A YANG Data Model for Routing Management

draft-ietf-netmod-routing-cfg-17

Ladislav Lhotka
⟨lhotka@nic.cz⟩

Acee Lindem
⟨acee@cisco.com⟩

24 March 2015

I-D Status

The document is now long overdue.

In -17, the data model was reduced to a bare minimum: features were left in the model only if they are required by all implementations and/or cannot be added by augmentation.

Discussions took place in the `rtg-yang-coord` mailing list.

Several issues have been resolved, some important ones still remain open.

Major Changes Between -16 and -17

- Route filters were completely removed. A data model of a route filtering framework will be addressed separately by routing experts.
- Numeric IDs of list entries in state data were removed – they can be added via augmentation when needed.
- Choice of next-hop cases was considerably reduced.
- Feature "multipath-routes" was removed.
- List `/if:interfaces-state/if:interface` [RFC 7223] was augmented with a leaf-list that shows assignments of interfaces to routing instances. See issue #5.

Next-Hop Choice

Only two next-hop cases remained in both RIBs (state data) and configuration of static routes.

Next-hop in RIB routes:

```
+--ro next-hop
|  +--ro (next-hop-options)
|     +--:(simple-next-hop)
|     |  +--ro outgoing-interface?
|     |  +--ro v6ur:next-hop-address?
|     |  +--ro v4ur:next-hop-address?
|     +--:(special-next-hop)
|        +--ro special-next-hop?
```

More complicated next-hop cases (ECMP, recursive next-hops) can by added by augmenting the `next-hop-options` schema node.

Open Issues

Most open issues follow from the existence of two major CLI configuration languages.

1. Routing-instance-centric versus routing-protocol-centric design;
2. RIB placement in the state data hierarchy;
3. Configurable connections between RIBs;
4. Assignment of interfaces to routing instances;
5. Placement of IPv6 RA parameters (configuration & state data);

Issue #1: Instance- v. Protocol-Centric Design

routing-instance-centric

```
+--rw routing-instance*
|  +--rw name
|  +--rw type?
|  …
|  +--rw routing-protocols
|     +--rw routing-protocol*
|        +--rw type
|        +--rw name
|        …
```

Used by Juniper, Alcatel-Lucent (and *ietf-routing*).

routing-protocol-centric

```
+--rw routing-protocol*
|  +--rw name
|  +--rw type?
|  …
|  +--rw routing-instances
|     +--rw routing-instance*
|        +--rw type
|        +--rw name
|        …
```

Instance means VRF. Used by Cisco, Brocade, HP, Huawei.

Parameters defined outside the inner list can serve as defaults for all entries of the inner list. Consequently, either design can be converted to the other but a round trip may result in a less compact configuration.

Stéphane Litkowski tried to run a poll but didn't get any representative feedback.

Issue #1, Continued

The decision between protocol- and instance-centric design really involves only a specific type of routing instance – VRF. However, some implementations support different types of routing instances – these couldn't be modeled if the top-level *routing-instance* list is removed.

Module *ospf* [draft-yeung-netmod-ospf-02] introduced a hybrid approach – an additional list of instances inside OSPF routing protocol configuration.

- protocol-centric implementations use only one (default) routing-instance,
- instance-centric implementations use only one instance inside `routing-protocol`.

This makes the data model complex and hard to understand.

Issue #2: RIB Placement

global

```
+--rw routing-instance*
|  +--rw name
|  +--rw type?
|  …
+--rw ribs
   +--rw rib*
      +--rw name
      +--rw address-family
      …
```

per routing-instance

```
+--rw routing-instance*
   +--rw name
   +--rw type?
   …
   +--rw ribs
      +--rw rib*
         +--rw name
         +--rw address-family
         …
```

Up to rev. -04, RIBs were per routing-instance.

Issue #2, Discussion

Acee: RIBs should be contained in routing-instances – default RIB per address family and additional RIBs for other purposes. This implicitly limits a RIB to a single routing-instance.

No requirement for connecting RIBs to routing protocol – all RIBs within routing-instance accessible.

Lada: Some implementations (Junos, BIRD) allow for assigning a (non-default) RIB to a routing protocol in configuration.

Issue #3: RIB Connections

configurable

```
+--rw ribs
   +--rw rib*
      +--rw name
      +--rw address-family
      +--rw description?
      +--rw recipient-ribs
         +--rw recipient-rib*
            +--rw rib-name
```

none

```
+--rw ribs
   +--rw rib*
      +--rw name
      +--rw address-family
      +--rw description?
```

Issue #3, Discussion

Acee: No concept of recipient RIBs – routes are only added, modified and deleted by routing protocols.

Lada: Again, some implementations (Junos, BIRD) have this concept.

Issue #4: Interface Assignment

in routing-instance

configuration

```
+--rw routing-instance*
|  …
|  +--rw interfaces
|  |  +--rw interface*
|  |     +--rw name
```

state data

```
+--ro if:interfaces-state
|   +--ro if:interface*
|      +--ro if:name
|      …
|      +--ro rt:routing-instances
|        +--ro rt:routing-instance*
```

in interface

configuration

```
+--rw if:interfaces
|  +--rw if:interface*
|     +--rw if:name
|     …
|     +--rw rt:routing-instance
```

Issue #4, Discussion

Acee: Separate list is unnecessary – `if:interface` should be augmented with a reference to a routing-instance.

Lada: Such an augment would then apply to all interface types (of all layers), because the `if:interface` list is flat. Assignment to routing instances makes sense only for L3 interfaces.

Acee: Configuration of IP addresses should not be disjoint from RIB definition that implies the IP/IPv6 address space.

Lada: The current organisation follows closely the Junos model:

```
set interface fe-0/0/2 unit 0 family inet address 6.6.6.5/24
set routing-instances blue-vr interface fe-0/0/2.0
```

Issue #4, Discussion (continued)

Acee: Associating an interface with multiple routing-instances is not supported by any implementation.

Lada: The idea was that such a constraint can be specified for a particular routing-instance type such as VRF. However, the current schema would also support routing-instance types that do not need this restriction.

Issue #5: IPv6 RA Parameters

IPv6 Router Advertisement parameters [RFC 4861] are currently configured inside routing-instance configuration, i.e. under `rt:interface`.

Acee: IPv6 RA parameters should augment the `ip:ipv6` container defined in the *ietf-ip* module [RFC 7277].

Lada: There was a discussion back in 2011 that considered both options:

http://www.ietf.org/mail-archive/web/netmod/current/msg06074.html

We didn't find substantial technical arguments to prefer one option over another.

These parameters are applicable only to (some) router interfaces.