

NETCONF
Internet-Draft
Intended status: Standards Track
Expires: July 15, 2019

E. Voit
R. Rahman
E. Nilsen-Nygaard
Cisco Systems
A. Clemm
Huawei
A. Bierman
YumaWorks
January 11, 2019

**Dynamic subscription to YANG Events and Datastores over RESTCONF
draft-ietf-netconf-restconf-notif-12**

Abstract

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

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 July 15, 2019.

Copyright Notice

Copyright (c) 2019 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
2.	Terminology	3
3.	Dynamic Subscriptions	3
3.1.	Transport Connectivity	4
3.2.	Discovery	4
3.3.	RESTCONF RPCs and HTTP Status Codes	4
3.4.	Call Flow for Server-Sent Events (SSE)	6
4.	QoS Treatment	8
5.	Notification Messages	9
6.	YANG Tree	9
7.	YANG module	9
8.	IANA Considerations	11
9.	Security Considerations	11
10.	Acknowledgments	12
11.	References	12
11.1.	Normative References	12
11.2.	Informative References	14
Appendix A.	Examples	14
A.1.	Dynamic Subscriptions	15
A.1.1.	Establishing Dynamic Subscriptions	15
A.1.2.	Modifying Dynamic Subscriptions	17
A.1.3.	Deleting Dynamic Subscriptions	19
A.2.	Subscription State Notifications	20
A.2.1.	subscription-modified	20
A.2.2.	subscription-completed, subscription-resumed, and replay-complete	21
A.2.3.	subscription-terminated and subscription-suspended	21
A.3.	Filter Example	21
Appendix B.	Changes between revisions	23
Authors' Addresses	26

[1.](#) Introduction

Mechanisms to support event subscription and push are defined in [I-D.[draft-ietf-netconf-subscribed-notifications](#)]. Enhancements to [I-D.[draft-ietf-netconf-subscribed-notifications](#)] which enable YANG datastore subscription and push are defined in [I-D.[ietf-netconf-yang-push](#)]. This document provides a transport specification for dynamic subscriptions over RESTCONF [[RFC8040](#)]. Driving these requirements is [[RFC7923](#)].

The streaming of notifications encapsulating 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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

The following terms use the definitions from [I-D.[draft-ietf-netconf-subscribed-notifications](#)]: dynamic subscription, event stream, notification message, publisher, receiver, subscriber, and subscription.

Other terms reused include datastore, which is defined in [[RFC8342](#)], and HTTP2 stream which maps to the definition of "stream" within [[RFC7540](#)], [Section 2](#).

[note to the RFC Editor - please replace XXXX within this document with the number of this document]

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 [I-D.[draft-ietf-netconf-subscribed-notifications](#)] [Section 2.4](#), the RPCs are done via RESTCONF POSTs. YANG datastore subscription is accomplished via augmentations to [I-D.[draft-ietf-netconf-subscribed-notifications](#)] as described within [I-D.[ietf-netconf-yang-push](#)] [Section 4.4](#).

As described in [[RFC8040](#)] [Section 6.3](#), a GET needs to be made against a specific URI on the publisher. Subscribers cannot pre-determine 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 which 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. A subscription does not move to the active state as per [Section 2.4.1](#). of [I-D.[draft-ietf-netconf-subscribed-notifications](#)] until the GET is received.

3.1. Transport Connectivity

For a dynamic subscription, where 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 [[RFC5246](#)] in order to secure the content in transit.

Without the involvement of additional protocols, HTTP sessions by themselves do not allow for a quick recognition of when the communication path has been lost with the publisher. Where quick recognition of the loss of a publisher is required, a subscriber SHOULD use a TLS heartbeat [[RFC6520](#)], just from receiver to publisher, to track HTTP session continuity.

Loss of the heartbeat MUST result in any subscription related TCP sessions between those endpoints being torn down. A subscriber can then attempt to re-establish the dynamic subscription by using the procedure described in [Section 3](#).

3.2. Discovery

Subscribers can learn what event streams a RESTCONF server supports by querying the "streams" container of ietf-subscribed-notification.yang in [I-D.[draft-ietf-netconf-subscribed-notifications](#)]. Support for the "streams" container of ietf-restconf-monitoring.yang in [[RFC8040](#)] is not required. If it is supported, the event streams which are in the "streams" container of ietf-subscribed-notifications.yang SHOULD also be in the "streams" container of ietf-restconf-monitoring.yang.

Subscribers can learn what datastores a RESTCONF server supports by following [Section 2](#) of [I-D.[draft-ietf-netconf-nmda-restconf](#)].

