Device Model of Bridged End Stations

Contribution to IEC/IEEE 60802

Authors:
Josef Dorr, Siemens AG
Günter Steindl, Siemens AG

CONTENTS

4 Device Model of Bridged End Stations 2
4.1 Terms and Definitions 2
4.2 Bridged End Station IA-Devices 2
4.3 YANG models 3
4.3.1 IEEE 802.1Q YANG model 4
4.3.2 IEEE 802.1Q Bridge management model 4
4.3.3 IEEE 802.1Q Bridge Port management model 5
4.3.4 IEEE 802.1AB YANG model 6
4.4 Internal LAN connection model 6
4.5 Topology discovery and verification requirements 3
4.5.1 Topology discovery 3
4.5.2 Topology verification 4
4.6 Topology discovery of complex IA-devices 7
4.7 Further complex IA device examples 9
4.8 Bridge Management of IA-device components 12
4.9 Generic IEEE 802.1AS management model 13
4.10 PTP Management of IA-device components 14

FIGURES

Figure 1 – complex IA device example 3
Figure 2 – Generic IEEE 802.1Q YANG Bridge management model 4
Figure 3 –IEEE 802.1Q YANG Bridge management model 5
Figure 4 –IEEE 802.1Q YANG Bridge Port management model 5
Figure 5 – Generic IEEE 802.1AB YANG model 6
Figure 6 – C-VLAN component internal LAN managed system 6
Figure 7 – YANG data scheme of LLDP local-system-data 7
Figure 8 – YANG data scheme excerpt of LLDP local port data 7
Figure 9 – complex IA-device example with ietf-interface and bridge-port data........ 8
Figure 10 – two ES components connected to the same Bridge component 9
Figure 11 – multiple end station components in one chassis 9
Figure 12 – cascaded Bridge with hidden internal Bridge components 10
Figure 13 – cascaded Bridge components with exposed internal connectivity 11
Figure 14 – IEEE 1588 YANG structure 13
Figure 15 – IA-device example with ietf-interfaces 14

4 Device Model of Bridged End Stations

4.1 Terms and Definitions

Important terms and definitions given in IEEE Std 802.1AB-2016

system	A managed collection of hardware and software components incorporating one or more chassis, stations and ports
chassis	A physical component incorporating one or more IEEE 802® LAN stations and their associated application functionality.

Important terms and definitions given in IEEE Std 802-2014:

station	An end station or bridge.
bridge	A functional unit that interconnects two or more IEEE 802® networks that use the same data link layer (DLL) protocols above the medium access control (MAC) sublayer, but can use different MAC protocols. Forwarding and filtering decisions are made on the basis of layer 2 information.
end station	A functional unit in an IEEE 802® network that acts as a source of, and/or destination for, link layer data traffic carried on the network.

Important terms and definitions given in IEEE Std 802.1Q-2018:

Bridge	A system that includes Media Access Control (MAC) Bridge or Virtual Local Area Network (VLAN) Bridge component functionality and that supports a claim of conformance to Clause 5 of IEEE Std 802.1Q-2018 for system behavior.

4.2 Bridged End Station IA-Devices

IA-devices very often include an end station component together with a Bridge component in one chassis. This kind of device is called bridged end-station. But even more complex IA-devices with multiple end station and Bridge components within one station and chassis can be found in industrial automation. The various components are connected by internal Ports and internal LANs. Figure 1 shows an example IA-device with two Bridge components and three end station components. All components are attached to one common management entity.

