

# Support of P802.1Xck, and more, on non-IEEE 802- specified interfaces

- Ludwig Pauwels
- 25 October 2017

# What is this contribution about?

IEEE 802 defines requirements for Ethernet interfaces and bridges

- At the physical layer
- For protocols that run on top of the physical layer

IEEE 802 also defines the management models for these requirements and specifies the management protocol-dependent models/syntax, e.g. YANG.

In the industry, other standardization organizations specify physical layer technologies. Some of these technologies define the transport of Ethernet frames, sometimes at a layer above the physical layer (\*)

We need the corresponding management models.

- The way of transporting Ethernet frames is different, hence this results in a different model / YANG
- For the layer transporting Ethernet frames, some protocols specified by IEEE 802 can be used
  - ➔ The question: can the IEEE 802-specified YANG models be used to manage protocols specified by IEEE 802 but running on top of a transport technology specified by other standardization organizations?

(\*) see slide 5 and backup for an example

# The YANG situation explained with an example

# Background on YANG modeling techniques

In a YANG model, data is defined for interfaces as either:

- Unconditional: applies to all interfaces; or
- Conditional by using “when” statement: applies to only the specified set of interfaces

An example for such YANG statement is:

```
augment "/if:interfaces/if:interface" {  
    when "if:type = :'ianaifethernetCsmacd' "" {
```

Consequences for data applicability:

Name	Type	enabled	...	Duplex	...	Configured-mode	...
Ethernet1	ethernetCsmacd	true	...	full	...	Not applicable	...
FASTline-user1	fastdsl	true	...	Not applicable	...	Mode-vdsl	...

Data defined without ‘when’ conditions, e.g. IETF RFC 7223

Data defined with “when= ethernetCsmacd”

Data defined with “when= fastdsl”

# Example for non-IEEE interface: VDSL2 running in “Packet Transfer Mode”

VDSL2 is specified in ITU-T G.922.3.

This specification specifies the transport of Ethernet frames as one of its modes (\*). This is called the “Packet Transfer Mode” (PTM) and is specified in G.922.3 annexes K.3 and N.

The corresponding YANG model is specified in BBF TR-355 and BBF TR-383.

Because the PTM mode is just one of the supported modes, the model splits the physical layer (BBF TR-355) from what is being transported (part of BBF TR-383).



- Data relating to the parsing of this bitstream as Ethernet frames is modeled as a higher layer “ptm” interface.
- Data relating to the transport of a bitstream over copper wire is modeled through an interface of “fastdsl” type

Name	Type	enabled	...	queues	...	Configured-mode	...
Ptm-user1	ptm	true	...	List of queues	...	Not applicable	...
FASTline-user1	fastdsl	true	...	Not applicable	...	Mode-vdsl	...

(\* The other mode defines the transport of ATM cells. See backup for some more information

# Applicability of IEEE 802 specifications to VDSL2-PTM interfaces

Applicable:

- IEEE 802.3 Frame structure, including the use of 802.1Q VLAN tags
- IEEE 802.3ah EFM OAM
- IEEE 802.1X
- (\*)

Not applicable:

- Physical layer aspects such as half/full duplex
- Link Aggregation
- (\*)

(\*) It is not the intention to be complete, the importance is to notice “some IEEE specifications apply” but “not all IEEE specifications apply”. Some references to applicability requirements are given in backup.

Next slide explains the 802.1Xck situation for the VDSL2 interface in PTM mode.

# P802.1Xck/D1.1: current YANG syntax and consequences for VDSL2-PTM

P802.1Xck/D1.1 contains the following statements:

```
augment "/if:interfaces/if:interface" {  
  when "if:type = 'ianaif:ethernetCsmacd' or if:type = 'ianaif:ilan'" {  
    description "Applies to the Controlled Port of SecY or PAC shim."; }  
augment "/if:interfaces-state/if:interface" {  
  when "if:type = 'ianaif:ethernetCsmacd' or if:type = 'ianaif:ilan'" {  
    description "Applies to the Controlled Port of SecY or PAC shim."; }  
}
```

Interface data				802.1X data		
Name	Type	enabled	...	Pae/pae-system	Pae/vp-enable	...
Ethernet1	ethernetCsmacd	true	...	A-name-value	false	x
FASTline-user1	fastdsl	true	...	NA	NA	NA
PTMoverFast1	<b>ptm</b>	true	...	<b>NA</b>	<b>NA</b>	<b>NA</b>

By YANG syntax definition, this data does not exist for PTM interfaces  
**But... this data is applicable!**

# Solutions and their impact on YANG modeling



# Alternative 1: add PTM interfaces to the when statements in P802.1Xck

Supporting 802.1X on PTM interfaces only requires adding “ptm” as an interface type to the “when” conditions in the P802.1Xck YANG data model.

