A YANG Data Model for Network Inventory

draft-ietf-ivy-network-inventory-yang-07

Authors:

Chaode Yu (yuchaode@huawei.com)

Sergio Belotti (sergio.belotti@nokia.com)

Jean-Francois Bouquier (jeff.bouquier@vodafone.com)

Fabio Peruzzini (fabio.peruzzini@telecomitalia.it)

Phil Bedard (phbedard@cisco.com)

Contributors:

Italo Busi (italo.busi@huawei.com) [Presenter]

Aihua Guo (aihuaguo.ietf@gmail.com)

Oscar Gonzalez de Dios (oscar.gonzalezdedios@telefonica.com)

Victor Lopez (victor.lopez@nokia.com)

Chenfang Zhang (zhangcf80@chinaunicom.com)

Nigel Davis (ndavis@ciena.com)

Bo Wu (lana.wubo@huawei.com)

Updates Since IETF 122

- Clarified how it is possible to extend the base inventory model
  - ✓ Defining new inventory objects (e.g., network inventory location)
  - ✓ Defining new types of NEs (e.g., virtual NEs)
  - ✓ Defining new types of components (e.g., SW components)
- Updated YANG model to support:
  - ✓ Removed component-specific containers
  - ✓ Added is-main attribute to indicate main chassis in multi-chassis Nes
  - ✓ Added a new re-usable grouping with component attributes
  - ✓ Split the common entity grouping into two groupings:
    - Basic common entity attributes: grouping reusable in other models
    - NE and component common attributes
- Completed appendixes providing some examples of ports, transceivers and breakouts configurations as well as examples of multi-chassis NEs
  - ✓ JSON instance examples are also provided
- Editorial improvements and fixed nits

Summary of the Open Issues (Priority 1)

- Attributes of Network Elements (#112)
  - ✓ System (software) name at NE (device) level (#44)
  - ✓ Attributes from RFC7317 (#2)
- Modelling of slots (#111)
  - ✓ Child and parent component navigation (#3)
- Breakout Port is potentially a logical inventory consideration (#126)
- Definition of port (#125)
- Impact on RFC8348 and IANA Hardware (#127)
  - ✓ Still to be assessed
- Definitions of chassis (#121) and slot/sub-slot (#79)
  - ✓ Addressed in the -08 update (submitted after IETF 123 cut-off)
- Security Considerations
- Editorial clean-up

Modelling of the slot (#111): 1/2

```
- chassis-A
  - class: chassis
- slot-A1
  - class: container
  - parent: chassis-A
  - parent-position: 1
- slot-A2
  - class: container
  - parent: chassis-A
  - parent-position: 2
- slot-A3
  - class: container
  - parent: chassis-A
  - parent-position: 3
- slot-A4
  - class: container
  - parent: chassis-A
  - parent-position: 4
- slot-A5
  - class: container
  - parent: chassis-A
  - parent-position: 5
- card-X
  - class: module
  - parent: slot-A1
- card-Y
  - class: module
  - parents: [
    slot-A3,
    slot-A4
   ]
```

Current Model

```
- chassis-A
  - class: chassis
  - slots: [
    1,
    2,
    3,
    4,
    5
   ]
- card-X
  - class: module
  - parent: chassis-A
  - occupied-slots: [
    1
   ]
- card-Y
  - class: module
  - parent: chassis-A
  - occupied-slots: [
    3,
    4
   ]
```

Proposed Alternative Model

Modelling of the slot (#111): 2/2

- Advantages of the proposed alternative model
  - ✓ Closer to the reality of the physical thing where the slots are part of the component and have no separated lifecycle (TMF holder concept)
  - ✓ Significantly reduce the number of component instances in the component list
  - ✓ Removes confusion about properties defined for the components but not applicable to the slots/containers
- Concerns
  - ✓ The container component class is already defined in IANA Hardware and supported by RFC8348
  - ✓ Less component instances, but new slot instances under a component (e.g., from 8 component instances to 3 component instances plus 5 slot instances)
  - ✓ All the component attributes are optional and not necessarily applicable to any component class or instance
  - ✓ Not clear how to support the logical (without any ad-hoc equipment) partitioning of the slot into sub-slots: corner case but defined and supported by ONF TR-547 that could be supported by the current model

Child and parent component navigation (#3)

```
+--ro child-component-ref
+--ro contains-child*
|       -> ../../component/component-id
+--ro parent-rel-pos?          int32
+--ro parent?
|       -> ../../component/component-id
```

