Subscribing to YANG datastore push updates draft-netconf-yang-push-00

IETF #94 Yokohama

A. Clemm <alex@cisco.com>
A. Gonzalez Prieto <albertgo@cisco.com>
E. Voit <evoit@cisco.com>

Refresher

- Subscription model that allows applications to receive datastore updates
- Define update messages that can be pushed via Netconf
- Purpose:
  - Alternative to polling
  - Make updates easy to consume
  - Avoid inundating clients with unnecessary data they are not interested in
    - Filtering
    - On change
- Addresses requirements stated in ietf-i2rs-pub-sub-requirements
- Adopted as WG document after Prague

Updates from last draft version

- Support for creating subscriptions via configuration
  - In addition to RPCs
- Clarify on-change subscription updates
- Support reusable filter definitions
- Specify RPCs using YANG (RFC 5277 predated YANG)
- Various editorial improvements
  - More examples
  - New overview section for the data model

1) Subscriptions created via configuration

- Allows subscriptions to be established by modifying a server's configuration
- A receiver needs to be specified
  - The receiver and subscriber need not be the same entity
- Allows to persist subscriptions

1) Subscriptions created via configuration

```
module: ietf-datastore-push
+--rw subscription-config
     |  +--rw datastore-push-subscription* [subscription-id]
     |     +--rw subscription-id      subscription-id
     |     +--rw target-datastore?    datastore
     |     +--rw stream?              system-stream
     |     +--rw encoding?            encoding
     |     +--rw start-time?          yang:date-and-time
     |     +--rw stop-time?           yang:date-and-time
     |     +--rw (update-trigger)?
     |     |  +--:(periodic)
     |     |  |  +--rw period?             yang:timeticks
     |     |  +--:(on-change)
     |     |     +--rw no-synch-on-start?   empty
     |     |     +--rw dampening-period     yang:timeticks
     |     |     +--rw excluded-change*     change-type
     |     +--rw (filterspec)?
     |     |  +--:(inline)
     |     |  |  +--rw (filter-type)?
     |     |  |     +--:(subtree)
     |     |  |     |  +--rw subtree-filter?      subtree-filter
     |     |  |     +--:(xpath)
     |     |  |        +--rw xpath-filter?        yang:xpath1.0
     |     |  +--:(by-reference)
     |     |     +--rw filter-ref?          filter-ref
     |     +--rw receiver-address
     |        +--rw (push-base-transport)?
     |           +--:(tcpudp)
     |              +--rw tcpudp
     |                 +--rw address?   inet:host
     |                 +--rw port?      inet:port-number
```

Same parameters as RPC-based subscription

A receiver needs to be specified

1) Subscriptions created via configuration

```
+--rw receiver-address
        +--rw (push-base-transport)?
           +--:(tcpudp)
              +--rw tcpudp
                 +--rw address?   inet:host
                 +--rw port?      inet:port-number
```

2) On-change subscription updates

- **XML** encoding rules correspond to input in **edit-config** operations [RFC6241]
- "operation" attributes in the data subtree:
  - **create**: element has been added since the last update.
  - **delete**: element has been deleted since the last update.
  - **merge**: element has been changed since the last update.
  - **replace**: element has been replaced since the last update.
  - **move**: (To be added in the next version)
  - **insert**: (To be added in the next version)

2) On-change subscription updates

- **JSON** encoding rules roughly analogous to **YANG-patch** operation
  - **No edit-ids** needed

3) Filter re-use

- Users can **configure filters independently from subscriptions**
- Those filters can be referenced in a subscription
- Rationale: facilitate **reuse of filter definitions**
  - important in case of complex filter conditions
- **Custom data streams** which had a similar objective have been removed

3) Filter re-use

```
module: ietf-datastore-push
      +--rw filter* [filter-id]
         +--rw filter-id          filter-id
         +--rw (filter-type)?
            +--:(subtree)
            |  +--rw subtree-filter?   subtree-filter
            +--:(xpath)
               +--rw xpath-filter?     yang:xpath1.0
```

4) New subscription property: datastore

- When creating a subscription, subscriber can specify the target datastore
- Mandatory to support “running”, which is the default one
- Optional to support startup

5) Other Yang model changes

- Rpc’s formalized
  - {create|modify|delete}-subscription
- Notifications
  - Modified: push-update
    - added leaf time-of-update
    - added “choice encoding” w/ datastore contents
    - before, just datastore-contents
  - New: push-change-update, subscription-started

5) Other Yang model changes

- grouping subscription-info
  - **start-time modified** definition
    - It now matches the requirements document
    - “Designates the time at which a subscription is supposed to start, or immediately, in case the start-time is in the past.
    - For **periodic subscription**, the start time also serves as **anchor time** from which the time of the next update is computed. (...)";
  - Added **choice filterspec**: **“**inline” or “by-reference”

Next Steps

- Add operations (move, insert) for on-change for efficiency
- Extend config subscription so that it can support multiple receivers
- Review security input

Backup Slides

On-change JSON encoding

```
<?xml version="1.0" encoding="UTF-8"?>
    <notification
        xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
      <subscription-id xmlns="urn:ietf:params:xml:ns:netconf:
          datastore-push:1.0">
        my-on-change-sub
      </subscription-id>
      <eventTime>2015-10-13T12:13:02Z</eventTime>
      <datastore-changes-json
          xmlns="urn:ietf:params:xml:ns:netconf:datastore-push:1.0">
     { "ietf-yang-patch:yang-patch": {
        "patch-id": [ null],
        "edit": [
          {
            "edit-id": "edit1",
            "operation": "merge",
            "target": "/alpha/beta",
            "value": {
              "beta": 1500
     } } ] } }
    </datastore-changes-json>
  </notification>
```

RPC create-subscription

```
rpc create-subscription {
      description
        "This RPC allows a subscriber to create a subscription
        on its own behalf.  If successful, the subscription
        remains in effect for the duration of the subscriber's
        association with the publisher, or until the subscription
        is terminated by virtue of a delete-subscription request.";
      input {
        uses subscription-info;
      }
      output {
        leaf subscription-id {
          type subscription-id;
          description
            "Identifier used for this subscription.";
    } } }
```

RPC modify-subscription

```
rpc modify-subscription {
      description
        "This RPC allows a subscriber to modify a subscription
        that was previously created using create-subscription.
        If successful, the subscription
        remains in effect for the duration of the subscriber's
        association with the publisher, or until the subscription
        is terminated by virtue of a delete-subscription request.";
      input {
        leaf subscription-id {
          type subscription-id;
          description
            "Identifier to use for this subscription.";
        }
        uses subscription-info;
    } }
```

RPC delete-subscription

```
rpc delete-subscription {
      description
        "This RPC allows a subscriber to delete a subscription that
        was previously created using create-subscription.";
      input {
        leaf subscription-id {
          type subscription-id;
          description
            "Identifier of the subscription that is to be deleted.
            Only subscriptions that were created using
            create-subscription can be deleted via this RPC.";
 } } }
```

More Yang model changes

- New features: on-change, json
- Identities
  - Added subscription-deleted (base subscription-errors)
  - Added system-streams (and derived from it datastore-push, operational-push, config-push)
  - Added datastore (and derived from it running, startup
- Typedefs:
  - datastore-contents has been refined into 4 typedefs: datastore-{contents|change}-{xml|json}
  - New: filter-id, datastore, system-stream, filter-ref
- New data nodes: container system-streams