For example:

```
augment "/if:interfaces/if:interface" {  
  when "if:type = 'ianaift:ethernetCsmacd' or  
        if:type = 'ianaift:ilan' or  
        if:type = 'ianaift:ptm'" {
```

Interface data				802.1X data		
Name	Type	enabled	...	Pae/pae-system	Pae/vp-enable	...
Ethernet1	ethernetCsmacd	true	...	A-name-value	false	x
FASTline-user1	fastdsl	true	...	NA	NA	NA
PTMoverFast1	<b>ptm</b>	true	...	A-name-value	false	x

# Is alternative 1 good enough?

In the industry, there are multiple physical layer technologies defined within multiple standardization organizations.

BBF TR-178 requires support of IEEE 802.1X on top of its access lines. For now, this includes amongst others:

- VDSL2 interfaces per the ITU-T G.993.2 Annex K.3 / Annex N (PTM mode) and G-FAST interfaces per the ITU-T G.9701 for use over copper wires
- GPON Interfaces per the ITU-T G.984 series, which is a point-to-multipoint fiber technology
- VDSL2 interfaces per ITU-T G.993.2 Annex K.2 (ATM mode)
- Ethernet interfaces per IEEE 802.3.

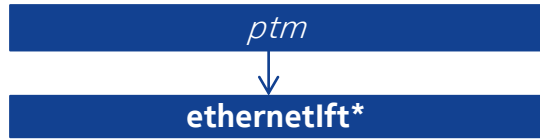
The GPON interfaces and interfaces in ATM mode will not be modeled in the interface list with type “ethernetCsmacd”, nor will it be with type “ptm”; this is because the applicability of the data will not be 100% identical.

Consequence, at the moment such interface stack has concluded other interface types will be used and the “when” statement needs to be adapted.

**→ Alternative 1 is not good enough**

## Alternative 2: use YANG 1.1 syntax and define the type PTM with inheritance

Define identity with inheritance:



← Redefine the identity in IANA, using inheritance

← New identity in IANA

(\*) Suggested name

Use YANG 1.1 syntax in the data models, this impacts on 802.1Xck:

```
augment "/if:interfaces/if:interface" {  
  when "if:type = 'ianaif:ethernetCsmacd' or if:type = 'ianaif:ilan'" or  
    derived-from-or-self(if:type, 'ianaif:ethernetlft')" {
```

Derived-from-or-self is a YANG 1.1 construct that means: **the data specified in this augment is applicable to:**

- All interfaces that are of the type ethernetlft (which will be none)
- **All interfaces that have a type that is derived from ethernetlft.** (This means the data is applicable for interfaces of the type "ptm".)

➔ Any future type defined based on ethernetlft automatically contains all 802.1x data and does not impact 802.1Xck

## Alternative 2: impact in IANA

Now:

- Add / redefine identities

```
identity ethernetlft {  
  base iana-interface-type;  
  description  
    "An ethernet interface. This is a base type that represents a logical interface transporting Ethernet frames, i.e. frames with  
    a destination and source MAC address, an Ethernet type or length field and a payload. This type is for use in modules that  
    specify data that is common for all Ethernet frame transporting technologies.  
    Specific derived interface types are defined for transporting technology specific data and for use in actual data  
    instantiations. ";  
}
```

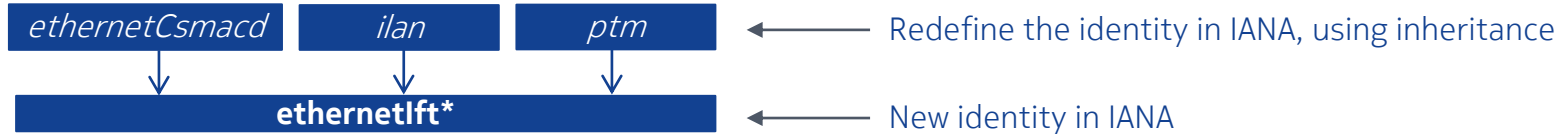
```
identity ptm {  
  base ethernetlft;  
  description  
    "Packet Transfer Mode.";  
}
```

Future:

- Define new identities with base ethernetlft

## Alternative 3: simplify the data models and use inheritance for all types

Define identity with inheritance



(\*) Suggested name

Use YANG 1.1 syntax in the data models, this impacts on 802.1Xck:

```
augment "/if:interfaces/if:interface" {  
  when "derived-from-or-self(if:type, 'ianaift:ethernetlft')" {
```

