ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 8650
Category: Standards Track
ISSN: 2070-1721

E. Voit
R. Rahman
E. Nilsen-Nygaard
Cisco Systems
A. Clemm
Futurewei
A. Bierman
YumaWorks

November 2019

Dynamic Subscription to YANG Events and Datastores over RESTCONF

Abstract

This document provides a RESTCONF binding to the dynamic subscription capability of both subscribed notifications and YANG-Push.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8650.

Table of Contents

1. Introduction

Mechanisms to support event subscription and YANG-Push are defined in [RFC8639]. Enhancements to [RFC8639] that enable YANG datastore subscription and YANG-Push are defined in [RFC8641]. This document provides a transport specification for dynamic subscriptions over RESTCONF [RFC8040]. Requirements for these mechanisms are captured in [RFC7923].

The streaming of notifications that encapsulate the resulting information push is done via the mechanism described in Section 6.3 of [RFC8040].

2. 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 use the definitions from [RFC8639]: dynamic subscription, event stream, notification message, publisher, receiver, subscriber, and subscription.

Other terms reused include datastore, which is defined in [RFC8342], and HTTP/2 stream, which maps to the definition of "stream" within [RFC7540], Section 2.

3. Dynamic Subscriptions

This section provides specifics on how to establish and maintain dynamic subscriptions over RESTCONF [RFC8040]. Subscribing to event streams is accomplished in this way via RPCs defined within [RFC8639], Section 2.4. The RPCs are done via RESTCONF POSTs. YANG datastore subscription is accomplished via augmentations to [RFC8639] as described within [RFC8641], Section 4.4.

As described in Section 6.3 of [RFC8040], a GET needs to be performed on a specific URI on the publisher. Subscribers cannot predetermine the URI against which a subscription might exist on a publisher, as the URI will only exist after the "establish-subscription" RPC has been accepted. Therefore, the POST for the "establish-subscription" RPC replaces the GET request for the "location" leaf that is used in [RFC8040] to obtain the URI. The subscription URI will be determined and sent as part of the response to the "establish-subscription" RPC, and a subsequent GET to this URI will be done in order to start the flow of notification messages back to the subscriber. As specified in Section 2.4.1 of [RFC8639], a subscription does not move to the active state until the GET is received.

3.1. Transport Connectivity

For a dynamic subscription, when a RESTCONF session doesn't already exist, a new RESTCONF session is initiated from the subscriber.

As stated in Section 2.1 of [RFC8040], a subscriber MUST establish the HTTP session over TLS [RFC8446] in order to secure the content in transit.

Without the involvement of additional protocols, HTTP sessions by themselves do not support quick recognition of the loss of the communication path to the publisher. Where quick recognition of the loss of a publisher is required, a subscriber SHOULD use a TLS heartbeat [RFC6520], just from subscriber to publisher, to track HTTP session continuity.

Loss of the heartbeat MUST result in the teardown of any subscription-related TCP sessions between those endpoints. A subscriber can then attempt to re-establish the dynamic subscription by using the procedure described in Section 3.4.

3.2. Discovery

Subscribers can learn which event streams a RESTCONF server supports by querying the "streams" container of ietf-subscribed-notifications.yang in [RFC8639]. Support for the "streams" container of ietf-restconf-monitoring.yang in [RFC8040] is not required. In the case when the RESTCONF binding specified by this document is used to convey the "streams" container from ietf-restconf-monitoring.yang (i.e., that feature is supported), any event streams contained therein are also expected to be present in the "streams" container of ietf-restconf-monitoring.yang.

Subscribers can learn which datastores a RESTCONF server supports by following Section 2 of [RFC8527].

3.3. RESTCONF RPCs and HTTP Status Codes

Specific HTTP response codes as defined in Section 6 of [RFC7231] will indicate the result of RESTCONF RPC requests with the publisher. An HTTP status code of 200 is the proper response to any successful RPC defined within [RFC8639] or [RFC8641].

If a publisher fails to serve the RPC request for one of the reasons indicated in Section 2.4.6 of [RFC8639] or Appendix A of [RFC8641], this will be indicated by an appropriate error code, as shown below, transported in the HTTP response.

