NOKIA

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

- Jessy Rouyer and Ludwig Pauwels

- 7 November 2017

What is this contribution about?

IEEE 802 specifies requirements and management models for Ethernet interfaces at the physical layer and for protocols running on top of the physical layer in bridges. IEEE 802 also specifies YANG data models specific to these protocols.

In the industry, other standards organizations than IEEE 802 specify physical layer technologies. Some transport Ethernet frames at a layer above the physical layer (*) using techniques that differ from 802.3. Some protocols specified by IEEE 802 can be used at this layer.

This contribution seeks to answer the following questions: can IEEE 802-specified YANG data models be used to manage IEEE 802-specified protocols running on top of a transport technology specified by other standardization organizations? If so, how?

NOKIA

(*) See slide 5 for examples

The YANG situation explained with an example

Background on YANG modeling techniques

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

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

An example of such “when” statement is:

```
augment "/if:interfaces/if:interface" {
    when "if:type = ‘ianaift:ethernetCsmacd’” {
```

Consequences on data applicability:

Data defined without “when” (e.g. per RFC 7223)

Data defined with “ethernetCsmacd” in “when”

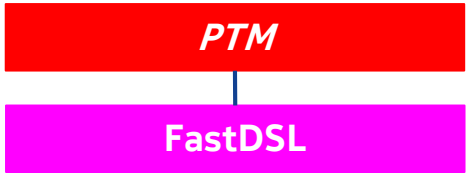
Data defined with “ptm” in “when”

Name	Type	enabled	…	duplex	…	queues	…
Ethernet1	ethernetCsmacd	true	…	full	…	Not applicable	…
PTMoverFast1	ptm	true	…	Not applicable	…	List of queues	…

Example of non-IEEE 802 interface: VDSL2 in “Packet Transfer Mode”

ITU-T G.993.2 specifies VDSL2, the transport of Ethernet frames being one of its modes (*) called the “Packet Transfer Mode” (PTM, as specified in G.992.3 annexes K.3 and N). BBF TR-355 and BBF TR-383 specify the corresponding YANG data model. Because PTM is just one of the supported modes, the YANG data model splits the physical layer (BBF TR-355) from what it transports (part of BBF TR-383).

This modeling is akin to that of an 802.3 Ethernet interface but without 802.3’s physical layer aspects.



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

				“ptm” in “when”		“fastdsl” in “when”	
Name	**Type**	**enabled**	**…**	**queues**	**…**	**configured-mode**	**…**
PTMoverFast1	ptm	true	…	List of queues	…	Not applicable	…
FASTline-user1	fastdsl	true	…	Not applicable	…	mode-vdsl	…

(*) The other mode defines the transport of ATM cells

NOKIA

Applicability of IEEE 802 specifications to VDSL2 interfaces in PTM

Some IEEE 802 specifications apply to VDSL2 interfaces in PTM (“VDSL2-PTM”):

- IEEE 802.1X (per BBF TR-178)
- IEEE 802.3 (frame structure, including the use of 802.1Q VLAN tags)
- IEEE 802.3ah (EFM OAM)
- (*)

But some don’t apply:

- IEEE 802.1AX Link Aggregation (combining multiple interfaces through Link Aggregation is irrelevant for VDSL2 interfaces as can be achieved through “bonding” of multiple FastDSL interfaces)
- IEEE 802.3’s physical layer aspects such as half/full duplex
- (*)

(*) There is no attempt to be exhaustive. BBF TR-178 and TR-301 provide requirements on IEEE 802 protocols for VDSL2-PTM / G.FAST interfaces

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."; }
```

Without a solution and by YANG syntax definition, 802.1X data does not exist for PTM interfaces but applies to them!

Interface data				802.1X data			
Name	**Type**	**enabled**	**…**	**pae/pae-system**	**pae/vp**	**nable**	**…**
Ethernet1	ethernetCsmacd	true	…	A-name-value	false		…
PTMoverFast1	ptm	true	…	**N/A**	**N/A**		**N/A**
FASTline-user1	fastdsl	true	…	N/A	N/A		N/A

NOKIA

Possible alternative solutions and their impact on YANG data 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'" {
```

802.1X data now exists for PTM interfaces!

