

Requirements for Subscription to YANG Datastores

draft-voit-i2rs-pub-sub-requirements-00

E. Voit

evoit@cisco.com

A. Clemm

alex@cisco.com

A. Gonzalez Prieto

alberto@cisco.com

Cisco Systems

draft-voit-i2rs-pub-sub-requirements

” Goal

- . Consolidate requirements for a "**pub/sub**" service for **YANG datastore** updates

” Why?

- . **Push-based** vs. Poll-based
- . I2RS needs more robust YANG object subscriptions

Context

- “ I2RS has requirements to interact with YANG datastores which extend beyond traditional configuration.
 - [i2rs-usecase](#)
 - [i2rs-arch](#)
- “ Existing YANG technology ecosystem is proving insufficient for those applications
 - relies on RPC-style interactions where data is configured or fetched on-demand by applications.
 - Netconf Event Notifications [RFC5277](#) is useful but does not follow the Pub/Sub paradigm. It also predates YANG.
 - Netconf Base Notifications [RFC6470](#) defines configuration change notifications, but is applicable for configuration information only.

Selected I2RS Requirements i2rs-usecase

- “ **L-Data-REQ-12:** The I2RS interface should support user subscriptions to data with the following parameters: push of data synchronously or **asynchronously via registered subscriptions...**
- “ **L-DATA-REQ-07:** The I2RS interface (protocol and IMs) should allow a subscribe to **select portions** of the data model.
- “ **PI-REQ01:** **monitor the available routes** installed in the RIB of each forwarding device, including near real time notification of route installation and removal.
- “ **BGP-REQ10:** I2RS client SHOULD be able instruct the I2RS agent(s) to notify the I2RS client when the BGP processes on an associated routing system **observe a route change to a specific set of IP Prefixes** and associated prefixes....The I2RS agent should be able to notify the client via publish or subscribe mechanism.
- “ **IGP-REQ-07:** The I2RS interface (protocol and IMs) should support a mechanism where the I2RS Clients can subscribe to the I2RS Agent's **notification of critical node IGP events.**
- “ **MPLS-LDP-REQ-03:** The I2RS Agent notifications should allow an I2RS client to subscribe to a **stream of state changes regarding the LDP sessions** or LDP LSPs from the I2RS Agent.
- “ **L-Data-REQ-01:** I2rs must be able to collect large data set from the network with **high frequency and resolution** with minimal impact to the device's CPU and memory.

Selected I2RS Requirements [i2rs-arch](#)

- “ **Section 6.4.2** identifies "subscribing to an information stream of route changes ... receiving notifications about peers coming up or going down"
- “ **Section 7.6** provides high level pub/sub (notification) guidance
 - needs to be able to receive notifications of changes in network state. Notifications here refers to changes which are unanticipated, represent events outside the control of the systems (such as interface failures on controlled devices)... a notification may also be due to a regular event.
 - I2RS Client needs to be able to register for just the events it is interested in. It is also possible that I2RS might provide a stream of notifications via a publish/subscribe mechanism that is not amenable to having the I2RS agent do the filtering.
- “ **Section 6.3** notes that when local config preempts I2RS, external notification might be necessary
- “ Additional references

