Transaction ID & Trace

2023-07-27

IETF 117

Jan Lindblad <jlindbla@cisco.com>

Roque Gagliano <rogaglia@cisco.com>

Transaction ID

draft-ieft-netconf-transaction-id-01

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

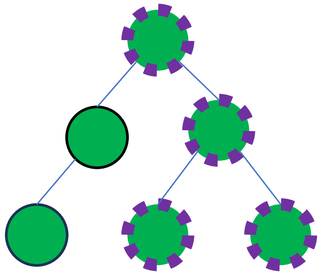
- YANG-Push mechanism updated to rely only on leafs, not XML attrs (makes it RESTCONF compatible)
- Added section on Txid in Datastore compare (RFC 9144) operations (also without XML attrs, for direct RESTCONF compatibility)
- Several clarifications (kudos to James Cumming and implementation team!)

- Fixed acl examples to include some YANG mandatory elements

Suggestion from Implementors

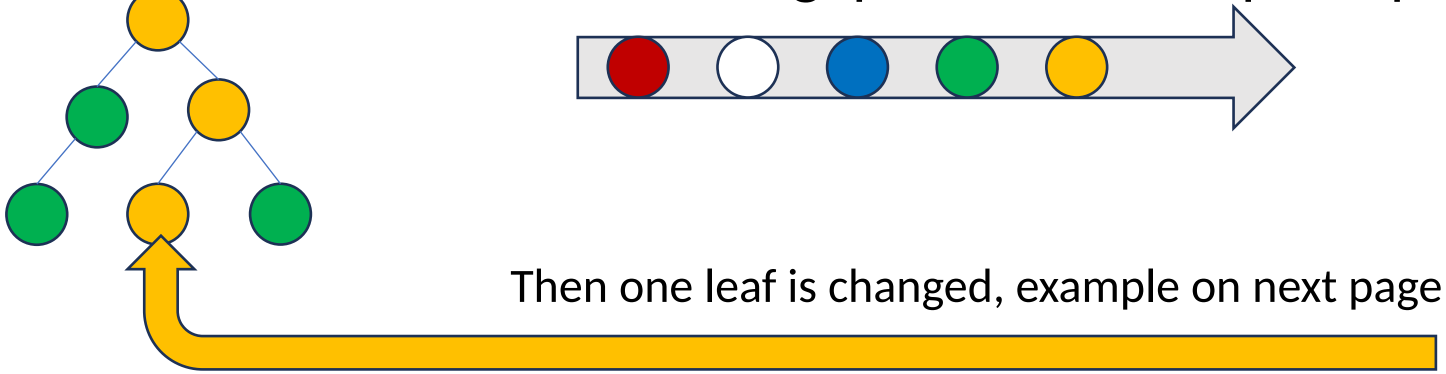
- Optional If-(un)modified-since functionality with Etag? "Etag+Seq"
  - Potential further savings
  - Somewhat more complex server implementation
  - Not available in RESTCONF today
- Add YANG extension to allow marking versioned nodes in YANG tree?
- Leverage RFC 7952 "Defining and Using Metadata with YANG"?
  - Considered early on, but then decided not to since this behavior does not match RESTCONF (RFC 8040)