3.3. RESTCONF RPCs and HTTP Status Codes

Specific HTTP responses codes as defined in [[RFC7231](#)] [section 6](#) will indicate the result of RESTCONF RPC requests with publisher. An HTTP status code of 200 is the proper response to any successful RPC defined within [I-D.[draft-ietf-netconf-subscribed-notifications](#)] or [I-D.[ietf-netconf-yang-push](#)].

If a publisher fails to serve the RPC request for one of the reasons indicated in [I-D.[draft-ietf-netconf-subscribed-notifications](#)] [Section 2.4.6](#) or [I-D.[ietf-netconf-yang-push](#)] [Appendix A](#), this will be indicated by "406" status code transported in the HTTP response.

When a "406" status code is returned, the RPC reply MUST include an "rpc-error" element per [\[RFC8040\] Section 7.1](#) with the following parameter values:

- o an "error-type" node of "application".
- o an "error-tag" node of "operation-failed".
- o an "error-app-tag" node with the value being a string that corresponds to an identity associated with the error, as defined in [I-D.[draft-ietf-netconf-subscribed-notifications](#)] [section 2.4.6](#) for general subscriptions, and [[I-D.ietf-netconf-yang-push](#)] [Appendix A.1](#), for datastore subscriptions. The tag to use depends on the RPC for which the error occurred. Viable errors for different RPCs are as follows:

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

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

In case of error responses to an "establish-subscription" or "modify-subscription" request there is the option of including an "error-info" node. This node may contain hints for parameter settings that might lead to successful RPC requests in the future. Following are the yang-data structures which may be returned:

establish-subscription returns hints in yang-data structure

```
-----  
target: event stream    establish-subscription-stream-error-info  
target: datastore       establish-subscription-datastore-error-info
```

modify-subscription returns hints in yang-data structure

```
-----  
target: event stream    modify-subscription-stream-error-info  
target: datastore       modify-subscription-datastore-error-info
```

The yang-data included within "error-info" SHOULD NOT include the optional leaf "error-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 (SSE)

The call flow is defined in Figure 1. The logical connections denoted by (a) and (b) can be a TCP connection or an HTTP2 stream (multiple HTTP2 streams can be carried in one TCP connection). Requests to [I-D.[draft-ietf-netconf-subscribed-notifications](#)] or [I-D.[ietf-netconf-yang-push](#)] augmented RPCs are sent on a connection indicated by (a). A successful "establish-subscription" will result in an RPC response returned with both a subscription identifier which uniquely identifies a subscription, as well as a URI which uniquely identifies the location of subscription on the publisher (b). This URI is defined via the "uri" leaf 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 initiates the publisher to initiate the flow of notification messages which are sent in SSE [[W3C-20150203](#)] as a response to the GET.