When an HTTP error code is returned, the RPC reply MUST include an <rpc-error> element per Section 7.1 of [RFC8040] with the following parameter values:

* an "error-type" node of "application".

* an "error-tag" node whose value is a string that corresponds to an identity associated with the error. This "error-tag" will come from one of two places and will correspond to the error identities either within Section 2.4.6 of [RFC8639] for general subscription errors (Table 1) or within Appendix A.1 of [RFC8641] for subscription errors specific to YANG datastores (Table 2).

* an "error-app-tag" node whose value is a string that corresponds to an identity associated with the error, as defined in Section 2.4.6 of [RFC8639] for general subscriptions or Appendix A.1 of [RFC8641] for subscription errors specific to YANG datastores. The tag to use depends on the RPC for which the error occurred. Viable errors for different RPCs are found in Table 3.

Error identity	Uses "error-tag"	HTTP code
dscp-unavailable	invalid-value	400
encoding-unsupported	invalid-value	400
filter-unsupported	invalid-value	400
insufficient-resources	resource-denied	409

no-such-subscription	invalid-value	404
replay-unsupported	operation-not-supported	501

Table 1: General Subscription Error Identities and Associated "error-tag" Use

Error identity	Uses "error-tag"	HTTP code
cant-include	operation-not-supported	501
datastore-not-subscribable	invalid-value	400
no-such-subscription-resync	invalid-value	404
on-change-unsupported	operation-not-supported	501
on-change-sync-unsupported	operation-not-supported	501
period-unsupported	invalid-value	400
update-too-big	too-big	400
sync-too-big	too-big	400
unchanging-selection	operation-failed	500

Table 2: Datastore-Specific Error Identities and Associated "error-tag" Use

RPC	Select an identity with a base
establish-subscription	establish-subscription-error
modify-subscription	modify-subscription-error
delete-subscription	delete-subscription-error
kill-subscription	delete-subscription-error
resync-subscription	resync-subscription-error

Table 3: RPC Errors and Associated Error Identities

Each error identity will be inserted as the "error-app-tag" using JSON encoding following the form <modulename>:<identityname>. An example of such a valid encoding would be "ietf-subscribed-notifications:no-such-subscription".

In the case of error responses to an "establish-subscription" or "modify-subscription" request, there is the option to include an "error-info" node. This node may contain hints for parameter settings that might lead to successful RPC requests in the future. Tables 4 and 5 show the yang-data structures that may be returned.

Target:	Return hints in yang-data structure
event stream	establish-subscription-stream-error-info
datastore	establish-subscription-datastore-error-info

Table 4: Optional "error-info" Node Hints for an "establish-subscription" Request

Target:	Returns hints in yang-data structure
event stream	modify-subscription-stream-error-info
datastore	modify-subscription-datastore-error-info

Table 5: Optional "error-info" Node Hints for an "modify-subscription" Request

The yang-data included within "error-info" SHOULD NOT include the optional leaf "reason", as such a leaf would be redundant with information that is already placed within the "error-app-tag".

In case of an <rpc-error> as a result of a "delete-subscription", a "kill-subscription", or a "resync-subscription" request, no "error-info" needs to be included, as the "subscription-id" is the only RPC input parameter, and no hints regarding this RPC input parameters need to be provided.