Comparing Variants



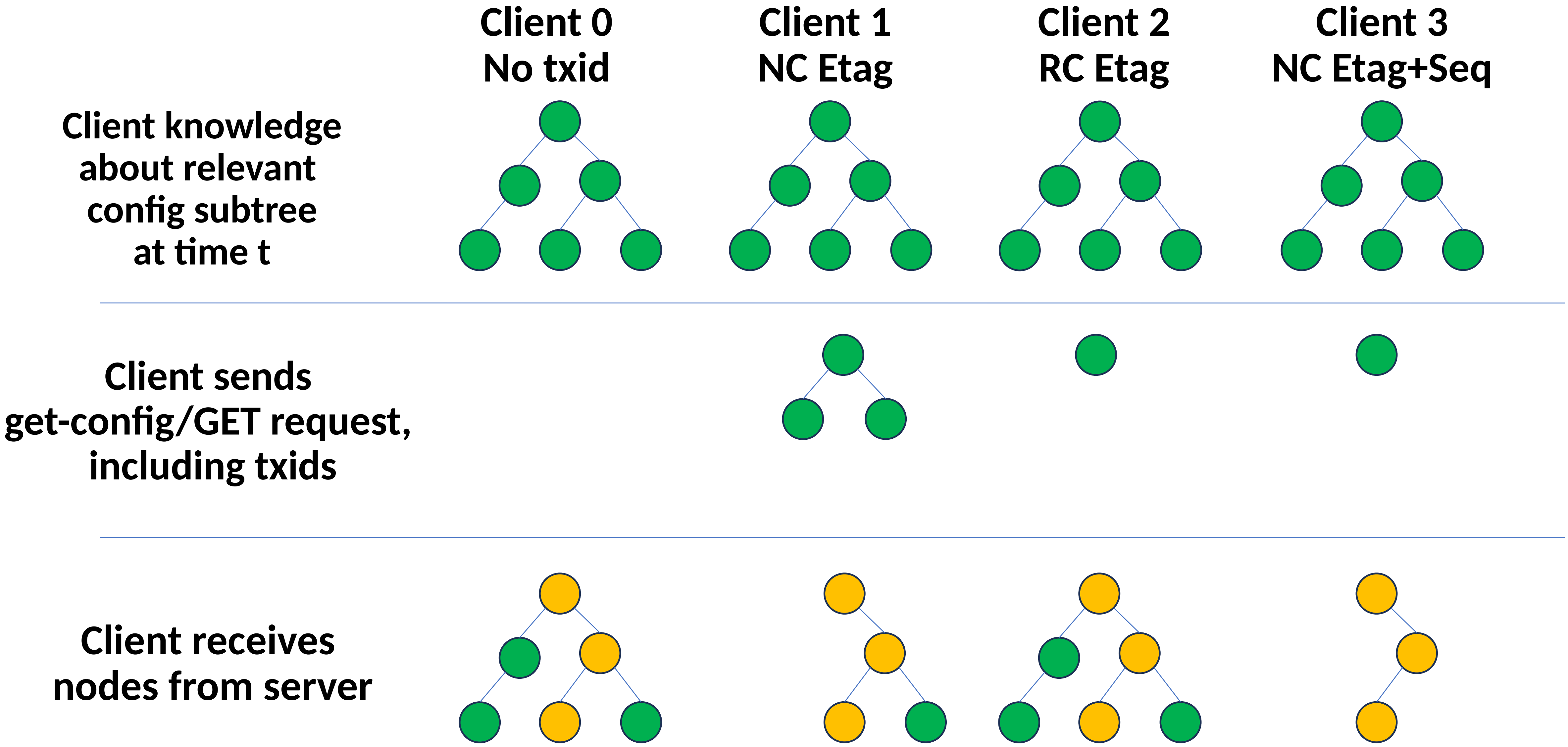
Server chooses some nodes to be Versioned nodes