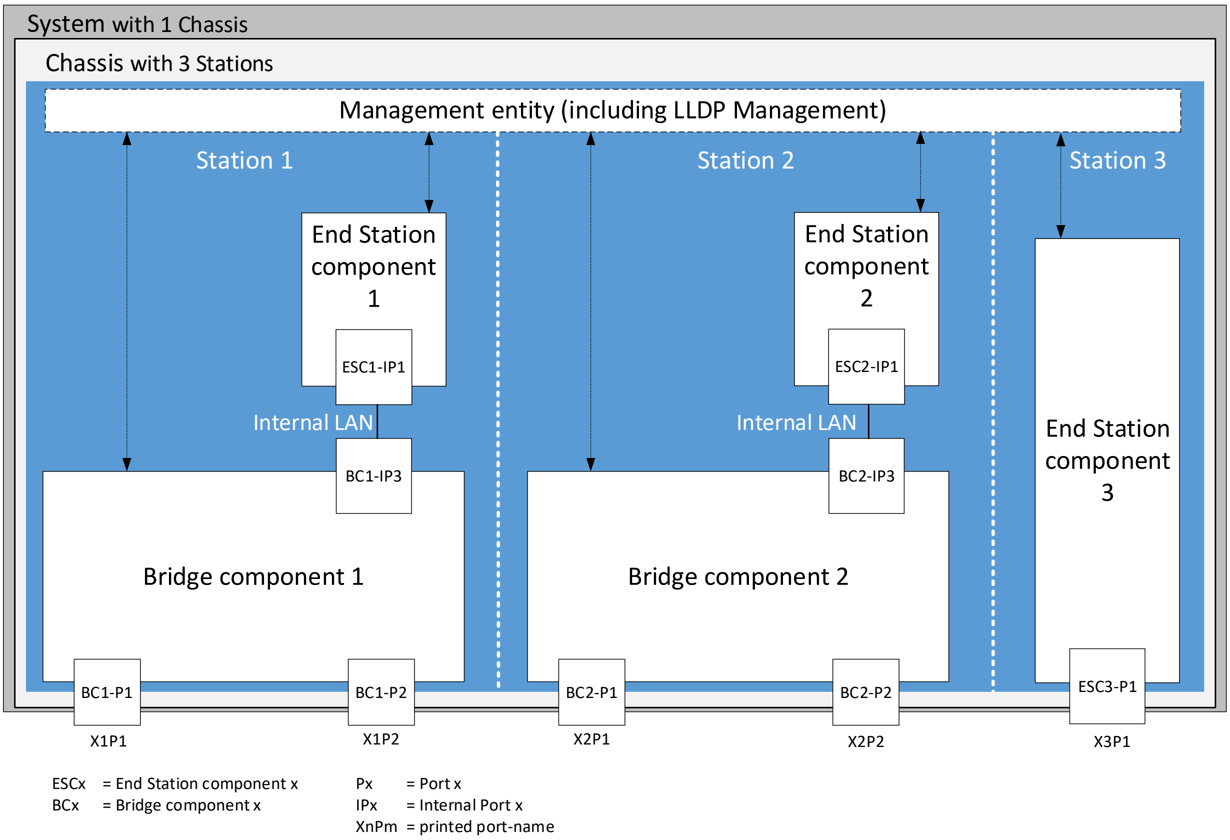

Figure 1 – complex IA device example

4.3 Topology discovery and verification requirements

4.3.1 Topology discovery

Machine electrical engineering includes the definition of the machine internal network topology including the assignment of names to stations, ports, and links.

Topology discovery must support in a vendor/organization independent way reporting of:

- CAE/CAD (e.g., EPLAN) defined human readable station, port, and link names to ensure that production planning and installation are identical.
  - Station names are usually written on a title block or printed on a sticker.
  - Port names (e.g. X1P1) are usually printed on the chassis of a station.
  - Link names are usually fastened as a label to the cables.

Maintenance staff (electricians) uses the human readable names of stations, ports, and links to easily identify failed stations, ports, or links.

The LLDP port-id can e.g., be used to convey the station and port names.

The topology discovery function within a Topology Discovery Entity (TDE) must be able to detect the following additional data, which is used by NPE and CNC for networking provisioning and stream establishment, from any IA device:

- Bridge or end station components with their names,
- external ports with their names and with their relation to the components,
- internal ports with their relation to the components, and
- internal LANs with their relation to internal ports.

4.3.2 Topology verification

Topology verification is an important use case in industrial automation. Checking discovered topologies against engineered topologies must be possible. The check includes the involved stations, ports, and links.

- Engineered topologies are based on data sheet provided information. This information together with the CAE/CAD assigned human readable names is used to create the reference topology for verification.

- Repair and replacement of a station shall not require an update of the reference topology for verification – or produce a verification error.

4.4 YANG models

The Bridged end station device model includes references to the IEEE 802.1Q YANG model and to the IEEE 802.1AB YANG model.

4.4.1 IEEE 802.1Q YANG model

Figure 2 shows the hierarchical structure that incorporates the IEEE 802.1Q YANG models.

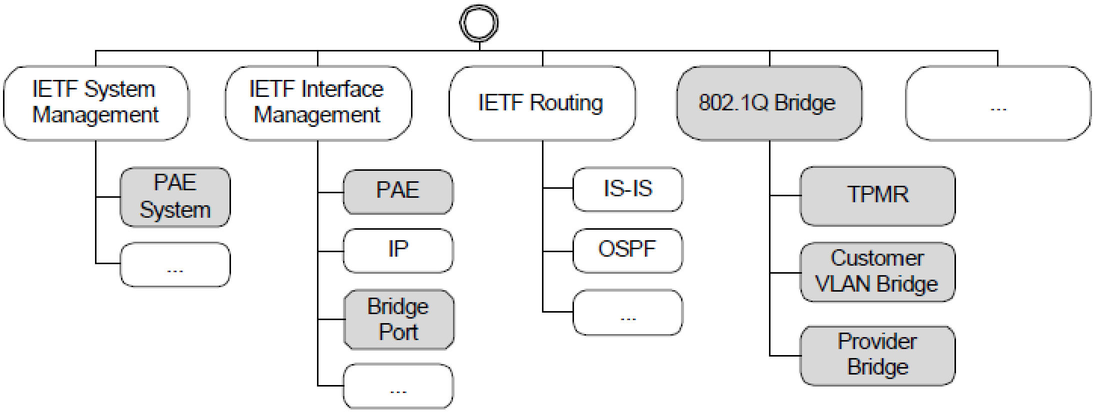

Figure 2 – Generic IEEE 802.1Q YANG Bridge management model

The 802.1Q Bridge model and the Bridge Port augmentation of the IETF Interface Management model are used for Bridged end station management.

4.4.2 IEEE 802.1Q Bridge management model

The 802.1Q Bridge model organizes a Bridge as a list of **components** (see IEEE 802.1Qcp-2018 Figure 48-4), where each component is identified by its unique component **name** (see Figure 3).