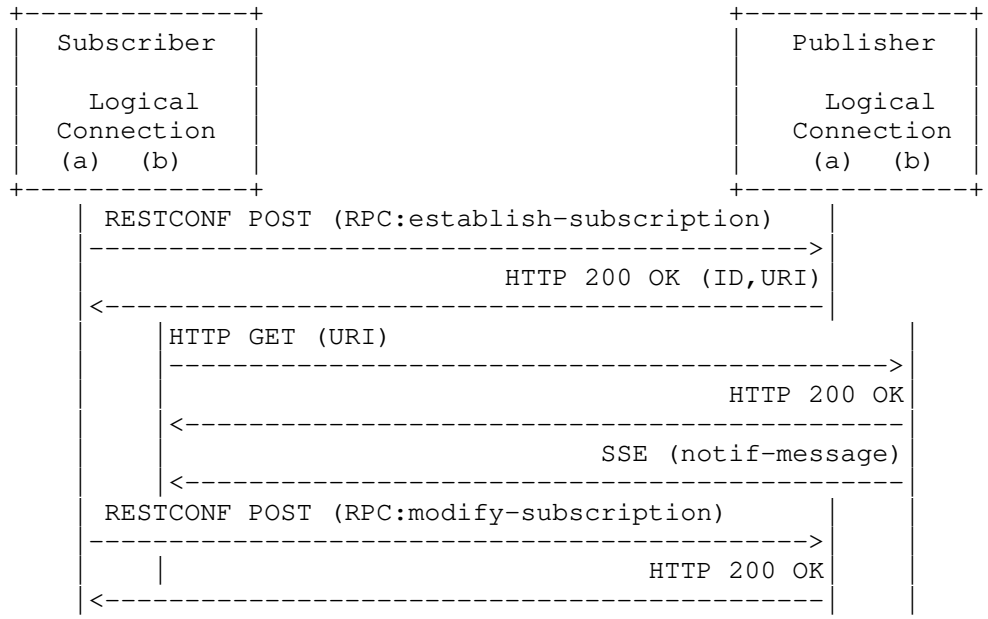
Note that "error-path" [RFC8040] does not need to be included with the <rpc-error> element, as subscription errors are generally associated with the choice of RPC input parameters.

3.4. Call Flow for Server-Sent Events

The call flow for Server-Sent Events (SSE) is defined in Figure 1. The logical connections denoted by (a) and (b) can be a TCP connection or an HTTP/2 stream (if HTTP/2 is used, multiple HTTP/2 streams can be carried in one TCP connection). Requests to RPCs as defined in [RFC8639] or [RFC8641] are sent on a connection indicated by (a). A successful "establish-subscription" will result in an RPC response returned with both a subscription identifier that uniquely identifies a subscription, as well as a URI that uniquely identifies the location of subscription on the publisher (b). This URI is defined via the "uri" leaf in the data model in Section 7.

An HTTP GET is then sent on a separate logical connection (b) to the URI on the publisher. This signals the publisher to initiate the flow of notification messages that are sent in SSE [W3C-20150203] as a response to the GET. There cannot be two or more simultaneous GET requests on a subscription URI: any GET request received while there is a current GET request on the same URI MUST be rejected with HTTP error code 409.

As described in Section 6.4 of [RFC8040], RESTCONF servers SHOULD NOT send the "event" or "id" fields in the SSE event notifications.



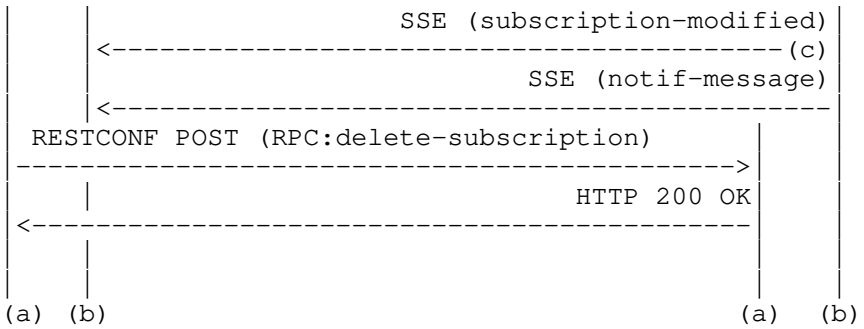


Figure 1: Dynamic Subscriptions with Server-Sent Events

Additional requirements for dynamic subscriptions over SSE include:

* A publisher MUST return all subscription state notifications in a separate SSE message used by the subscription to which the state change refers.

* Subscription RPCs MUST NOT use the connection currently providing notification messages for that subscription.

* In addition to an RPC response for a "modify-subscription" RPC traveling over (a), a "subscription-modified" state change notification MUST be sent within (b). This allows the receiver to know exactly when, within the stream of events, the new terms of the subscription have been applied to the notification messages. See arrow (c).

