NETCONF
Internet-Draft
Intended status: Standards Track
Expires: April 7, 2019

E. Voit
Cisco Systems
A. Clemm
Huawei
A. Gonzalez Prieto
Microsoft
E. Nilsen-Nygaard
A. Tripathy
Cisco Systems

October 4, 2018

NETCONF Support for Event Notifications
draft-ietf-netconf-netconf-event-notifications-13

Abstract

This document provides a NETCONF binding to the dynamic subscription capability of both subscribed notifications and YANG push.

RFC Editor note: please replace the four references to pre-RFC normative drafts with the actual assigned RFC numbers.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Compatibility with RFC-5277's create-subscription 3
4. Mandatory XML, event stream and datastore support 3
5. NETCONF connectivity and the Dynamic Subscriptions 4
6. Notification Messages 4
7. Dynamic Subscriptions and RPC Error Responses 4
8. Security Considerations 5
9. Acknowledgments 6
10. Notes to the RFC Editor 6
11. References 6
11.1. Normative References 6
11.2. Informative References 7
Appendix A. Examples 7
A.1. Event Stream Discovery 7
A.2. Dynamic Subscriptions 8
A.3. Subscription State Notifications 12
A.4. Filter Examples 14
Appendix B. Changes between revisions 15
B.1. v11 to v13 15
B.2. v10 to v11 16
B.3. v09 to v10 16
B.4. v08 to v09 16
B.5. v07 to v08 16
B.6. v06 to v07 16
B.7. v05 to v06 16
B.8. v03 to v04 16
B.9. v01 to v03 17
B.10. v00 to v01 17
Authors' Addresses 17

1. Introduction

This document provides a binding for events streamed over the NETCONF protocol [RFC6241] for dynamic subscriptions as defined in [I-D.draft-ietf-netconf-subscribed-notifications]. In addition, as [I-D.ietf-netconf-yang-push] is itself built upon [I-D.draft-ietf-netconf-subscribed-notifications], this document

enables a NETCONF client to request via a dynamic subscription and receive updates from a YANG datastore located on a NETCONF server.

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 are defined in [I-D.draft-ietf-netconf-subscribed-notifications]: dynamic subscription, event stream, notification message, publisher, receiver, subscriber, subscription.

3. Compatibility with RFC-5277's create-subscription

A publisher is allowed to concurrently support dynamic subscription RPCs of [I-D.draft-ietf-netconf-subscribed-notifications] at the same time as [RFC5277]'s "create-subscription" RPC. However a single NETCONF transport session cannot support both this specification and a subscription established by [RFC5277]'s "create-subscription" RPC. To protect against any attempts to use a single NETCONF transport session in this way:

- A solution must reply with the [RFC6241] error "operation-not-supported" if a "create-subscription" RPC is received on a NETCONF session where an [I-D.draft-ietf-netconf-subscribed-notifications] established subscription exists.
- A solution must reply with the [RFC6241] error "operation-not-supported" if an "establish-subscription" request is been received on a NETCONF session where the "create-subscription" RPC has successfully [RFC5277] created a subscription.

If a publisher supports this specification but not subscriptions via [RFC5277], the publisher MUST NOT advertise "urn:ietf:params:netconf:capability:notification:1.0".

4. Mandatory XML, event stream and datastore support

The "encode-xml" feature of [I-D.draft-ietf-netconf-subscribed-notifications] is mandatory to support. This indicates that XML is a valid encoding for RPCs, state change notifications, and subscribed content.

A NETCONF publisher supporting event stream subscription via [I-D.draft-ietf-netconf-subscribed-notifications] MUST support the "NETCONF" event stream identified in that draft.

5. NETCONF connectivity and the Dynamic Subscriptions

For a dynamic subscription, if the NETCONF session involved with the "establish-subscription" terminates, the subscription MUST be terminated.

For a dynamic subscription a "modify-subscription", "delete-subscription", or "resynch-subscription" RPC MUST be sent using same the NETCONF session upon which the referenced subscription was established.

6. Notification Messages

Notification messages transported over the NETCONF protocol MUST be encoded in a <notification> message as defined within [RFC5277], Section 4. And per [RFC5277]'s "eventTime" object definition, the "eventTime" MUST be populated with the event occurrence time.