Concern (if aligning with RFC8348): duplicated information which is needed only when a card occupy more than one slot

- Option 1: update the parent to be a list
  - ✓ What about adding other-parent*?
- Option 2: use the alternative solution for modelling the slots

Attributes of Network Elements (#112)

Rename as product-rev

```
+--rw ne-id                    string
+--ro ne-type?                 identityref
+--ro uuid?                    yang:uuid
+--rw name?                    string
+--rw description?             string
+--rw alias?                   string
+--ro hardware-rev?            string
+--ro software-rev?            string
+--ro software-patch-rev*      string
+--ro mfg-name?                string
+--ro mfg-date?                yang:date-and-time
+--ro serial-number?           string
+--ro product-name?            string
```

Open issue:

- Keep it to represent the OS in case of NEs with a single OS
- Remove it and always use the SW inventory model to report the OS modules of a NE

Note: during SW upgrade, there could be multiple versions of the OS even if only one is running (active)

Hardware and Software Revisions

```
+--rw component-id              string
+--ro class                     union
+--ro uuid?                     yang:uuid
+--rw name?                     string
+--rw description?              string
+--rw alias?                    string
+--ro hardware-rev?             string
+--ro software-rev?             string
+--ro software-patch-rev*       string
+--ro firmware-rev?             string
.....................................
```

An HW component can report HW, SW, FW and path revisions: same issue as before for SW upgrade:

1. One HW component with both HW revision and running SW revision
2. One HW component with HW revision and a SW component with SW revision

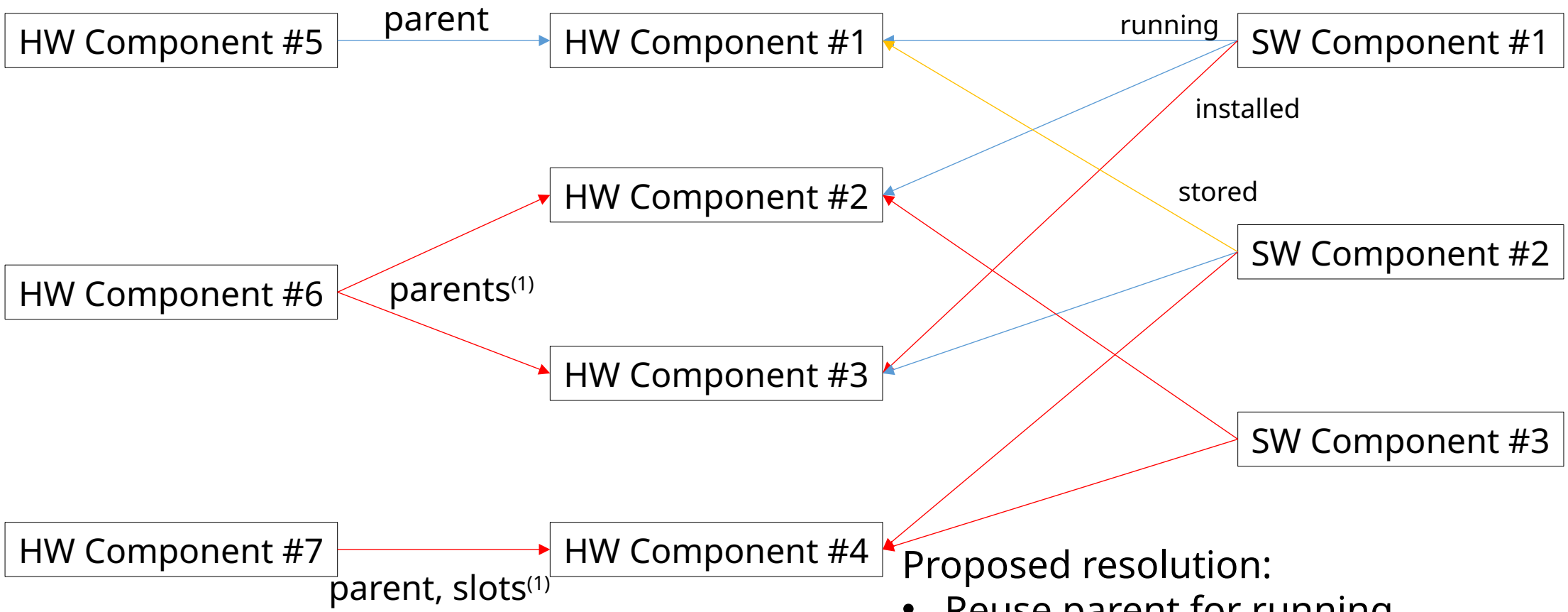
Notes:

- What about if an HW component comes with only one running SW module with no separated part number?
- What about reporting multiple running SW modules? Do we need a list of running SW modules within the same HW component?