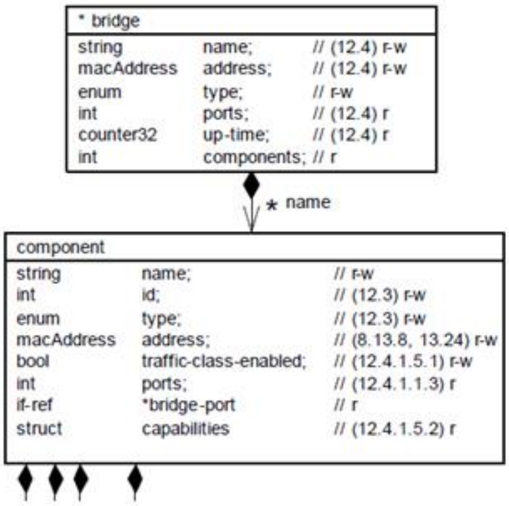

Figure 3 –IEEE 802.1Q YANG Bridge management model

Various component types are defined, to classify a particular Bridge component - including the c-vlan-component type.

<MIBs: The ieee8021BridgeBaseTable of the IEEE8021-BRIDGE-MIB contains generic information about every component, where each component is identified by its unique componentId.>

4.4.3 IEEE 802.1Q Bridge Port management model

The YANG model of IEEE 802.1Qcp-2018 models **Bridge Ports** as augmentation of the generic IETF Interface Management data model (`ietf-interfaces`) defined in RFC 8343. An interface is identified by its unique interface **name**. The bridge-port augmentation includes the name of the component to which the port belongs. Figure 4 shows an excerpt of the UML representation of this model.

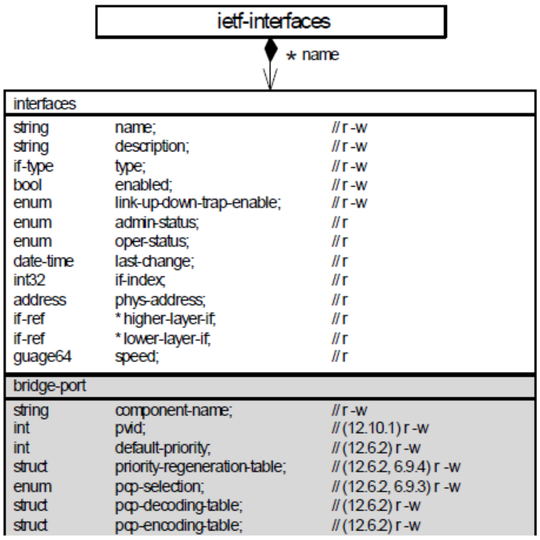

Figure 4 –IEEE 802.1Q YANG Bridge Port management model

<MIBs: The ieee8021BridgeBasePortTable of the IEEE8021-BRIDGE-MIB contains generic information, about every port of a Bridge. Each Port is identified by the unique tuple: componentId/portNumber.>

4.4.4 IEEE 802.1AB YANG model

Figure 5 shows the hierarchical structure that incorporates the IEEE 802.1AB YANG model.

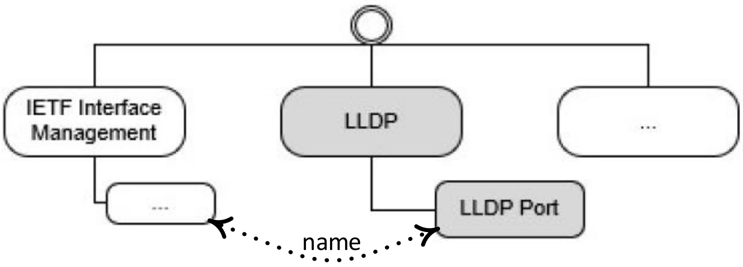

Figure 5 – Generic IEEE 802.1AB YANG model

Different to the 802.1Q Bridge port management model the LLDP Port model does not augment the IETF Interface Management model but uses the name of the associated interface as a common key to the LLDP port list.

4.5 Internal LAN connection model

IEEE 802.1Q Figure 17-1 describes a concept to model internal connections between C-VLAN components with the help of the IETF interface objects and definitions. Figure 6 shows this concept from IEEE 802.1Q.

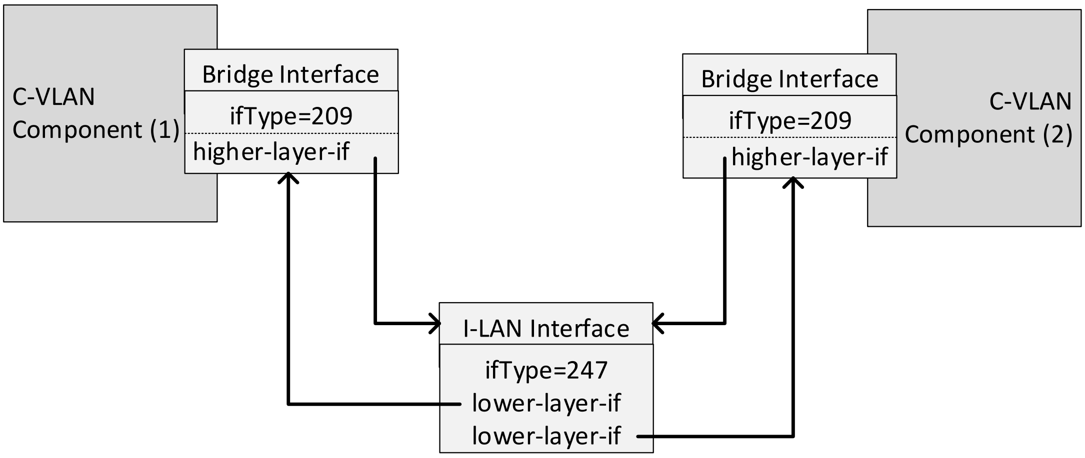

Figure 6 – C-VLAN component internal LAN managed system

The internal LAN connection model comprises three configuration steps:

- The internal Ports of the C_VLAN components are modeled as Bridge ports and interfaces with `ifType=209` (`transparent bridge interface`).
- An additional I-LAN interface is created with `ifType=247` (`Internal LAN on a bridge per IEEE 802.1ap`).
- The I-LAN interface references the Bridge interfaces of the connected C_VLAN components as `lower-layer-if`, and

  the Bridge interfaces of the connected C_VLAN components reference the I-LAN interface as `higher-layer-if`.

