

# Interface extensions YANG & VLAN sub-interface YANG Status update

draft-ietf-netmod-intf-ext-yang-05,  
draft-ietf-netmod-sub-intf-vlan-model-02, &  
draft-wilton-netmod-interface-properties-00

Rob Wilton (Cisco)

[rwilton@cisco.com](mailto:rwilton@cisco.com)

IETF 99, NETMOD WG

# draft-ietf-netmod-intf-ext-yang status:

Since last IETF:

- Updated based on feedback of a few issues discussed at the last IETF.
- YANG doctor review from Andy – thanks:
- Nearly all issues raised have been fixed, but just two remaining:
- The first issue is that I need to provide examples in the draft.

# Issue 2: ethSubInterface property

- ethSubInterface is meant to be a generic way of doing interface properties (full example later).
- Alas, it doesn't really work or help very much (not extensible).
- I think that I have a better solution in draft-wilton-netmod-interface-properties-00.
- But waiting for that draft to complete will probably delay this draft for too long (L2VPN models are dependent on these)

# Issue 2: ethSubInterface - Resolution

My Ideal outcome:

- WG adoption for the approach described in draft-wilton-netmod-interface-properties.
- draft-ietf-netmod-intf-ext-yang-05 proceeds now, model is updated (in a backwards compatible way) as draft-wilton-netmod-interface-properties-00 completes.

*Will ask for opinions after presenting draft-wilton-netmod-interface-properties.*

# draft-ietf-netmod-sub-intf-vlan-model status:

- Updated after feedback from last IETF.
- Model structure simplified ...
- Further simplification once groupings from IEEE are updated.
- Same issue regarding *ethSubInterface* also applies here.
- Also need to fix issue raised by Vladamir.

# Previous VLAN draft tree output:

```
module: ietf-if-l3-vlan
  augment /if:interfaces/if:interface/if-cmn:encapsulation/
                                          if-cmn:encaps-
type:
  +--:(vlan)
    +--rw vlan
      +--rw tag* [index]
        +--rw index          uint8
        +--rw dot1q-tag
          +--rw tag-type     dot1q-tag-type
          +--rw vlan-id      ieee:vlanid
```

# Current VLAN draft tree output:

```
+--:(dot1q-vlan)
```

```
  +--rw dot1q-vlan
```

```
    +--rw outer-tag!
```

```
      | +--rw dot1q-tag
```

```
        | +--rw tag-type      dot1q-tag-type
```

```
        | +--rw vlan-id      ieee:vlanid
```

```
    +--rw second-tag!
```

```
      +--rw dot1q-tag
```

```
        +--rw tag-type      dot1q-tag-type
```

```
        +--rw vlan-id      ieee:vlanid
```

# Once groupings are fixed:

```
+---:(dot1q-vlan)
```

```
  +---rw dot1q-vlan
```

```
    +---rw outer-tag!
```

```
      | +---rw tag-type      dot1q-tag-type
```

```
      | +---rw vlan-id      ieee:vlanid
```

```
    +---rw second-tag!
```

```
      | +---rw tag-type      dot1q-tag-type
```

```
      | +---rw vlan-id      ieee:vlanid
```



# draft-wilton-netmod-interface-properties-00

- New -00 draft.
- Aims to solves interface properties issue.
- Defines new interface property identities.
- IANA if-types also derives from one or more of these property identities.
- Interface configuration is conditional on these identities.

# Interface properties

- Perhaps owned by IANA.
- Example properties:
  - Physical
  - Virtual
  - Sub-interface
  - Point-to-point
  - Multicast
  - Ethernet-like
- New properties can be defined in future.
- **Issue: How do we get the right set or properties, who controls new ones?**

# IANA if-types updated.

- Backwards compatible update.
- 2 example identities:

```
identity ethernetCsmacd {  
    base iana-interface-type;  
    base ianaifp:physical;  
    base ianaifp:multicast;  
    base ianaifp:ethernet-like;  
    description "Ethernet ...";  
}
```

```
identity ieee8023adLag {  
    base iana-interface-type;  
    base ianaifp:virtual;  
    base ianaifp:multicast;  
    base ianaifp:ethernet-like;  
    description "IEEE 802.3ad Link Aggregate.";  
}
```

- Issue: How do we get the mapping right? Who policies updates?

# Before (without interface properties)

```
augment "/if:interfaces/if:interface" {  
    when "derived-from-or-self(if:type,  
'ianaift:ethernetCsmacd') or  
        derived-from-or-self(if:type, 'ianaift:ieee8023adLag')  
or  
        derived-from-or-self(if:type, 'ianaift:l2vlan') or  
        derived-from-or-self(if:type, 'ianaift:ifPwType')" {  
        description "Applies to all Ethernet-like interfaces";  
    }  
}
```

# With proposed solution:

```
augment "/if:interfaces/if:interface" {  
    when "derived-from(if:type, 'ianaifp:ethernet-like')" {  
        description "Applies to all Ethernet-like interfaces";  
    }  
}
```

- This is the same that I was trying to achieve before, but think that I now have a method that AFAIK fully works.
- **But it probably requires management by IANA, is this realistic?**

# Interface properties summary

- Introduce new interface property identities – owned by IANA?
- iana-if-types derives from these properties – owned by IANA?
- Defining new properties is backwards compatible.
- Add properties to new or existing interfaces is backwards compatible.
- Think we can migrate from OLD to NEW in backwards compatible way.
- **Is WG interested in adopting?**
- **Do we delay extensions and VLAN drafts, or (I prefer) finish more quickly then revise later.**