

Subscribing to YANG datastore push updates
draft-ietf-netconf-yang-push-02
NETCONF WG
IETF #95 Buenos Aires
7-April-2015

Alexander Clemm
Alberto Gonzalez Prieto
Eric Voit
Ambika Prasad Tripathy
Einar Nilsen-Nygaard

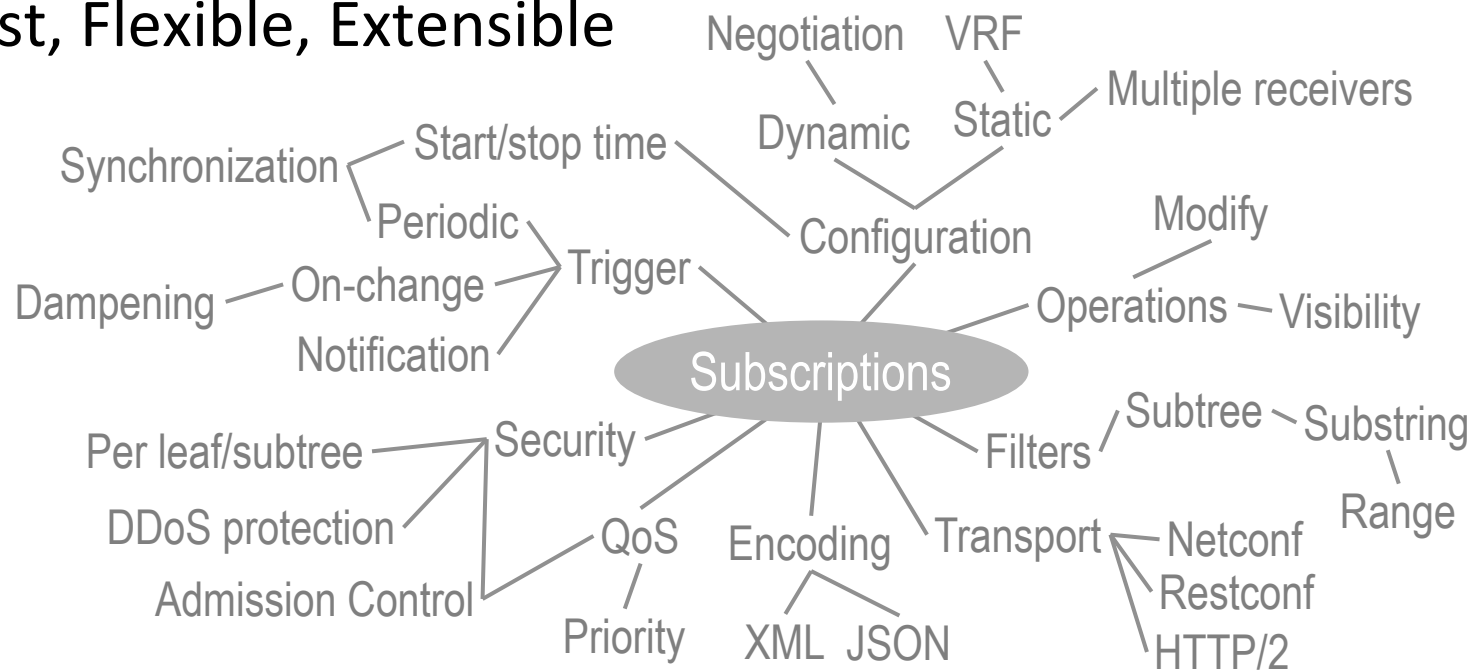
<alex|albertgo|evoit|ambripa|einarnn@cisco.com>

Updates since IETF #94

- Two revision updates (00→01→02)
- Establish-subscription replaces create-subscription
 - Use for negotiated subscription vs unnegotiated (RFC 5277)
- Updated RPC definitions for establish, modify, delete subscription
 - e.g. error return codes
- Added push source configuration for static subscriptions (e.g., VRF)
- Incorporated content from draft-voit-netconf-restconf-yang-push
 - Subscription prioritization, state machine, YANG
 - NOT transport (RESTCONF, HTTP, HTTP/2)
- Adopted capabilities from OC-Telemetry.yang
 - Multiple static receivers, DSCP
- Made associated YANG updates

