Network Device YANG Organizational Model
draft-rtgyangdt-rtgwg-device-model-03

Authors: Acee Lindem, Christian Hopps, Dean Bogdanovic,
Lou Berger (Ed.)

Contributors: Anees Shaikh, Kevin D'Souza, Luyuan Fang, Qin Wu,
Rob Shakir, Stephane Litkowski, Yan Gang

Repo: https://github.com/ietf-rtg-area-yang-arch-dt/meta-model.git

Topics

- Brief Review of Models, LNEs, and NIs
- Challenges
- Use of Schema Mount
- Draft Changes since 01
- Model Disposition
- Open issues
- Next steps

Defined Models

1. module: network-device
   - Overall structure for any network device type
     - From small router to Carrier Class
     - Covers relations amongst models –
       Not to be implemented directly
2. module: logical-network-element
   - Separates management/resource domains
     - Commonly called logical system or router, and
       virtual switch, chassis, or fabric,
       virtual device contexts
3. module: network-instance
   - Separates routing or switching domain
     - e.g., VRF or VSI

Details
Covered in
RTGWG

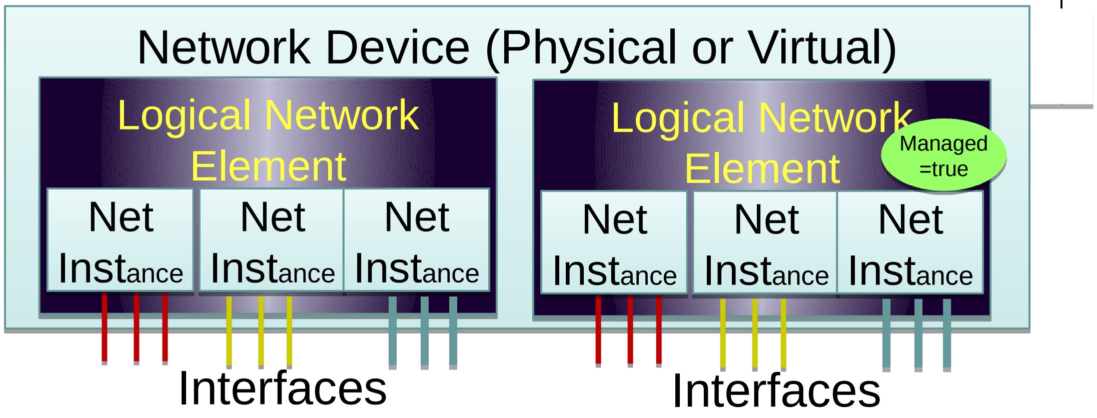
- Will eventually be broken into three documents
  - 2 and 3 will separate standards track RTGWG drafts

Current (draft -03) Approach

The term *schema mount* is used to be solution neutral

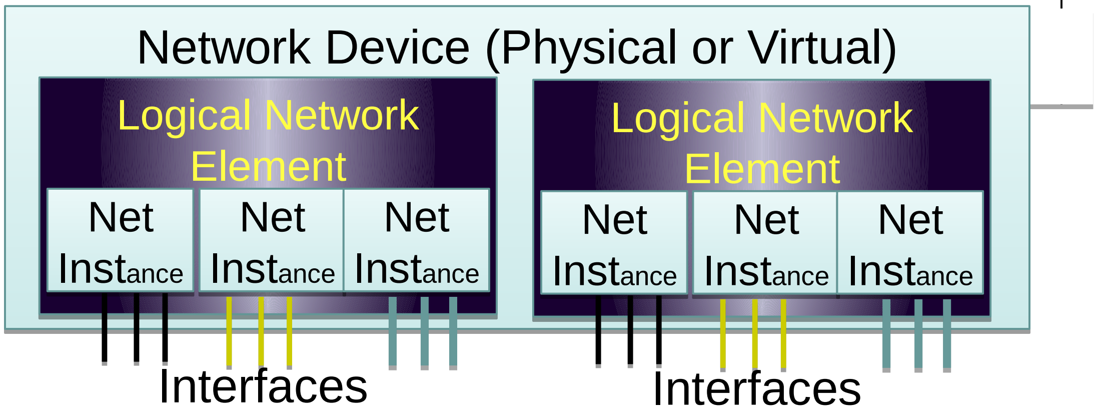
- Rely on “schema” mount
  - Works for any module – ***without modification***
- Adds two tables
  - LNE: logical-network-inventory
  - NI: network-instance
- Each table defines a per {LNE, NI} instance root
  - Under which any top-level model may be *instantiated*
    - Note this is defined in the schema
  - Choice of available model is up to the implementation
    - Some type of device profile definition is expected
  - ietf-yang-library is used to enumerate available models