* In addition to any required access permissions (e.g., Network Configuration Access Control Model (NACM)), the RPCs "modify-subscription", "resync-subscription", and "delete-subscription" SHOULD only be allowed by the same RESTCONF username [RFC8040] that invoked "establish-subscription". Such a restriction generally serves to preserve users' privacy, but exceptions might be made for administrators that may need to modify or delete other users' subscriptions.

* The "kill-subscription" RPC can be invoked by any RESTCONF username with the required administrative permissions.

A publisher MUST terminate a subscription in the following cases:

* Receipt of a "delete-subscription" or a "kill-subscription" RPC for that subscription

* Loss of TLS heartbeat

A publisher MAY terminate a subscription at any time as stated in Section 1.3 of [RFC8639].

4. QoS Treatment

Qos treatment for event streams is described in Section 2.3 of [RFC8639]. In addition, if HTTP/2 is used, the publisher MUST:

* Take the "weighting" leaf node in [RFC8639] and copy it into the HTTP/2 stream weight, Section 5.3 of [RFC7540], and

* Take any existing subscription "dependency", as specified by the "dependency" leaf node in [RFC8639], and use the HTTP/2 stream for the parent subscription as the HTTP/2 stream dependency (as described in Section 5.3.1 of [RFC7540]) of the dependent subscription.

* Set the exclusive flag (Section 5.3.1 of [RFC7540]) to 0.

For dynamic subscriptions with the same Differentiated Services Code Point (DSCP) value to a specific publisher, it is recommended that the subscriber sends all URI GET requests on a common HTTP/2 session

(if HTTP/2 is used). Conversely, a subscriber cannot use a common HTTP/2 session for subscriptions with different DSCP values.

5. Notification Messages

Notification messages transported over RESTCONF will be encoded according to [RFC8040], Section 6.4.

6. YANG Tree

The YANG module defined in Section 7 has one leaf that augments three nodes of [RFC8639].

```
module: ietf-restconf-subscribed-notifications
  augment /sn:establish-subscription/sn:output:
    +--ro uri?   inet:uri
  augment /sn:subscriptions/sn:subscription:
    +--ro uri?   inet:uri
  augment /sn:subscription-modified:
    +--ro uri?   inet:uri
```

7. YANG Module

This module references [RFC8639].

```
<CODE BEGINS>
  file "ietf-restconf-subscribed-notifications@2019-11-17.yang"
module ietf-restconf-subscribed-notifications {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:"
          + "ietf-restconf-subscribed-notifications";
  prefix rsn;

  import ietf-subscribed-notifications {
    prefix sn;
  }
  import ietf-inet-types {
    prefix inet;
  }

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

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

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

     Editor:   Reshad Rahman
               <mailto:rrahman@cisco.com>";
  description
    "Defines RESTCONF as a supported transport for subscribed
     event notifications.

     Copyright (c) 2019 IETF Trust and the persons identified
     as authors of the code.  All rights reserved.

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

     This version of this YANG module is part of RFC 8650; see the
     RFC itself for full legal notices.";
```

```
  revision 2019-11-17 {
    description
      "Initial version";
    reference
      "RFC 8650: Dynamic Subscription to YANG Events and Datastores
       over RESTCONF";
  }

  grouping uri {
    description
      "Provides a reusable description of a URI.";
    leaf uri {
      type inet:uri;
      config false;
      description
        "Location of a subscription-specific URI on the publisher.";
    }
  }

  augment "/sn:establish-subscription/sn:output" {
    description
      "This augmentation allows RESTCONF-specific parameters for a
       response to a publisher's subscription request.";
    uses uri;
  }

  augment "/sn:subscriptions/sn:subscription" {
    description
      "This augmentation allows RESTCONF-specific parameters to be
       exposed for a subscription.";
    uses uri;
  }

  augment "/sn:subscription-modified" {
    description
      "This augmentation allows RESTCONF-specific parameters to be
       included as part of the notification that a subscription has
       been modified.";
    uses uri;
  }
}
<CODE ENDS>
```

8. IANA Considerations

This document registers the following namespace URI in the "ns" subregistry of the "IETF XML Registry" [RFC3688]:

URI:
urn:ietf:params:xml:ns:yang:ietf-restconf-subscribed-notifications

Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

This document registers the following YANG module in the "YANG Module Names" registry [RFC6020]:

Name: ietf-restconf-subscribed-notifications

Namespace:
urn:ietf:params:xml:ns:yang:ietf-restconf-subscribed-notifications

Prefix: rsn

Reference: RFC 8650

9. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management transports

such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

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

The one new data node introduced in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to this data node. These are the subtrees and data nodes and their sensitivity/vulnerability:

Container: "/subscriptions"

* "uri": leaf will show where subscribed resources might be located on a publisher. Access control must be set so that only someone with proper access permissions, i.e., the same RESTCONF [RFC8040] user credentials that invoked the corresponding "establish-subscription", has the ability to access this resource.