- 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 0
No txid

Client 1
NC Etag

Client 2
RC Etag

Client 3
NC Etag+Seq

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] …

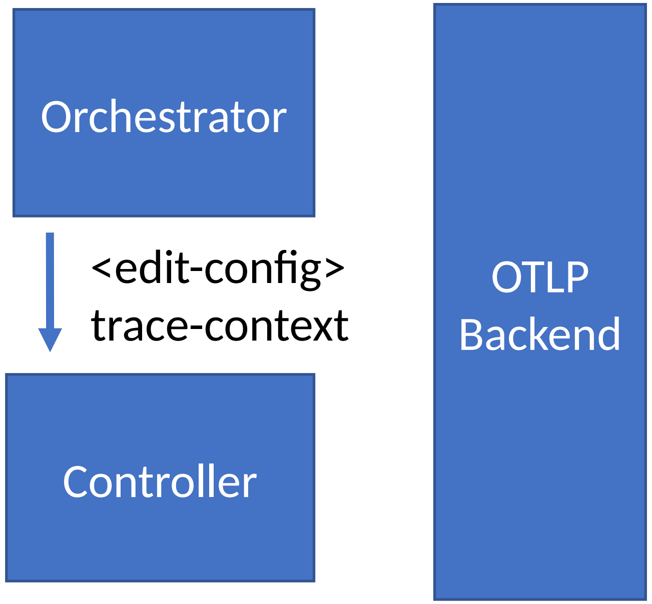
2.2. Trace Context header versioning

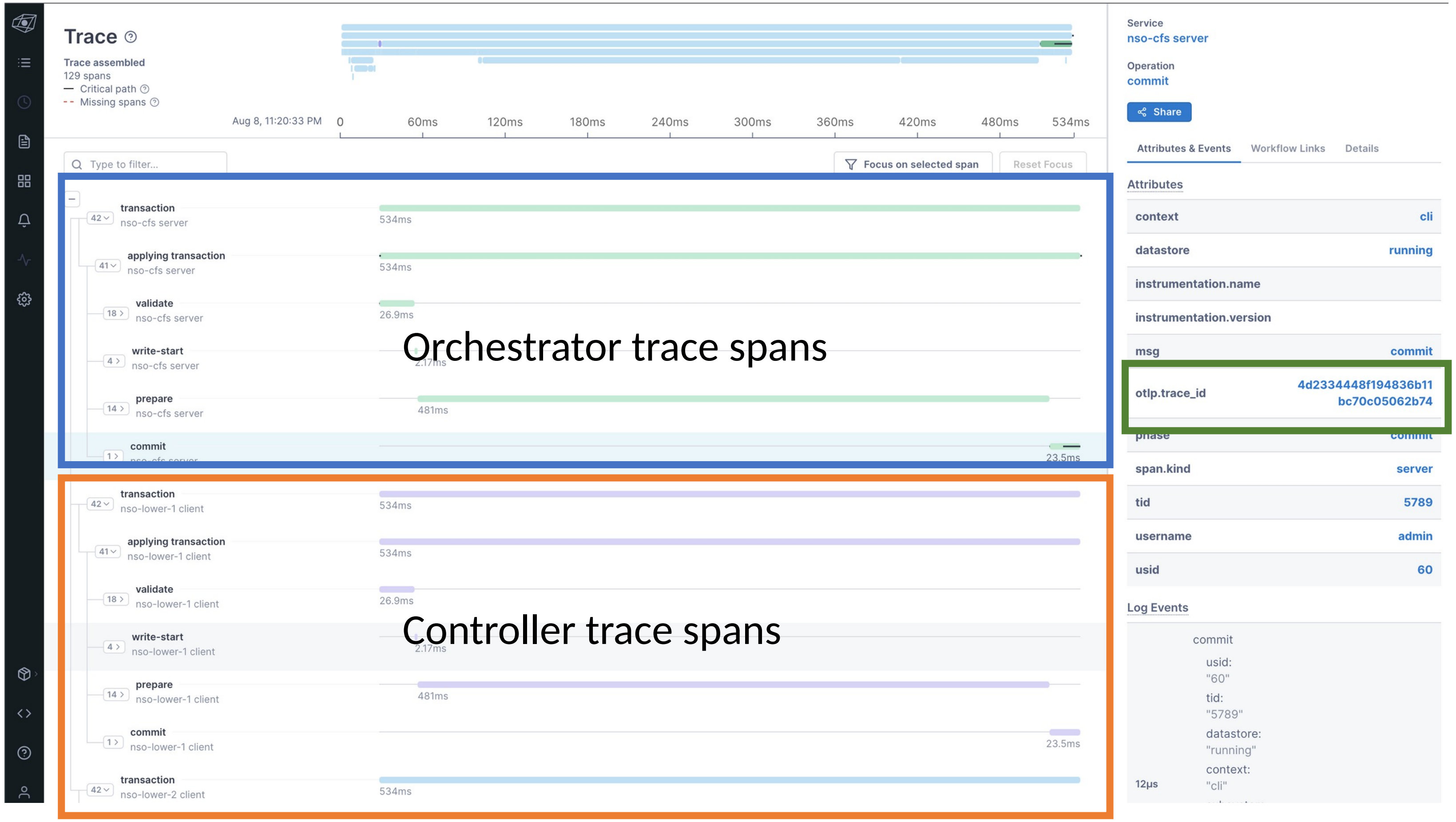
… [I-D.draft-rogaglia-netconf-trace-ctx-extension-03] defines a pair YANG modules that SHOULD be included in the YANG library per [RFC8525]

