

802.1Q

YANG bridge-address

Stephen Haddock

October 19, 2024

Introduction

- In the 802.1Q standard, the bridge-address is read-only.
 - Enables reading the value of a globally unique address allocated by the manufacturer and embedded in the bridge.
- In the YANG model, the bridge(name)/macAddress is read-write and mandatory.
 - Requires a NETCONF server (administrator) to enter a globally unique address to be used by the bridge before any other configuration can be done.
 - Where does this address come from?
- This seems like a problem! Is it?

802.1Q Bridge-Address

- In clause 8.13 Bridge Operation: Addressing.
 - **8.13.8 Unique identification of a Bridge**
A unique EUI-48 Universally Administered MAC address, termed the Bridge Address, shall be assigned to each Bridge. The Bridge Address may be the individual MAC address of a Bridge Port; in which case, use of the address of the lowest numbered Bridge Port (Port 1) is recommended.
 -
- In clause 12.4 Bridge Management Entity.
 - Specifies a number of “management operations” that can be performed, each with “inputs” that are sent to the bridge, and “outputs” provided by the bridge.
 - The Bridge Address appears in the outputs, not the inputs.
 - I interpret this to mean it is read-only (there is no operation that sets the Bridge Address).
 - 802.1cp-2018 has Table 48-6 that cross-references managed objects to YANG nodes that says the Bridge Address managed object is “r-w” with a reference to 12.4.
- In Clause 17.7.2 Bridge MIB
 - ieee8021BridgeBaseBridgeAddress is read-create

YANG bridge model

```
container bridges {
  description
    "Contains the Bridge(s) configuration information.";
  list bridge {
    key "name";
    unique "address";
    description
      "Provides configuration data in support of the Bridge Configuration
      resources. There is a single bridge data node per Bridge.";
    leaf name {
      type dot1qtypes:name-type;
      description
        "A text string associated with the Bridge, of locally determined
        significance.";
      reference
        "12.4 of IEEE Std 802.1Q";
    }
    leaf address {
      type ieee:mac-address;
      mandatory true;
      description
        "The MAC address for the Bridge from which the Bridge Identifiers
        used by the STP, RSTP, and MSTP are derived.";
      reference
        "12.4 of IEEE Std 802.1Q";
    }
  }
}
```

What happened?

- This is pure speculation, but it appears this is the result of trying to create a list of bridges with unique addresses.
 - The use of “unique macAddress” in the list requires that the macAddress is config-true and mandatory-true.
 - Why do we have a list of bridges, rather than single leaf, in the first place?
 - Maybe idea is that you can sub-divide a bridge into multiple smaller bridges? Note that each bridge already contains a list of bridge components.

Can anything be done?

- I think what we want is either:
 1. macAddress is “config false”
 - Bridge only uses the address provided by the manufacturer.
 2. macAddress is “config true” but “mandatory false”
 - Provides ability to overwrite the initial value provided by the manufacturer.
- The significant change is to remove the “unique” statement
 - Assume that if have multiple entries in the list, the bridge is smart enough to provide different addresses.
 - If “config true” and a configuration overwrites the address, the configuration assumes responsibility for maintaining uniqueness.
 - I believe this change would be backwards compatible.
 - Could avoid the whole uniqueness issue if change the list to a leaf, but this would not be backwards compatible.

Back up slides

Thank You