In case of option 2: have only one revision (or version/revision)?

- HW revision if the component is an HW component
- SW revision if the component is a SW module component
- OS revision if the component is an OS component
- Patch revision if the component is a SW patch

Hardware and Software Components

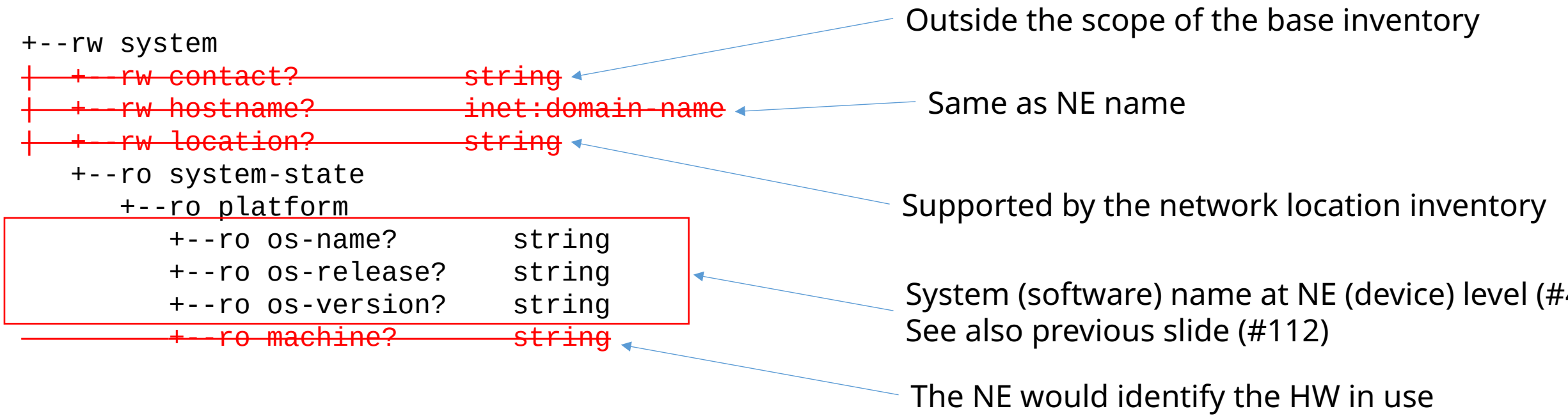


Proposed resolution:

- Reuse parent for running
- Add installed and stored in SW inventory model

(1) Pending resolution of issue #3

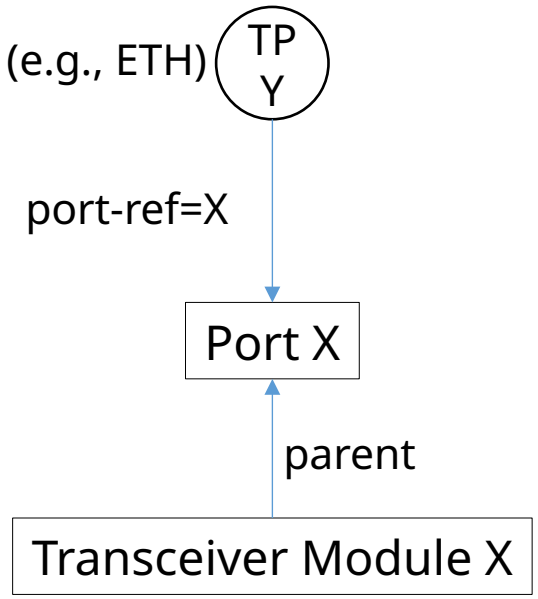
Attributes from RFC7317 (#2)



