

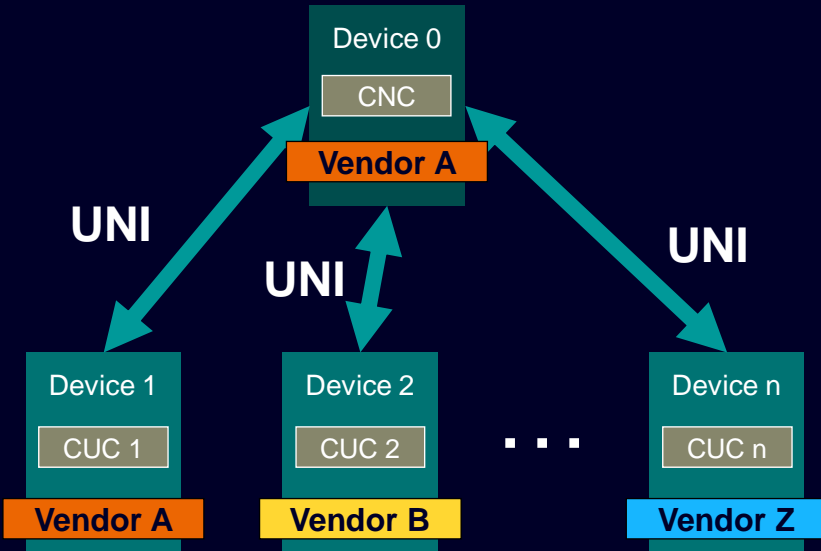
YANG module ieee802-dot1dj-tsn-config-uni

Rodrigo Ferreira Coelho [Siemens AG]

Contributors:

Günter Steindl [Siemens AG]

Motivation



Multivendor interoperability, i.e. behavior of UNI and its exchanged data, is defined by the YANG module `ieee802-dot1dj-tsn-config-uni`

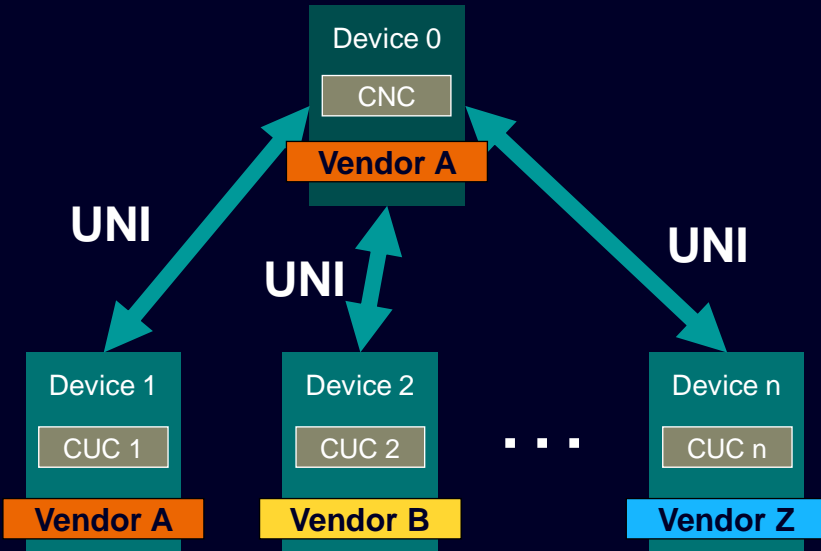
This contribution details multiple **ballot comments** and their respective **proposed changes** regarding the YANG module `ieee802-dot1dj-tsn-config-uni`

Agenda

- Ensuring assignment of **unique StreamIDs** and **unique stream dst-MAC address**
- **ieee802-dot1dj-tsn-config-uni** YANG module
 - **Access of CUC** to containers **via RPCs**
 - Account for unique StreamID assignment
 - Account for unique dst-MAC address assignment
 - **Multiple streams** per request
 - AddStreams **with vs. without scheduling** semantics
 - AddStreams, ScheduleStreams, AddAndScheduleStreams
 - JoinStream (\Leftrightarrow AddListeners)
 - RemoveStreams, RemoveListeners

