Contribution for the IEC/IEEE 60802 -- Extended onboarding model

Mark Hantel, Rockwell Automation

Marcel Kießling, Beckhoff Automation GmbH & Co KG

Christophe Mangin, Mitsubishi Electric Corp.

Marius Stanica, ABB AG

Günter Steindl, Siemens AG

V03

Problem statement

Constraint devices are often limited in possible code and data sizes.

Analysis of available NETCONF/YANG implementations assume that the available code and data space in constraint devices is not sufficient.

Thus, to include these constraint devices as IEC/IEEE 60802as ccB devices some extensions seem to be necessary.

Note: The functionality of a ccA device is a superset of a ccB device. Thus, each manufacturer may define each of its ccA device as ccB devices and implement the named extension.

Possible solutions

Constraint devices that meet the technical requirements for ccB are already configured today using organization- or manufacturer-specific protocols. If these procedures are integrated into the 60802 configuration model, these devices can also be integrated into a 60802 configuration domain.

This integration could be implemented via a configuration proxy. This function translates the NETCONF/YANG spoken by the CNC bidirectionally into the protocol specific to the constraint devices.

In this way, the configuration model of the 60802 configuration domain is preserved, and the constraint devices are integrated.

But how does the CNC know if proxies are present, which ones are available, and which devices they represent?

1. Network Discovery Protocols: These protocols help the CNC identify and communicate with proxies on the network.
2. Configuration Data: Proxies can be pre-configured with information about the devices they represent, which they then share with the CNC.
3. Registration Mechanisms: Proxies might register themselves and their associated devices with the CNC upon initialization.

2. would make the Plug&Produce case impossible; and 3. would require a new protocol. Therefore, 1. seems promising, as we already have a discovery mechanism with NETCONF/YANG.

What would be needed?

As part of the discovery process, the CNC must be able to find the proxies, the devices they represent, and the constraint devices.

Proxies

The 60802 YANG modules need to be extended to include information about the proxy role. In principle, a module could be defined that documents the following:

1. The station providing this YANG module is a proxy.

2. This proxy function is provided by this station for the following list of devices.

Constraint devices

Since the constraint devices do not natively support NETCONF, the proxy identification information must be transported via another mechanism. It makes sense to use LLDP for this purpose.

What needs to be defined for LLDP?

1. A TLV that serves to transport the proxy identification information (e.g., from the IEEE802.1 number space).

2. A coding of the data that works across organizations or manufacturers (e.g., already by organization/vendor defined OIDs).

3. (Long-term) An LLDP YANG module extension for this additional TLV.

4. The constraint devices must provide, evaluate, and deliver this TLV via their organization- or manufacturer-specific protocol to the proxy.

Why not using the Management TLV of LLDP?

This TLV may already be defined by organizations or vendors in a different way. Particular, because the OID field is intended to define a link to the responsible MIB entry for this device.

CNC

The CNC must evaluate the proxy information to configure them appropriately. During the configuration, it ensures that the proxies can reach newly connected devices.

Additionally, during the onboarding process, the CNC must identify the responsible proxy through the LLDP information from the neighbor already in the configuration domain and find/address the corresponding NETCONF instance there.

Once this process/preparation is completed, the existing onboarding process is used. The proxy then translates the configuration instructions into the configuration language of the constraint devices.

Security

Deployment of certificates between the proxies and the constraint devices is specified organization or manufacturer specific. National and international standards specify requirements which applies for constraint devices and their configuration protocols too.

Thus, it seems that any concern about a decrease in security beyond a customer accepted point is unrealistic.

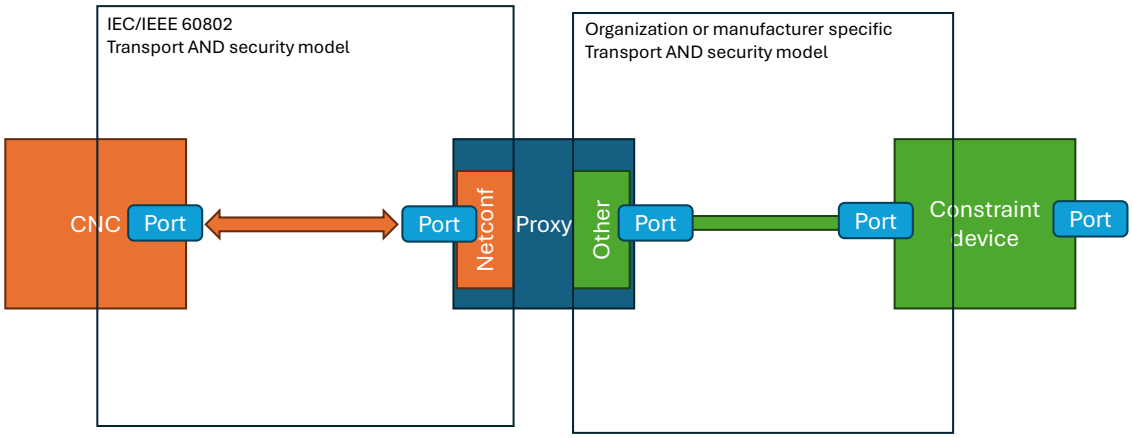

Figure 1: Security principle

An ccB constraint device may only differ configuration transport (NETCONF vs. organization or manufacturer specific) from the existing definition in the IEC/IEEE 60802.

Open issues

- What are the properties of a constrained device (can any device be declared as constrained device)?

ccB – only difference is the supported configuration protocol

- Is the constrained device an end station, a bridge or could it be both?

ccB – End station or bridged end station (don't see a bridge as pure network component)

- Where can the constrained device be connected to the TSN configuration domain (anywhere, behind a NETCONF proxy only)?