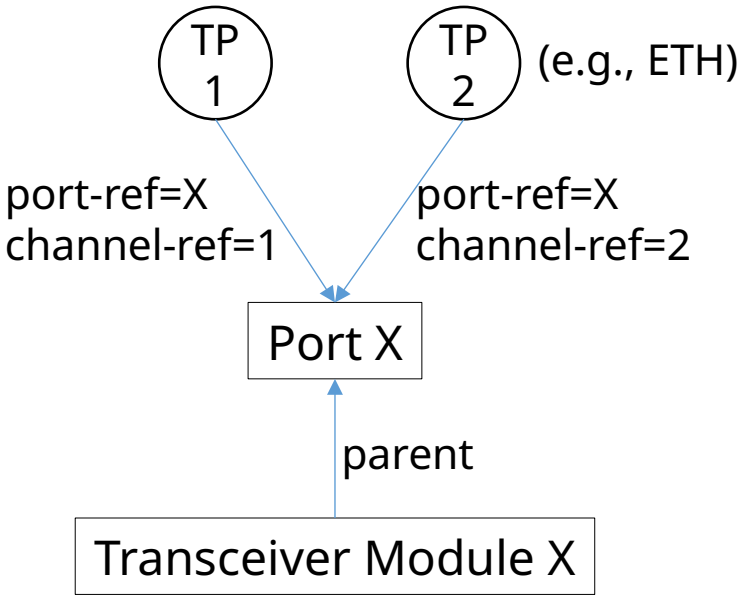
How to represent SW module name, release, patch and build?
(if in the scope of base inventory model)

Breakout Port Modelling (#126) – 1/5

- Issue: The breakout port model appears to be logical inventory and outside the scope of the base inventory model
  - ✓ Current Approach:



No breakout configuration

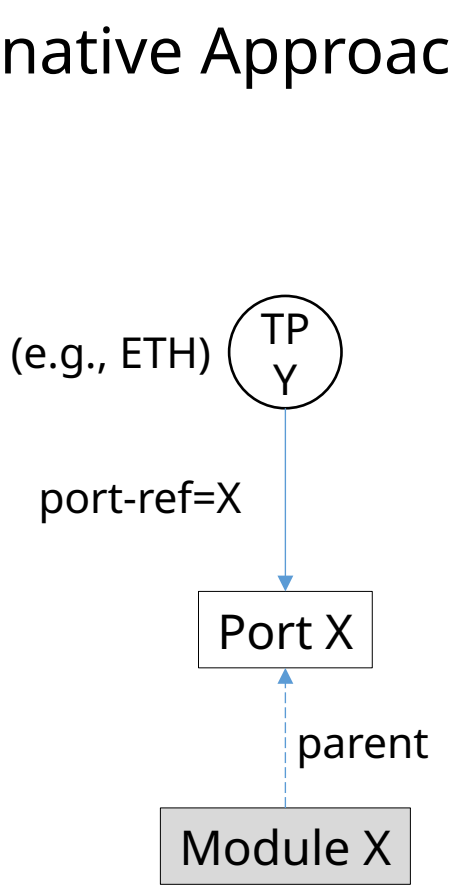


Breakout configuration

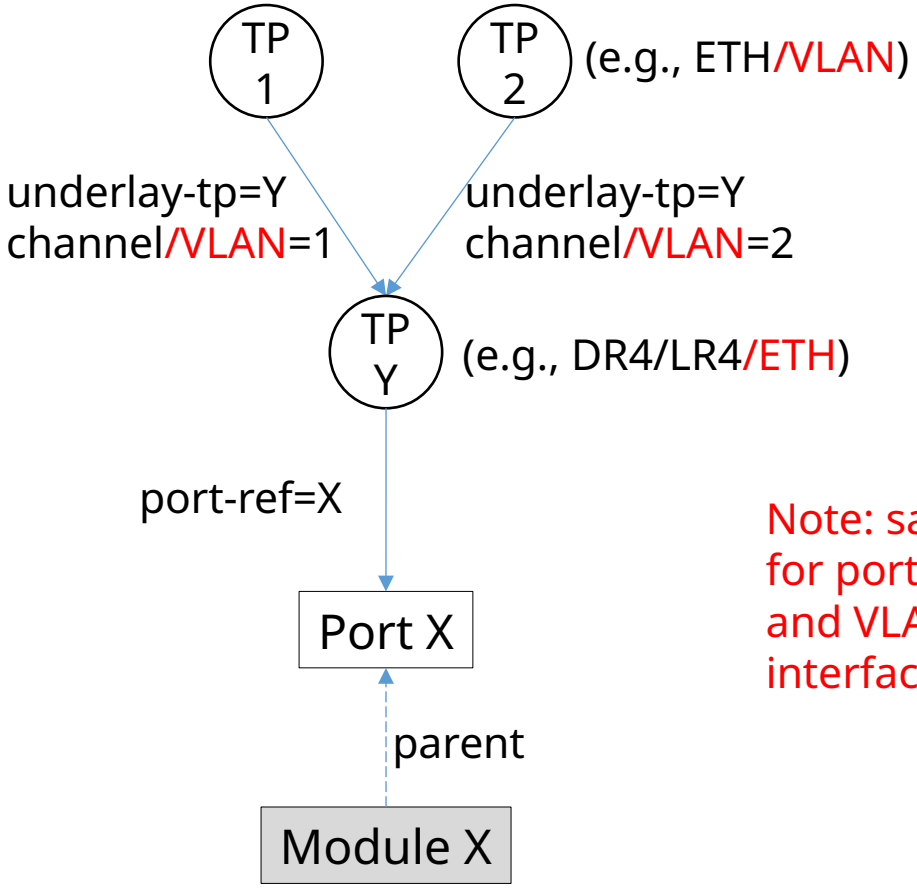
lote: List of available breakout channels in the transceiver module

