

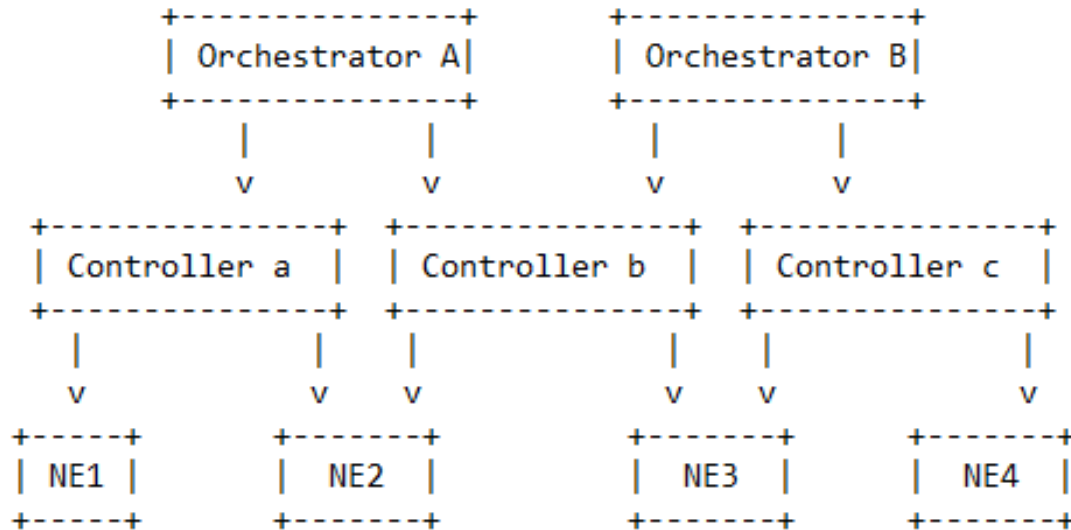
External Transaction ID for Configuration Tracing

[draft-quilbeuf-opsawg-configuration-tracing-00](#)

J. Quilbeuf (Huawei) , B. Claise (Huawei) , T. Graf
(Swisscom), D. Lopez (Telefonica) , S. Qiong (China
Telecom)

IETF 115, NETCONF & OPSAWG

Motivation: Tracing Configuration



Something wrong with last change of configuration on NE2.

Where does the change comes from?

Use cases:

- **Configuration Mistake** *“Which service request, if any, caused the mistake?”*
- **Concurrent NMS modification** *“Both NMS assume that they are in charge of the NE and regularly overwrite each other configuration.”*
- **Conflicting Intents** *“Conflicting configuration changes are cause by two conflicting service requests.”*

Existing: Saving configuration information

- Usually devices, controllers and orchestrators have an history of configuration changes:
 - With a single id (commit-id, checkpoint, label, ...), called local-commit-id here
 - With configuration metadata: timestamp, user, protocol, ... used to configure
 - With the changes themselves (diff of CLI configuration, of XML)

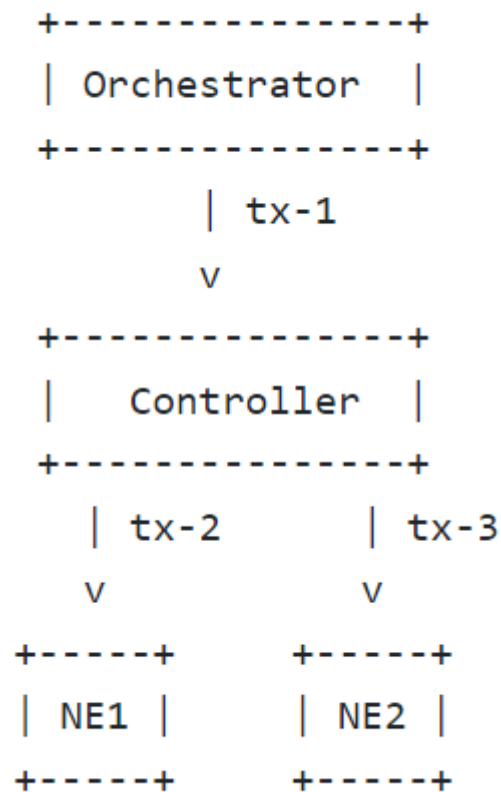
We assume that the “local-commit-id” of the bad configuration can be found based on device specific information.