Vision

- Common subscription control mechanisms
 - YANG-Push for YANG Datastore updates
 - Event Notifications for General Notifications
- Robust, Flexible, Extensible



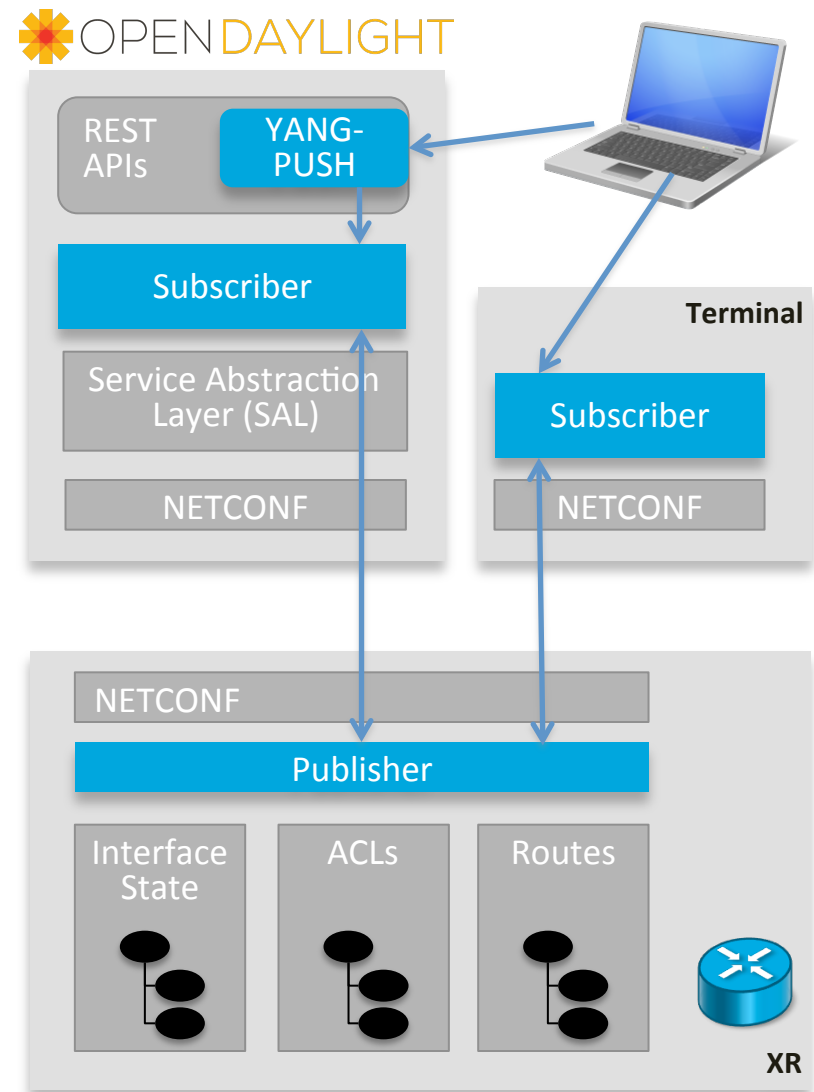
- Incrementally addressable components / drafts

NETCONF Event Notification & YANG Push Drafts - Key Elements

				draft-ietf-netconf-yang-push
		Event Notifications		YANG Push
		1.0	1.1	
Subscription	Types of Subscription	Dynamic	Dynamic and Static	
	Subscriptions per Session	one	many	
	Negotiation	No	Yes	
	RPCs	create	establish, modify, delete	
	Control Plane Notifications	None	started, suspended, resumed, terminated, modified	
Data Plane Notifications		notification	+subscription-id	push-update, push-change-update
Transport	NETCONF	Yes		
	RESTConf, HTTP, HTTP2	No	Yes	