Breakout Port Modelling (#126) – 2/5

✓ Alternative Approach:



No breakout configuration



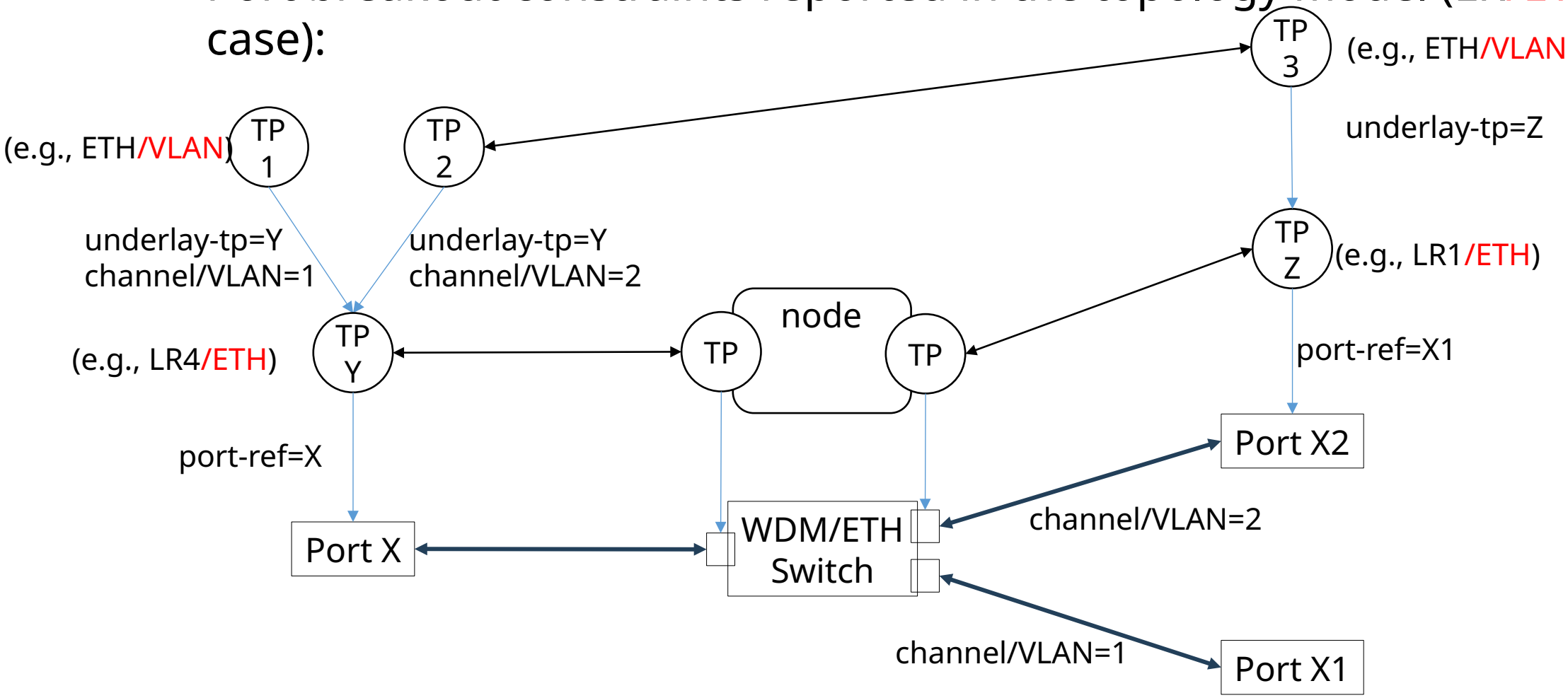
Breakout configuration

Note: same pattern for port breakout and VLAN sub-interface modelling

Note: List of available breakout channels in the DR4/LR4 TP/Link. No