Logical Network Elements



- Separate management sub-domains
  - Sub-domains can be managed independently and by a top level manager (managed=true)
- Differs from multiple logical devices and VMs
  - Where top level management of subdomains not supported

Network Instances



- Separate routing / switching domains
- Can represent of an RFC 4364 VRF or a Layer 2 Virtual Switch Instance (VSI) or a bridge/router (i.e., both)
- General virtualized instance implying a separate L2, L3, or L2/L3 context.
  - For L3, this implies a unique IPv4/IPv6 address space.

Schema Mount Usage

- Allows device hierarchy to vary for different classes of devices.
  - All modules present in the top level may also be mounted within an LNE.
  - Modules supported within an LNE is implementation dependent.
  - Network Instances can be mounted at top or within LNE.
  - All modules can also be mounted with in LNE though for many it doesn’t make sense.
  - Modules supported by a device learned through ietf-yang-library.

Model 1: A Top-Level Device

```
Namespace "urn:ietf:params:xml:ns:yang:...";
   +--rw ietf-yang-library
   |
   +--rw interfaces
   +--rw hardware
   +--rw qos
   |
   +--rw system-management
   +--rw network-services
   +--rw oam-protocols
   |
   +--rw routing
   +--rw mpls
   +--rw ieee-dot1Q
   |
   +--rw ietf-acl
   +--rw ietf-key-chain
   |
   +--rw logical-network-element
   +--rw network-instance
```

```
module: network-device
   +--rw system-management
      +--rw system-management-global
      |  +--rw statistics-collection
      |  ...
      +--rw system-management-protocol* [type]
      |  +--rw type=syslog
      |  +--rw type=dns
      |  +--rw type=ntp
      |  +--rw type=ssh
      |  +--rw type=tacacs
      |  +--rw type=snmp
      |  +--rw type=netconf
```

```
module: network-device
   +--rw network-services
      +--rw network-service* [type]
         +--rw type=ntp-server
         +--rw type=dns-server
         +--rw type=dhcp-server
```

```
module: network-device
   +--rw oam-protocols
      +--rw oam-protocol* [type]
         +--rw type=bfd
         +--rw type=cfm
         +--rw type=twamp
```

```
module: network-device
   +--rw routing
      +--rw control-plane-protocols
      |  +--rw control-plane-protocol* [type]
      |     +--rw type      identityref
      |     +--rw policy
      +--rw ribs
         +--rw rib* [name]
            +--rw name            string
            +--rw description?    string
            +--rw policy
```

```
module: network-device
  +--rw mpls
     +--rw global
     +--rw lsps* [type]
       +--rw type=static
       +--rw type=constrained-paths
       +--rw type=igp-congruent
```

Model 1: Open Issue

- Question is what to do with the device model?
  - Keep it informational and it will not necessary dictate model hierarchy or inter-module relationships?
    - Risk is that the work will not have impact
  - Make it standards track and move to NETMOD WG?
    - Would dictate where other models fit in the hierarchy
    - Hard to get consensus on overall device layout – “Haters gonna hate!”

ietf-routing Alignment

- ietf-routing no longer includes routing-instance list
- ietf-routing is now a module that would be mounted at the top, LNE, or NI level.
- ietf-routing includes its own list of routing protocols since this is needed for static routing definition.
  - Should this list be elsewhere?
- ietf-routing includes a list of interface – this would not be needed with LNE and NI bindings.

Open Issues/Plans

- Relying on Standardized Schema Mount Solution from NETMOD
  - Instantiation of LNEs and NIs triggered simply by list addition?
- Alignment with OpsState Requirements,
- Clarification of relationship with different policy containers
- Hardware/QoS structuring
- System management, network services, and OAM protocol base models