4.6 Topology discovery of complex IA-devices

The device discovery function is based on the LLDP local system and port data.

Basic LLDP **local system data** provides the chassis-id value and two lists of supported and enabled system capabilities. Figure 7 shows the YANG data scheme definition of local-system-data as defined in IEEE P802.1ABcu D1.7.

```
+--ro local-system-data
| +--ro chassis-id-subtype?             ieee:chassis-id-subtype-type
| +--ro chassis-id?                     ieee:chassis-id-type
| +--ro system-name?                    string
| +--ro system-description?             string
| +--ro system-capabilities-supported?  lldp-types:system-capabilitiesmap
| +--ro system-capabilities-enabled?    lldp-types:system-capabilitiesmap
```

Figure 7 – YANG data scheme of LLDP local-system-data

<MIBs: The lldpV2LocalSystemData of the LLDP-V2-MIB provide equivalent data.>

IA-devices with Bridge and end station components signal `bridge` and `station-only` capabilities within the map of enabled system capabilities.

<open issue: IEEE 802.1AB states, that the station-only bit "… should therefore not be set in conjunction with any other bits.">

Basic LLDP **local port data** provides for **all external ports** – among a lot of other data - the interface name (`name`), that is a reference to the port's associated entry in the `ietf-interfaces` YANG module (IETF RFC 8343). Figure 8 shows an excerpt of the YANG data scheme definition of local-port-data as defined in IEEE P802.1ABcuD1.7.

```
+--rw port*   [name dest-mac-address]
       +--rw name                       if:interface-ref
       +--rw dest-mac-address           ieee:mac-address
       +--rw admin-status?              enumeration
       +--rw notification-enable?       boolean
       +--rw tlvs-tx-enable?            bits
       +--…
```

Figure 8 – YANG data scheme excerpt of LLDP local port data

<MIBs: The lldpV2LocPortTable of the LLDP-V2-MIB provides equivalent data.>

When the TDE detects a complex IA-device (i.e., `bridge` and `station-only` capabilities are declared in the system capabilities maps), it has to read additional data from the `ietf-interfaces` YANG module to be able to identify the single components with their internal ports and internal LANs. Figure 4 shows the UML representation of the Interface Management (IETF RFC 8343) model with the bridge-port augmentation of IEEE 802.1Qcp-2018.

Discovery of a complex IA-device with multiple components can be accomplished, when the IA-device's LLDP local system data and ietf-interfaces data are organized as described in 4.5:

– All external ports are represented in the LLDP list of local port data. External ports are also represented in the ietf-interfaces list with `ifType=6` (`ethernetCsmacd`).

– The internal ports can be identified in the ietf-interfaces list by `ifType=209` (`transparent bridge interface`).

– The internal LANs (I-LANs) can be identified in the ietf-interfaces list by `ifType=247` (`Internal LAN on a bridge per IEEE 802.1ap`).

– The internal ports which are connected by an internal LAN can be identified by the `lower-layer-if` references of the I-LAN interface object.

- The `ieee802-dot1q-bridge` augmentation of the interface model provides for all external and internal ports the name of the corresponding component (`component-name`).

With this data it is possible for the device discovery function of a TDE to detect the internal structure of any combined device as required.

Figure 9 shows the relevant LLDP, ietf-interface and bridge-port data of the example IA-device from Figure 1.