For dynamic subscriptions, all notification messages MUST use the NETCONF transport session used by the "establish-subscription" RPC.

7. Dynamic Subscriptions and RPC Error Responses

Management of dynamic subscriptions occurs via RPCs as defined in [I-D.ietf-netconf-yang-push] and [I-D.draft-ietf-netconf-subscribed-notifications]. When an RPC error occurs, the NETCONF RPC reply MUST include an "rpc-error" element per [RFC6241] with the error information populated as follows:

- an "error-type" node of "application".
- an "error-tag" node of "operation-failed".
- an "error-severity" of "error" (this MAY but does not have to be included).
- 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 specific identity to use depends on the RPC for which the error occurred. Viable errors for different RPCs are as follows:

RPC	use base identity
establish-subscription	establish-subscription-error
modify-subscription	modify-subscription-error
delete-subscription	delete-subscription-error
kill-subscription	kill-subscription-error
resynch-subscription	resynch-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 XML-encoded data with 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 resulting from a "delete-subscription", "kill-subscription", or "resynch-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.

8. Security Considerations

If a malicious or buggy NETCONF subscriber sends a number of establish-subscription requests, then these subscriptions accumulate and may use up system resources. In such a situation, subscriptions MAY be terminated by terminating the underlying NETCONF session. The

publisher MAY also suspend or terminate a subset of the active subscriptions on that NETCONF session.

9. Acknowledgments

We wish to acknowledge the helpful contributions, comments, and suggestions that were received from: Andy Bierman, Yan Gang, Sharon Chisholm, Hector Trevino, Peipei Guo, Susan Hares, Tim Jenkins, Balazs Lengyel, Martin Bjorklund, Mahesh Jethanandani, Kent Watsen, and Guangying Zheng.

10. Notes to the RFC Editor

This section can be removed by the RFC editor after the requests have been performed.

RFC 6241 need to be updated. RFC-6241 refers to RFC-5277 which says that a notification message can only be sent after a successful "create-subscription". This text must be modified to also allow notification messages be sent after a successful "establish-subscription".

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, "Customized Subscriptions to a Publisher's Event Streams", draft-ietf-netconf-subscribed-notifications-17 (work in progress), September 2018.

[I-D.ietf-netconf-yang-push]
Clemm, Alexander., Voit, Eric., Gonzalez Prieto, Alberto., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", September 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>.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <https://www.rfc-editor.org/info/rfc5277>.

[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>.

[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>.

11.2. Informative References

[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.

A.1. Event Stream Discovery

As defined in [I-D.draft-ietf-netconf-subscribed-notifications] an event stream exposes a continuous set of events available for subscription. A NETCONF client can retrieve the list of available event streams from a NETCONF publisher using the "get" operation against the top-level container "/streams" defined in [I-D.draft-ietf-netconf-subscribed-notifications] Section 3.1.

The following example illustrates the retrieval of the list of available event streams:

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <streams
     xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"/>
    </filter>
  </get>
</rpc>
```

Figure 1: Get streams request

After such a request, the NETCONF publisher returns a list of event streams available, as well as additional information which might exist in the container.

A.2. Dynamic Subscriptions

A.2.1. 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 "id" of 22, the second, an "id" of 23.

```
      +------------+                 +-----------+
      | Subscriber |                 | Publisher |
      +------------+                 +-----------+
            |                              |
            |    Capability Exchange       |
            |<---------------------------->|
            |                              |
            |                              |
            |    establish-subscription    |
            |----------------------------->|  (a)
            | RPC Reply: OK, id = 22       |
            |<-----------------------------|  (b)
            |                              |
            | notification message (for 22)|
            |<-----------------------------|
            |                              |
            |                              |
            |    establish-subscription    |
            |----------------------------->|
            | notification message (for 22)|
            |<-----------------------------|
            | RPC Reply: OK, id = 23       |
            |<-----------------------------|
            |                              |
            |                              |
            | notification message (for 22)|
            |<-----------------------------|
            | notification message (for 23)|
            |<-----------------------------|
            |                              |
```

Figure 2: Multiple subscriptions over a NETCONF session

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

