Notification Message Headers and Bundles

draft-voit-netconf-notification-messages-01

Eric Voit, Andy Bierman, Alexander Clemm, Tim Jenkins
### Functionality per Draft

<table>
<thead>
<tr>
<th><strong>Subscribed Notifications</strong></th>
<th><strong>YANG Datastore Push</strong></th>
<th><strong>YANG Notifications2</strong></th>
<th><strong>NETCONF Transport for Subscribed Notifications</strong></th>
<th><strong>RESTCONF &amp; HTTP2 Transport for Subscribed Notifications</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Dynamic &amp; Configured subscriptions</td>
<td>• Datastore on-change and periodic triggers</td>
<td>• Encapsulation Headers objects: Signature, de-duplication, severity, originator</td>
<td>• Transport mapping</td>
<td></td>
</tr>
<tr>
<td>• Multiple subscriptions / transport</td>
<td>• Filtering objects within a notification</td>
<td>• Bundled records and record types</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Multiple configured receivers</td>
<td>• Authorization model per object</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Establish, modify, delete, kill RPC</td>
<td>• Sending of full YANG trees or yang-patch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• State change notifications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Suspend/resume</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Filtering full notifications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Stream discovery</td>
<td>• Tagging of partial updates</td>
<td>• Transport mappings (including HTTP2 QoS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Replay (and start time negotiation)</td>
<td>• Tagging of on-change object support</td>
<td>• Heartbeats and clean-up</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Prioritization</td>
<td>• Negotiation of filters and period lengths</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Monitoring / reporting</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• QoS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Error responses</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- draft-ietf-netconf-yang-push
- draft-ietf-netconf-subscribed-notifications
- draft-yang-netmod-yang-notifications2
- draft-ietf-netconf-event-netconf
- draft-ietf-netconf-event-restconf

4
Purpose

• Create a way to “bundle” multiple update and notification records into a single notification message
• Allow to combine updates from multiple subscriptions into a single notification
• Distinguish between what goes into the “record”, and what goes into the “bundle”
  • Record contents distinguished from wrapper that it is contained in
  • Wrapper can be single notification message (one record), or bundled-notification message (multiple records)
#1 Transport Agnostic Header Objects

```plaintext
+---n notification-message
  +--ro notification-message-header
    | +--ro record-time
    | +--ro record-type?
    | +--ro subscription-id*
    | +--ro record-id?
    | +--ro observation-domain-id?
    | +--ro notification-id?
    | +--ro notification-time?
    | +--ro previous-notification-id?
    | +--ro dscp?
    | +--ro message-generator-id?
    | +--ro signature?
  +--ro receiver-record-contents?
```

To deal with the possibility of distributed subscriptions involving notifications/streams from multiple line cards.

Touch points w/ draft-zheng-netconf-udp-pub-channel-00
#2 bundling multiple notifications into a single transportable message

```
+---n bundled-notification-message
    +---ro bundled-notification-message-header
        +---ro notification-id?
        +---ro notification-time
        +---ro previous-notification-id?
        +---ro dscp?
        +---ro message-generator-id?
        +---ro signature?
        +---ro record-count?
    +---ro notification-records*
        +---ro notification-record-header
            +---ro record-time
            +---ro record-type?
            +---ro subscription-id*
            +---ro record-id?
            +---ro observation-domain-id?
        +---ro receiver-record-contents?
```
Next steps

• Refinements to data model
  • Some fields are overspecified and may not be needed → cleanup
e.g. various time stamps

• Synchronization/alignment with the concept of multiple message
generators for a subscription
  • Separation of concerns: decoupling bundling, common headers, and loss detection

• Ask for WG adoption