

# LLDP Extensions in 802.3

Leveraging 802.1ABcu for LLDP Extensions

Introduction and Examples

Scott Mansfield (Ericsson)

2023-05-09

# IEEE Std 802.1ABcu-2019

- Defines the YANG for IEEE Std 802.1AB
- The YANG is found in two modules
  - `ieee802-dot1ab-lldp.yang`
  - `ieee802-dot1ab-types.yang`
- The YANG is designed to be augmented when Extension TLVs are defined.
- For "system" level TLVs augment the `lldp:lldp` branch
- For "port" level TLVs augment the `lldp:port` branch

# Example from 802.1

- IEEE Std 802.1-2022 defines a set of "basic set" of tlvs in Clause D.1
- There is a bit string that allows each TLV to be enabled
- For each TLV there is a container that uses a YANG grouping that has the details of the information needed by the TLV

```
grouping port-vlan-id-tlv {  
  description  
    "Port VLAN ID TLV";  
  reference  
    "D.2.1 of IEEE Std 802.1Q";  
  leaf port-vlan-id {  
    type dot1qtypes:vlanid;  
    description  
      "Advertise port VLAN identifier, Send 0 if unknown or unsupported";  
    reference  
      "D.2.1 of IEEE Std 802.1Q";  
  }  
}
```

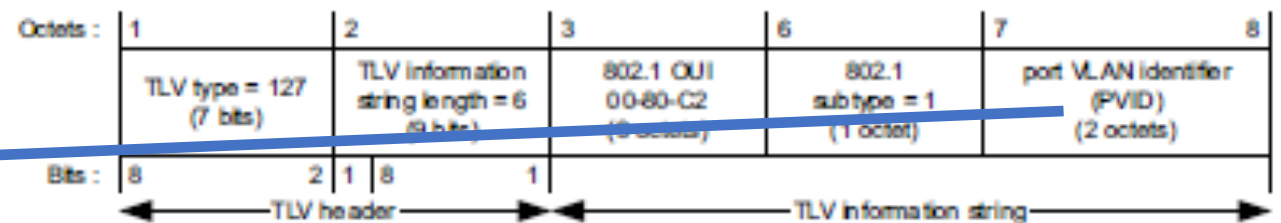


Figure D-1—Port VLAN ID TLV format

# YANG version of LLDP-EXT-DOT3-MIB

- For example in IEEE Std 802.3 this MIB OBJECT would turn into (something like this) in YANG

```
lldpV2Xdot3PortConfigTLVsTxEnable OBJECT-TYPE
    SYNTAX      BITS {
        macPhyConfigStatus(0),
        powerViaMDI(1),
        unused(2), --avoids re-use of the old link agg bit number
        maxFrameSize(3)
    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The lldpV2Xdot3PortConfigTLVsTxEnable, defined as a bitmap,
        includes the IEEE 802.3 organizationally defined set of LLDP
        TLVs whose transmission is allowed by the local LLDP agent by
        the network management. Each bit in the bitmap corresponds
        to an IEEE 802.3 subtype associated with a specific IEEE
        802.3 optional TLV.

        The bit 'macPhyConfigStatus(0)' indicates that the LLDP agent
        should transmit 'MAC/PHY configuration/status TLV'.

        The bit 'powerViaMDI(1)' indicates that the LLDP agent should
        transmit 'Power via MDI TLV'.

        The bit 'unused(2)' is no longer used; this was used for
        the 'Link Aggregation TLV' in the previous version.

        The bit 'maxFrameSize(3)' indicates that the LLDP agent should
        transmit 'Maximum-frame-size TLV'.

        The default value for lldpV2Xdot3PortConfigTLVsTxEnable object
        is an empty set, which means no enumerated values are set.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system."
    REFERENCE
        "IEEE Std 802.3 30.12.1.1.1"
    DEFVAL      { {} }
    ::= { lldpV2Xdot3PortConfigEntry 1 }
```

```
augment "/lldp:lldp/lldp:port" {
    description
        "Augments port with port config tlvs";
    leaf tlvs-port-config-enable {
        type bits {
            bit mac-phy-config-status {
                position 0;
                description
                    "30.12.1.1.1 of IEEE Std 802.3-2022";
            }
            bit power-via-mdi {
                position 1;
                description
                    "30.12.1.1.1 of IEEE Std 802.3-2022";
            }
            bit unused {
                position 2;
                description
                    "30.12.1.1.1 of IEEE Std 802.3-2022";
            }
            bit max-frame-size {
                position 3;
                description
                    "30.12.1.1.1 of IEEE Std 802.3-2022";
            }
        }
    }
    description
        "Bitmap that corresponds to an IEEE 802.3 subtype associated
        with a specific IEEE 802.3 port config TLVs";
    reference
        "30.12.1.1.1 of IEEE Std 802.3-2022";
}
```

# Other YANG Needed

- If the TLVs themselves need to be described in YANG, the container/grouping style used in 802.1Qcz can be used
- If all that is needed from 802.3 is to indicate that the TLV is enabled or not
  - Then modeling the bit strings as shown in the example is what is needed
  - Finding where to anchor the bit strings should be evaluated
- Determine if there any issues with using 802.1AB and 802.1ABcu as is or if some other modification is necessary