I.e. it is no longer needed to specify ethernetCsmacd and ilan as both are defined as inheriting from ethernetlft

➔ Any future type defined based on ethernetlft automatically contains all 802.1x data and does not impact 802.1Xck

# Alternative 3: impact on IANA

Now:

- Add / redefine identities

```
identity ethernetlft {  
  base iana-interface-type;  
  description
```

“An ethernet interface. This is a base type that represents a logical interface transporting Ethernet frames, i.e. frames with a destination and source MAC address, an Ethernet type or length field and a payload. This type is for use in modules that specify data that is common for all Ethernet frame transporting technologies.

Specific derived interface types are defined for transporting technology specific data and for use in actual data instantiations.  
”;

```
}
```

```
identity ethernetCsmacd {  
  base ethernetlft;  
  description
```

“For all Ethernet-like interfaces, regardless of speed, as per RFC 3635.”;

```
reference
```

“RFC 3635 - Definitions of Managed Objects for the Ethernet-like Interface Types”;

```
}
```

```
identity ptm {  
  base ethernetlft;  
  description
```

“Packet Transfer Mode.”;

```
}
```

And similarly for identity “ilan”

Future:

- Define new identities with base ethernetlft

## Alternative 2 and 3: impact on other IEEE 802 data models

All IEEE 802 models must be carefully specified with the right “when” statements, and similarly with the right “must” statements. If not consistently done for all, then the effort has little value.

Use of these identities in YANG data models:

- ieee802-ethernet-interface.yang: models the **physical** Ethernet interfaces
  - ➔ Define when/must statements using ethernetCsmacd, i.e. no impact
- ieee802-dot1ax.yang: link aggregation is defined for **physical** Ethernet interfaces
  - ➔ Define when/must statements using ethernetCsmacd, i.e. no impact
- ieee802-dot1x.yang, applies also to **logical** “ethernetIft” interfaces that are non-IEEE specified
  - ➔ Define when/must statements using “ethernetIft”, i.e. impacted

Future models

- EFM OAM: applies also to **logical** “ethernetIft” interfaces that are non-IEEE specified
  - ➔ Define when/must statements using “ethernetIft”

# Impact on P802.1Xck



## Summary: what can we do to support PTM interfaces in P802.1Xck?

**Alternative 0:** leaves P802.1Xck unchanged, consequence:

- The P802.1Xck data model cannot be used industry-wide as it excludes some “EthernetIft” interfaces such as PTM
- Every organization that specifies the use of 802.1X on a particular Ethernet interface is forced to specify its own YANG data model for it

**Alternative 1:** extends the “when” statement with “ptm” interface:

- This is not future-proof as needs more changes for other transport technologies

**Alternatives 2 or 3:** redefine existing types in IANA and update YANG data models with the proper YANG (syntax 1.1) statements

- This solves the issues above
- This is future-proof: new technologies only need the definition of new IANA identities

➔ **Proposal: alternatives 2 or 3 based on IEEE preference**

**NOKIA**

# Backup

## Example for non-IEEE interface: VDSL2 running in “ATM Mode”

VDSL2 is specified in ITU-T G.922.3. Annex K.2 specifies the transport of ATM cells.

ITU-T I.363.5 specifies an ATM Adaptation Layer, i.e. how to transport frames over ATM

The type of frame can be various, e.g. transport an Ethernet frame, or transport an IP frame. The way this frame is transported depends on the type of frame.

For ethernet frames this can be:

- Use LLC/SNAP bridged encapsulation technique as defined in IETF RFC 2684

The method has two sub-variants:

- Transport the Ethernet frame with padding and FCS
- Transport the Ethernet frame without padding and FCS
- Use VC-Mux bridged encapsulation as defined in IETF RFC 2684

# Protocol requirements for VDSL2-PTM / G.FAST interfaces

- BBF TR-178
  - Requires the use of IEEE 802.1X on access lines  
[R-145] The EAN MUST support IEEE802.1x.
- BBF TR-301
  - Requires support of EFM OAM on G.FAST lines (IEEE 802.3ah / 802.3 clause 57)  
R-107 The DPU MUST be able to establish an IEEE 802.3ah EFM OAM session with the G.fast NT Module/CPE (i.e. OAM discovery and exchange of state and configuration information).  
R-108 The DPU MUST support IEEE802.3ah OAM clause 57.2.9 active mode.
  - In existing access networks EFM OAM is supported also on VDSL2 lines