Ensuring assignment of unique StreamIDs in a TSN Domain

Motivation: Operation modes in IEC/IEEE 60802



According to IEC/IEEE 60802, three operation modes possible

1. Offline engineered

1. StreamIDs assigned offline, no further change allowed

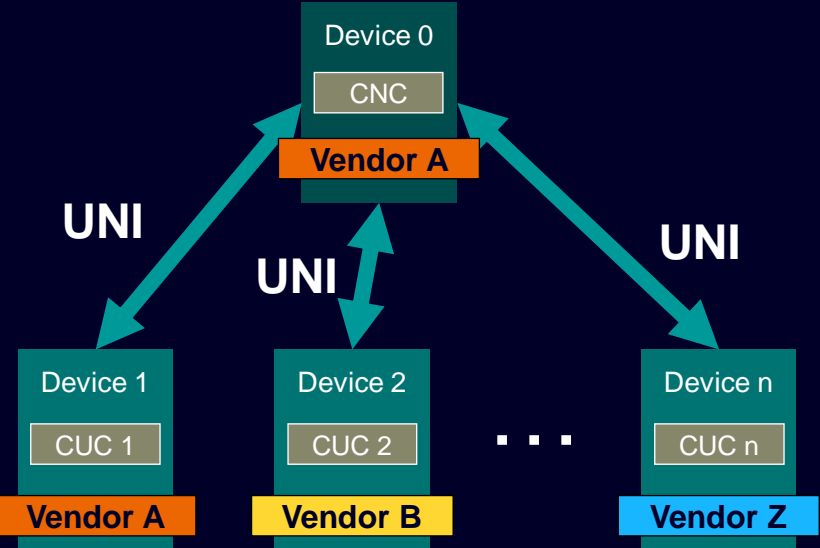
2. Dynamic topology

1. StreamIDs assigned online
2. Requests sent per CUCs (CUCs independent of each other)

3. Offline engineered & dynamic topology

1. Some StreamIDs assigned offline, some requested by CUCs (CUCs independent of each other)

Current Qcc/ Qdj approach to assign StreamIDs



Qcc/ Qdj approach

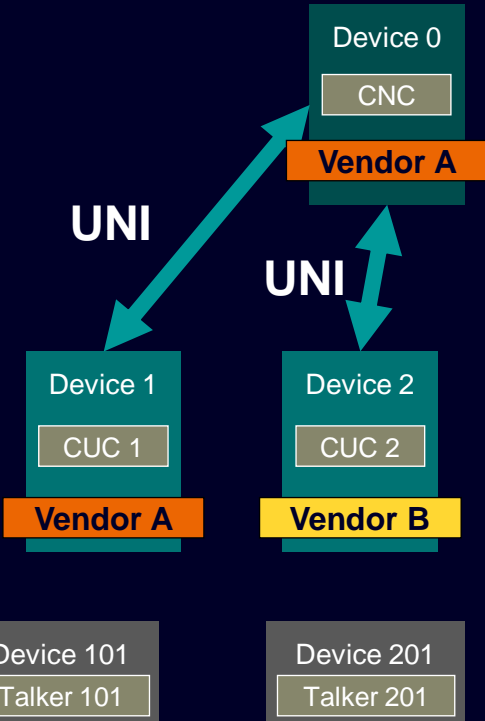
- StreamIDs assigned by CUCs when requesting a stream

Table 46-1—StreamID elements

Name	Data type	Reference
MacAddress	mac-address-type	46.2.3.1.1
UniqueID	uint16	46.2.3.1.2

- MacAddress
 - Talker MAC address
 - Pool of MAC addresses managed by CUC
- UniqueID
 - To distinguish streams within end station (i.e. same MAC address)

Issues with current Qcc/ Qdj approach for IEC/IEEE 60802 operation modes



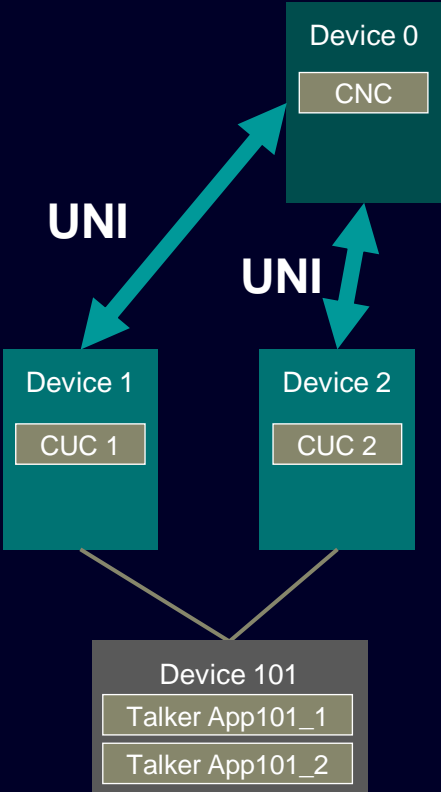
Issue a)

In operation mode 3: **Offline & dynamic topology**

1. How to ensure MACAddr used by engineering tool will not be reused by other CUCs (example CUC1 or CUC2)?
 1. Vendor independent



Issues with current Qcc/ Qdj approach for IEC/IEEE 60802 operation modes



Issue b)

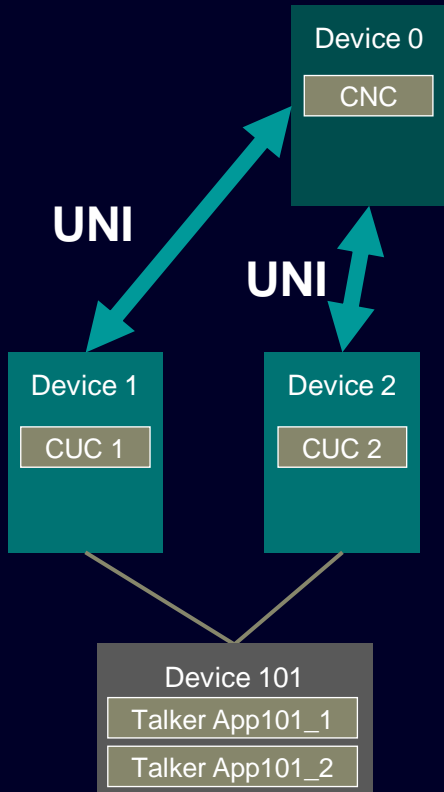
In operation mode 2, 3: (Offline &) Dynamic topology

Example:

- Talker MAC address used for StreamID
- 1 stream requested by CUC1. Talker := Device101

StreamID	Talker(Dev)	Talker(app)	CUC
MACDev101:01	Dev101	Tapp101_1	CUC1

Issues with current Qcc/ Qdj approach for IEC/IEEE 60802 operation modes



Issue b)

In operation mode 2, 3: (Offline &) Dynamic topology

Example:

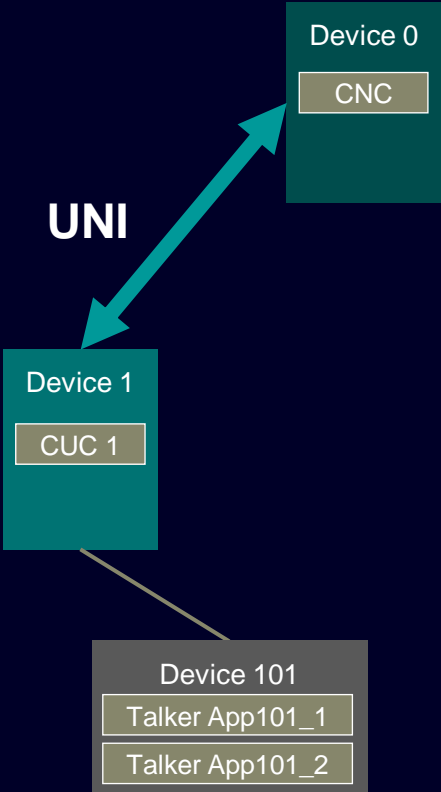
- Talker MAC address used for StreamID
- 1 stream requested by CUC1. Talker := Device101
- 1 stream requested by CUC2. Talker := Device101

StreamID	Talker(Dev)	Talker(app)	CUC
MACDev101:01	Dev101	Tapp101_1	CUC1
MACDev101:01	Dev101	Tapp101_2	CUC2



Duplicated StreamID assignment

Issues with current Qcc/ Qdj approach for IEC/IEEE 60802 operation modes



Issue c)