```
<rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <establish-subscription
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <stream>NETCONF</stream>
    <stream-xpath-filter xmlns:ds="http://example.com/events">
      /ds:foo/
    </stream-xpath-filter>
    <dscp>10</dscp>
  </establish-subscription>
</rpc>
```

Figure 3: establish-subscription request (a)

As NETCONF publisher was able to fully satisfy the request (a), the publisher sends the subscription "id" of the accepted subscription within message (b):

```
  <rpc-reply message-id="102"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <id
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
      22
    </id>
  </rpc-reply>
```

Figure 4: establish-subscription success (b)

If the NETCONF 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:

```
   <rpc-reply message-id="102"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <rpc-error>
      <error-type>application</error-type>
      <error-tag>operation-failed</error-tag>
      <error-severity>error</error-severity>
      <error-app-tag>
        ietf-subscribed-notifications:dscp-unavailable
      </error-app-tag>
     </rpc-error>
   </rpc-reply>
```

Figure 5: an unsuccessful establish subscription

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

A.2.2. 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.

```
      +------------+                 +-----------+
      | Subscriber |                 | Publisher |
      +------------+                 +-----------+
            |                              |
            | notification message (for 23)|
            |<-----------------------------|
            |                              |
            | modify-subscription (id = 23)|
            |----------------------------->|  (c)
            | RPC error (with hint)        |
            |<-----------------------------|  (d)
            |                              |
            | modify-subscription (id = 23)|
            |----------------------------->|
            | RPC Reply: OK                |
            |<-----------------------------|
            |                              |
            | notification message (for 23)|
            |<-----------------------------|
            |                              |
```

Figure 6: Interaction model for successful subscription modification

If the subscription being modified in Figure 6 is a datastore subscription as per [I-D.ietf-netconf-yang-push], the modification request made in (c) may look like that shown in Figure 7. 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.

```
<rpc message-id="303"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <modify-subscription
       xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"
       xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <id>23</id>
    <yp:datastore-xpath-filter xmlns:ds="http://example.com/datastore">
        /ds:foo/ds:bar
    </yp:datastore-xpath-filter>
    <yp:periodic>
      <yp:period>500</yp:period>
    </yp:periodic>
  </modify-subscription>
</rpc>
```

Figure 7: Subscription modification request (c)

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

```
   <rpc-reply message-id="303"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <rpc-error>
       <error-type>application</error-type>
       <error-tag>operation-failed</error-tag>
       <error-severity>error</error-severity>
       <error-app-tag>
           ietf-yang-push:period-unsupported
       </error-app-tag>
       <error-info>
          xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
          <modify-subscription-datastore-error-info>
            <period-hint>
                3000
            </period-hint>
          </modify-subscription-datastore-error-info>
       </error-info>
     </rpc-error>
   </rpc-reply>
```

Figure 8: Modify subscription failure with Hint (d)

A.2.3. Deleting Dynamic Subscriptions

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

```
<rpc message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <delete-subscription
    xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <id>22</id>
  </delete-subscription>
</rpc>
```

Figure 9: Delete subscription

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

If the NETCONF publisher cannot satisfy the request, the publisher sends an error-rpc element indicating the modification didn't work. Figure 10 shows a valid response for existing valid subscription "id", but that subscription "id" was created on a different NETCONF transport session:

```
<rpc-reply message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rpc-error>
    <error-type>application</error-type>
    <error-tag>operation-failed</error-tag>
    <error-severity>error</error-severity>
    <error-app-tag>
        ietf-subscribed-notifications:no-such-subscription
    </error-app-tag>
  </rpc-error>
</rpc-reply>
```

Figure 10: Unsuccessful delete subscription

A.3. Subscription State Notifications

A publisher will send subscription state notifications for dynamic subscriptions according to the definitions within [I-D.draft-ietf-netconf-subscribed-notifications]).

A.3.1. subscription-modified

As per Section 2.7.2 of [I-D.draft-ietf-netconf-subscribed-notifications], a "subscription-modified" might be sent if over NETCONF if the definition of a configured filter changes. A subscription state notification encoded in XML would look like:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-09-01T10:00:00Z</eventTime>
  <subscription-modified
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <id>39</id>
    <transport xmlns:nsn=
    "urn:ietf:params:xml:ns:yang:ietf-netconf-subscribed-notifications">
      nsn:netconf
    <transport>
    <stream-xpath-filter xmlns:ex="http://example.com/events">
      /ex:foo
    </stream-xpath-filter>
    <stream>NETCONF</stream>
  </subscription-modified>
</notification>
```

