Routing Area Yang Architecture Design Team Draft Status

Members: Acee Lindem, Anees Shaikh, Christian Hopps,
Dean Bogdanovic, Ebban Aries, Lou Berger,
Qin Wu, Rob Shakir, Xufeng Liu, Yingzhen Qu

Wiki: http://trac.tools.ietf.org/area/rtg/trac/wiki/RtgYangArchDT

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

Design Team Status

- 3 standards track drafts
  - YANG Logical Network Elements – draft-ietf-rtgwg-lne-model-03
  - YANG Network Instances – draft-ietf-rtgwg-ni-model-03
  - Rtg Area Common YANG Data Types – draft-ietf-rtgwg-routing-types-08
- Other drafts
  - Module Tags – draft-rtgyangdt-netmod-module-tags
  - Device Logical Organization – draft-ietf-rtgwg-device-model

Individual Draft in NetMod

Gated by tags

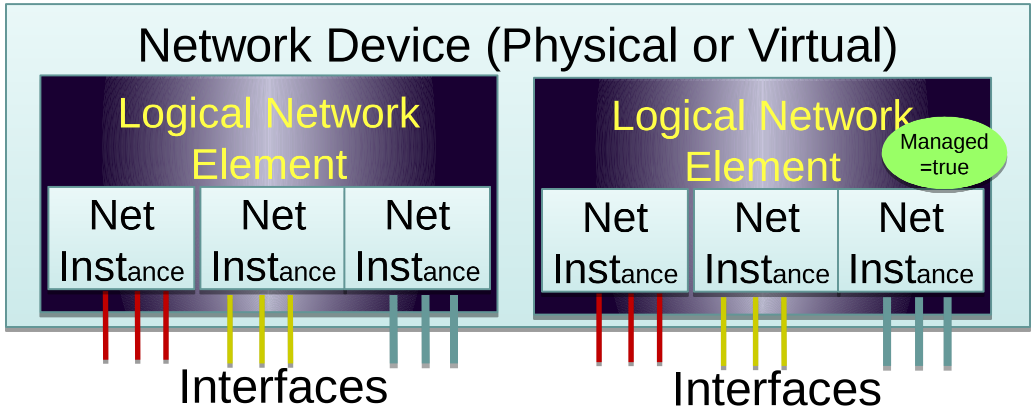
- NMDA Next Steps
  - RFC 8022 Re-spin with solely NDMA restructuring will be done

draft-ietf-rtgwg-lne-model and draft-ietf-rtgwg-ni-model

Status Summary

- Drafts previously blocked by open issues in Schema Mount
  - Previous blocking issues have been settled
  - Drafts use new tree representation for mounts as described in the tree mount presentation
  - Hope to see WG LC soon
  - Schema Mount Limitation - Uses XPath syntax, no filtering based on data, e.g., NI/VRF Name
  - Schema Mount limitation – Design time mounts are currently out of scope though there are cases where it could be useful
- Both drafts have been updated and are ready for Last Call in RTG WG

LNEs and NIs: Reminder



Logical Network Element

- Separate management sub-domains
  - Sub-domains can be managed independently and by a top level manager (managed=true)
  - Commonly called logical system or router; or virtual switch, chassis, fabric, or device context
- Can be supported via multiple logical devices and VMs
  - Where only limited top level management of subdomains is supported

Network Instance

- 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.

draft-ietf-rtgwg-lne-model-03 Update

- Text aligned with the current (and hopefully final) definition of schema mount
- Cleaned up intro an other editorial issues identified in rtg dir review*
- Cleaned up yang layout per YANG DR review
- Added/filled in missing sections
  - e.g., Terminology and Security Consideration
- Added errors and notifications
- Use new tree representation
- Tried to improve narrative based on comments and questions
- Consolidated and expanded examples in new Appendix B.

LNE: Module Tree

```
module: ietf-logical-network-element
    +--rw logical-network-elements
       +--rw logical-network-element* [name]
          +--rw name                 string
          +--rw managed?             boolean
          +--rw description?         string
          +--mp root
  augment /if:interfaces/if:interface:
    +--rw bind-lne-name?
            -> /logical-network-elements/logical-network-element/name
```

New tree representation

Added to cover cases of asynchronous interface ㄹㅌ NI bind failures

```
  notifications:
    +---n bind-lne-name-failed
       +--ro name                 -> /if:interfaces/interface/name
       +--ro bind-lne-name        -> /if:interfaces/interface/lne:bind-lne-name
       +--ro error-info?          string
```

LNE: Module Example

```
module: ietf-logical-network-element
   +--rw logical-network-elements
      +--rw logical-network-element* [name]
         +--rw managed?       boolean
         +--rw name           string
         +--mp root
            ...
```