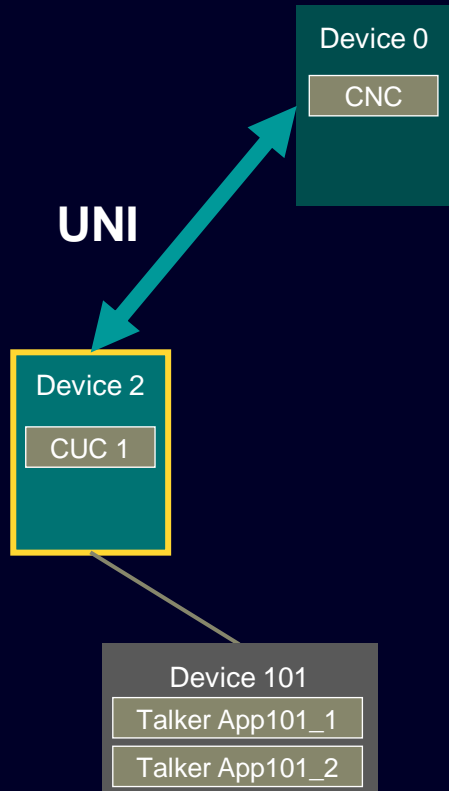
In operation mode 2, 3: (Offline &) Dynamic topology

Example:

- CUC pool MAC address used for StreamID
1. Stream 1 requested by CUC1. TalkerApp :=Tapp101_1

StreamID	CUC(Dev)	CUC(app)	Talker(app)
MACDev1:01	Dev1	CUC1	Tapp101_1

Issues with current Qcc/ Qdj approach for IEC/IEEE 60802 operation modes



Issue c)

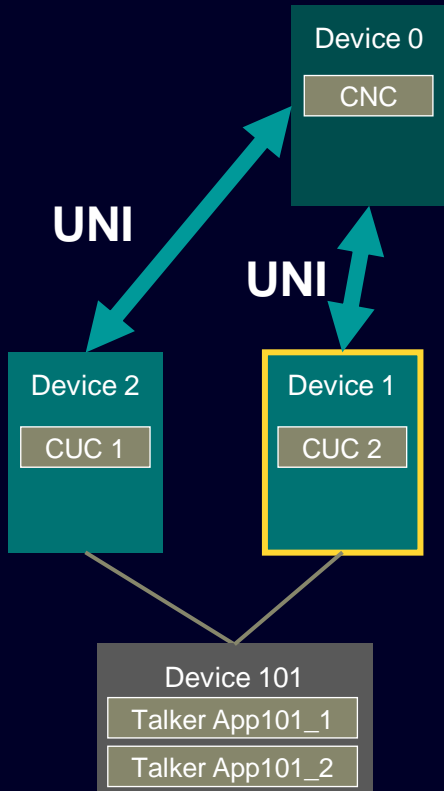
In operation mode 2, 3: (Offline &) Dynamic topology

Example:

- CUC pool MAC address used for StreamID
1. Stream 1 requested by CUC1. TalkerApp :=Tapp101_1
 2. Device 2 replaces Device 1, i.e. Dev2 hosts CUC1

StreamID	CUC(Dev)	CUC(app)	Talker(app)
MACDev1:01	Dev1	CUC1	Tapp101_1

Issues with current Qcc/ Qdj approach for IEC/IEEE 60802 operation modes



Issue c)

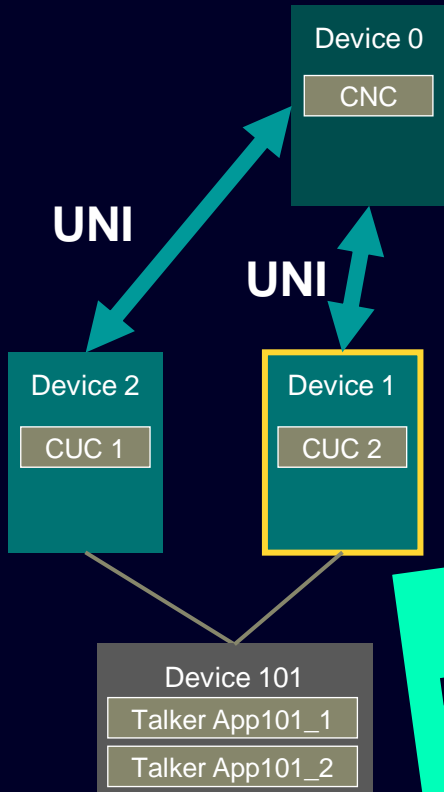
In operation mode 2, 3: (Offline &) Dynamic topology

Example:

- CUC pool MAC address used for StreamID
1. Stream 1 requested by CUC1. TalkerApp :=Tapp101_1
 2. Device 2 replaces Device 1, i.e. Dev2 hosts CUC1
 3. Device 1 comes back to host CUC2

StreamID	CUC(Dev)	CUC(app)	Talker(app)
MACDev1:01	Dev1	CUC1	Tapp101_1

Issues with current Qcc/ Qdj approach for IEC/IEEE 60802 operation modes



There are possibly more use cases where this problem occurs

Issue c)

In operation mode 2, 3: (Offline &) Dynamic topology

Example:

- CUC pool MAC address used for StreamID
1. Stream 1 requested by CUC1. TalkerApp :=Tapp101_1
 2. Device 2 replaces Device 1, i.e. Dev2 hosts CUC1
 3. Device 1 comes back to host CUC2
 4. Stream 2 requested by CUC2. TalkerApp :=Tapp101_2

StreamID	CUC(Dev)	CUC(app)	Talker(app)
MACDev1:01	Dev1	CUC1	Tapp101_1
MACDev1:01	Dev1	CUC2	Tapp101_2

! Duplicated StreamID assignment

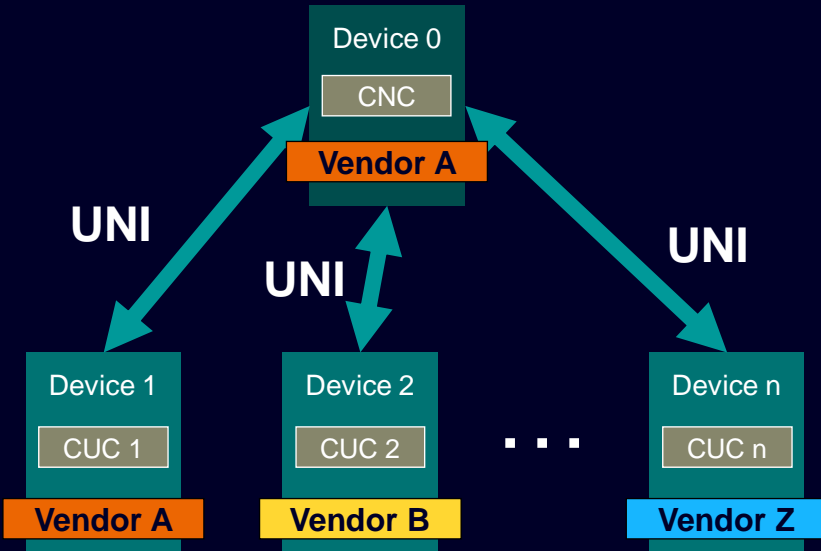
How to ensure unique StreamID assignment in a TSN Domain?

