

Subscription Drafts

IETF #102 - NETCONF WG

Eric Voit, Reshad Rahman & Alexander Clemm (who is in transit)

16-Jul-2018

With Thanks to...

Authors on at least 1 WG draft

Andy Bierman

Alexander Clemm

Tim Jenkins

Balazs Lengyel

Einar Nilsen-Nygaard

Alberto Gonzalez Prieto

Reshad Rahman

Ambika Prasad Tripathy

Eric Voit

+ DeSIGN Team 1

Sharon Chisholm

Yan Gang

Peipei Guo

Susan Hares

Michael Scharf

Hector Trevino

Kent Watsen

Guangying Zheng (Walker)

+ new with DeSIGN Team 2

Henk Birkholz

Igor Bryskin

Xufeng Liu

Tianran Zhou

NETCONF WG Subscription Drafts

Draft	
Custom Subscription to Event Streams	draft-ietf-netconf-subscribed-notifications
YANG Datastore Subscription	draft-ietf-netconf-yang-push
NETCONF Support for Event Notifications	draft-ietf-netconf-netconf-event-notifications
RESTCONF & HTTP Transport for Event Notifications	draft-ietf-netconf-restconf-notif
Notification Message Headers and Bundles	draft-ietf-netconf-notification-messages
UDP based Publication Channel for Streaming Telemetry	draft-ietf-netconf-udp-pub-channel
Subscription to Multiple Stream Originators	draft-zhou-netconf-multi-stream-originators
YANG PUSH Based Generalized Network Control Automation Problem Stmt.	draft-bryskin-netconf-automation-framework
Coap Transfer	draft-birkholz-yang-push-coap-problem-statement
Smart filters for Push Updates - Problem Statement	draft-clemm-netconf-push-smart-filters-ps
YangPush Notification Capabilities	draft-lengyel-netconf-notification-capabilities
Concise YANG Telemetry	draft-birkholz-yang-core-telemetry

WGLC

Pending

This session

Updated with WGLC so far...

- v11 to v14: based on comments
 - Receiver “address” removed (transport parameters restricted to transport drafts.)
 - Added “replay-previous-event-time” to “subscription-started” to simplify loss discovery.
 - Renamed the event counters
 - DSCP now an optional feature
 - Wording tweaks
- Open
 - Mechanism for replay for configured subscriptions between Boot & “subscription-started” (next slide)

Open - Replay for configured subscriptions

- Without replay, events are just sent once transport is available.
- There are classes of applications (e.g., [IMA](#)) which require visibility into all events placed into a event stream since boot.
- What is the visibility mechanism for configured subscriptions when there are meaningful events between Publisher boot & the “subscription-started” notification?
- Without a mechanism, only events created after transport is available are visible, which doesn’t meet the requirement for that class of applications.

Lets get WG feedback to hopefully close here...

Options - Replay for configured subscriptions

Option 1: Configured Replay

Current draft

The empty leaf “configured-replay” requires features {configured, replay}.
Events beginning with boot are placed at the front of the stream.

Option 2: Do not explicitly support replay for a configured subscription

To fill functional gap, each receiver needing prior stream info will create a dynamic replay subscription.

- With lots of receivers, this could result in an large number of temporary dynamic replay subscriptions coordinated to boot time.
- Delayed initial stream processing at receiver:
 - At “subscription-started” recognize missing events,
 - pause event processing and buffer incoming events,
 - request missing events via the dynamic subscription,
 - and insert them into the stream in the proper order.
- Receiver won't know when boot occurred, and therefore will subscribe to events pre-boot, and then interpret from the events themselves when boot occurred.
- Receiver must always support dynamic subscription.
 - This may be a new function needed for receivers where network loss is not an issue.
 - For Option 1, if it is not an issue, receiver RPCs can be locked-out (resulting in tighter security)

These functions already needed where you need to recover from packet loss. But often a higher quantity of events might be in play:

- Boot time often longer than network loss
- Event quantities at boot are high
- Dynamic subscription availability delay

PRO

CON

Note: Where there are independent receiver transport sessions for a subscription, these will be established at different times. And different initial events will go to each receiver.

draft-ietf-netconf-yang-push

Updated with WGLC so far...

- v16 to v17: based on review comments
 - Minor updates to text and YANG module

draft-ietf-netconf-netconf-event-notifications

Updated with WGLC so far...

- v09 - v10:
 - Wording updates per LC.
 - Tweaked examples based on subscribed-notification changes.
 - Proposed example YANG augmentation for NETCONF call home receiver to ietf-netconf-server.yang. This can be done subsequently to WGLC for either a bis or this document, or by placing it actually into ietf-netconf-server.yang
- Unresolved (next slide)
 - Do we progress only the dynamic subscription requirements through WGLC, and hold off on a –bis once ietf-netconf-server.yang is available for configured subscriptions. (A2 on next slide)

YANGPush Now thread: Three WGLC drafts: How do we close?

Hum A: Do we do progress Dynamic & Configured together

	Progression Option	Implication	
A1	Dynamic & configured together <ul style="list-style-type: none">• Current three drafts	<ul style="list-style-type: none">• Done	< Preferred
A2	Dynamic & configured together <ul style="list-style-type: none">• Current SN & YP• Update NETCONF-Notif so it just supports dynamic• Support configured via a -bis of NETCONF-Notif when ietf-netconf-server.yang completes	<ul style="list-style-type: none">• Minimal time delta	< Would be ok
A3	Configured after Dynamic Subscribed notifications & YANG Push	<ul style="list-style-type: none">• Refactoring YANG model and all drafts text places timeframe beyond business relevance• Open authorship	

YANGPush Now thread: Three WGLC drafts: How do we close?

Hum B: Do we do progress Subscribed Notifications & YANG Push together

	Progression Option	Implication	
B1	Subscribed Notifications & YANG Push together	<ul style="list-style-type: none">• Done• WG direction since adoption	< Preferred
B2	Subscribed Notifications, then YANG Push	<ul style="list-style-type: none">• No business driver	

Current status

- v04 to v06
 - Error mechanisms updated to match embedded RESTCONF mechanisms
 - Restructured format and sections of document.
 - Added a YANG data model for HTTP specific parameters.
 - Mirrored the examples from the NETCONF transport draft to allow easy comparison.
- Upcoming v07 changes
 - Model leafref updates to ietf-restconf-server.yang for call home.
- When last call?

draft-ietf-netconf-notification-messages

Updates since IETF #100

- No new version since v03.

- Awaiting completion of drafts in WGLC

Thank you!