Managed=true

```
+--ro yanglib:modules-state/
|  ...
+--rw sys:system/
|  ...
+--ro sys:system-state/
|  ...
+--ro rt:routing-state/
|  +--ro router-id? quad
|  +--ro control-plane-protocols
|     +--ro control-plane-protocol* []
|        +--ro ospf:ospf/
|           +--ro instance* [af]
|           ...
+--rw rt:routing/
|  ...
+--rw if:interfaces/
|  ...
+--ro if:interfaces-state/
   ...
```

LNE Next Steps

- More feedback
- WG LC?

draft-ietf-rtgwg-ni-model-03 Update

- Text aligned with the current (and hopefully final) definition of schema mount, impacts types & roots
- Resolved open policy question by providing a structure for LxVPN-specific augmentations
- Cleaned up intro an other editorial issues identified in rtg dir review
- Cleaned up yang layout per YANG DR review
- Added/filled in missing sections
  - e.g., Terminology and Security Consideration
- Added errors and notifications
- Use new tree representation
- Tried to improve narrative based on comments and questions
- Consolidated and expanded examples in new Appendix B.

LxVPN Support

Old:

```
module: ietf-network-instance
  +--rw network-instances
     +--rw network-instance* [name]
        +--rw name               string
        +--rw enabled?           boolean
        +--rw description?       string
        +--rw network-instance-policy
        |  ...
        +--mp root
           ...
```

- NI Policy
  - Container for core instance configuration information
  - Place holder, with details pending
- Root
  - Single mount point for use by any NI type

New:

```
module: ietf-network-instance
  +--rw network-instances
     +--rw network-instance* [name]
        +--rw name               string
        +--rw enabled?           boolean
        +--rw description?       string
        +--rw (ni-type)?
        +--rw (root-type)?
           ...
```

- NI Type
  - Empty choice statement
  - Place holder for LxVPN technology specific augmentation
- Root Type
  - Well known mount point for VRF/VSIs

LxVPN Technology Specific Information

- This is for VRF or VSI related information *in the core instance*
- Differs based on LxVPN technology
  - L2VPN – VPLS, VxLAN, EVPN, …
  - L3VPN – MPLS, IP tunnels, …
- Supported via ***ni-types*** choice statement
  - With technology specific case augmentations
  - Also provides explicit indication of ni-type
- Alternatively, augmentations can be made in other top-level module(s)

```
module: ietf-network-instance
  +--rw network-instances
     +--rw network-instance* [name]
        +--rw name           string
        +--rw enabled?       boolean
        +--rw description?   string
        +--rw (ni-type)?
        |  +--:(l3vpn)  //augmentation
        |     +--rw l3vpn:l3vpn
        |     |  ...    // config data
        |     +--ro l3vpn:l3vpn-state
        |     |  ...    // state data
```

Well Known Mount Points

- Supported module list *within* an NI (VRF/VSI) is likely to be determined based on L2 or L3 type and common across different types of L2 or L3 VPN technologies
- Three types of Nis have been identified
  1. VRFs for L3VPNs
  2. VSIs for L2VPNs
  3. VSI+VRF for L2+L3VPNs (bridge/routers)
- Schema mount defines the schema (i.e., module list) on a per mount point *name* basis
  - So need named mount point per type

```
module: ietf-network-instance
  +--rw network-instances
     +--rw network-instance* [name]
        +--rw name              string
        +--rw enabled?          boolean
        +--rw description?      string
        +--rw (ni-type)?
        +--rw (root-type)?
           +--:(vrf-root)
           |  +--mp vrf-root?
           +--:(vsi-root)
           |  +--mp vsi-root?
           +--:(vv-root)
              +--mp vv-root?
                //one root required per NI
```

Notifications

- Added to cover cases of asynchronous interface ㄹㅌ NI bind failures
- Interface may be bound multiple ways {base, IPv4 & IPv6}
  - Failure can occur on one or more

```
notifications:
  +---n bind-ni-name-failed
     +--ro name              -> /if:interfaces/interface/name
     +--ro interface
     |  +--ro bind-ni-name?   -> /if:interfaces/interface/ni:bind-ni-name
     +--ro ipv4
     |  +--ro bind-ni-name?   -> /if:interfaces/interface/ip:ipv4/ni:bind-ni-name
     +--ro ipv6
     |  +--ro bind-ni-name?   -> /if:interfaces/interface/ip:ipv6/ni:bind-ni-name
     +--ro error-info?   string
```