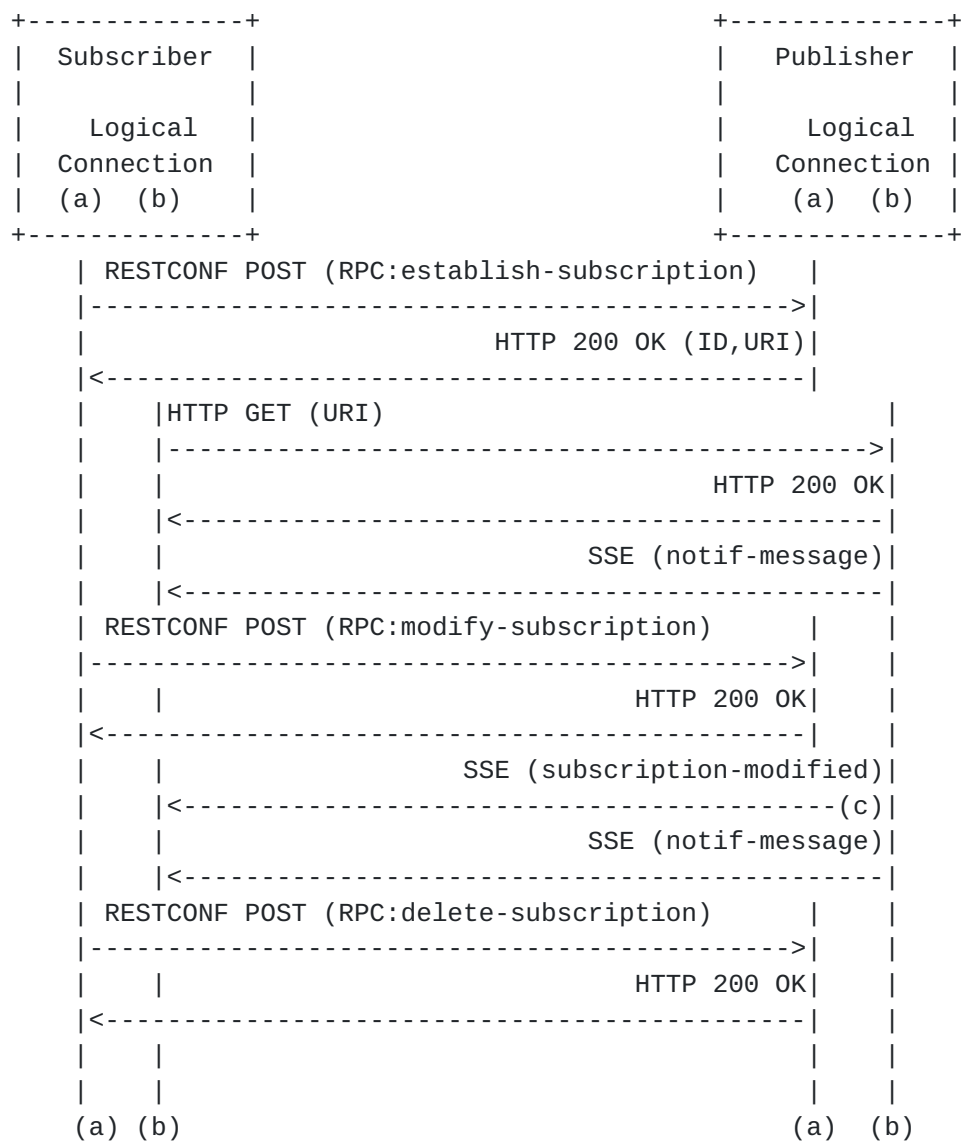


Figure 1: Dynamic with server-sent events

Additional requirements for dynamic subscriptions over SSE include:

- o All subscription state notifications from a publisher MUST be returned in a separate SSE message used by the subscription to which the state change refers.
- o Subscription RPCs MUST NOT use the connection currently providing notification messages for that subscription.
- o 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 the new terms of the subscription have been applied to the notification messages. See arrow (c).

- o In addition to any required access permissions (e.g., NACM), RPCs modify-subscription, resync-subscription and delete-subscription SHOULD only be allowed by the same RESTCONF username [[RFC8040](#)] which invoked establish-subscription.
- o 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:

- o Receipt of a "delete-subscription" or a "kill-subscription" RPC for that subscription.
- o Loss of TLS heartbeat

A publisher MAY terminate a subscription at any time as stated in [I-D.[draft-ietf-netconf-subscribed-notifications](#)] [Section 1.3](#)

4. QoS Treatment

To meet subscription quality of service promises, the publisher MUST take any existing subscription "dscp" and apply it to the DSCP marking in the IP header.

In addition, where HTTP2 transport is available to a notification message queued for transport to a receiver, the publisher MUST:

- o take any existing subscription "priority", as specified by the "weighting" leaf node in [I-D.[draft-ietf-netconf-subscribed-notifications](#)], and copy it into the HTTP2 stream weight, [[RFC7540](#)] [section 5.3](#), and
- o take any existing subscription "dependency", as specified by the "dependency" leaf node in [I-D.[draft-ietf-netconf-subscribed-notifications](#)], and use the HTTP2 stream for the parent subscription as the HTTP2 stream dependency, [[RFC7540](#)] [section 5.3.1](#), of the dependent subscription.
- o set the exclusive flag, [[RFC7540](#)] [section 5.3.1](#), to 0.

5. Notification Messages

Notification messages transported over RESTCONF will be encoded according to [\[RFC8040\], section 6.4](#).

6. YANG Tree

The YANG model defined in [Section 7](#) has one leaf augmented into three places of [\[I-D.draft-ietf-netconf-subscribed-notifications\]](#).

```
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
[\[I-D.draft-ietf-netconf-subscribed-notifications\]](#).

