NETCONF
Simultaneous Requests

Nemanja Stamenic [Siemens AG]
IEC/IEEE 60802 Ad Hoc Call 17.12.2021
Single Domain – Multiple CUCs
CUC – UNI - CNC dynamics

1. Stream Request
2. NETCONF client request
3. NETCONF protocol message
4. UNI-RPC call (e.g. add_stream)
5. Datastore update notification
6. Datastore update (stream conf)
7. UNI-RPC response
8. NETCONF protocol message
9. NETCONF client response
10. Stream confirmation
CNC Southbound IF

11. NETCONF client request
12. NETCONF protocol message
13. RPC e.g. <edit config>
14. Candidate DS update
15. Datastore commit
16. Configuration by management entity
Multiple CNCs within one IA-Station

Domain 1

- CUC 1.1
- CUC 1.2
- ... CUC 1.N_i

- Security
- HMI / Diagnostics

Domain 2

- CUC 2.1
- CUC 2.2
- ... CUC 2.N_i

- Security
- HMI / Diagnostics

Domain 3

- CUC 3.1
- CUC 3.2
- ... CUC 3.N_i

- Security
- HMI / Diagnostics

Chassis

Management Entity

NMDA NACM

Domain 1 IF
Domain 2 IF
Domain 3 IF

NETCONF Server Interface:
- \( N_1 + N_2 + N_3 + 6 \) Read/Write Clients
- Access control is required
NETCONF Server Interface:
- 9 Read/Write Clients
- Access control is required
Problem Statement – NETCONF Limitations

- Multiple clients will access (R/W) NETCONF datastores simultaneously
- Up to 30 clients access the NETCONF server implementing CNC northbound and southbound interfaces
- Serial access (global lock) is not acceptable due to unpredictable time behavior
- Working directly on <running> data store without consistency is not acceptable
- Partial-lock (RFC 5717) is possible only on the <running> data store, not on <candidate>
- There is no standard NETCONF mechanism to support these use cases
Further Steps – NETCONF Limitations

• The problem was presented at the YANGSTERS meeting on 28.09.2021
• NETCONF experts from IETF were present
• Statement: standardization effort at IETF is needed to support IA use cases
• Suggested technical solution: NETCONF Transactions
  [https://www.ietf.org/archive/id/draft-lhotka-netconf-restconf-transactions-00.txt](https://www.ietf.org/archive/id/draft-lhotka-netconf-restconf-transactions-00.txt)
• Draft expired in December 2018
NETCONF Transactions

• Support of the <candidate> datastore is required

• A new configuration datastore named <staging> is introduced

• It represents a staging area private to each user, it is subject to access control

• Essentially, each client gets its own non-shared <candidate> data-store

• Each client does not have to get a separate copy since efficient implementation methods exist (persistent data, copy-on-write – up to the implementors to decide)

• Commit operation automatically merges the content of the client’s data store into <intended>. Merge conflicts result in an „operation failed – merge conflict“ response
UNI Limitations

- Access control of the stream list on per TSN domain and on per client basis is necessary
- ieee802-dot1q-tns-config-uni.yang proposes a „flat“ stream list - Streams[1..N]
- A hierarchical data model would fit better for the required access control.
  - Domain[1..n]
  - Client[1..k] -> NACM (write access limited to “this” client)
  - Stream [1...l]
- Root access to the stream list only via RPC
UNI Further Steps

• When can the next PQdj draft be expected?

• Current state of UNI (IEEE PQdj) does not cover all IA use cases
  • Hierarchical stream list
  • RPCs to avoid polling
  • Notifications
  • Stream ID management
  • CNC and CUC requirements and informative text
  • …

• Work on a textual contribution is in progress, the goal is to have it integrated in the draft
THANK YOU!
QUESTIONS?