

NETCONF
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2019

E. Voit
Cisco Systems
A. Clemm
Huawei
A. Gonzalez Prieto
VMware
E. Nilsen-Nygaard
A. Tripathy
Cisco Systems
July 2, 2018

NETCONF Support for Event Notifications
draft-ietf-netconf-netconf-event-notifications-10

Abstract

This document provides a NETCONF binding to subscribed notifications and to 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

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 January 3, 2019.

Copyright Notice

Copyright (c) 2018 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. Compatibility with RFC-5277's create-subscription	3
4. Mandatory XML, event stream and datastore support	4
5. NETCONF connectivity and the subscription lifecycle	4
5.1. Dynamic Subscriptions	4
5.2. Configured Subscriptions	4
6. Notification Messages	5
7. Dynamic Subscriptions and RPC Error Responses	5
8. YANG module	7
9. IANA Considerations	9
10. Security Considerations	9
11. Acknowledgments	9
12. References	9
12.1. Normative References	10
12.2. Informative References	11
Appendix A. Examples	11
A.1. Event Stream Discovery	11
A.2. Dynamic Subscriptions	12
A.3. Configured Subscriptions	16
A.4. Subscription State Notifications	22
Appendix B. Changes between revisions	24
B.1. v09 to v10	24
B.2. v08 to v09	24
B.3. v07 to v08	24
B.4. v06 to v07	24
B.5. v05 to v06	25
B.6. v03 to v04	25
B.7. v01 to v03	25
B.8. v00 to v01	25
Authors' Addresses	25

1. Introduction

This document provides a binding for events streamed over the NETCONF protocol [RFC6241] as per [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 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]: notification message, event stream, publisher, receiver, subscriber, subscription, configured subscription.

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

A publisher is allowed to concurrently support configured subscriptions and 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:

- o A solution must reply with the [RFC6241] error "operation-not-supported" if a "create-subscription" RPC is received on a NETCONF session where any other [I-D.draft-ietf-netconf-subscribed-notifications] or [RFC5277] subscription exists.
- o It is prohibited to send updates or state change notifications for a configured subscription on a NETCONF session where the create-subscription RPC has successfully [RFC5277] created subscription.
- o A "create-subscription" RPC MUST be rejected if any [I-D.draft-ietf-netconf-subscribed-notifications] or [RFC5277]subscription is active across that NETCONF transport session.

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.

A NETCONF publisher supporting [I-D.ietf-netconf-yang-push] MUST support the operational state datastore as defined by [RFC8342].

5. NETCONF connectivity and the subscription lifecycle

This section describes how the availability of NETCONF transport impacts the establishment and lifecycle of different types of [I-D.draft-ietf-netconf-subscribed-notifications] subscriptions.

5.1. Dynamic Subscriptions

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

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."

5.2. Configured Subscriptions

When a configured subscription enters the "valid" state, there is no guarantee a usable NETCONF transport session is currently in place with each associated receiver. As a result, the first configured subscription to a specific receiver MUST establish a NETCONF transport session via NETCONF call home [RFC8071], section 4.1. This transport session MUST then be used by additional configured subscriptions targeting that the same receiver. This same receiver is identifiable on the publisher as one which targets the same IP address and port used to establish the existing NETCONF call home connection. This transport session MAY also be used by dynamic subscriptions and/or non-subscription related NETCONF operations originated by the NETCONF client.

The method of identifying the targeted receiver IP address, port, and security credentials are left up to implementers of this

specification. For implementation guidance and a YANG model for this function, please look to [I-D.draft-ietf-netconf-netconf-client-server].

If the call home fails because the publisher receives receiver credentials which are subsequently declined per [RFC8071], Section 4.1, step S5 authentication, then that receiver MUST be placed into the "timeout" state.

If the call home fails to establish for any other reason, the publisher MUST NOT progress the receiver to the "active" state. Additionally, the publisher SHOULD place the receiver into the "timeout" state after a predetermined number of either failed call home attempts or NETCONF sessions remotely terminated by the receiver.

NETCONF transport session connectivity SHOULD be verified as described in [RFC8071], Section 4.1, step S7.

Until NETCONF transport with a receiver has been established, and a "subscription-started" state change notification has been successfully sent for a configured subscription, that subscription's receiver MUST remain in either the "connecting" or the "timeout" state.