NI: Module Example

```
module: ietf-network-instance
  +--rw network-instances
     +--rw network-instance* [name]
        +--rw name            string
        +--rw enabled?        boolean
        +--rw description?    string
        +--rw (ni-type)?
        |  +--:(l3vpn)
        |     +--rw l3vpn:l3vpn
        |     |  ... // config data
        |     +--ro l3vpn:l3vpn-state
        |     |  ... // state data
        +--rw (root-type)?
           +--:(vrf-root)
              +--mp vrf-root
                 ...
```

```
+--ro rt:routing-state/
|  +--ro router-id?                 yang:dotted-
quad
|  +--ro control-plane-protocols
|     +--ro control-plane-protocol* [type name]
|        +--ro ospf:ospf/
|           …
+--rw rt:routing/
|  +--rw router-id?                 yang:dotted-
quad
|  +--rw control-plane-protocols
|     +--rw control-plane-protocol* [type name]
|     +--rw ospf:ospf/
|        +--rw instance* [af]
|           +--rw areas
|              +--rw area* [area-id]
|                 +--rw interfaces
|                    +--rw interface* [name]
|                       +--rw name if:interface-ref
|                       +--rw cost?   uint16
+--ro if:interfaces@
|  ...
+--ro if:interfaces-state@
|  ...
```

NI: Full Module Tree

```
module: ietf-network-instance
  +--rw network-instances
     +--rw network-instance* [name]
        +--rw name           string
        +--rw enabled?       boolean
        +--rw description?   string
        +--rw (ni-type)?
        +--rw (root-type)?
           +--:(vrf-root)
           |  +--mp vrf-root?
           +--:(vsi-root)
           |  +--mp vsi-root?
           +--:(vv-root)
              +--mp vv-root?
augment /if:interfaces/if:interface:
  +--rw bind-ni-name?    -> /network-instances/network-instance/name
augment /if:interfaces/if:interface/ip:ipv4:
  +--rw bind-ni-name?    -> /network-instances/network-instance/name
augment /if:interfaces/if:interface/ip:ipv6:
  +--rw bind-ni-name?    -> /network-instances/network-instance/name
notifications:
  +---n bind-ni-name-failed
     +--ro name          -> /if:interfaces/interface/name
     +--ro interface
     |  +--ro bind-ni-name?   -> /if:interfaces/interface/ni:bind-ni-name
     +--ro ipv4
     |  +--ro bind-ni-name?   -> /if:interfaces/interface/ip:ipv4/ni:bind-ni-name
     +--ro ipv6
     |  +--ro bind-ni-name?   -> /if:interfaces/interface/ip:ipv6/ni:bind-ni-name
     +--ro error-info?   string
```

Open Issues

- Schema mount currently doesn't allow parent-reference filtering on the instance of the mount point, but rather just the schema.
- This means it is not possible to filter based on actual data, e.g., bind-network-instance-name="green".
- Recommended resolution:
  - Accept limitation
  - Implementations may choose to impose a limitation on parent references
    - But not required

NI Next Steps

- More feedback
- WG LC?

draft-ietf-rtgwg-routing-types

Recent Changes and History (1/2)

- February - YANG Doctor Review from Lada Lahotka
- Addition of BGP Subsequent Address family types as per comment from Sue Hares.
- Addition of percentage and timeticks64 types as per comment from Rob Wilton.
  - From OpenConfig types – used in BGP Model
- April - Second YANG Doctor review from Radek Krejčí.
  - Use boilerplate for YANG model headers as per Apendix C of RFC 6087BIS
- Split of IANA based types as per comment from Martin Bjorklund similar to RFC 7224

Recent Changes and History (2/2)

- May – Routing Directorate review from Stewart Bryant
- June 13th – Working Group Last Call
- Add ipv6-route-target, route-origin, and ipv6-route-orgin as per comments from Jeff Haas
- Add geo-coordinates type as per comments from Robert Razuk
  - Protocol encodings reviewed and discussed across LISP, OSPF, IS-IS and BGP drafts
  - However, significant risk of change since the protocol drafts are new and it is not clear all the experts have reviewed the encoding
  - Will be moved to a separate ietf-geo-location module in separate draft for further review and independent progression.
  - Should new draft go immediately to WG document?

Routing Types Summary & Next Steps

- Handle a few pending changes and re-issue the draft
  - Remove geo-location
  - Update route-target with flexible type
  - Clarify description of label-stack grouping
- It is time to progress the updated version of the model and limit further comments to the existing types as opposed to suggestions for new types.
  - Exceptions may be made for reviewed YANG types provided as code snippets.
- RTG WG Co-chairs will request publication after IETF and progress the document.