IETF95 Hackathon – YANG Push

- YANG-PUSH Client from OpenDaylight Beryllium
- XR Router Demo Code
- Extended implementation: draft-ietf-netconf-yang-push
 - Create Subscription
 - Delete Subscription
 - On-Change
 - Periodic
- Hands-on Demo at tonight's Bits-N-Bytes



Model overview: streams + filters

```
module: ietf-yang-push
  +--ro update-streams
  |   +--ro update-stream*   update-stream
  +--rw filters
  |   +--rw filter* [filter-id]
  |       +--rw filter-id           filter-id
  |       +--rw (filter-type)?
  |           +--:(subtree)
  |               |   +--rw subtree-filter
  |               +--:(xpath)
  |                   |   +--rw xpath-filter?           yang:xpath1.0
  |                   +--:(rfc5277)
  |                       +--rw filter
```

Model overview:

Subscription configuration

```
+-rw subscription-config {configured-subscriptions}?
| +-rw yang-push-subscription* [subscription-id]
|   +-rw subscription-id          subscription-id
|   +-rw stream?                  update-stream
|   +-rw encoding?                encoding
|   +-rw subscription-start-time? yang:date-and-time
|   +-rw subscription-stop-time?  yang:date-and-time
|   +-rw (filterspec)?
|     |--:(inline)
|     | | +-rw (filter-type)?
|     | | | +-:(subtree)
|     | | | | +-rw subtree-filter
|     | | | | +-:(xpath)
|     | | | | | +-rw xpath-filter?          yang:xpath1.0
|     | | | | +-:(rfc5277)
|     | | | | +-rw filter
|     | | +-:(by-reference)
|     | +-rw filter-ref?                  filter-ref
```

Optional feature,
only applies to static
subscriptions

Model overview:

Subscription configuration (contd)

```
+-rw subscription-config {configured-subscriptions}?
| +-rw yang-push-subscription* [subscription-id]
...
|   +-rw (update-trigger)?
|   |   +--:(periodic)
|   |   |   +-rw period                               yang:timeticks
|   |   +--:(on-change) {on-change}?
|   |       +-rw no-synch-on-start?                 empty
|   |       +-rw dampening-period                   yang:timeticks
|   |       +-rw excluded-change*                   change-type
|   +-rw receiver* [address]
|   |   +-rw address      inet:host
|   |   +-rw port?        inet:port-number
|   |   +-rw protocol?    transport-protocol
|   +-rw (push-source)?
|   |   +--:(interface-originated)
|   |   |   +-rw source-interface?                   if:interface-ref
|   |   +--:(address-originated)
|   |       +-rw source-vrf?                         uint32
|   |       +-rw source-address                       inet:ip-address-no-zone
|   +-rw dscp?          inet:dscp {configured-subscriptions}?
|   +-rw subscription-priority?    uint8
|   +-rw subscription-dependency?  String
```


Model overview:

Subscription monitoring

```
+--ro subscriptions
  +--ro yang-push-subscription* [subscription-id]
    +--ro subscription-id          subscription-id
    +--ro configured-subscription?empty {configured-subscriptions}?
    +--ro subscription-status?     identityref
    +--ro stream?                  update-stream
    +--ro encoding?                encoding
    +--ro subscription-start-time? yang:date-and-time
    +--ro subscription-stop-time?  yang:date-and-time
    +--ro (filterspec)?
      | +--:(inline)
      |   ...
      | +--:(by-reference)
      |   +--ro filter-ref?          filter-ref +--ro (update-trigger)?
      |   +--:(periodic)
      |     | +--ro period            yang:timeticks
      |     +--:(on-change) {on-change}?
      |       +--ro no-synch-on-start? empty
      |       +--ro dampening-period  yang:timeticks
      |       +--ro excluded-change*  change-type
    +--ro receiver* [address]
      | +--ro address      inet:host
      | +--ro port?       inet:port-number
      | +--ro protocol?   transport-protocol
    +--ro (push-source)?
      | +--:(interface-originated)
      |   | +--ro source-interface?  if:interface-ref
      |   +--:(address-originated)
      |     +--ro source-vrf?        uint32
      |     +--ro source-address     inet:ip-address-no-zone
    +--ro dscp?              inet:dscp {configured-subscriptions}?
    +--ro subscription-priority? uint8
    +--ro subscription-dependency? string
```

