Core Routing Module

`draft-ietf-netmod-routing-cfg-00`

Ladislav Lhotka
⟨*lhotka@cesnet.cz*⟩

26 July 2011

Objectives

- The data model should be suitable for the common address families, in particular IPv4 and IPv6, unicast and multicast.
- Simple routing setups, such as static routing, should be configurable in a simple way, ideally without the need to write additional YANG modules.
- The framework must allow for complicated setups including multiple routing tables and multiple routing protocols, and controlled redistribution of routing information.
- Vendors should be able to map data models using this framework to their proprietary data models and configuration interfaces.

Main Changes

(compared to `draft-lhotka-netmod-routing-00`):

- corrections and simplifications
- new RPC method for FIB queries - *get-route*
- attempt to allow for multiple address families – the *ietf-routing* module contains only top-level containers, AFI/SAFI-specific contents are to be provided by other modules via augmentation. A module for IPv4 unicast (*ietf-ipv4-unicast-routing*) is a part of the same draft.

The module *ietf-routing* also contains enumeration typedefs with IANA-registered address families (`ipV4`, `ipV6`, `nsap`, `hdlc`, ...) and SAFI (`nlri-unicast`, `nlri-multicast`, `nlri-mpls`, ...).

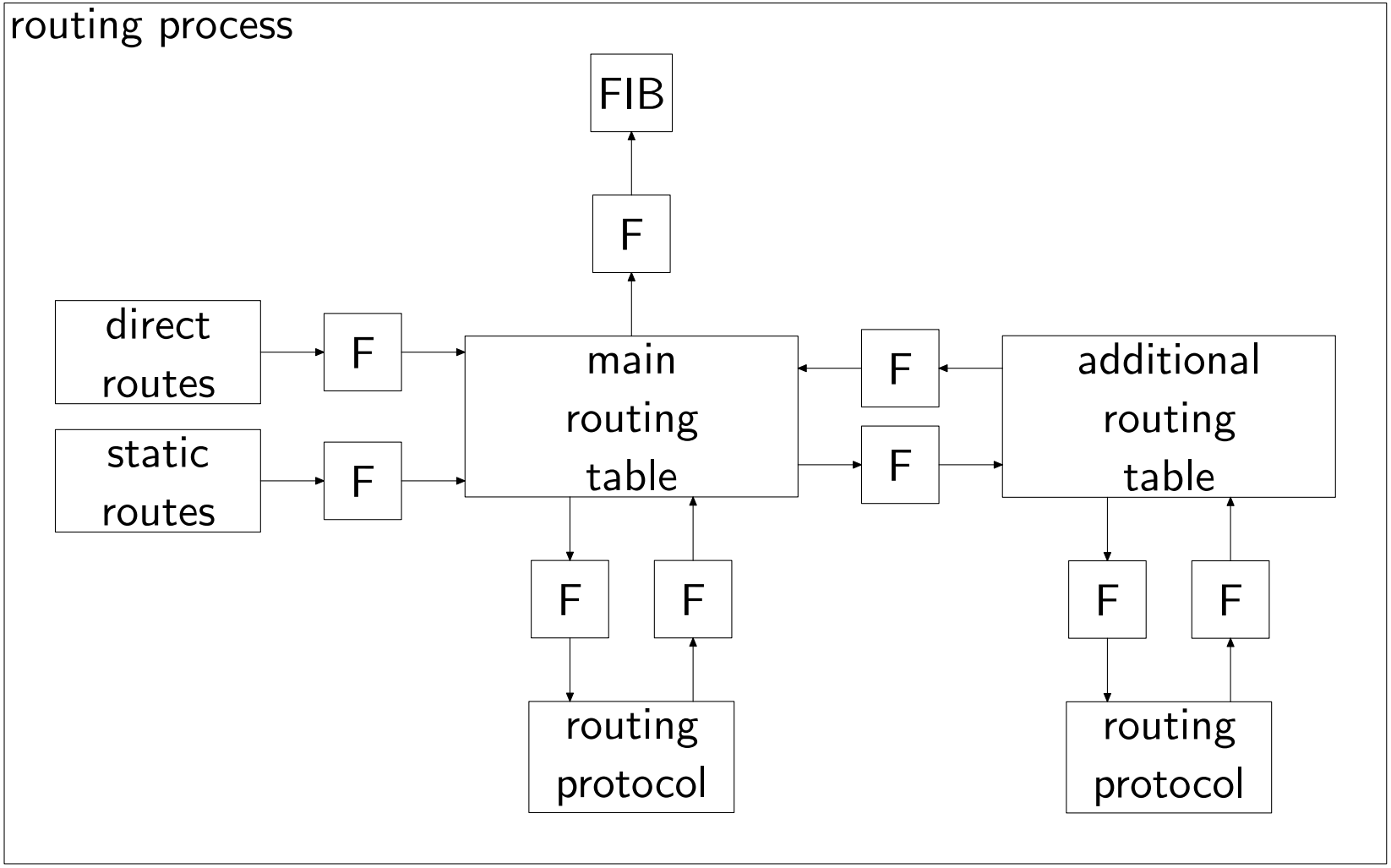
Module *ietf-routing*

```
+--rw routing
   +--rw routing-process [name]
      +--rw name               string
      +--rw address-family?    address-family
      +--rw safi?              subsequent-address-family
      +--rw description?       string
      +--rw enabled?           boolean
```

