CNC-CUC Interaction for IEC/IEEE 60802

Nemanja Stamenic (Siemens AG)
Günter Steindl (Siemens AG)
Josef Dorr (Siemens AG)
Martin Mittelberger (Siemens AG)
Rodrigo Coelho (Siemens AG)

6 Required functions for an industrial network

6.9 CNC

6.9.3 CNC-CUC Interaction

6.9.3.1 General

The following text describes in detail the actions 2 to 5 shown in Figure 2. These CNC-CUC interactions specified in 6.9.3.2, 6.9.3.3 and 6.9.3.4 are based on the following preconditions:

a) CNC is activated to manage the configuration domain.

b) CUC knows the IP address of the CNC.

c) Security setup has been performed.

d) CUC is part of the configuration domain.

e) CUC knows the domain-id of the configuration domain.

f) CUC is registered by cuc-id at CNC.

6.9.3.2 Add Stream Operation

Add stream operation is performed when an application requests an establishment of one or more new streams. CNC shall handle each add stream operation as "establish all or no streams".

EDITOR'S NOTE : Should CUC be able to choose if the add stream operation is to be performed as "establish all or no streams" or not? P802.1Qdj does not support this choice currently.

CUC shall use the following steps:

a) CUC receives the stream requirements from the middleware.

b) CUC requests a stream-id for each stream that needs to be established using the RPC defined in the ieee802-dot1q-tsn-config-uni module. There are two possible outcomes:

   1) CUC receives a stream-id for each stream as a response from CNC.

   2) There is no free stream-id for one or more streams. The strategy for CUC behaviour in this case depends on the way the pool of streams is managed (one pool per CUC or one pool per configuration domain).

c) The CUC creates and locks the candidate configuration datastore. There are two possible outcomes:

   1) Candidate configuration datastore is available and locked by the CUC.

   2) Candidate configuration datastore is not available due to the lock imposed by another CUC. Wait and retry is the only possible strategy. Concurrent requests and/or pipelining is not supported.

EDITOR'S NOTE : non-blocking requests might be possible depending on the outcome of the currently ongoing discussion in the P802.1Qdj. Steps c), d), e) and f) would be merged in one RPC call in this case. Risk: starvation problem occurs.

d) CUC creates an entry for each stream at the CNC using the ieee802-dot1q-tsn-config-uni module and sets the stream parameters.

e) CUC calls <validate> and <commit> operations for the candidate configuration datastore. The changes are validated by the CNC before the commit is accepted or refused.

f) CUC calls compute_planned_and_modified_streams RPC from the ieee802-dot1q-tsn-config-uni module. There are two possible outcomes:

   1) OK: compute of at least one stream has started. CUC awaits the configure_streams_completed notification.

   2) NOK: compute has not started due to invalid input parameters or similar error.

g) The CUC receives the configure_streams_completed notification from the CNC, which contains the failure-code for each requested stream.

h) CUC informs the middleware that the streams have been established.

6.9.3.3 Remove Stream Operation

Remove stream operation is performed when an application requests removal of one or more streams. CUC shall use the following steps:

a) CUC receives the stream removal request from the middleware.

b) CUC calls remove_streams RPC from the ieee802-dot1q-tsn-config-uni module. There are two possible outcomes:

   1) OK: the removal of at least one stream has started. CUC awaits remove_streams_completed notification.

   2) NOK: stream removal has not started due to invalid input parameters or similar error.

c) The CUC receives the remove_streams_completed notification from the CNC, which contains the failure-code for each stream whose removal was requested.

d) The CUC informs the middleware that the streams have been removed.

6.9.3.4 Modify Stream Operation

Modify stream operation shall be implemented as a sequence of the remove stream operation as defined in 6.9.3.3 and the add stream operation as defined in 6.9.3.2.

6.9.3.5 Topology Changes

6.9.3.5.1 IA-Station Fault

In the case of a malfunction of an IA station, CNC discovers the topology change and recognizes that the path used for a stream is broken. The status-info node of the affected streams in the ieee802-dot1q-tsn-config-uni module is updated accordingly by the CNC. CUC is informed of the status-info change via YANG subscription. It is the responsibility of the middleware to deal with this situation e.g., by requesting the stream removal.

6.9.3.5.2 IA-Station Fault Clearance

After an IA-Station fault described in 6.9.3.5.1, the IA-station can be repaired or replaced and reintroduced into the network. In this case CNC discovers the topology change. Streams are not reconfigured automatically. The value of the status-info node in the ieee802-dot1q-tsn-config-uni module is changed after the repair/replacement. CUC is informed of the status-info change via YANG subscription. It is the responsibility of the middleware to deal with this situation e.g., by calling the compute_and_configure_streams RPC in the ieee802-dot1q-tsn-config-uni module.