Interface data				802.1X data			
Name	**Type**	**enabled**	**…**	**pae/pae-system**	**pae/vp**	**nable**	**…**
Ethernet1	ethernetCsmacd	true	…	A-name-value	false		…
PTMoverFast1	ptm	true	…	**A-name-value**	**false**		**…**
FASTline-user1	fastdsl	true	…	N/A	N/A		N/A

NOKIA

Is alternative 1 good enough?

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

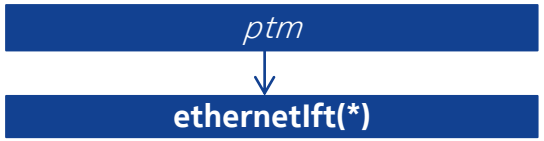
BBF TR-178 requires support of IEEE 802.1X on access lines. For now, this includes amongst others:

- Ethernet interfaces (per IEEE 802.3)
- GPON interfaces (per the ITU-T G.984 series), which is a point-to-multipoint fiber technology
- VDSL2 interfaces in ATM mode (per ITU-T G.992.3 Annex K.2)
- VDSL2 interfaces in PTM and G-FAST interfaces (per ITU-T G.9701) for use over copper wires

GPON interfaces and VDSL2 interfaces in ATM mode will not be modeled with either “ethernetCsmacd” or “ptm” interface types as the applicability of the data will not be 100% identical. As a consequence, for the moment, other interface types will be used and the “when” statement will need to be adapted.

➔ **Alternative 1 is not good enough**

Alternative 2: redefine PTM identity via inheritance and use YANG 1.1 construct



Redefine the PTM identity in IANA via inheritance

Create a new ethernetIft(*) identity in IANA

Use the derived-from-or-self YANG 1.1 construct in P802.1Xck to define the PTM identity via inheritance from the new ethernetIft(*) identity:

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

derived-from-or-self means that the data specified in the **augment** is applicable to :

- All interfaces of the ethernetIft type (there would be no such interfaces); and
- All interfaces of a type derived from ethernetIft (the data would apply to interfaces of the “ptm” type)

➔ **Any future type derived via inheritance from ethernetIft automatically contains all 802.1X data and there is no need to update 802.1Xck to support it**

(*) Suggested name

NOKIA

Alternative 2: impact on IANA

Now:

Add and redefine identities:

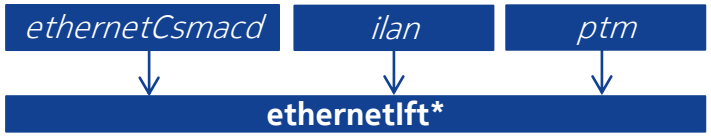
```
identity ethernetIft {
  base iana-interface-type;
  description
    "An Ethernet interface. This is a base type that represents a logical interface transporting Ethernet frames, i.e. frames
     with a destination and source MAC address, an Ethernet type or length field and a payload. This type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

Future:

Define new identities based on ethernetIft

NOKIA

Alternative 3: redefine **all** identities via inheritance and use YANG 1.1 construct



Redefine all identities in IANA via inheritance

Create a new ethernetIft(*) identity in IANA

Same as alternative 2 but for all identities, use the derived-from-or-self YANG 1.1 construct in P802.1Xck to define all identities via inheritance from the new ethernetIft(*) identity:

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

There is no longer a need to specifically include ethernetCsmacd and ilan interfaces in the “when” statement as both are redefined as inheriting from ethernetIft (as are ptm interfaces).

➔ **As in alternative 2, any future type derived via inheritance from ethernetIft automatically contains all 802.1X data and there is no need to update 802.1Xck to support it. The YANG is also simpler.**

(*) Suggested name

Alternative 3: impact on IANA

Now:

Add and redefine identities:

```
identity ethernetIfft {
  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 specified for transporting technology-specific data and for use in actual data
     instantiations.";
}
identity ptm {
  base ethernetIfft;
  description
    "Packet Transfer Mode.";
}
identity ethernetCsmacd {
  base ethernetIfft;
  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";
}
/*  And similarly for identity “ilan” */
```

Future:

Define new identities based on ethernetIfft

NOKIA

Alternatives 2 and 3: impact on other IEEE 802 YANG data models

All IEEE 802 YANG data 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 **physical** Ethernet interfaces
  → No impact as defines when/must statements using ethernetCsmacd
- **ieee802-dot1ax.yang**: models Link Aggregation for **physical** Ethernet interfaces
  → No impact as defines when/must statements using ethernetCsmacd
- **ieee802-dot1x.yang**: also applies to **logical** “ethernetIft” interfaces that are non-IEEE specified
  → Impact as should define when/must statements using “ethernetIft”

Potential future YANG data model

- **For 802.3ah EFM OAM**: also applies to **logical** “ethernetIft” interfaces that are non-IEEE specified
  → Should define when/must statements using “ethernetIft”

Alternative 4: draft-wilton-netmod-interface-properties-00

Alternative shared by Robert Wilton as a follow up to 25 October 2017 presentation on Security TG and YANGsters calls of a first version of this contribution:

- Addresses exactly the same issue as this contribution
- Also uses YANG 1.1 inheritance for defining interface types
- Differs from alternatives 2 or 3 in how to define what to inherit from:
  - Alternatives 2 or 3 propose the new “ethernetItf” interface type from which other interface types can inherit
  - Draft Wilton proposes a base “iana-if-property-type” set of interface properties (“ethernet-like”, “physical”, “virtual”, “point-to-point”, etc., being some of them), from which interface types can inherit
- Gives examples needing discussion among organizations defining interface types and YANG modules
  - For example, the definition of the “point-to-point” property suggests that on a point-to-point interface there are neither broadcast nor multicast packets, which contradicts the 802.1-specified use of PDUs with a group address on such interfaces.

➔ **Generic approach but good definitions of properties, interface types and modules will take work / coordination time among the relevant standards organizations**

Impact on P802.1Xck

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

Alternative 0: leave P802.1Xck unchanged, consequence:

- The P802.1Xck YANG data model cannot be used industry-wide as it excludes some “ethernetIft” interfaces such as PTM
- Every standards organization that specifies the use of 802.1X on a non-IEEE 802-specified Ethernet interface is forced to specify its own YANG data model for it

Alternative 1: extend 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

- These solve the issues above, and are redundant yet backward compatible with alternative 1
- These are future-proof: new technologies only need the definition of new IANA identities

Alternative 4: pursue draft-wilton-netmod-interface-properties-00

- This has the same advantages as alternatives 2 or 3 but is a more generic approach, the difficult and time consuming part being in defining properties fitting multiple transport technologies, the interface types using these properties, and the YANG modules with these properties

Comparison of alternative solutions and suggestion

Alternative →	0: do nothing	1: add "PTM"	2: add "ethernetItf" and redefine "ptm"	3: as 2 + redefine "ethernetCsmacd" / "ilan"	4: Draft Wilton
Can module be used on PTM interfaces?	NO	YES	YES	YES	YES
Future-proof for other non-IEEE 802 interfaces?	NO	NO	YES	YES	YES
Can ieee802-dot1x.yang remain in YANG 1.0 syntax?	YES	YES	NO	NO	NO
Can be done with existing IANA type definitions?	YES	YES	NO	NO	NO
Can iana-if-type.yang remain in YANG 1.0 syntax	YES	YES	YES	YES	NO
Estimated time to get solution in place	0	0	WEEKS	WEEKS	LONG TERM

➔ **While alternative 1 is acceptable in P802.1Xck, one of alternatives 2, 3 or 4, once in place, should be preferably used as needed in this or other 802 YANG modules going forward**


NOKIA