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




Trace

Trace assembled
129 spans
— Critical path
-- Missing spans

Aug 8, 11:20:33 PM

0 60ms 120ms 180ms 240ms 300ms 360ms 420ms 480ms 534ms

Type to filter...

Focus on selected span

Reset Focus

Orchestrator trace spans

```
42  transaction            nso-cfs server      534ms
41  applying transaction   nso-cfs server      534ms
18  validate               nso-cfs server      26.9ms
4   write-start            nso-cfs server      2.17ms
14  prepare                nso-cfs server      481ms
1   commit                 nso-cfs server      23.5ms
```

Controller trace spans

```
42  transaction            nso-lower-1 client  534ms
41  applying transaction   nso-lower-1 client  534ms
18  validate               nso-lower-1 client  26.9ms
4   write-start            nso-lower-1 client  2.17ms
14  prepare                nso-lower-1 client  481ms
1   commit                 nso-lower-1 client  23.5ms
42  transaction            nso-lower-2 client  534ms
```

Service
nso-cfs server

Operation
commit

Share

Attributes & Events | Workflow Links | Details

Attributes

context	cli
datastore	running
instrumentation.name	
instrumentation.version	
msg	commit
otlp.trace_id	4d2334448f194836b11 bc70c05062b74
phase	commit
span.kind	server
tid	5789
username	admin
usid	60

Log Events

```
      commit
        usid:
        "60"
        tid:
        "5789"
        datastore:
        "running"
        context:
12µs    "cli"
```

Same trace-id

Thank you

Overview of Related Drafts

TRANS-ID	draft-ietf-netconf-transaction-id-**01**
CFG-TRACE	draft-quilbeuf-opsawg-configuration-tracing-**02**
W3C-TRACE	draft-rogaglia-netconf-trace-ctx-extension-**03**
W3C-HEADERS	**draft-rogaglia-netconf-restconf-trace-ctx-headers-**00
PRIV-CAND	draft-jgc-netconf-privcand-**02**
ETAGS	RFC 8040 (RESTCONF)

Massive Use Case Overlaps

* = Not in recent versions

	TRANS-ID	CFG-TRACE	W3C-TRACE W3C-HEADERS	PRIV-CAND	ETAGS
Increase transaction throughput by reducing lock time	✔	-	-	✔	-
Allow clients to get config changes at top level or within subtree ("Sync")	✔	Assumed in other doc	-	-	✔
Allow clients to make config changes conditional on no conflicts ("No overwrite")	✔	Assumed in other doc	-	✔	✔
Allow clients to recognize their own echo in YANG Push updates	✔	-	-	-	-
Map transaction ids from client to server and server controlled children	*	✔	✔	-	-
Finding Source of configuration mistakes	*	✔	✔	-	-
Detecting conflicting intents	*	✔	✔	✔	-
Provisioning root cause analysis	*	✔	✔	-	-
System performance profiling	*	✔	✔	-	-
Billing and auditing	*	✔	✔	-	-

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