Existing: Transaction ID in NETCONF

- From [draft-lindblad-netconf-transaction-id-02](#)
- Idea: common id (transaction id or tx-id) between the client and server, assigned to subtree of configuration. The goal is to speed up synchronization by skipping subtree for which the transaction id did not change since last synchronization.

Exploitation in our case: for a given pair of NETCONF client/server, the transaction id matches a configuration change on the server with the corresponding change on the client.

Mapping transaction-ids to commit-ids



For NETCONF clients, configuration of NETCONF service is a **southbound** transaction. Here tx-1 is a southbound transaction for the Orchestrator.

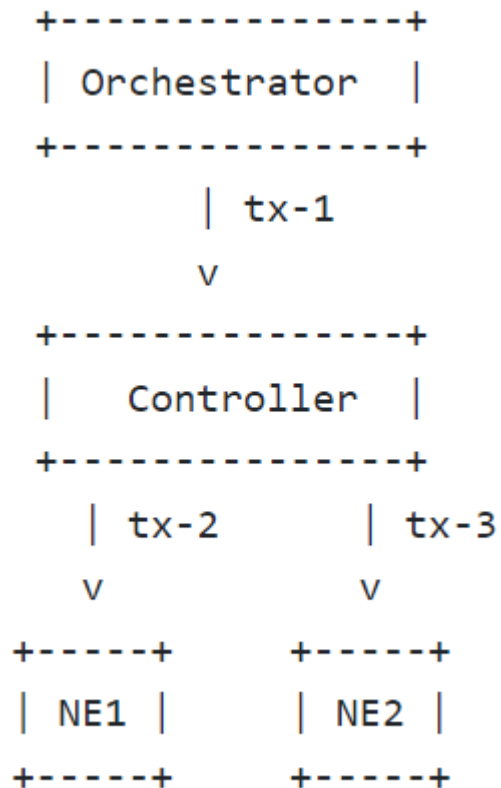
For NETCONF servers, receiving configuration from a NETCONF client is a **northbound** transaction.

Here tx-1 is a northbound transaction for the Controller.

If a local configuration change is caused by a northbound transaction, we map the local commit-id to the northbound transaction id. We also record the ID of the client in that case.

If a local configuration change triggers some southbound transactions, we map the local commit-id to the southbound transaction ids.

YANG Module



```
module: ietf-external-transaction-id
  +--ro external-transactions-id
    +--ro configuration-change* [local-commit-id]
      +--ro local-commit-id          string
      +--ro northbound-transaction-id? ietf-netconf-txid:etag-t
      +--ro northbound-client-id      string
      +--ro southbound-transaction-id* ietf-netconf-txid:etag-t
```

Both northbound and southbound transaction ids can be mapped to the same local-commit-id.

For example in the Controller, the same local-commit-id will be mapped to:

- tx-1 as the northbound-transaction-id (northbound-client-id: id of Orchestrator)
- tx-2 and tx-3 as the southbound-transaction-ids

Open questions/items to consider

- Feature from draft-lindblad-netconf-transaction-id-00 removed in -01: possibility for the client to set the transaction id when sending the configuration to the server. Would simplify this draft, any objection to adding that feature?
- Risk of collision between southbound transaction-ids from different southbound elements
- Is NETCONF the right scope? Should we include RESTCONF as well? Other configuration protocols?
- Link with draft-rogaglia-netconf-trace-ctx-extension-00

Conclusion

- Thanks for reviews (Med!)
- Is the problem of mapping configuration changes in NEs to a service request a valid problem?
- Is our solution a good approach?

Draft repo is at <https://github.com/JeanQuilbeufHuawei/draft-quilbeuf-opsawg-configuration-tracing>