The subscription URI is implementation specific and is encrypted via the use of TLS. Therefore, even if an attacker succeeds in guessing the subscription URI, a RESTCONF username [RFC8040] with the required administrative permissions must be used to be able to access or modify that subscription. It is recommended that the subscription URI values not be easily predictable.

The access permission considerations for the RPCs "modify-subscription", "resync-subscription", "delete-subscription", and "kill-subscription" are described in Section 3.4.

If a buggy or compromised RESTCONF subscriber sends a number of "establish-subscription" requests, then these subscriptions accumulate and may use up system resources. In such a situation, the publisher MAY also suspend or terminate a subset of the active subscriptions from that RESTCONF subscriber in order to reclaim resources and preserve normal operation for the other subscriptions.

10. References

10.1. 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>.

[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>.

[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>.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport

Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, DOI 10.17487/RFC6520, February 2012, <https://www.rfc-editor.org/info/rfc6520>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540>.

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

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

[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>.

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

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

[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>.

[W3C-20150203] Hickson, I., "Server-Sent Events", W3C Recommendation, 3 February 2015, <https://www.w3.org/TR/2015/REC-eventsource-20150203/>. Latest version available at <https://www.w3.org/TR/eventsource/>.

10.2. Informative References

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <https://www.rfc-editor.org/info/rfc7923>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[RFC8347] Liu, X., Ed., Kyparlis, A., Parikh, R., Lindem, A., and M. Zhang, "A YANG Data Model for the Virtual Router Redundancy Protocol (VRRP)", RFC 8347, DOI 10.17487/RFC8347, March 2018, <https://www.rfc-editor.org/info/rfc8347>.

[RFC8527] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K.,

and R. Wilton, "RESTCONF Extensions to Support the Network Management Datastore Architecture", RFC 8527, DOI 10.17487/RFC8527, March 2019, <https://www.rfc-editor.org/info/rfc8527>.

[RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", RFC 8640, DOI 10.17487/RFC8640, September 2019, <https://www.rfc-editor.org/info/rfc8640>.

[XPATH] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", W3C Recommendation, 16 November 1999, <http://www.w3.org/TR/1999/REC-xpath-19991116>. Latest version available at <https://www.w3.org/TR/ xpath/>.

Appendix A. Examples

This section is non-normative. To allow easy comparison, this section mirrors the functional examples shown with NETCONF over XML within [RFC8640]. In addition, HTTP/2 vs HTTP/1.1 headers are not shown as the contents of the JSON encoded objects are identical within.

The subscription URI values used in the examples in this section are purely illustrative, and are not indicative of the expected usage that is described in Section 9.

The DSCP values are only for example purposes and are all indicated in decimal since the encoding is JSON [RFC7951].

A.1. Dynamic Subscriptions

A.1.1. Establishing Dynamic Subscriptions

The following figure shows two successful "establish-subscription" RPC requests as per [RFC8639]. The first request is given a subscription identifier of 22, and the second, an identifier of 23.