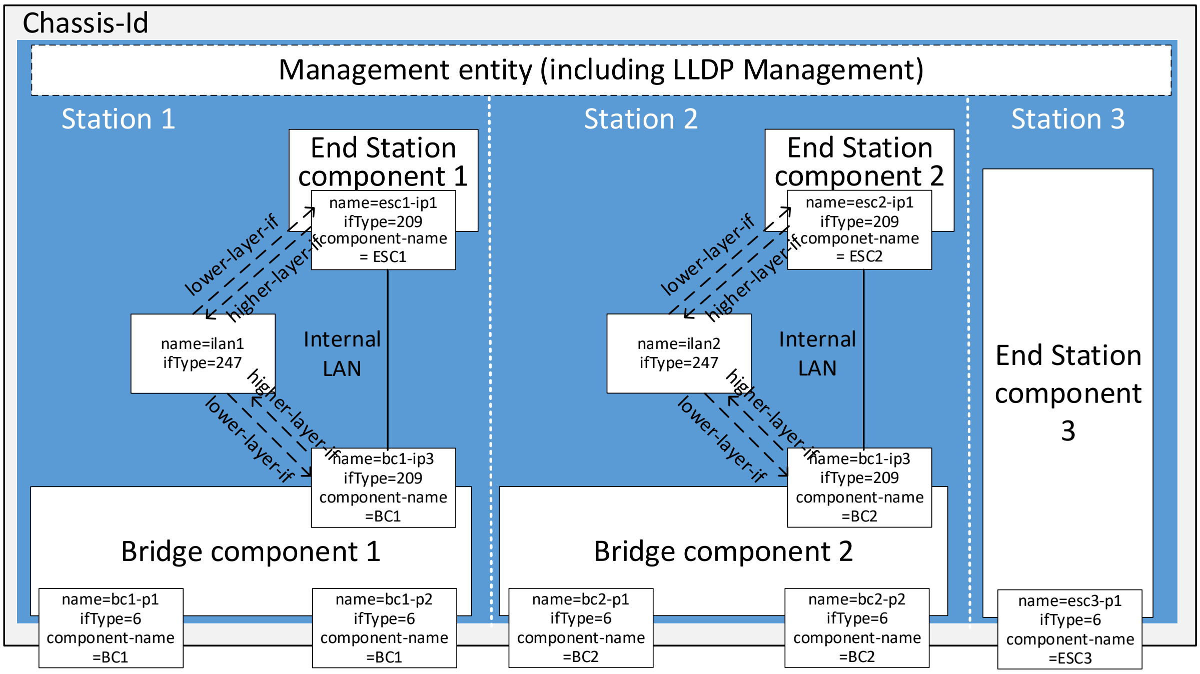

Figure 9 – complex IA-device example with ietf-interface and bridge-port data

4.7 Further complex IA device examples

Figure 10 shows two end station components connected to the same Bridge component:

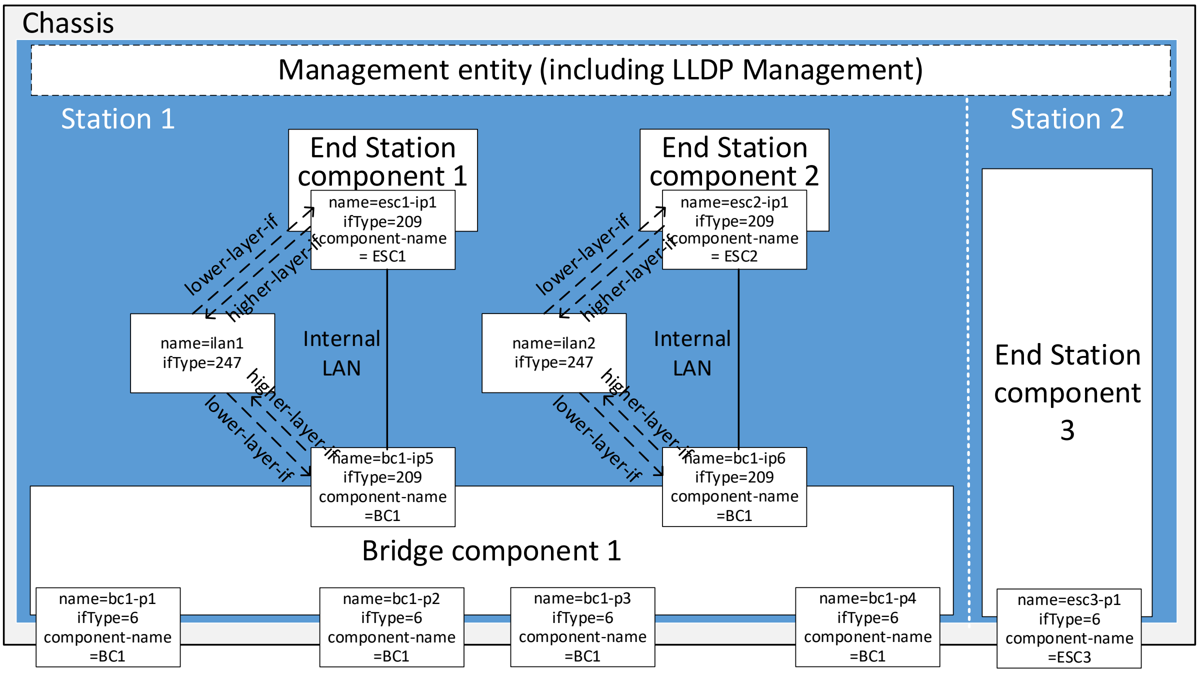

Figure 10 – two ES components connected to the same Bridge component

Figure 11 shows multiple end stations without Bridge component in one chassis:

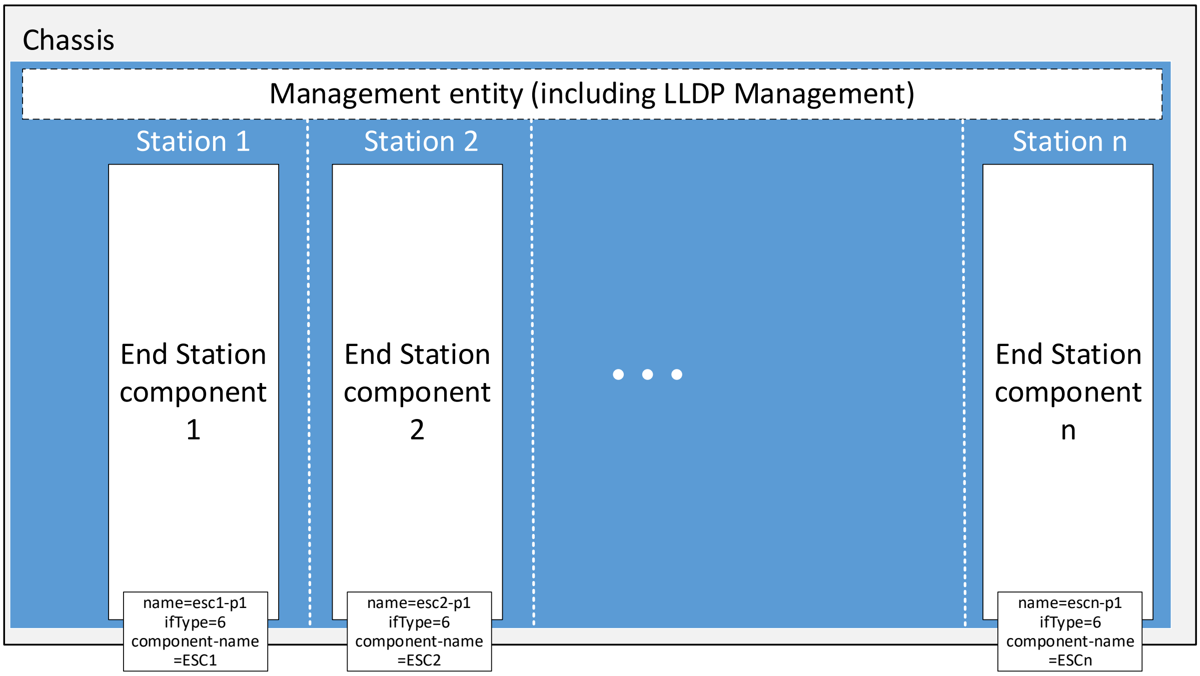

Figure 11 – multiple end station components in one chassis

Figure 12 shows a cascaded Bridge component with three not externally visible internal Bridge components:

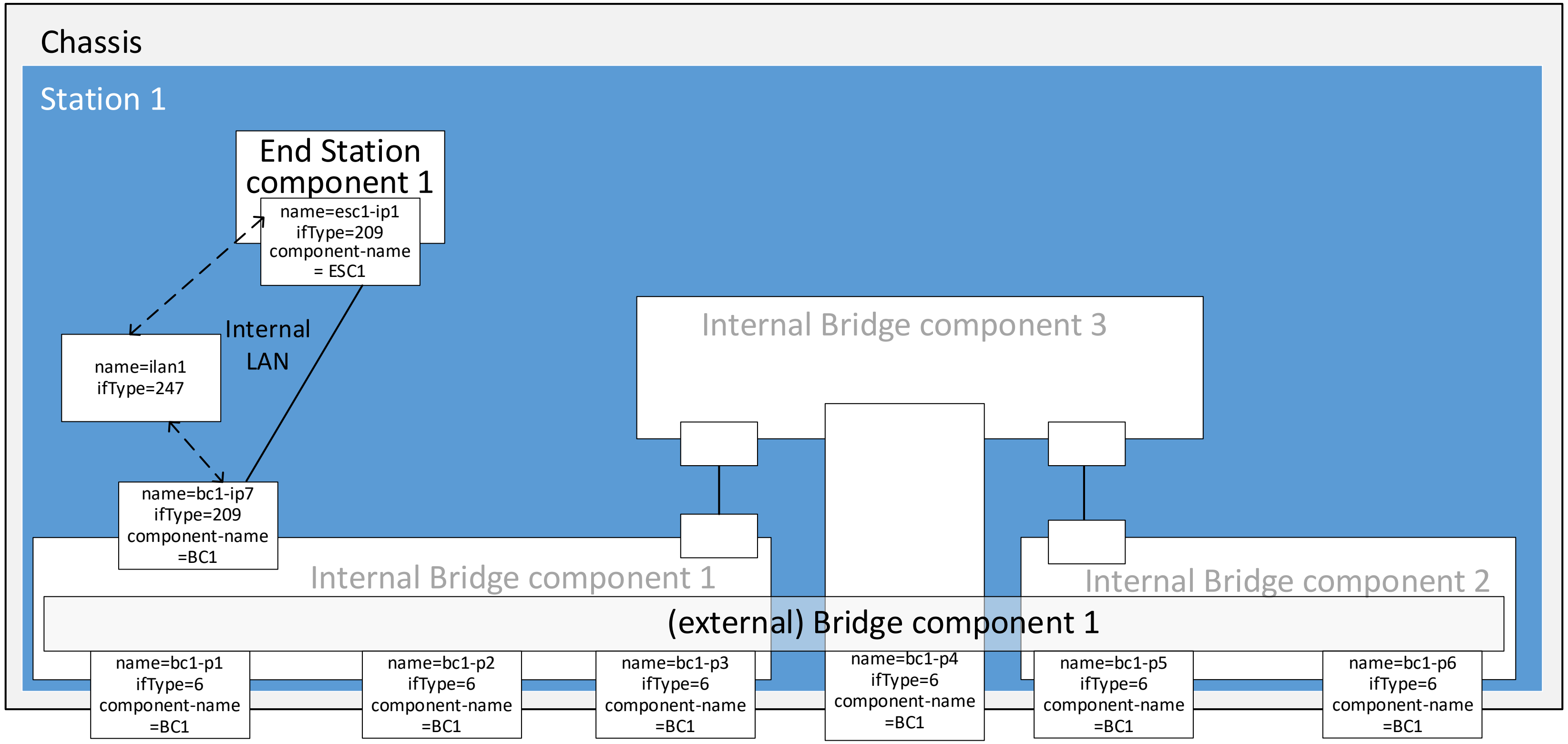

Figure 12 – cascaded Bridge with hidden internal Bridge components

< note: this will probably be the preferred option for cascaded Bridges>

Figure 13 shows three externally visible cascaded Bridge components:

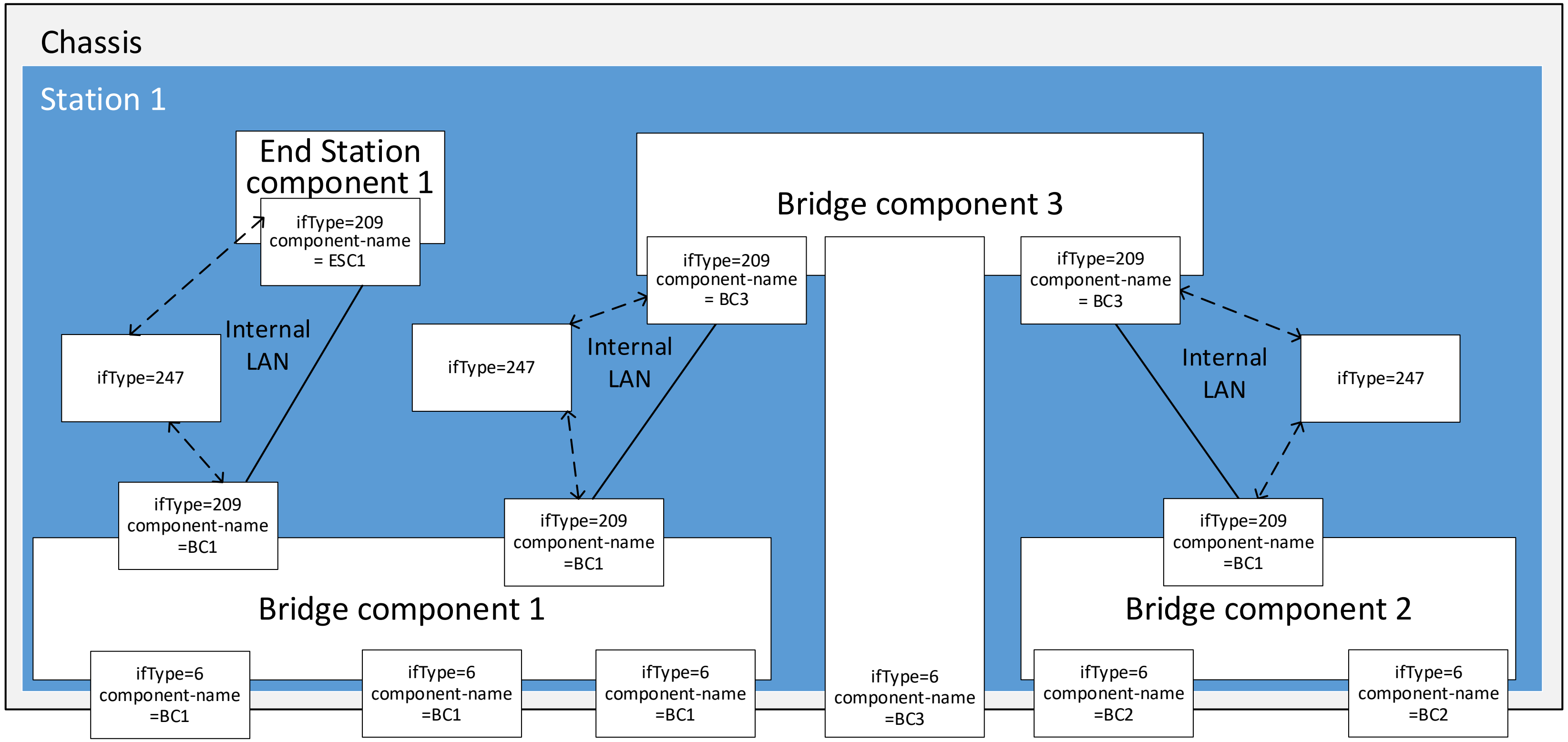

Figure 13 – cascaded Bridge components with exposed internal connectivity

4.8 Bridge Management of IA-device components

The `ieee802-dot1q-bridge` management model organizes a Bridge as a list of components with the component `name` as unique identification of the single components. Each component includes:

- a component type (e.g., `c-vlan-component`),
- component specific parameters (e.g., `traffic-classes`),
- a list of references to the component's ports in the `ietf-interfaces` list,

Port specific parameters (e.g., `traffic-class-table`) are located in the `ieee802-dot1q-bridge` augmentation of the `ietf-interfaces` list. Interfaces are uniquely addressed by `name`.

All components and all ports of an IA-device can effectively be identified and managed by the unique `names` of the components and the interfaces, which can be discovered by a Topology Discovery Entity (TDE) as described in 4.6.