- Applies to both dynamic and static subscriptions
- Mirrors subscription parameters from RPC or config

Note: items in blue are new or modified since -00

Model overview: RPCs (input)

```
+---x establish-subscription
| +---w input
| | +---w stream?                update-stream
| | +---w encoding?              encoding
| | +---w subscription-start-time? yang:date-and-time
| | +---w subscription-stop-time? yang:date-and-time
| | +---w (filterspec)?
| | | +---:(inline)
| | | | +---w (filter-type)?
| | | | | +---:(subtree)
| | | | | | +---w subtree-filter
| | | | | | +---:(xpath)
| | | | | | +---w xpath-filter?          yang:xpath1.0
| | | | | | +---:(rfc5277)
| | | | | | +---w filter
| | | | +---:(by-reference)
| | | | | +---w filter-ref?              filter-ref
| | +---w (update-trigger)?
| | | +---:(periodic)
| | | | +---w period                      yang:timeticks
| | | | +---:(on-change) {on-change}?
| | | | | +---w no-synch-on-start?        empty
| | | | | +---w dampening-period          yang:timeticks
| | | | | +---w excluded-change*          change-type
| | +---w dscp?                          inet:dscp {configured-subscriptions}?
| | +---w subscription-priority?          uint8
| | +---w subscription-dependency?        string
```

- Input parameters mirror those of static subscription, except for receiver specification
- Same underlying groupings

Model overview: RPCs (output)

```
+---x establish-subscription
|   +---ro output
|       +---ro subscription-result          subscription-result
|       +---ro (result)?
|           +---:(success)
|               | +---ro subscription-id          subscription-id
|               +---:(no-success)
|                   +---ro stream?              update-stream
|                   +---ro encoding?           encoding
|                   +---ro subscription-start-time? yang:date-and-time
|                   +---ro subscription-stop-time? yang:date-and-time
|                   +---ro (filterspec)?
|                       | +---:(inline)
|                       | | +---ro (filter-type)?
|                       | | | +---:(subtree)
|                       | | | | +---ro subtree-filter
|                       | | | | +---:(xpath)
|                       | | | | | +---ro xpath-filter?          yang:xpath1.0
|                       | | | | | +---:(rfc5277)
|                       | | | | | +---ro filter
|                       | | | | +---:(by-reference)
|                       | | | | +---ro filter-ref?              filter-ref
|                       +---ro (update-trigger)?
|                           | +---:(periodic)
|                           | | +---ro period                yang:timeticks
|                           | | +---:(on-change) {on-change}?
|                           | | | +---ro no-synch-on-start?    empty
|                           | | | +---ro dampening-period      yang:timeticks
|                           | | | +---ro excluded-change*      change-type
|                           +---ro dscp?          inet:dscp {configured-subscriptions}?
|                           +---ro subscription-priority?      uint8
|                           +---ro subscription-dependency?    string
```

- “No success” returns parameter settings that should be changed to lead to future success (subscription negotiation)

Model overview (contd.)

- Not depicted:
 - RPCs for modify, delete subscription
 - Notifications
 - Push updates
 - Push-update, push-change-update
 - Control notifications
 - Subscription started, stopped, suspended, resumed

Next Steps

- At this point, YANG-push is “self-sufficient” for YANG Subscriptions over NETCONF
- At the end of WG, discuss compartmentalizing functionality in different drafts:
 - Transport independence (NETCONF, RESTCONF, HTTP, IPFIX?, other?)
 - Common subscription model for Event Notifications and YANG Datastore Push
- Recommendation: split subscription mgmt and transports

Thank you!

NETCONF Event Notification & Datastore Push Drafts - Context

complementary

What you need	Consume a stream of Publisher generated messages at the cadence determined by the Publisher	Consume a stream of Publisher generated YANG data updates at a cadence negotiated with the Subscriber
Proposed	RFC5277 draft-gonzalez-netconf-5277bis	draft-ietf-netconf-yang-push draft-voit-netconf-restconf-yang-push