StreamIDs must be
centrally managed, i.e.
assigned by the CNC

Ensuring assignment of unique stream dst-MAC address to streams in a TSN Domain

Motivation: Operation modes in IEC/IEEE 60802

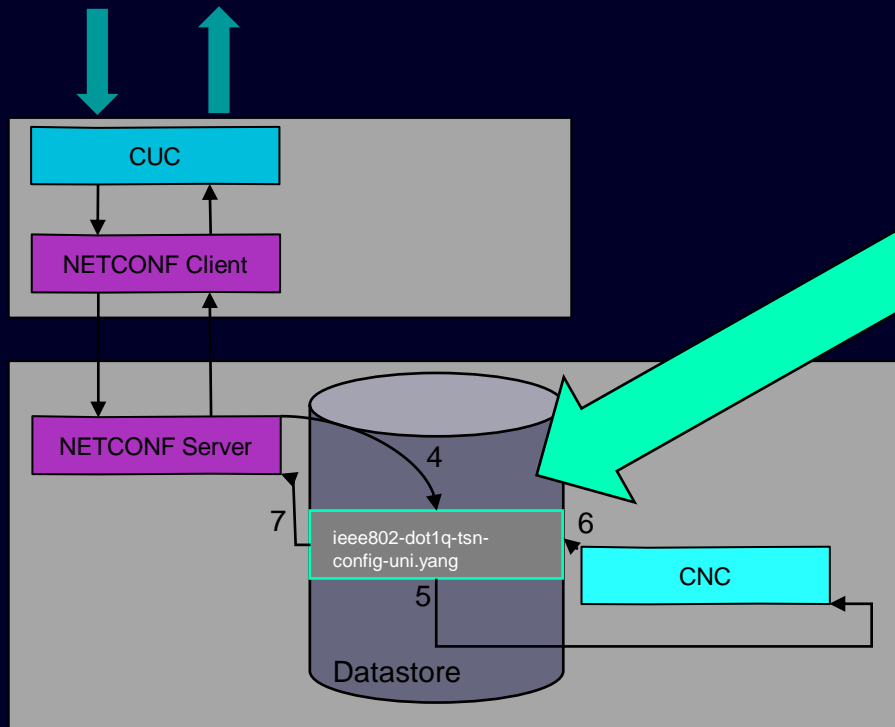
Similar to StreamID assignment, local management of stream dst-MAC, i.e. by CUCs, cannot ensure uniqueness



Stream dst-MAC
addresses must be
centrally managed, i.e.
assigned by the CNC

**Requesting a stream using
ieee802-dot1dj-tsn-config-uni
without pre-defining a StreamID?
(i.e. CNC assigns StreamIDs)**

YANG module overview



Extracted from 60802-Stamenic-NETCONF-Simultaneous-Requests-12-21-v01.pdf

ieee802-dot1q-tsn-config-uni

- YANG Containers
- YANG Remote Procedure Calls (RPCs)
- YANG Notifications
- Not addressed in this contribution

YANG Containers overview

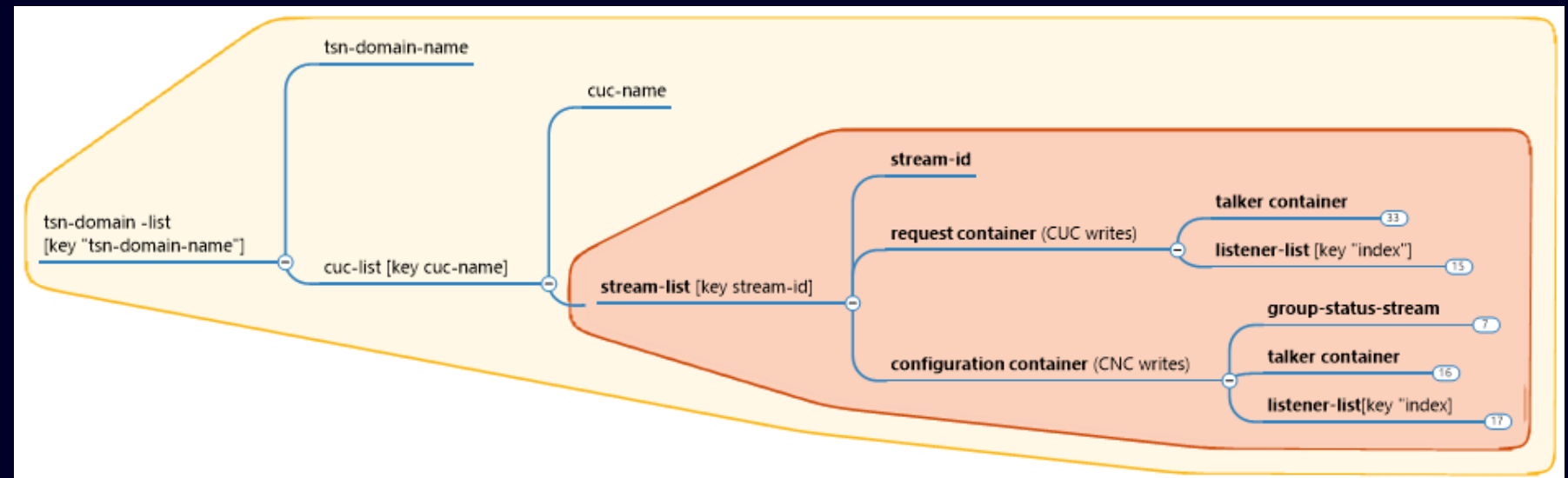
“A hierarchical data model would fit better for the required access control.