If a NETCONF session is disconnected but the "stop-time" of a subscription being transported over that session has not been reached, the publisher restarts the call home process and return the receiver to the "connecting" state.

6. Notification Messages

Notification messages transported over the NETCONF protocol will use the "notification" message defined within [RFC5277], section 4.

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:

- o an "error-type" node of "application".
- o an "error-tag" node of "operation-failed".

- o an "error-severity" of "error" (this MAY but does not have to be included).
- 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 identityname 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".

- o 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 as a result of a "delete-subscription", a "kill-subscription", or a "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. YANG module

This module references

[I-D.draft-ietf-netconf-subscribed-notifications].

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

```
<CODE BEGINS>file
"ietf-netconf-subscribed-notifications@2018-04-20.yang"
module ietf-netconf-subscribed-notifications {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-netconf-subscribed-notifications";

  prefix nsn;

  import ietf-subscribed-notifications {
    prefix sn;
  }

  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: Alberto Gonzalez Prieto
<mailto:agonzalezpri@vmware.com>

Editor: Ambika Prasad Tripathy
<mailto:ambtripa@cisco.com>

Editor: Einar Nilsen-Nygaard
<mailto:einarnn@cisco.com>";

description

"Defines NETCONF as a supported transport for subscribed event notifications.

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

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

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

revision 2018-04-20 {

description

"Initial version";

reference

"RFC XXXX: NETCONF Support for Event Notifications";

}

identity netconf {

base sn:transport;

base sn:inline-address;

description

"NETCONF is used as a transport for notification messages and state change notifications."

}

}

<CODE ENDS>

9. IANA Considerations

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

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

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

Prefix: nsn

Reference: RFC XXXX: NETCONF Support for Event Notifications

10. Security Considerations

Notification messages (including state change notifications) are never sent before the NETCONF capabilities exchange has completed.

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.