4.9 Generic IEEE 802.1AS management model

For the management of IEEE 802.1AS time-aware-systems a MIB module is defined in IEEE 802.1AS-2020. The IEEE P802.1ASdn project to define a corresponding YANG data model that allows configuring and state reporting for all IEEE 802.1AS-2020 managed objects was created. This project is intended to augment the IEEE Std 1588 YANG data model, which is currently defined in the IEEE P1588e project. Experimental revisions of the 1588 model and the 802.1AS augmentation are currently available:

- https://github.com/YangModels/yang/blob/master/experimental/ieee/1588/ni-ieee1588-ptp.yang

- https://github.com/YangModels/yang/blob/master/experimental/ieee/802.1/ni-ieee802-dot1as.yang

In the following paragraph references to these experimental YANG models are included.

The 1588 YANG model essentially is constructed as a list of PTP Instances. Each entry in the PTP Instance list includes:

- the PTP Instance specific data sets,
- a list of associated PTP Ports with their PTP Port specific data sets.

Additionally, a single instance of the Common Mean Link Delay Service (cmlds) container is integrated. This container includes a list of PTP Ports with their cmlds specific objects and data sets.

Figure 14 shows the generic layout of the experimental 1588 YANG model (see https://www.ieee802.org/1/files/public/docs2020/dn-cummings-update-on-1588-and-802-1ASdn-1120-v01.pdf and https://github.com/YangModels/yang/blob/master/experimental/ieee/1588/ni-ieee1588-ptp.yang).

```
+--rw instance-list*       [instance-number]
      +--rw instance-number            uint32
      +--rw default-ds
      +--rw current-ds
      +--rw …
      +--rw port-ds-list* [port-number]
            +--rw port-number          uint16
            +--rw port-state           enumeration
            +--rw underlying-interface if:interface-ref
            +-- rw …
```

Figure 14 – IEEE 1588 YANG structure

A PTP Instance is identified by an instance-Number that is unique within the PTP Node.

A PTP Port is identified by a port-Number that is unique within a PTP Instance. Each port-ds-list entry includes a reference to the ietf interface.

The experimental 802.1AS YANG module keeps the structure of the 1588 YANG model and augments specific data sets with 802.1AS specific data where needed.

Figure 15 shows the example IA-device from Figure 1 with PTP Instance and PTP Port identifiers.

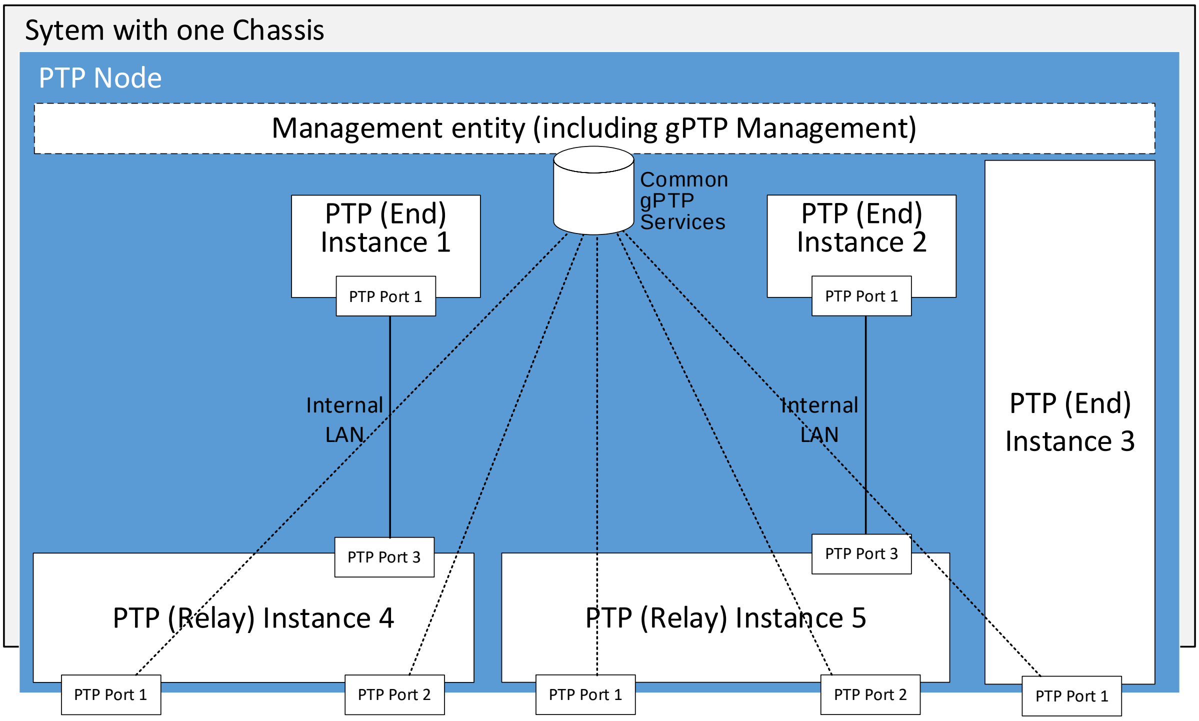

Figure 15 – IA-device example with ietf-interfaces

4.10 PTP Management of IA-device components

PTP Instances, which are identified by instance-numbers, are related to Bridge components, which are identified by names.

PTP Ports, which are identified by port-numbers, are related to Bridge and internal Ports, which are identified by names.

Open issues:

1. How can the PTP Instance numbers of a Bridge component be determined?
2. How can the PTP Port number of a Bridge or internal Port be determined?
3. How are internal Ports and LANs modelled, concerning:
    a. Path delay
    b. Engineered sync tree
    c. BMCA