draft-voit-i2rs-pub-sub-requirements

Abstract

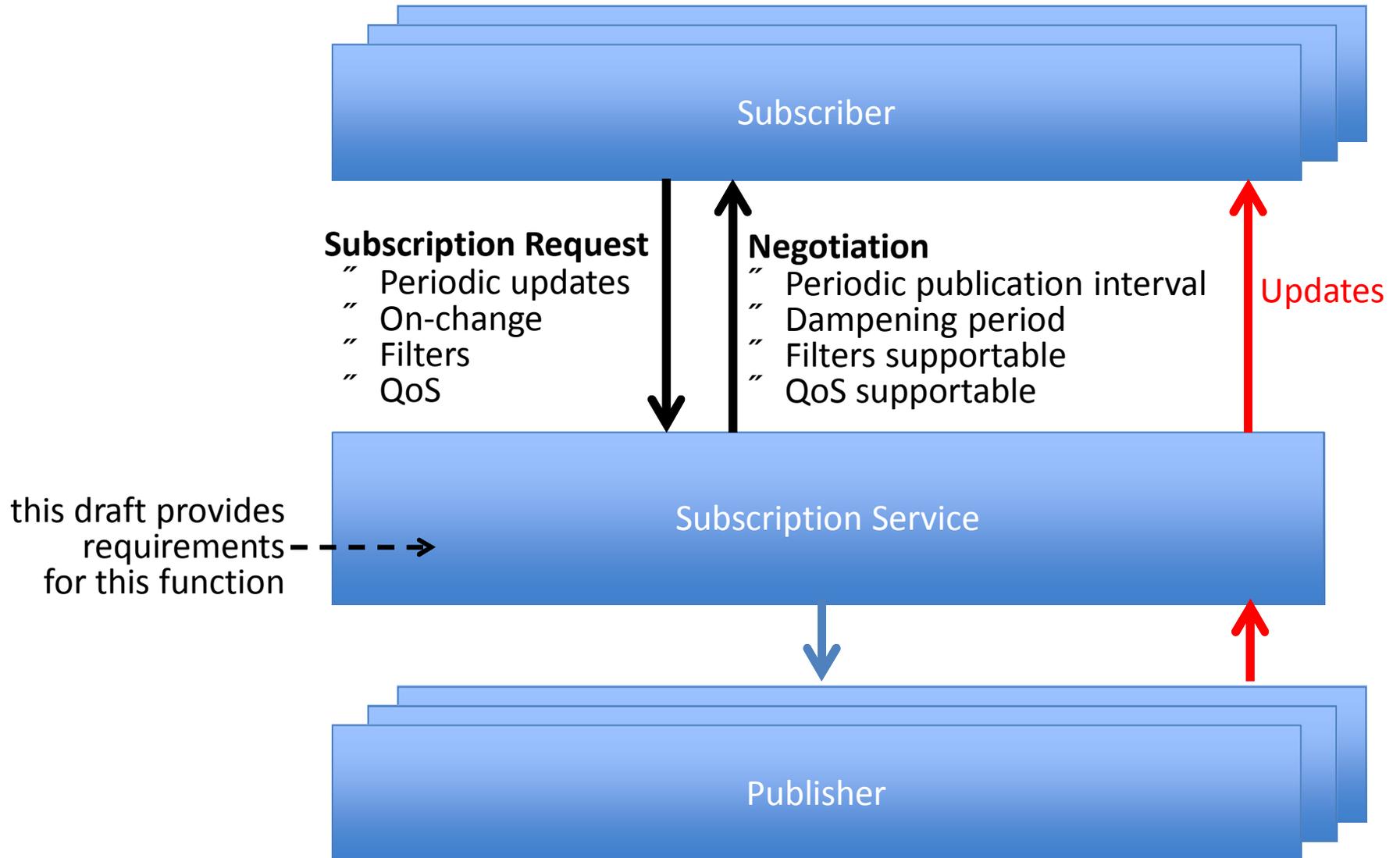
- “ ...allows client applications to subscribe to updates of a YANG datastore. Based on criteria negotiated as part of a subscription, updates will be pushed to targeted recipients. Such a capability eliminates the need for periodic polling of YANG datastores by applications and fills a functional gap in existing YANG transports.
- “ Beyond a set of basic requirements for the service, various refinements are addressed. These refinements include: periodicity of object updates, filtering out of objects underneath a requested a subtree, and delivery QoS guarantees.

draft-voit-i2rs-pub-sub-requirements

Terminology

- “ A **Subscriber** makes requests for set(s) of YANG object data. The Subscriber is the owner of the Subscription.
- “ A **Publisher** is responsible for distributing subscribed YANG object data per the terms of a Subscription. In general, a Publisher is the owner of the YANG datastore that is subjected to the Subscription.
- “ A **Subscription Service** provides Subscriptions to Subscribers of YANG data. A Subscription Service interacts with the Publisher of the YANG data as needed to provide the data per the terms of the Subscription.
- “ A **Subscription Request** for one or more YANG subtrees made by the Subscriber of a Publisher and targeted to a Receiver. A Subscription MAY include constraints which dictates how often or under what conditions YANG subtree updates might be sent.
- “ A **Subscription** is a contract between a Subscription Service and a Subscriber that stipulates the data to be pushed and the associated terms.