- Domain[1..n]
- Client[1..k] -> NACM (write access limited to “this” client)
 - Stream [1...l]

Hierarchy level for TSN Domain and CUC

- Not scope of this contribution

Extracted from 60802-Stamenic-NETCONF-Simultaneous-Requests-12-21-v01.pdf



YANG RPCs

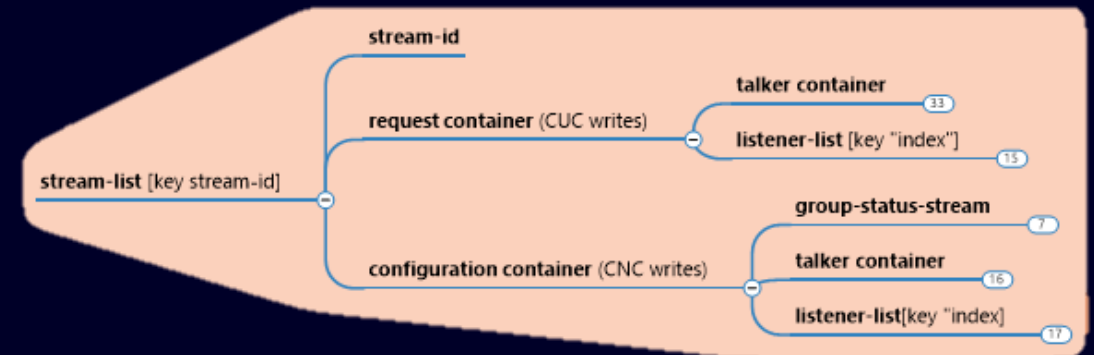
“A hierarchical data model would fit better for the required access control.

- 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”

Extracted from 60802-Stamenic-NETCONF-Simultaneous-Requests-12-21-v01.pdf

Access to stream list *only* via YANG RPCs

- Facilitates Network Configuration Access Control Model (NACM)
- Allows for **decoupling stream request from datastore structure** when needed



Stream requests: AddStreams

RPC AddStreams

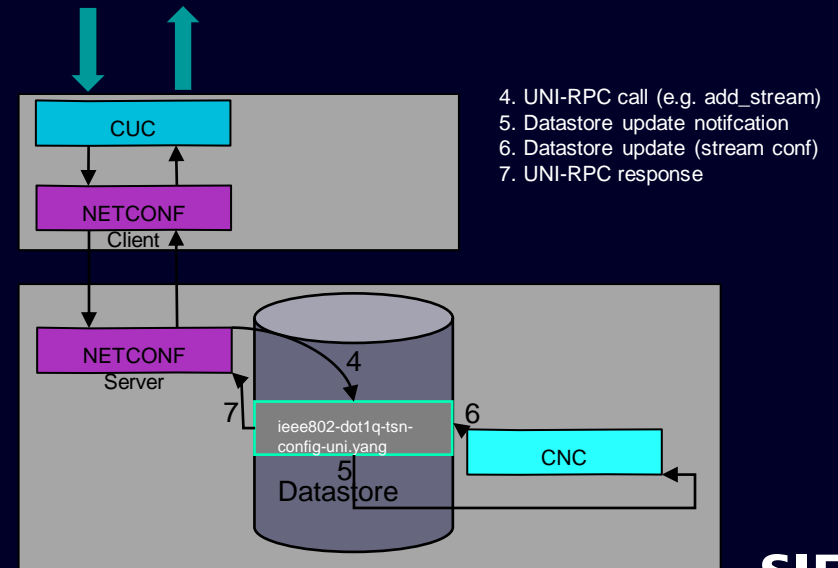
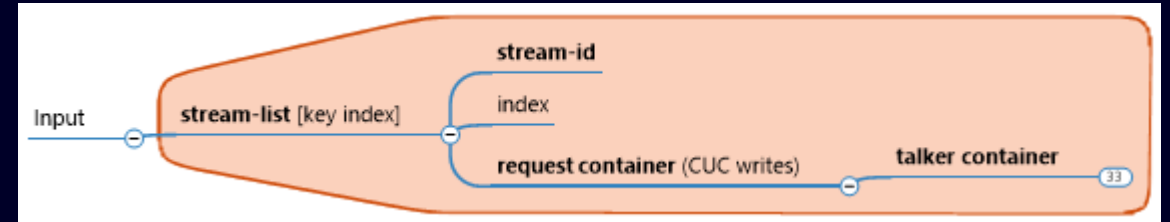
- Description
 - Adds multiple streams to CNC datastore
 - Assigns StreamIDs to streams
- Input (list of (StreamID, request container))

where StreamID := unknown

- Output (list of (StreamID))

where StreamID assigned by CNC

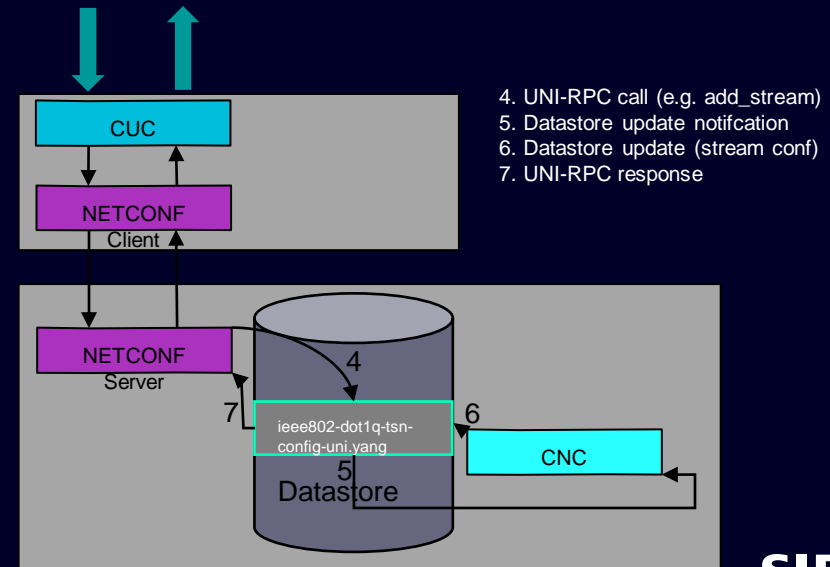
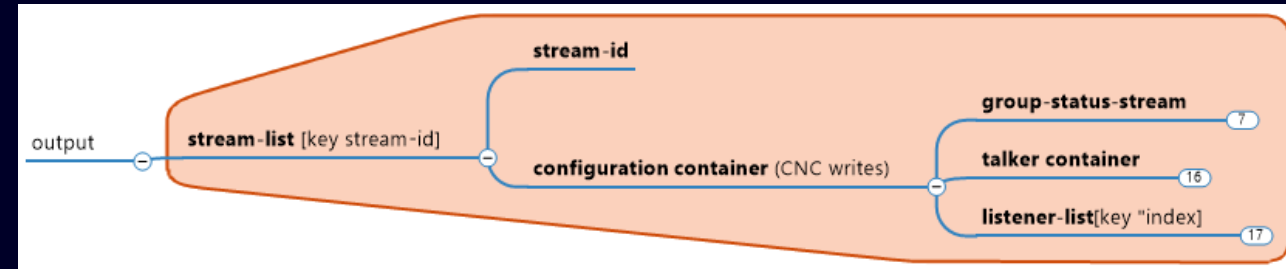
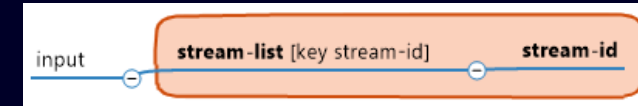
- RPC notifies CNC of new AddStream request
- CNC assigns StreamIDs to these streams
- RPC stores stream with StreamIDs in YANG container
- RPC responses with list of assigned StreamIDs