Module *ietf-ipv4-unicast-routing*

```
+--rw v4ur:ipv4-unicast-routing
   +--rw v4ur:routing-protocol-instances
   |  +--rw v4ur:routing-protocol-instance [name]
   |     +--rw v4ur:static-routes
   |     |  +--rw v4ur:static-route [id]
   |     |     +-- ...
   |     +--rw v4ur:name              string
   |     +--rw v4ur:description?      string
   |     +--rw v4ur:type              identityref
   |     +--rw v4ur:routing-table?    leafref
   |     +--rw v4ur:import-filter?    leafref
   |     +--rw v4ur:export-filter?    leafref
   +--rw v4ur:route-filters
   |  +--rw v4ur:route-filter [name]
   |     +-- ...
   +--rw v4ur:routing-tables
      +--rw v4ur:routing-table [name]
         +--ro v4ur:routes
         |  +--ro v4ur:route
         |     +-- ...
         +--rw v4ur:name                          string
         +--rw v4ur:description?                  string
         +--rw v4ur:recipient-routing-tables [recipient-name]
            +--rw v4ur:recipient-name    leafref
            +--rw v4ur:filter?           leafref
```

Example Setup



Comments Received So Far

Martin Björklund:

- Move the skeletons of generic components (route tables and filters) back to the *ietf-routing* module and augment them from other modules only with AF-dependent contents.

Joel Halpern:

- objection against *delete-route* RPC method;
- RIBs and routing processes are orthogonal concepts, the same BGP instance may be used for different address families;
- central, protocol-independent, RIB, à la RIB2

Tom Petch:

- Different types of routers: default-free zone, CPE, MPLS

Quotes from the Discussion

Andy Bierman: "The domain experts don't really know the data modeling stuff, and the data modeling experts don't really know the domain stuff."

Tom Petch (responding to Joel Halpern): "In passing, I do not share your view of routers and routing terminology, and nor do I share Lada's, but do not know if any of us are right."

Next Steps

Short term (-01 revision):

1. Change the name of *routing-process* container.
2. Remove the restriction of one address family per routing process.
3. Remove RPC method *delete-route*.
4. Move generic framework components (routing protocol instance, routing tables and filters) back to the *ietf-routing* module.

Longer term:

1. Cooperation with IETF routing experts: improvements of the existing module(s), development of "production" modules for routing protocols.
2. Cooperation with vendors' routing experts: data models should remain reasonably compatible with the logic of their data models and configuration interfaces.

Open Issues

1. A separate I-D for the *ietf-ipv4-unicast-module*, or perhaps a common draft for IPv4 and IPv6 unicast?
2. A better name for *routing-process*.
3. Cooperation with external experts.