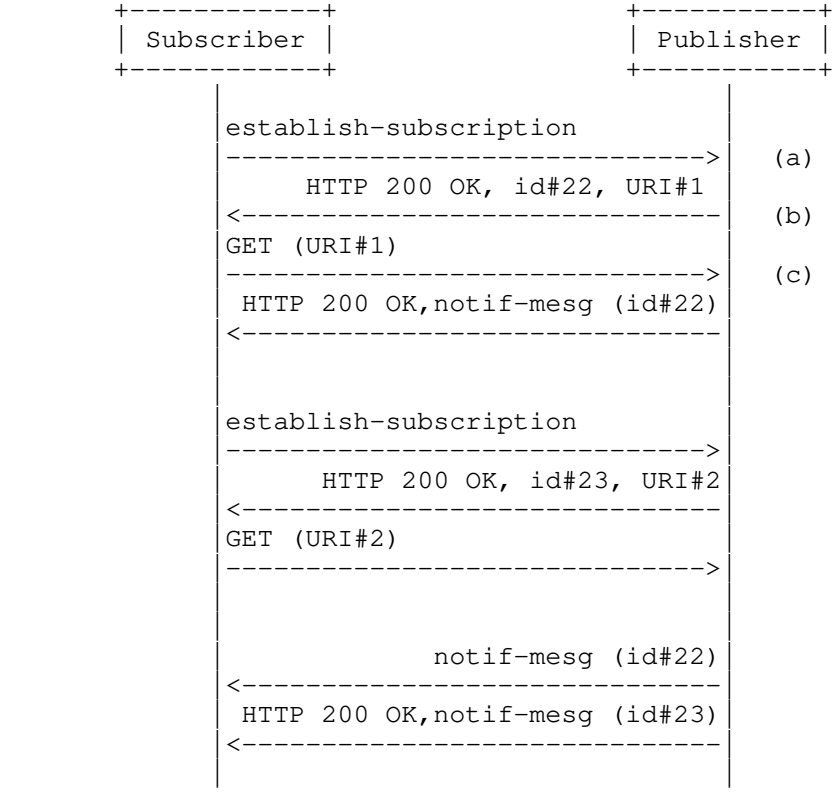


Figure 2: Multiple Subscriptions over RESTCONF/HTTP

To provide examples of the information being transported, example messages for interactions in Figure 2 are detailed below:

```
POST /restconf/operations
     /ietf-subscribed-notifications:establish-subscription

{
   "ietf-subscribed-notifications:input": {
      "stream-xpath-filter": "/example-module:foo/",
      "stream": "NETCONF",
      "dscp": 10
   }
}
```

Figure 3: "establish-subscription" Request (a)

As the publisher was able to fully satisfy the request, the publisher sends the subscription identifier of the accepted subscription and the URI:

```
HTTP status code - 200

{
   "id": 22,
   "uri": "https://example.com/restconf/subscriptions/22"
}
```

Figure 4: "establish-subscription" Success (b)

Upon receipt of the successful response, the subscriber does a GET to the provided URI to start the flow of notification messages. When the publisher receives this, the subscription is moved to the active state (c).

```
GET /restconf/subscriptions/22
```

Figure 5: "establish-subscription" Subsequent POST

While not shown in Figure 2, if the publisher had not been able to fully satisfy the request, or the subscriber has no authorization to establish the subscription, the publisher would have sent an RPC error response. For instance, if the "dscp" value of 10 asserted by the subscriber in Figure 3 proved unacceptable, the publisher may have returned:

```
HTTP status code - 400

{ "ietf-restconf:errors" : {
    "error" : [
      {
        "error-type": "application",
        "error-tag": "invalid-value",
        "error-severity": "error",
        "error-app-tag":
            "ietf-subscribed-notifications:dscp-unavailable"
      }
    ]
  }
}
```

Figure 6: An Unsuccessful "establish-subscription"

The subscriber can use this information in future attempts to establish a subscription.

A.1.2. Modifying Dynamic Subscriptions

An existing subscription may be modified. The following exchange shows a negotiation of such a modification via several exchanges between a subscriber and a publisher. This negotiation consists of a failed RPC modification request/response followed by a successful one.