Stream requests: ScheduleStreams

RPC ScheduleStreams

- Description
 - Schedules/ plans multiple streams
- Input (list of (StreamID))
- Output (stream list)
- Modified configuration containers after scheduling/ planning requested streams
 - Stream dst-MAC addr assigned by CNC (configuration container)
 - Configuration information (relevant to CUC and talker) of streams impacted by requested streams



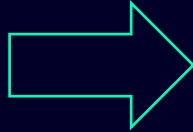
Scheduling multiple streams with one request is an IA requirement

Stream requests: ScheduleStreams

RPC ScheduleStreams

- ...
- **Stream dst-MAC** addr assigned by CNC (configuration container)
- ...

This restriction must be removed from the standard



15 **46.2.5.3.1 IEEE802-MacAddresses**

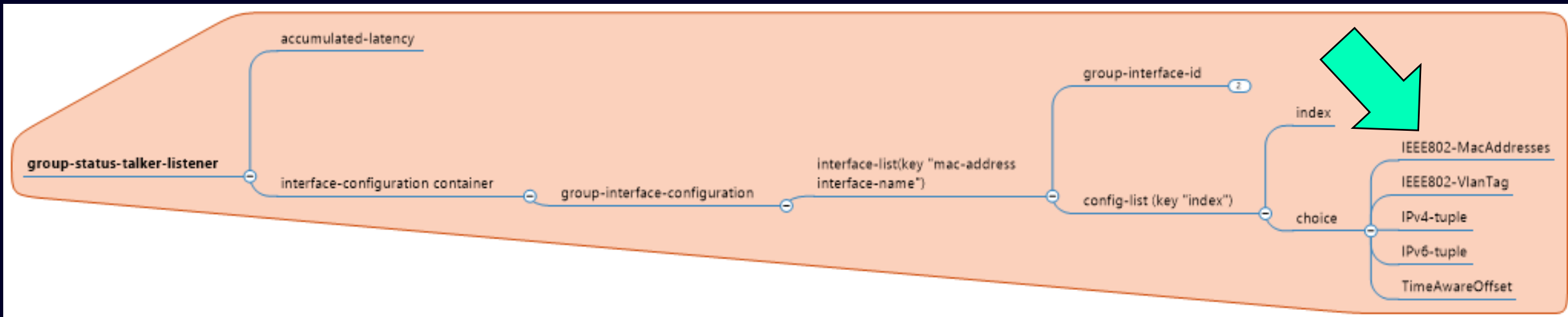
16 The IEEE802-MacAddresses group provides the source and destination MAC addresses that apply to the
17 network side of the user/network boundary.

18 NOTE 1—On the user side, the MAC addresses are in DataFrameSpecification.IEEE802-MacAddresses.

19 NOTE 2—The source MAC address of the network is typically the same as the user. The destination MAC address can
20 be different. For example, the user can use an individual address, but the network can use a group (multicast) address.

21 This group uses the specifications from DataFrameSpecification.IEEE802-MacAddresses (46.2.3.4.1).

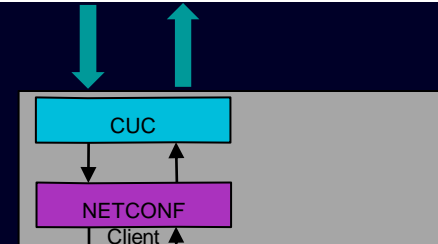
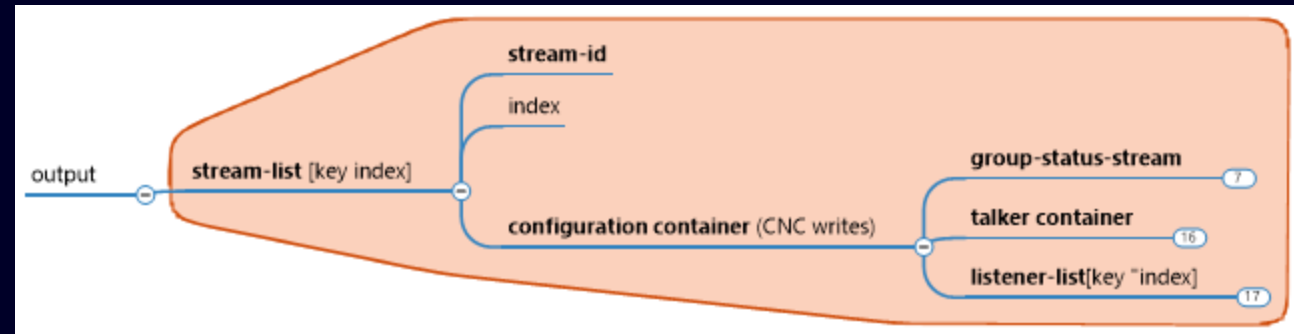
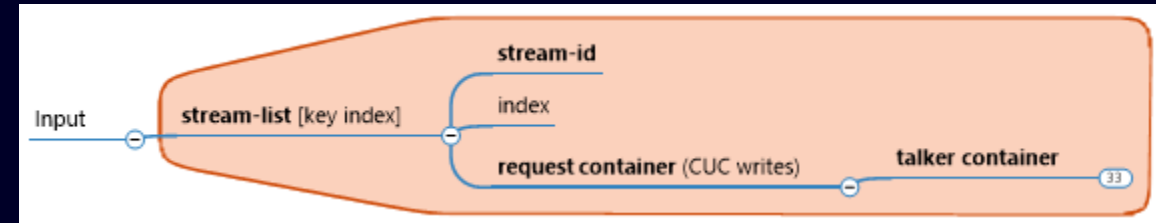
22 This configuration value is not provided unless IEEE Std 802.1CB is supported and a value for Active
23 Destination MAC and VLAN Stream identification is provided in CB-StreamIdenTypeList of
24 InterfaceCapabilities.