Figure 11: subscription-modified subscription state notification

A.3.2. subscription-resumed, and replay-complete

A "subscription-resumed" would look like:

```
  <notification
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2007-09-01T10:00:00Z</eventTime>
    <subscription-resumed
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
      <id>39</id>
    </subscription-resumed>
  </notification>
```

Figure 12: subscription-resumed notification in XML

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

A.3.3. subscription-terminated and subscription-suspended

A "subscription-terminated" would look like:

```
<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-09-01T10:00:00Z</eventTime>
  <subscription-terminated
    xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <id>39</id>
    <reason>
       suspension-timeout
    </reason>
  </subscription-terminated>
</notification>
```

Figure 13: subscription-terminated subscription state notification

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

A.4. Filter Examples

This section provides examples which illustrate both xpath and subtree methods of filtering event record contents. The examples are based on the YANG notification "vrrp-protocol-error-event" as defined per the ietf-vrrp.yang model within [RFC8347]. Event records based on this specification which are generated by the publisher might appear as:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2018-09-14T08:22:33.44Z</eventTime>
  <vrrp-protocol-error-event
       xmlns="urn:ietf:params:xml:ns:yang:ietf-vrrp">
     <protocol-error-reason>checksum-error</protocol-error-reason>
  </vrrp-protocol-error-event>
</notification>
```

Figure 14: 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:

```
<rpc message-id="601" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <establish-subscription
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <stream>NETCONF</stream>
    <stream-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-vrrp">
      /vrrp-protocol-error-event[protocol-error-reason="checksum-error"]
    </stream-xpath-filter>
  </establish-subscription>
</rpc>
```

Figure 15: Establishing a subscription error reason via XPATH

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

Suppose the "establish-subscription" in Figure 15 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:

```
 <rpc message-id="602" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <modify-subscription
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
     <id>99</id>
     <stream-subtree-filter>
      <vrrp-protocol-error-event
             xmlns="urn:ietf:params:xml:ns:yang:ietf-vrrp"/>
     </stream-subtree-filter>
   </modify-subscription>
 </rpc>
```

Figure 16

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)

B.1. v11 to v13

o Subscription identifier renamed to id.
o Appendix A.4 for filter examples
o for v13, Tweak of example to /foo/bar

B.2. v10 to v11

- Configured removed.

B.3. v09 to v10

- Tweaks to examples and text.
- Downshifted state names.
- Removed address from examples.

B.4. v08 to v09

- Tweaks based on Kent's comments.
- Updated examples in Appendix A. And updates to some object names based on changes in the subscribed-notifications draft.
- Added a YANG model for the NETCONF identity.

B.5. v07 to v08

- Tweaks and clarification on :interleave.

B.6. v06 to v07

- XML encoding and operational datastore mandatory.
- Error mechanisms and examples updated.

B.7. v05 to v06

- Moved examples to appendices
- All examples rewritten based on namespace learnings
- Normative text consolidated in front
- Removed all mention of JSON
- Call home process detailed
- Note: this is a major revision attempting to cover those comments received from two week review.

B.8. v03 to v04

- Added additional detail to "configured subscriptions"
- Added interleave capability
- Adjusted terminology to that in draft-ietf-netconf-subscribed-notifications
- Corrected namespaces in examples

B.9. v01 to v03

- Text simplifications throughout
- v02 had no meaningful changes

B.10. v00 to v01

- Added Call Home in solution for configured subscriptions.
- Clarified support for multiple subscription on a single session. No need to support multiple create-subscription.
- Added mapping between terminology in yang-push and [RFC6241] (the one followed in this document).
- Editorial improvements.

Authors' Addresses

Eric Voit
Cisco Systems

Email: evoit@cisco.com

Alexander Clemm
Huawei

Email: ludwig@clemm.org

Alberto Gonzalez Prieto
Microsoft

Email: alberto.gonzalez@microsoft.com

Einar Nilsen-Nygaard
Cisco Systems

Email: einarnn@cisco.com

Ambika Prasad Tripathy
Cisco Systems

Email: ambtripa@cisco.com