need to differentiate transceiver modules from other modules nor to report them for non-pluggable ports. **No need to update iana-hardware**

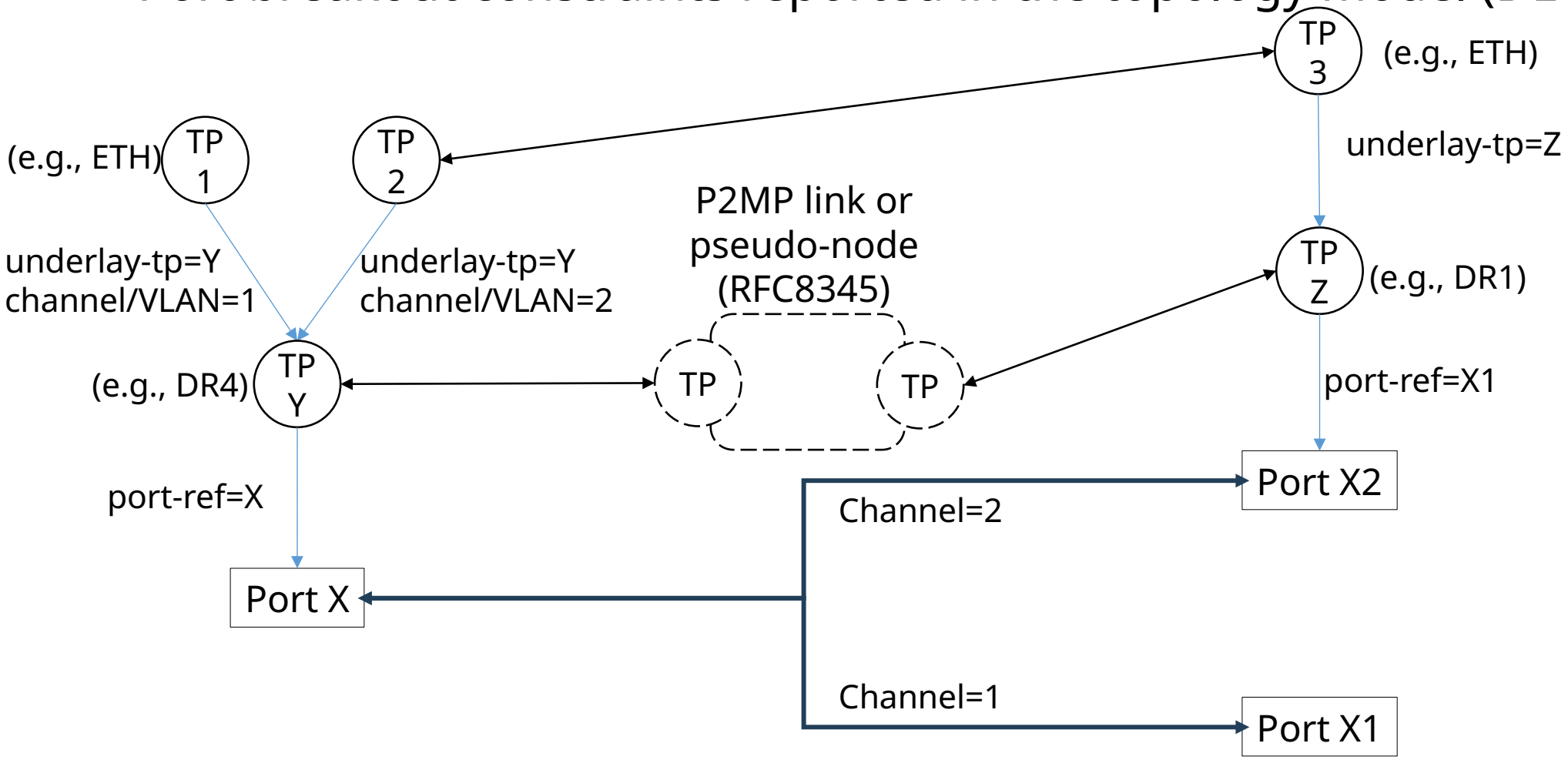
Breakout Port Modelling (#126) – 3/5

- ✓ Port breakout constraints reported in the topology model (LR/ETH case):