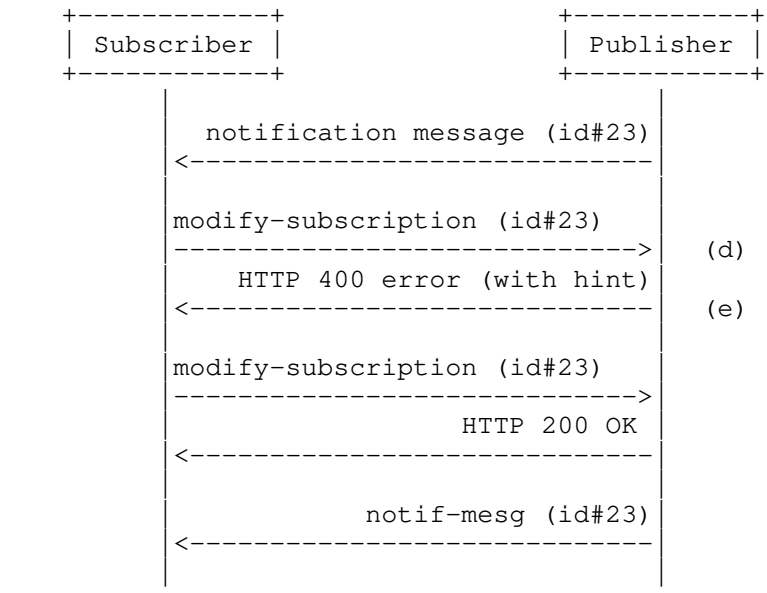


Figure 7: Interaction Model for Successful Subscription Modification

If the subscription being modified in Figure 7 is a datastore subscription as per [RFC8641], the modification request made in (d) may look like that shown in Figure 8. As can be seen, the modifications being attempted are the application of a new XML Path Language (XPath) filter as well as the setting of a new periodic time interval.

```
POST /restconf/operations
     /ietf-subscribed-notifications:modify-subscription

{
 "ietf-subscribed-notifications:input": {
    "id": 23,
    "ietf-yang-push:datastore-xpath-filter":
       "/example-module:foo/example-module:bar",
    "ietf-yang-push:periodic": {
       "ietf-yang-push:period": 500
    }
  }
}
```

Figure 8: Subscription Modification Request (c)

If the publisher can satisfy both changes, the publisher sends a positive result for the RPC. If the publisher cannot satisfy either of the proposed changes, the publisher sends an RPC error response (e). The following is an example RPC error response for (e) that includes a hint. This hint is an alternative time period value that might have resulted in a successful modification:

```
HTTP status code - 400

{ "ietf-restconf:errors" : {
    "error" : [
      "error-type": "application",
      "error-tag": "invalid-value",
      "error-severity": "error",
      "error-app-tag": "ietf-yang-push:period-unsupported",
      "error-info": {
        "ietf-yang-push":
        "modify-subscription-datastore-error-info": {
           "period-hint": 3000
        }
      }
    ]
  }
}
```

Figure 9: "modify-subscription" Failure with Hint (e)

A.1.3. Deleting Dynamic Subscriptions

The following demonstrates deleting a subscription. This subscription may have been to either a stream or a datastore.

```
POST /restconf/operations
     /ietf-subscribed-notifications:delete-subscription

{
 "delete-subscription": {
    "id": "22"
 }
}
```

Figure 10: "delete-subscription" Request

If the publisher can satisfy the request, the publisher replies with success to the RPC request.

If the publisher cannot satisfy the request, the publisher sends an <rpc-error> element indicating the modification didn't work. Figure 11 shows a valid response for an existing valid subscription identifier, but that subscription identifier was created on a different transport session:

```
HTTP status code - 404

{
  "ietf-restconf:errors" : {
    "error" : [
      "error-type": "application",
      "error-tag": "invalid-value",
      "error-severity": "error",
      "error-app-tag":
         "ietf-subscribed-notifications:no-such-subscription"
    ]
  }
}
```

Figure 11: Unsuccessful "delete-subscription"

A.2. Subscription State Notifications

A publisher will send subscription state notifications according to the definitions within [RFC8639].