```
<CODE BEGINS> file
  "ietf-restconf-subscribed-notifications@2019-01-11.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:   <http://tools.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 XXXX; see the RFC itself for full legal notices.";

```
revision 2019-01-11 {
  description
    "Initial version";
  reference
    "RFC XXXX: RESTCONF Transport for Event Notifications";
}

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 "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 XXXX: RESTCONF Transport for Event Notifications

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 [[RFC5246](#)].

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 nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

Container: "/subscriptions"

- o "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, and perhaps even HTTP session 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.

10. 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.

11. References

11.1. Normative References

[I-D.[draft-ietf-netconf-subscribed-notifications](#)]

Voit, E., Clemm, A., Gonzalez Prieto, A., Tripathy, A., and E. Nilsen-Nygaard, "Custom Subscription to Event Streams", [draft-ietf-netconf-subscribed-notifications-21](#) (work in progress), January 2019.

[I-D.ietf-netconf-yang-push]

Clemm, A., Voit, E., Gonzalez Prieto, A., Prasad Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", [draft-ietf-netconf-yang-push-20](#) (work in progress), October 2018, <<https://datatracker.ietf.org/doc/draft-ietf-netconf-yang-push/>>.

[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>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [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>>.
- [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>>.
- [W3C-20150203] "Server-Sent Events, World Wide Web Consortium CR CR-eventsource-20121211", February 2015, <<https://www.w3.org/TR/2015/REC-eventsource-20150203/>>.

11.2. Informative References

- [I-D.[draft-ietf-netconf-netconf-event-notifications](#)]
Clemm, Alexander., Voit, Eric., Gonzalez Prieto, Alberto., Nilsen-Nygaard, E., and A. Tripathy, "NETCONF support for event notifications", May 2018,
<<https://datatracker.ietf.org/doc/draft-ietf-netconf-netconf-event-notifications/>>.
- [I-D.[draft-ietf-netconf-nmda-restconf](#)]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "RESTCONF Extensions to Support the Network Management Datastore Architecture", April 2018,
<<https://datatracker.ietf.org/doc/draft-ietf-netconf-nmda-restconf/>>.
- [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>>.
- [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>>.
- [XPath] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", November 1999,
<<http://www.w3.org/TR/1999/REC-xpath-19991116>>.

Appendix A. Examples

This section is non-normative. To allow easy comparison, this section mirrors the functional examples shown with NETCONF over XML within [I-D.[draft-ietf-netconf-netconf-event-notifications](#)]. In addition, HTTP2 vs HTTP1.1 headers are not shown as the contents of the JSON encoded objects are identical within.

A.1. Dynamic Subscriptions

A.1.1. Establishing Dynamic Subscriptions

The following figure shows two successful "establish-subscription" RPC requests as per [I-D.[draft-ietf-netconf-subscribed-notifications](#)]. The first request is given a subscription identifier of 22, the second, an identifier of 23.

+-----+	+-----+
Subscriber	Publisher
+-----+	+-----+
establish-subscription	
----->	(a)
HTTP 200 OK, id#22, URI#1	
<-----	(b)
GET (URI#1)	
----->	(c)
HTTP 200 OK,notif-mesg (id#22)	
<-----	
establish-subscription	
----->	
HTTP 200 OK, id#23, URI#2	
<-----	
GET (URI#2)	
----->	
notif-mesg (id#22)	
<-----	
HTTP 200 OK,notif-mesg (id#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 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 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 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 - 406

```
{ "ietf-restconf:errors" : {  
  "error" : [  
    {  
      "error-type": "application",  
      "error-tag": "operation-failed",  
      "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.

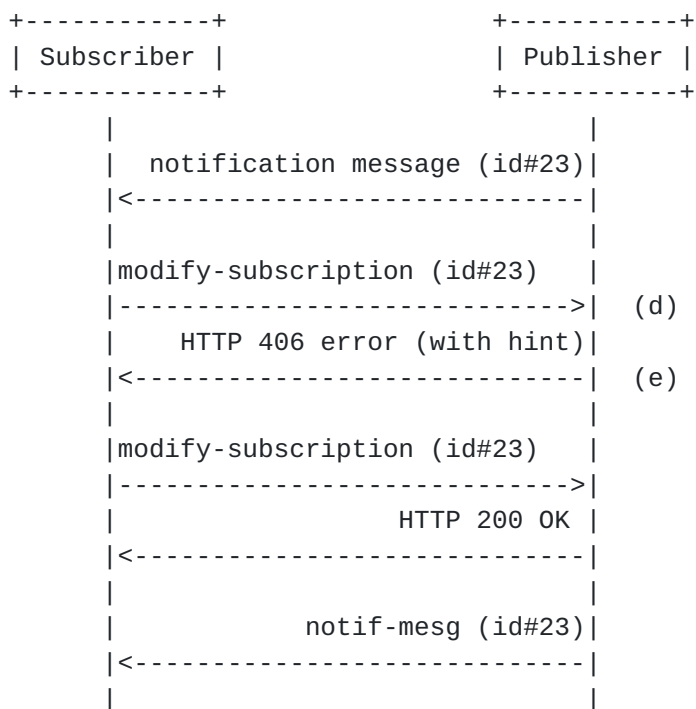


Figure 7: Interaction model for successful subscription modification

If the subscription being modified in Figure 7 is a datastore subscription as per [\[I-D.ietf-netconf-yang-push\]](#), 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 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) which includes a hint. This hint is an alternative time period value which might have resulted in a successful modification:

HTTP status code - 406

```
{ "ietf-restconf:errors" : {
  "error" : [
    "error-type": "application",
    "error-tag": "operation-failed",
    "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

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 error-rpc element indicating the modification didn't work. Figure 11 shows a valid response for existing valid subscription identifier,

but that subscription identifier was created on a different transport session:

```
HTTP status code - 406

{
  "ietf-restconf:errors" : {
    "error" : [
      "error-type": "application",
      "error-tag": "operation-failed",
      "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

[I-D.[draft-ietf-netconf-subscribed-notifications](#)]).

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-complete

A "subscription-completed" 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 which illustrate 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 which 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 which only passes instances of event records where there is a "checksum-error" as part of a 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 which replaces the XPath filter with a subtree filter which 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

For more examples of subtree filters, see [\[RFC6241\]](#), [section 6.4](#).

[Appendix B](#). Changes between revisions

(To be removed by RFC editor prior to publication)

v11 - v12

- o Added text in 3.2 for expected behavior when ietf-restconf-monitoring.yang is also supported.
- o Added [section 2](#) to the reference to [draft-ietf-netconf-nmda-restconf](#).
- o Replaced kill-subscription-error by delete-subscription-error in [section 3.3](#).
- o Clarified vertical lines (a) and (b) in Figure 1 of [section 3.4](#)
- o [Section 3.4](#), 3rd bullet after Figure 1, replaced "must" with "MUST".
- o Modified text in [section 3.4](#) regarding access to RPCs modify-subscription, resync-subscription, delete-subscription and kill-subscription.
- o [Section 4](#), first bullet for HTTP2: replaced dscp and priority with weighting and weight.
- o [Section 6](#), added YANG tree diagram and fixed description of the module.
- o [Section 7](#), fixed indentation of module description statement.

- o [Section 7](#), in YANG module changed year in copyright statement to 2019.
- o [Section 8](#), added text on how server protects access to the subscription URI.
- o Fixed outdated references and removed unused references.
- o Fixed the instances of line too long.
- o Fixed example in Figure 3.

v10 - v11

- o Per Kent's request, added name attribute to artwork which need to be extracted

v09 - v10

- o Fixed typo for resync.
- o Added text wrt RPC permissions and RESTCONF username.

v08 - v09

- o Addressed comments received during WGLC.

v07 - v08

- o Aligned with RESTCONF mechanism.
- o YANG model: removed augment of subscription-started, added restconf transport.
- o Tweaked [Appendix A.1](#) to match [draft-ietf-netconf-netconf-event-notifications-13](#).
- o Added [Appendix A.3](#) for filter example.

v06 - v07

- o Removed configured subscriptions.
- o Subscription identifier renamed to id.

v05 - v06

- o JSON examples updated by Reshad.

v04 - v05

- o Error mechanisms updated to match embedded RESTCONF mechanisms
- o Restructured format and sections of document.
- o Added a YANG data model for HTTP specific parameters.
- o Mirrored the examples from the NETCONF transport draft to allow easy comparison.

v03 - v04

- o Draft not fully synched to new version of subscribed-notifications yet.
- o References updated

v02 - v03

- o Event notification reframed to notification message.
- o Tweaks to wording/capitalization/format.

v01 - v02

- o Removed sections now redundant with [I-D.[draft-ietf-netconf-subscribed-notifications](#)] and [I-D.[ietf-netconf-yang-push](#)] such as: mechanisms for subscription maintenance, terminology definitions, stream discovery.
- o 3rd party subscriptions are out-of-scope.
- o SSE only used with RESTCONF and HTTP1.1 dynamic subscriptions
- o Timeframes for event tagging are self-defined.
- o Clean-up of wording, references to terminology, section numbers.

v00 - v01

- o Removed the ability for more than one subscription to go to a single HTTP2 stream.
- o Updated call flows. Extensively.
- o SSE only used with RESTCONF and HTTP1.1 dynamic subscriptions

- o HTTP is not used to determine that a receiver has gone silent and is not Receiving Event Notifications
- o Many clean-ups of wording and terminology

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
Huawei

Email: ludwig@clemm.org

Andy Bierman
YumaWorks

Email: andy@yumaworks.com