Anywhere – constraint devices extend an existing configuration domain

- Which boundary state does the port has to which the constrained device is connected to the configuration domain and once being configured for the configuration domain?

No difference – same as already mentioned

- In case the constrained device is a bridge, how is the boundary port state managed?

The proxy is more or less a configuration “gateway” between NETCONF/YANG and something else. The same actions as already mentioned happen during onboarding

- Does the CNC have any possibility to verify the secure device identity of the constrained device?

Can the CNC trust the proxy to report the correct secure device identity. That’s the backbone of the shown concept.

- Can IEC/IEEE 60802 put any security requirements to the user-defined configuration protocol?

Good question

- How is the interaction between constrained device, CNC and NETCONF proxy during onboarding?

See this contribution.

Proxy - Open issues

The proxy's task is to represent itself and many other IA stations via NETCONF/YANG. It needs to be clarified how this should be done. Here are a few possibilities:

1. The IA station of the proxy hosts many IP addresses, and behind each IP address is its own NETCONF/YANG instance.

2. The IA station of the proxy hosts one IP address with many TCP ports, and behind each port is its own NETCONF/YANG instance.

3. The IA station of the proxy hosts exactly one NETCONF/YANG instance, and the different IA stations are separated using YANG mechanisms.

Onboarding sequence

A new IA station is connected. The neighbor, which is already part of the configuration domain, reports a change (link status has changed) to the CNC.

The CNC reads the stored LLDP information of the neighbor (either directly or through its proxy) from the YANG-modulated data via NETCONF related to the connected device.

From this LLDP data, it finds out whether the connected station:

- natively speaks NETCONF/YANG (no information regarding proxy operation is included),

- requires a proxy, and if so, which one.

Then, the CNC contacts the NETCONF server responsible for this device.
Note: In the case of a proxy, it may take some time for the proxy to set up the necessary NETCONF/device instance. The CNC can only proceed once this is set up.

After that, the CNC, as previously described, communicates with the appropriate NETCONF server and configures the device accordingly. The proxy instance handles the NETCONF parameterization and translates it into the required organization-defined or vendor-specific protocol, and configures the constrained device.

Then the neighbor can be configured to lift the isolation mode on this port, and the connected device becomes part of the configuration domain.

Addressing scheme

The CNC must find the responsible Netconf instance. For this, appropriate information must be available.

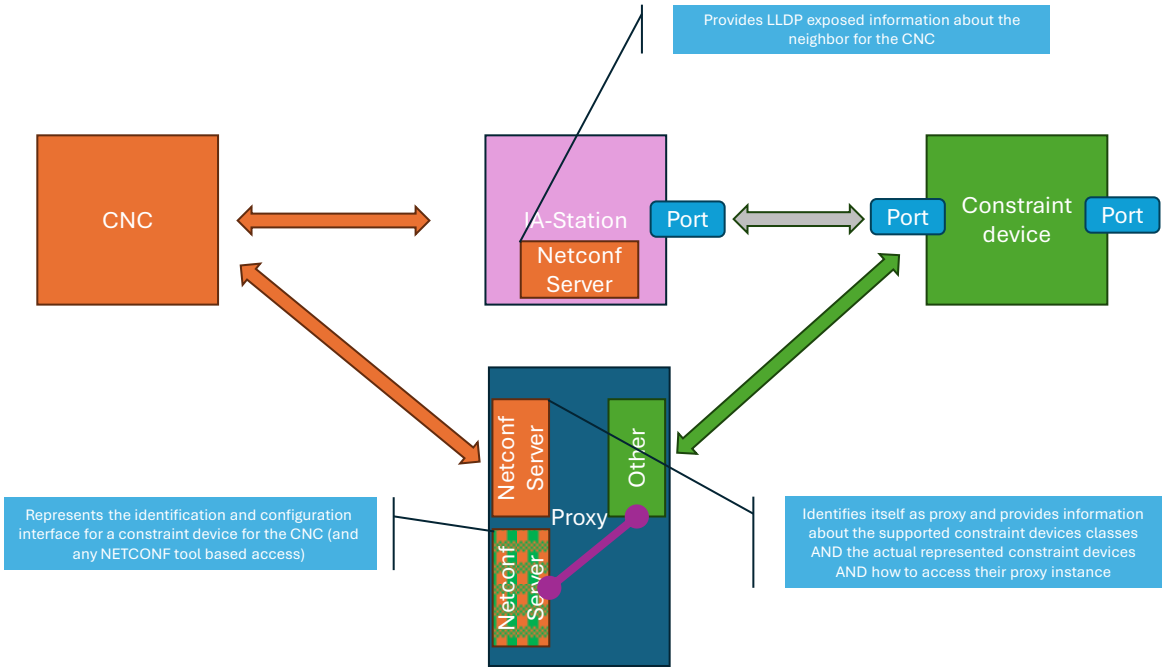

Figure 2: Addressing scheme

Step 1: Which proxy is responsible?

The constraint device communicates which proxy it needs through the information distributed by LLDP. The proxy describes, via its YANG module, which constraint devices it is responsible for.

Step 2: Which proxy instance is responsible and how is it addressed via Netconf?

The proxy describes, via its YANG module, which constraint devices it "represents" and how the corresponding instance is addressed. This list is dynamic, meaning it expands with each added constraint device represented by this proxy.

Note: The constraint device provides sufficient information for its proxy instance (its YANG module) to be uniquely identified.

Step 3: Using the proxy instance by the CNC if information needs to be read from or configuration needs to be written to this constraint device