Stream requests: AddAndScheduleStreams

RPC AddAndScheduleStreams

- Description
 - Adds multiple streams to CNC datastore
 - Assigns StreamIDs to streams
 - Schedules multiple streams
- Input (list of (StreamID, request container)) where StreamID := unknown
- Output (stream list)
 - Modified configuration containers after scheduling/ planning requested streams
 - StreamID assigned by CNC
 - Stream dst-MAC addr assigned by CNC
 - Configuration information (relevant to CUC and talker) of streams impacted by requested streams



4. UNI-RPC call (e.g. add_stream)
5. Datastore update notification
6. Datastore update (stream conf)
7. UNI-RPC response

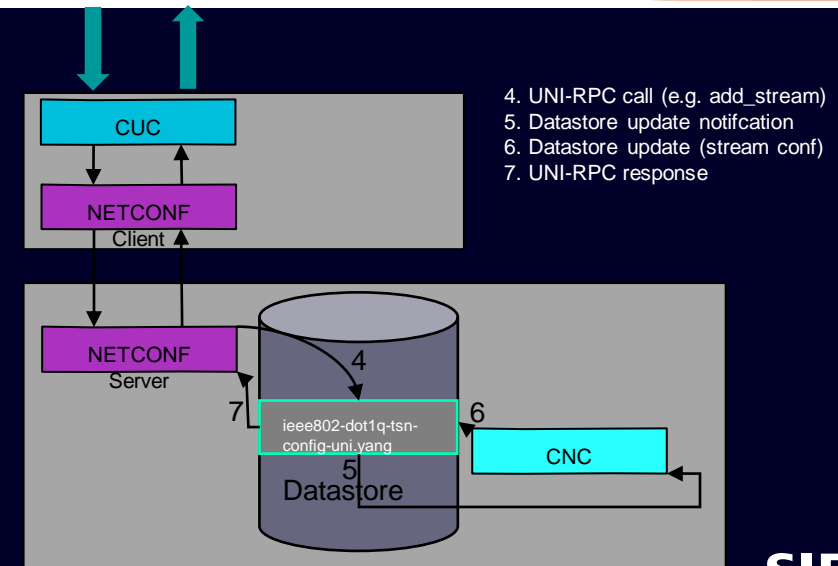
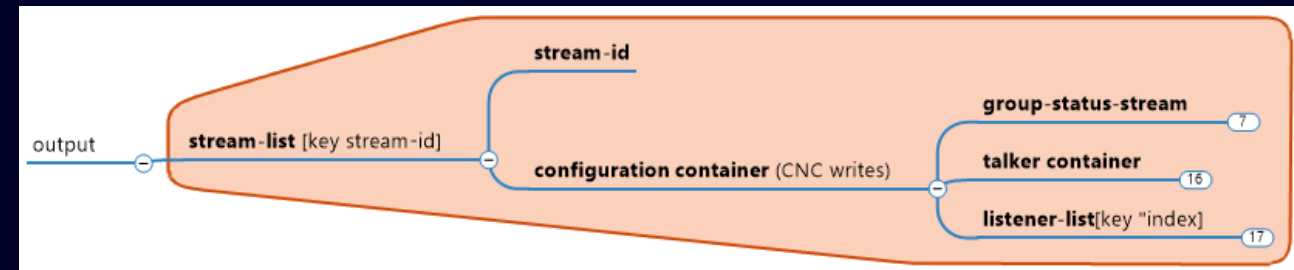
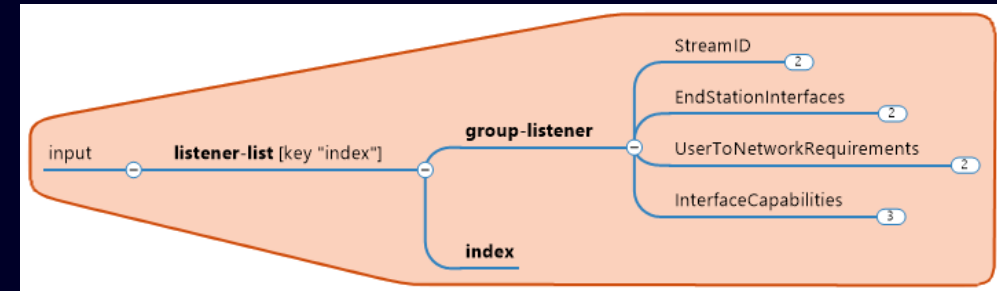
Scheduling multiple streams with one request is an IA requirement

Further **ieee802-dot1dj-tsn-config-uni** RPCs for stream management

Add listeners to stream

RPC JoinStream ⇔ AddAndScheduleListeners

- Description
 - **Adds** listeners to a stream into CNC datastore
 - **Schedules** listeners of stream
- **Input** (list of (Listener))
- **Output** (stream list)
- Modified configuration containers after scheduling/ planning requested streams
 - Configuration information (relevant to CUC and talker) of streams impacted by requested streams