This draft has a YANG module which consists of a single identity. As a result additional security concerns beyond those of the imported modules are not introduced.

11. 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.

12. References

12.1. 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-14 (work in progress), July 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", June 2018,
<[https://datatracker.ietf.org/doc/
draft-ietf-netconf-yang-push/](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>>.
- [RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event
Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008,
<<https://www.rfc-editor.org/info/rfc5277>>.
- [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>>.
- [RFC8071] Watsen, K., "NETCONF Call Home and RESTCONF Call Home",
RFC 8071, DOI 10.17487/RFC8071, February 2017,
<<https://www.rfc-editor.org/info/rfc8071>>.
- [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>>.

[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>>.

12.2. Informative References

[I-D.draft-ietf-netconf-netconf-client-server] Watsen, K. and G. Wu, "NETCONF Client and Server Models", draft-ietf-netconf-netconf-client-server-06 (work in progress), June 2018.

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 identifier of 22, the second, an identifier of 23.

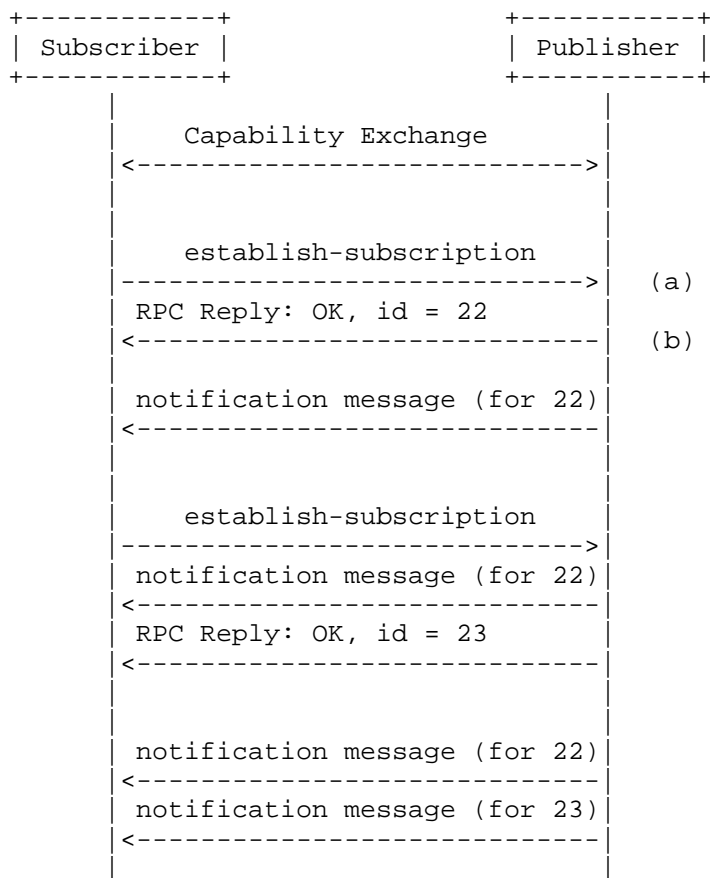


Figure 2: Multiple subscriptions over a NETCONF session

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:ex="http://example.com/events">
      /ex: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 identifier of the accepted subscription within message (b):

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

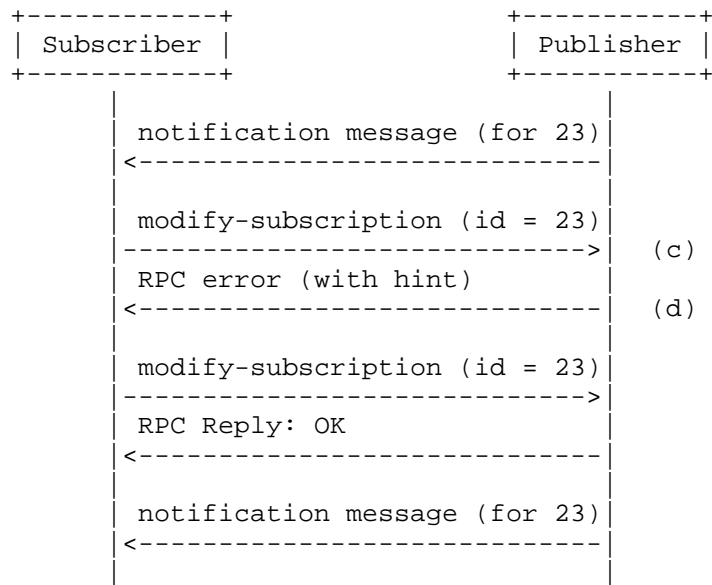


Figure 6: Interaction model for successful subscription modification

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">
    <identifier>23</identifier>
    <yp:datastore-xpath-filter xmlns="http://example.com/datastore">
      /interfaces-state/interface/oper-status
    </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">
    <identifier>22</identifier>
  </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 identifier, but that subscription identifier 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. Configured Subscriptions

Configured subscriptions may be established, modified, and deleted using configuration operations against the top-level subtree of [I-D.draft-ietf-netconf-subscribed-notifications] or [I-D.ietf-netconf-yang-push].

In this section, we present examples of how to manage the configuration subscriptions using a NETCONF client.

A.3.1. Creating Configured Subscriptions

For subscription creation, a NETCONF client may send:

