Transaction ID & Trace

2023-07-27
IETF 117
Jan Lindblad <jlinlbla@cisco.com>
Roque Gagliano <rogaglia@cisco.com>
Transaction ID

draft-ietf-netconf-transaction-id-01
draft-ietf-netconf-transaction-id-01 changes vs. -00

• YANG-Push mechanism updated to rely only on leafs, not XML attrs (makes it RESTCONF compatible)
• Added section on Txid in Datastore compare (RFC 9144) operations (also without XML attrs, for direct RESTCONF compatibility)
• Several clarifications (kudos to James Cumming and implementation team!)

• Fixed acl examples to include some YANG mandatory elements
Suggestion from Implementors

- Optional If-(un)modified-since functionality with Etag? "Etag+Seq"
  - Potential further savings
  - Somewhat more complex server implementation
  - Not available in RESTCONF today

- Add YANG extension to allow marking versioned nodes in YANG tree?

- Leverage RFC 7952 "Defining and Using Metadata with YANG"?
  - Considered early on, but then decided not to since this behavior does not match RESTCONF (RFC 8040)
Comparing Variants

Server chooses some nodes to be Versioned nodes

• Last-modified Txid mechanism
  • Server keeps timestamp for each versioned node

• Etag Txid mechanism
  • Server keeps opaque string for each versioned node

• Etag+Seq proposed Txid mechanism
  • As Etag, plus server keeps sequence of previous Txid:s

Then one leaf is changed, example on next page
Comparing Txid Variants

Client knowledge about relevant config subtree at time t

Client sends get-config/GET request, including txids

Client receives nodes from server
Next Steps

- Implementation experience
- Add "Etag+Seq", optional If-(un)modified-since functionality?
- Add YANG extension to allow marking versioned nodes in YANG tree?
- Further Implementation experience
Trace Context Extension

draft-rogaglia-netconf-trace-ctx-extension-03
draft-rogaglia-netconf-restconf-trace-ctx-headers-00
draft-rogaglia-netconf-trace-ctx-extension-03 changes vs. -02

• Developed section on error handling (sx:structure, example)
• Improve introduction per IETF 115 feedback
• Included IANA early feedback

draft-rogaglia-netconf-restconf-trace-ctx-headers-00 (new)

• Same objectives as above but for RESTCONF
• Simpler document as W3C already defines HTTP headers
Trace-ctx Definition Very Simple in RESTCONF

2. RESTCONF Extensions

A RESTCONF server SHOULD support trace context traceparent header as defined in [W3C-Trace-Context].

A RESTCONF server SHOULD support trace context tracestate header as defined in [W3C-Trace-Context].

2.1. Error handling

The RESTCONF server SHOULD follow [the W3C specs and trace-ctx-extension-03] ...

2.2. Trace Context header versioning

... [I-D.draft-rogaglia-netconf-trace-ctx-extension-03] defines a pair YANG modules that SHOULD be included in the YANG library per [RFC8525]
Next Steps

- More details around error handling
- Publish RESTCONF document “should” implement W3C headers
- Figure out process for W3C protocol registration
- WG adoption of both trace drafts?
- Implementation experience
- Monitor W3C “baggage headers” development (still W3C draft state)
The end goal is to tap into OTLP ecosystem
Example with ServiceNow/LightStep backend

Orchestrator

Controller

<edit-config>
trace-context

OTLP Backend

Orchestrator trace spans

Controller trace spans

Same trace-id
Thank you
# Overview of Related Drafts

<table>
<thead>
<tr>
<th>Category</th>
<th>Draft Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANS-ID</td>
<td>draft-ietf-netconf-transaction-id-01</td>
</tr>
<tr>
<td>CFG-TRACE</td>
<td>draft-quilbeuf-opsawg-configuration-tracing-02</td>
</tr>
<tr>
<td>W3C-TRACE</td>
<td>draft-rogaglia-netconf-trace-ctx-extension-03</td>
</tr>
<tr>
<td>W3C-HEADERS</td>
<td>draft-rogaglia-netconf-restconf-trace-ctx-headers-00</td>
</tr>
<tr>
<td>PRIV-CAND</td>
<td>draft-jgc-netconf-privcand-02</td>
</tr>
<tr>
<td>ETAGS</td>
<td>RFC 8040 (RESTCONF)</td>
</tr>
</tbody>
</table>
# Massive Use Case Overlaps

* = Not in recent versions

<table>
<thead>
<tr>
<th>Use Case</th>
<th>TRANS-ID</th>
<th>CFG-TRACE</th>
<th>W3C-TRACE</th>
<th>PRIV-CAND</th>
<th>ETAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase transaction throughput by reducing lock time</td>
<td>✔</td>
<td>-</td>
<td>-</td>
<td>✔</td>
<td>-</td>
</tr>
<tr>
<td>Allow clients to get config changes at top level or within subtree (&quot;Sync&quot;)</td>
<td>✔</td>
<td>Assumed in other doc</td>
<td>-</td>
<td>-</td>
<td>✔</td>
</tr>
<tr>
<td>Allow clients to make config changes conditional on no conflicts (&quot;No overwrite&quot;)</td>
<td>✔</td>
<td>Assumed in other doc</td>
<td>-</td>
<td>✓</td>
<td>✔</td>
</tr>
<tr>
<td>Allow clients to recognize their own echo in YANG Push updates</td>
<td>✔</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Map transaction ids from client to server and server controlled children</td>
<td>*</td>
<td>✔</td>
<td>✔</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Finding Source of configuration mistakes</td>
<td>*</td>
<td>✔</td>
<td>✔</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Detecting conflicting intents</td>
<td>*</td>
<td>✔</td>
<td>✔</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Provisioning root cause analysis</td>
<td>*</td>
<td>✔</td>
<td>✔</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>System performance profiling</td>
<td>*</td>
<td>✔</td>
<td>✔</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Billing and auditing</td>
<td>*</td>
<td>✔</td>
<td>✔</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Direction Going Forward (IMHO)

Create framework of separate documents
• Each one optional to implement, optional to use
• Use cases, terminology and behavior aligned

CFG-TRACE defines use cases and terminology