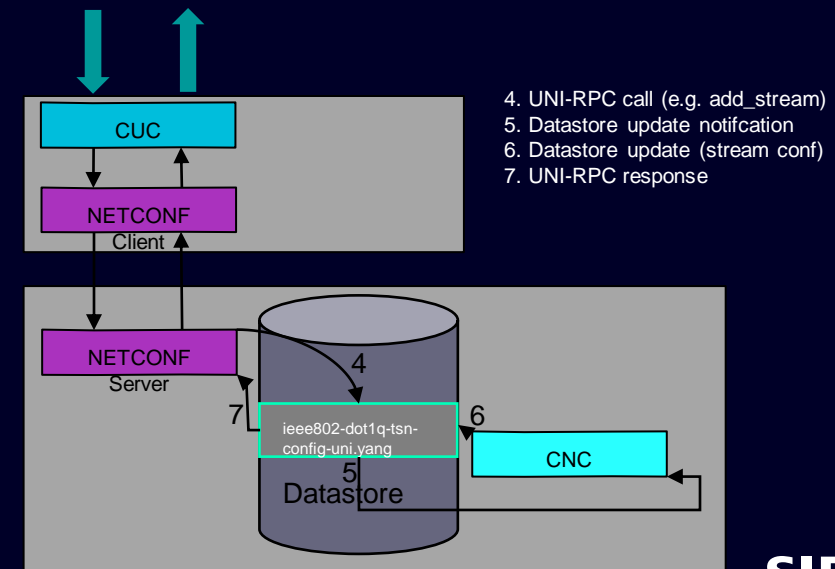
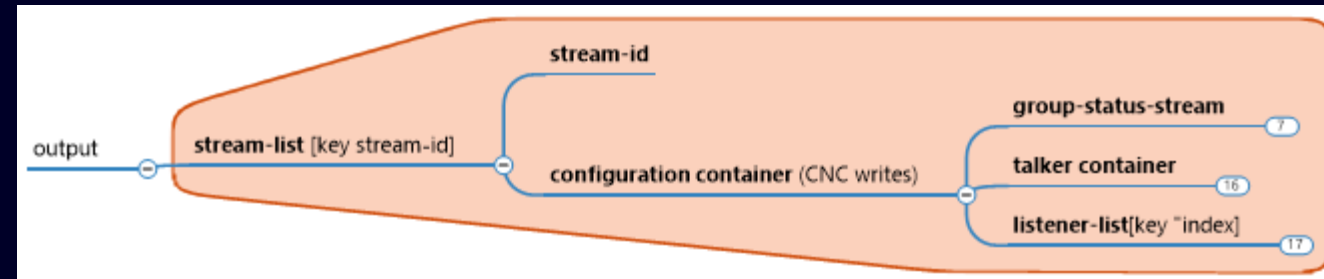
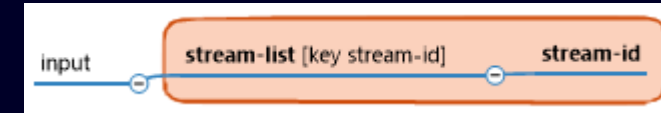


- UNI-RPC call (e.g. add_stream)
- Datastore update notification
- Datastore update (stream conf)
- UNI-RPC response

Remove Streams

RPC RemoveStreams

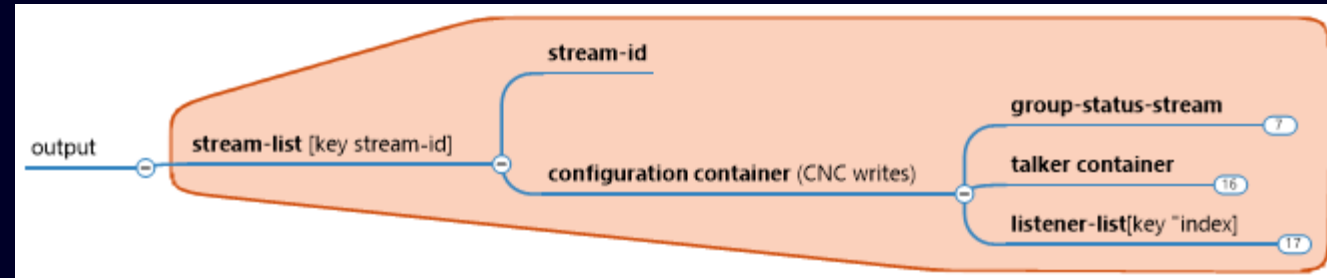
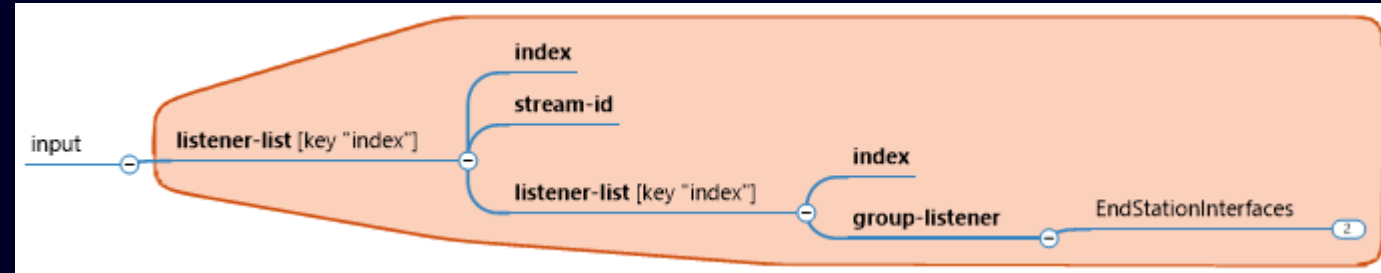
- Description
 - **Removes** streams from CNC datastore
 - Updates stream configurations
- **Input** (list of (StreamID))
- **Output** (stream list)
- Modified configuration containers after removing requested streams
- Configuration information (relevant to CUC and Talker) of streams impacted by requested streams



Remove listeners of a stream

RPC RemoveListeners

- Description
 - **Removes** listeners of a stream
 - Updates stream configurations
- Input (StreamID, list of (Listener))
- **Output** (stream list)
 - Modified configuration containers after removing listeners of requested streams
 - Configuration information (relevant to CUC, Talker, and Listeners) of streams impacted by requested streams



Summary

- StreamIDs and Stream dst-MAC addresses managed by the CNC
- Stream requests via YANG RPCs
 - Requesting multiple streams in one request is an IA requirement
 - Add & Schedule
 - Separated
 - In a single request
 - AddListeners, RemoveListeners, RemoveStream

Next steps

- YANG module refinement and textual contribution

| Contact

Dr. Rodrigo Ferreira Coelho

System Architect

DI FA CTR ICO ARC

Siemenspromenade 1

91058 Erlangen

Deutschland

Phone: +49 9131 17-45546

E-mail: rodrigo.ferreira_coelho@siemens.com