EDITOR'S NOTE : Update of the status-info of the affected stream is required in order to inform the CUC about the fault clearance. The failure code must provide the information if the failed streams might be successfully recomputed.

6.9.3.5.3 CUC Removal and Addition

CUC removal and addition can occur in the case of a CUC fault followed by fault clearance or CUC replacement. CUC addition and/or removal occurs also in the context of plug and produce of machine modules.

There are two possible behaviours of the CNC:

a) CNC removes the whole CUC tree including all streams from its configuration domain

b) CNC keeps the whole CUC tree including all streams

New CUC can use the operation described in 6.9.3.2. to request establishment of new streams.

EDITOR'S NOTE: Should 60802 specify a mechanism that handles CUC removals and additions? Should P802.1Qdj enable CUC to choose between CNC behaviours a) and b) e.g., during onboarding? Should the scenario with CUCs that are not part of the configuration domain be considered?

6.9.3.5.4 CNC Fault

The streams are not interrupted in the case the CNC is removed, but resource allocation cannot be changed.

6.9.2.4.5. CNC Fault Clearance

Reinitialization of the configuration domain occurs as soon as the new CNC is added to the configuration domain.

EDITOR'S NOTE : Is mechanism for CNC replacement required for the edition 2 of the profile?

6.9.4 CNC-Behavior

6.9.4.1 General

RFC 8342 specifies the Network Management Datastore Architecture, which is the base for the NETCONF client and NETCONF server interactions.

6.9.4.2 Add Stream Operation

Add stream operation is not atomic. It is processed in multiple stages. The following list represents an example of such a sequence.

a) The CUC calls <edit-config> to add stream parameters as defined in 6.9.3.2

b) The CUC calls <validate> operation on the candidate configuration datastore.

c) CNC receives <commit> operation call from the CUC.

d) CNC sets the running configuration to the current contents of the candidate configuration.

e) CNC receives compute_planned_and_modified_streams RPC call.

f) CNC validates the added stream parameters. There are two possible outcomes:

   1) OK: Compute starts, RPC returns OK

   2) NOK: computes is not possible, RPC returns NOK. The streams are not added to the operational state datastore. The add stream operation is finished.

g) CNC computes all streams. There are two possible outcomes:

   3) OK: All streams have been computed.

   4) NOK: Computation of one or more streams failed. CNC provides the configure_streams_completed notification to the CUC. Add stream operation is finished.

h) CNC configures the paths for all streams. There are two possible outcomes:

   5) OK: The path for all streams are configured. The streams are added to the operational state datastore. The stream-status in the running configuration datastore.is set to “configured”.
   6) NOK: Configuration of one or more streams failed. CNC provides the configure_streams_completed notification to the CUC. Add stream operation is finished.

i) CNC provides the configure_streams_completed notification to the CUC.

A stream can be present only in running configuration datastore, in which case its status is unknown, or it can be present in both running configuration datastore and operational state datastore in which case its status (status-info node) is shown in operational state datastore.

6.9.4.3 Remove Stream Operations

Remove stream operation is not atomic. It is processed in multiple stages. The following list represents an example of such a sequence.

The state of the configuration datastore is irrelevant for the remove stream operation.

a) The CNC receives the remove_streams RPC call defined in the ieee802-dot1q-tsn-config-uni module from the CUC.

b) CNC validates the parameters of the RPC call. There are two possible outcomes:

   1) OK: CNC starts removal of the streams. CNC returns OK.
   2) NOK: stream removal has not started due to invalid input parameters or similar error. CNC return NOK.

c) CNC removes the streams from the running configuration datastore.

d) CNC removes the paths from the IA-stations. There are two possible outcomes:

   1) OK: The paths are removed.
   2) NOK: Removal of paths failed. The status-info node is updated in the and operational state datastore. CNC provides the remove_streams_completed notification with status NOK to the CUC. The remove stream operation is finished.

e) CNC removes the streams from the operational state datastore.

f) CNC provides the remove_streams_completed notification with status OK to the CUC.

After the successful stream removal, all resources are freed. The stream-ids are still assigned to the corresponding CUC.

6.9.4.4 RPCs and Datastore Handling

For each stream that has been added to the operational state datastore, the stream-status in running configuration datastore is set to “configured” by the CNC. The streams whose status-info is not equal to “configured” are always (re)configured after the compute_planned_and_modified_streams RPC call.