Breakout Port Modelling (#126) – 4/5

- ✓ Port breakout constraints reported in the topology model (DE case):



Breakout Port Modelling (#126) – 5/5

- Advantages of the proposed alternative model
  - ✓ No need to update iana-hardware to define the transceiver-module component
  - ✓ Can be applied also to RFC8348 implementations
  - ✓ Re-use same modelling for physical and logical breakouts
  - ✓ No need to mandate reporting the module/transceiver-module under the port in case of non-pluggable ports (in line with most of the existing implementations of non-pluggable ports)
- Implications
  - ✓ Move the YANG model (and JSON examples) for breakouts to the network inventory topology model/I-D
  - ✓ Remove the definition of the transceiver-module
  - ✓ Clean-up some text and definitions no longer needed/valid

Definitions of port (#125)

- Need to improve the definition of port and add the definition of a connector?
  - ✓ Port
    A component where networking traffic can be received and/or transmitted, e.g., by attaching networking cables. : In case of pluggable ports, the port may be empty when no transceiver module is plugged in.
  - ✓ Physical interface
    An interface associated to a ~~physical~~ port. A physical interface is always in the lowest layer of the interface stack.
  - ✓ Logical interface
    An interface which is not associated to a ~~physical~~ port.
  - ✓ Breakout Port
    A port is usually associated with a single physical interface. A breakout port is a port which is broken down and associated into multiple physical interfaces. Those physical interfaces can have the same or different speeds and the same or different number of breakout channels.
  - ✓ Trunk Port
    A trunk port is a port which is associated with one and only physical interface.
  - ✓ Port Breakout
    The configuration of a breakout port.

Note: the definitions of physical and logical interfaces have been slightly updated in draft-ietf-ivy-network-

Definitions of chassis (#121) and slot/sub-slot (#79)

- See draft-ietf-ivy-network-inventory-yang-08
  - ✓ Chassis

    A field replaceable equipment with a particular structural format and dimensions. A chassis can include spaces to take cards (called slots). Elsewhere, a chassis can be called shelf, sub-rack, etc.
  - ✓ Card

    A pluggable equipment with a particular structural format and dimensions which can be inserted into one or more slots (or sub-slots). A card can have spaces to take other cards (called sub-slots). Elsewhere, a card can be called board, module, circuit pack, etc.
  - ✓ Slot

    A space in a chassis that can be equipped with one card, which may be chosen from a limited range of types of cards. A slot can be subdivided into smaller spaces (called sub-slots)

Summary of the Open Issues (Priority 2)

- Issues which can be closed:
  - ✓ Missing component location attribute (#42)
  - ✓ Hardware and Firmware revisions for NE (#40)
    - Can be closed with Attributes of Network Elements (#112)
- Issue that still need some discussion
  - ✓ Add "asset-id" attribute to the NE (#36)
  - ✓ Missing attributes from RFC8348 (#11)
  - ✓ Management of functional dependencies between components (#70)

Missing component location attribute (#42)

- The proposal is not to add this attribute to the base inventory model
  - ✓ A similar attribute was deprecated in OpenConfig
  - ✓ The location string, using the pattern defined in ONF TR-547, can be calculated by the client using the information already provided in the base inventory model
  - ✓ An alternative option is to have this information being calculated by the server and reported to the client. The drawbacks of this approach are:
    - o duplicated information in the YANG data model
    - o need to standardize one (or a limited set of) string patterns to ensure different servers provides the same pattern
  - ✓ If the string is computed by the client, it can use the same pattern regardless of the different server implementations

Next Step

- Resolve the priority 1 open issues
  - ✓ Address some of the priority 2 open issue, time permitting
- Editorial clean-up
- Further review and comments from the WG is appreciated

- The target is to be ready for WG LC by end of September 2025
  - ✓ Need to agree on the resolution of priority 1 issues by end of August 2025