANG Data Types";
  }

  import ietf-subscribed-notifications {

    prefix sn;
    reference
      "draft-ietf-netconf-subscribed-notifications:
       Customized Subscriptions to a Publisher's Event Streams

       NOTE TO RFC Editor: Please replace above reference to
       draft-ietf-netconf-subscribed-notifications with RFC number
       when published (i.e. RFC xxxx).";
  }

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

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

     Editor:   Eric Voit
               <mailto:evoit@cisco.com>

     Editor:   Xufeng Liu
```

```
                <mailto:xufeng.liu.ietf@gmail.com>

      Editor:   Igor Bryskin
                <mailto:igor.bryskin@huawei.com>

      Editor:   Tianran Zhou
                <mailto:zhoutianran@huawei.com>

      Editor:   Guangying Zheng
                <mailto:zhengguangying@huawei.com>

      Editor:   Henk Birkholz
                <mailto:henk.birkholz@sit.fraunhofer.de>";

   description
     "This module contains YANG specifications for smart filter.

     Copyright (c) 2018 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
     (https://trustee.ietf.org/license-info).

         This version of this YANG module is part of
         draft-clemm-netmod-push-smart-filters-01; see the RFC itself
                 for full legal notices.

         NOTE TO RFC EDITOR: Please replace above reference to
         with RFC number when
         published (i.e. RFC xxxx).";

   revision 2018-10-22 {
     description
       "Initial revision.
       NOTE TO RFC EDITOR:
       (1)Please replace the above revision date to
       the date of RFC publication when published.
       (2) Please replace the date in the file name
       (ietf-smart-filter@2018-10-22.yang) to the date of RFC
       publication.
       (3) Please replace the following reference to
       draft-clemm-netmod-push-smart-filters-01 with RFC number when
       published (i.e. RFC xxxx).";
     reference
       "draft-clemm-netmod-push-smart-filters-01";
```

```
   }

  /*
   * IDENTITIES
   */

   /* Smart-filter type identities  */

   identity smart-filter {
      description
       "A base identity that represents the smart filter types. ";
   }

   identity smart-filter-threshold {
      base smart-filter;
      description
       "An identity instance based on smart-filter, which support
        filter the push data by fix threshold value.";
   }

  /*
   * TYPE DEFINITIONS
   */
   typedef sf-op-type {
     type enumeration {
       enum eq {
         description "equal to";
       }
       enum gt {
         description "greater than";
       }
       enum ge {
         description "greater than or equal to";
       }
       enum lt {
         description "less than";
       }
       enum le {
         description "less than or equal to";
       }
     }
     description "A boolean comparator for an object and a data value.
              Include: eq, gt, ge, lt, le.";
   }

  /*
   * GROUP DEFINITIONS
   */
```

```
   grouping sf-threshold{
     description
       "the group for threshold filter";
     leaf filter-node {
       if-feature "sn:xpath";
       type yang:xpath1.0;
       description
         "This parameter contains an XPath expression identifying
          the node of the target filter.";
       }

     leaf threshold-value {
       type string;
       description "threshold value";
     }

     leaf op-type {
       type sf-op-type;
       description "comparison operator";
     }
   }

   //augment statements
   augment "/sn:subscriptions/sn:subscription" {
     description "add the smart filter container";
     container smart-filter {
       description "It concludes filter configurations";

       choice filter-type {
         description
           "Select different smart filter";
         case threshold-filter {
           description
             "threshold-filter";
           uses sf-threshold;
         }
       }
     }
   }
 }
 <CODE ENDS>
```

6. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

IANA is requested to assign a new URI from the IETF XML Registry [RFC3688]. The following URI is suggested:

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

This document also requests a new YANG module name in the YANG Module Names registry [RFC7950] with the following suggestion:

```
name: ietf-ioam
namespace: urn:ietf:params:xml:ns:yang:ietf-smart-filter
prefix: sf
reference: RFC XXXX
```

7. Security Considerations

The application of Smart Filters requires a certain amount of processing resources at the server. An attacker could attempt to attack a server by creating YANG-push subscriptions with a large number of complex Smart Filters in an attempt to diminish server resources. Server implementations can guard against such scenarios in several ways. For one, they can implement NACM in order to require proper authorization for requests to be made. Second, server implementations can reject requests made for a a larger number of Smart Filters than the implementation can reasonably sustain.

8. Normative References

[notif-sub] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Custom Subscriptions to a Publisher's Event Streams", June 2018, <https://datatracker.ietf.org/doc/draft-ietf-netconf-subscribed-notifications/>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

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

[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>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[yang-push] Clemm, A., Voit, E., Gonzalez Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", July 2018, <https://datatracker.ietf.org/doc/draft-ietf-netconf-yang-push/>.

Authors' Addresses

Alexander Clemm
Huawei - Futurewei Technologies, Inc.
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: ludwig@clemm.org

Eric Voit
Cisco Systems

Email: evoit@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Igor Bryskin
Huawei

Email: igor.bryskin@huawei.com

Tianran Zhou
Huawei

Email: zhoutianran@huawei.com

Guangying Zheng
Huawei

Email: zhengguangying@huawei.com

Henk Birkholz
Fraunhofer SIT

Email: henk.birkholz@sit.fraunhofer.de