Pub/Sub Subscription Service



Subscription Service Requirements

“ Subscription Management

- . MUST support the ability to **create and to terminate a Subscription**
- . MUST be able to support and independently track one or more Subscription Requests by the same Subscriber
- . MUST be able to support an add/change/delete of one or more YANG subtrees as part of the same Subscription Request
- . SHOULD support the ability to **suspend and to resume a Subscription** on request of a client
- . MAY at its discretion **revoke or suspend an existing subscription**
- . MUST **send an indication** to the Subscriber when a Subscription undergoes a **state change**, i.e. when it is started, suspended, resumed, or terminated
- . MUST allow **Subscriptions to be monitored**

“ Types of Subscriptions Supported

- . MUST support the ability to **subscribe to periodic updates**
- . SHOULD support the ability to **subscribe to updates "on-change"**
- . SHOULD be able to interpret Subscription Requests **QoS Policy** requests

Subscription Service Requirements (2)

” Subscription Negotiation

- . MUST be able to negotiate the **terms of a Subscription** (interval, dampening period, policy, filters)
- . SHOULD be able to negotiate **QoS criteria** for a Subscription
- . (When a Subscription Request cannot be fulfilled) MUST include in its decline a set of criteria that would have been acceptable when the Subscription Request was made

” Update Distribution

- . For "**on-change**" updates, the Subscription Service MUST **only send deltas** to the object data for which a change occurred
- . For each object needs to include an indication whether it was removed, added, or changed
- . MUST **publish only data nodes that meet the filter criteria**

Subscription Service Requirements (3)

“ Transport

- . Starting point: **Netconf**

- “ Based on current I2RS requirements

- . Long term:

- “ Ensure YANG subscription mechanisms can be generalized to allow for additional transports (e.g., point to multipoint)

- . SHOULD support **different transports**
- . SHOULD support **different encodings of payload**

Subscription Service Requirements (4)

“ Security

- Mutual **authentication**
- **Versioning** MUST be supported
- data pushed MUST be **authorized** in the same way that regular data retrieval operations are
- MUST filter Subscriptions to suppress object updates where the Receiver has no read authorization
- A Subscription Service SHOULD decline a Subscription Request if it would deplete its resources

Subscribing to datastore push updates draft-netmod-clemm-datastore-push-00.txt

Alexander Clemm, Alberto Gonzalez Prieto, Eric Voit

Subscription Negotiation

- “ Leverage RFC 5277 <create-subscription>
- “ Server may reject a subscription request
 - . Implementation limitations (e.g. on-change)
 - . Resource limitations (e.g. update size, frequency)
- “ Response to include “acceptable” parameter settings (no guarantee)
- “ Additional notifications to indicate if server cannot keep “subscription promise)
- “ Optional: client throttling of subscription via suspend/resume

Selected discussion items

- “ <create-subscription> vs edit-config
- “ Additional methods for subscription throttling via suspend/resume

Subscription Data Model

```
module: ietf-datastore-push
  +--ro datastore-push-subscription
    +--ro stream string
    +--ro subscription-id subscription-identifier
    +--ro (filter)?
      | +--:(subtree)
      | | +--ro subtree-filter
      | +--:(xpath)
      |   +--ro xpath-filter yang:xpath1.0
    +--ro (notification-trigger)
      | +--:(periodic)
      | | +--ro period yang:timeticks
      | +--:(on-change)
      |   +--ro (change-policy)
      |     +--:(update-dampening) / (next revision)
      |     | +--ro period yang:timeticks
      |     +--:(delta-policy)
      |       +--ro delta uint32
    +--ro start-time? yang:date-and-time
    +--ro stop-time? yang:date-and-time
```

Selected discussion items

- “ Subscriptions created as result of <create-subscription>
- “ On-change subpolicies options or choices

Push Data Stream and Transport

- “ Push-update notifications
 - . Notification “push-update”
 - “ Subscription correlator
 - . Ties update to a specific subscription
 - “ Data node with datastore update
 - . Per subscription
 - . Filtered per NACM rules
 - . Other notifications in case notifications are suspended, resumed, or terminated
- “ Leverage <notification> element (per RFC 5277)
- “ Alternative transport mappings conceivable but outside scope

Q & A