A.2.1. "subscription-modified"

A "subscription-modified" encoded in JSON would look like:

```
{
  "ietf-restconf:notification" : {
    "eventTime": "2007-09-01T10:00:00Z",
    "ietf-subscribed-notifications:subscription-modified": {
      "id": 39,
      "uri": "https://example.com/restconf/subscriptions/22"
      "stream-xpath-filter": "/example-module:foo",
      "stream": {
         "ietf-netconf-subscribed-notifications" : "NETCONF"
      }
    }
  }
}
```

Figure 12: "subscription-modified" Subscription State Notification

A.2.2. "subscription-completed", "subscription-resumed", and "replay-completed"

A "subscription-completed" notification would look like:

```
{
  "ietf-restconf:notification" : {
    "eventTime": "2007-09-01T10:00:00Z",
    "ietf-subscribed-notifications:subscription-completed": {
      "id": 39,
    }
  }
}
```

Figure 13: "subscription-completed" Notification in JSON

The "subscription-resumed" and "replay-complete" are virtually identical, with "subscription-completed" simply being replaced by "subscription-resumed" and "replay-complete".

A.2.3. "subscription-terminated" and "subscription-suspended"

A "subscription-terminated" would look like:

```
{
  "ietf-restconf:notification" : {
    "eventTime": "2007-09-01T10:00:00Z",
    "ietf-subscribed-notifications:subscription-terminated": {
      "id": 39,
      "error-id": "suspension-timeout"
    }
  }
}
```

Figure 14: "subscription-terminated" Subscription State Notification

The "subscription-suspended" is virtually identical, with "subscription-terminated" simply being replaced by "subscription-suspended".

A.3. Filter Example

This section provides an example that illustrates the method of filtering event record contents. The example is based on the YANG notification "vrrp-protocol-error-event" as defined per the ietf-vrrp.yang module within [RFC8347]. Event records based on this specification that are generated by the publisher might appear as:

```
data: {
data:   "ietf-restconf:notification" : {
data:     "eventTime" : "2018-09-14T08:22:33.44Z",
data:     "ietf-vrrp:vrrp-protocol-error-event" : {
data:       "protocol-error-reason" : "checksum-error"
data:     }
data:   }
data: }
```

Figure 15: RFC 8347 (VRRP) - Example Notification

Suppose a subscriber wanted to establish a subscription that only passes instances of event records where there is a "checksum-error" as part of a Virtual Router Redundancy Protocol (VRRP) protocol event. Also assume the publisher places such event records into the NETCONF stream. To get a continuous series of matching event records, the subscriber might request the application of an XPath filter against the NETCONF stream. An "establish-subscription" RPC to meet this objective might be:

```
POST /restconf/operations
     /ietf-subscribed-notifications:establish-subscription
{
   "ietf-subscribed-notifications:input": {
      "stream": "NETCONF",
      "stream-xpath-filter":
```

```
          "/ietf-vrrp:vrrp-protocol-error-event[
            protocol-error-reason='checksum-error']/",
      }
   }
```

Figure 16: Establishing a Subscription Error Reason via XPath

For more examples of XPath filters, see [XPATH].

Suppose the "establish-subscription" in Figure 16 was accepted. And suppose later a subscriber decided they wanted to broaden this subscription cover to all VRRP protocol events (i.e., not just those with a "checksum error"). The subscriber might attempt to modify the subscription in a way that replaces the XPath filter with a subtree filter that sends all VRRP protocol events to a subscriber. Such a "modify-subscription" RPC might look like:

```
POST /restconf/operations
     /ietf-subscribed-notifications:modify-subscription
{
   "ietf-subscribed-notifications:input": {
      "stream": "NETCONF",
      "stream-subtree-filter": {
        "/ietf-vrrp:vrrp-protocol-error-event" : {}
      }
   }
}
```

Figure 17: Example "modify-subscription" RPC

For more examples of subtree filters, see [RFC6241], Section 6.4.

Acknowledgments

We wish to acknowledge the helpful contributions, comments, and suggestions that were received from Ambika Prasad Tripathy, Alberto Gonzalez Prieto, Susan Hares, Tim Jenkins, Balazs Lengyel, Kent Watsen, Michael Scharf, Guangying Zheng, Martin Bjorklund, Qin Wu, and Robert Wilton.

Authors' Addresses

Eric Voit
Cisco Systems

Email: evoit@cisco.com

Reshad Rahman
Cisco Systems

Email: rrahman@cisco.com

Einar Nilsen-Nygaard
Cisco Systems

Email: einarnn@cisco.com

Alexander Clemm
Futurewei

Email: ludwig@clemm.org

Andy Bierman
YumaWorks

Email: andy@yumaworks.com