```
<rpc message-id="201" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config>
      <subscriptions
        xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
        <subscription>
          <identifier>1922</identifier>
          <transport xmlns:nsn=
"urn:ietf:params:xml:ns:yang:ietf-netconf-subscribed-notifications">
            nsn:netconf
          <transport>
            <stream>NETCONF</stream>
          <receivers>
            <receiver>
              <name>receiver1</name>
            </receiver>
          </receivers>
        </subscription>
      </subscriptions>
    </config>
  </edit-config>
</rpc>
```

Figure 11: Create a configured subscription

If the request is accepted, the publisher will indicate this. If the request is not accepted because the publisher cannot serve it, no configuration is changed. In this case the publisher may reply:

```
<rpc-reply message-id="201"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rpc-error>
    <error-type>application</error-type>
    <error-tag>resource-denied</error-tag>
    <error-severity>error</error-severity>
    <error-message xml:lang="en">
      Temporarily the publisher cannot serve this
      subscription due to the current workload.
    </error-message>
  </rpc-error>
</rpc-reply>
```

Figure 12: Response to a failed configured subscription establishment

After a subscription has been created, NETCONF connectivity to each receiver will be established if it does not already exist. This will be accomplished through the association on the publisher with the networking parameters needed to establish connectivity with each configured receiver. These parameters will be used as needed by NETCONF call home [RFC8071].

The following figure shows the interaction model for the successful creation of a configured subscription.

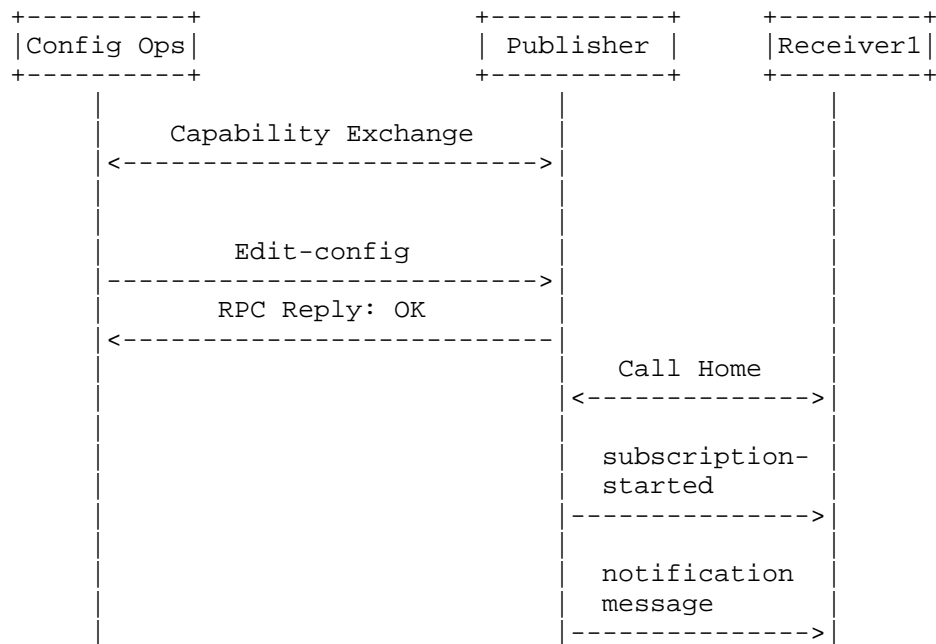


Figure 13: Interaction model for configured subscription establishment

A.3.2. Modifying Configured Subscriptions

Configured subscriptions can be modified using configuration operations against the top-level container `"/subscriptions"`.

For example, the subscription established in the previous section could be modified as follows, here a adding a second receiver:

```
<rpc message-id="202" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <subscriptions
        xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
        <subscription>
          <identifier>
            1922
          </identifier>
          <receivers>
            <receiver>
              <name>receiver2</name>
            </receiver>
          </receivers>
        </subscription>
      </subscriptions>
    </config>
  </edit-config>
</rpc>
```

Figure 14: Modify configured subscription

If the request is accepted, the publisher will indicate success. The result is that the interaction model described in Figure 13 may be extended as follows.

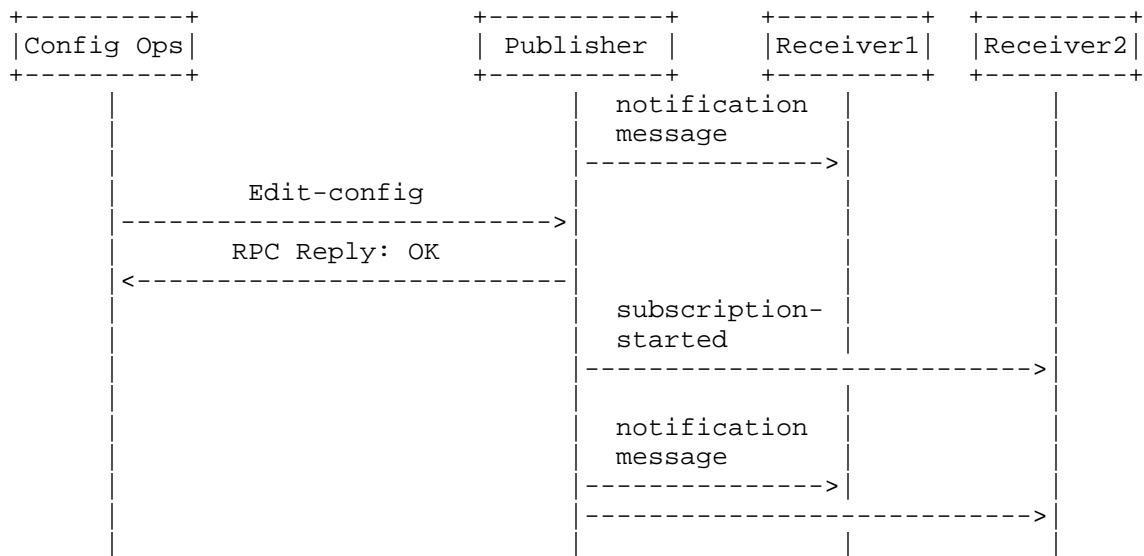


Figure 15: Interaction model for configured subscription modification

Note in the above that in the specific example above, modifying a configured subscription actually resulted in "subscription-started" notification. And because of an existing NETCONF session, no additional call home was needed. Also note that if the edit of the configuration had impacted the filter, a separate modify-subscription would have been required for the original receiver.

A.3.3. Deleting Configured Subscriptions

Configured subscriptions can be deleted using configuration operations against the top-level container `/subscriptions`. Deleting the subscription above would result in the following flow impacting all active receivers.

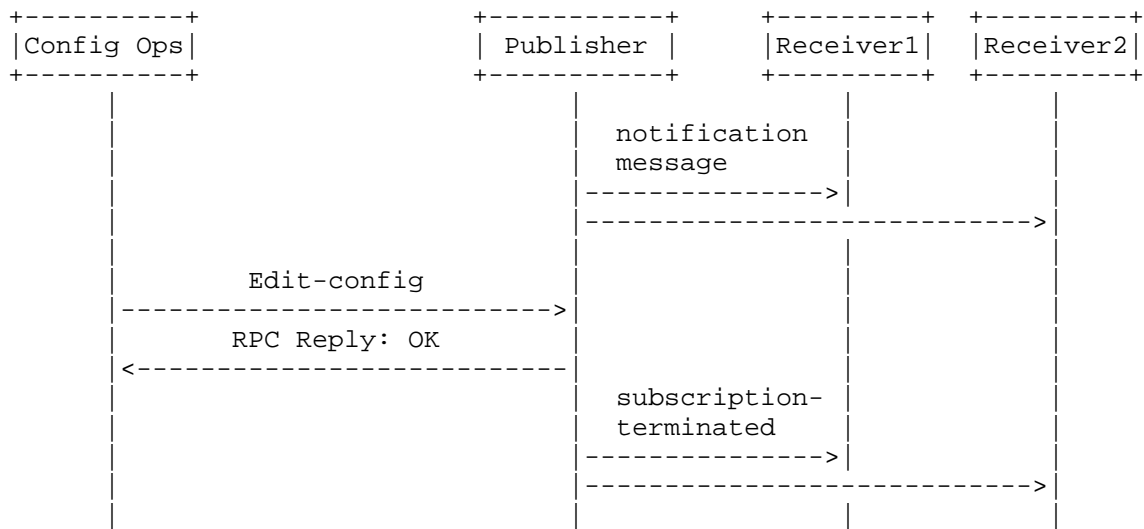


Figure 16: Interaction model for configured subscription deletion

A.4. Subscription State Notifications

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

A.4.1. subscription-started and subscription-modified

A "subscription-started" over NETCONF 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-started
    xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <identifier>39</identifier>
    <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-started>
</notification>
```

Figure 17: subscription-started subscription state notification

The "subscription-modified" is identical to Figure 17, with just the word "started" being replaced by "modified".

A.4.2. subscription-completed, subscription-resumed, and replay-complete

A "subscription-completed" would look like:

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

Figure 18: subscription-completed notification in XML

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

A.4.3. 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">
    <identifier>39</identifier>
    <error-id>
      suspension-timeout
    </error-id>
  </subscription-terminated>
</notification>
```

Figure 19: subscription-terminated subscription state notification

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

Appendix B. Changes between revisions

(To be removed by RFC editor prior to publication)

B.1. v09 to v10

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

B.2. v08 to v09

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

B.3. v07 to v08

- o Tweaks and clarification on :interleave.

B.4. v06 to v07

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

B.5. v05 to v06

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

B.6. v03 to v04

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

B.7. v01 to v03

- o Text simplifications throughout
- o v02 had no meaningful changes

B.8. v00 to v01

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

Authors' Addresses

Eric Voit
Cisco Systems

Email: evoit@cisco.com

Alexander Clemm
Huawei

Email: ludwig@clemm.org

Alberto Gonzalez Prieto
VMware

Email: agonzalezpri@vmware.com

Einar Nilsen-Nygaard
Cisco Systems

Email: einarnn@cisco.com

Ambika Prasad Tripathy
Cisco Systems

Email: ambtripa@cisco.com