Workgroup: Network Working Group
Internet-Draft:
draft-ietf-rtgwg-yang-rib-extend-24
Published: 16 October 2023
Intended Status: Standards Track
Expires: 18 April 2024
Authors: A. Lindem
LabN Consulting LLC
Y. Qu
Futurewei

RIB Extension YANG Data Model

Abstract

A Routing Information Base (RIB) is a list of routes and their corresponding administrative data and operational state.

RFC 8349 defines the basic building blocks for the RIB data model, and this model augments it to support multiple next-hops (aka, paths) for each route as well as additional attributes.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 18 April 2024.

Table of Contents

1. Introduction

This document defines a YANG [RFC7950] data model which extends the RIB data model defined in the ietf-routing YANG module [RFC8349] with more route attributes.

A RIB is a collection of routes with attributes controlled and manipulated by control-plane protocols. Each RIB contains only routes of one address family [RFC8349]. Within a protocol, routes are selected based on the metrics in use by that protocol, and the protocol installs the routes to RIB. The RIB selects the preferred or active route by comparing the route-preference (aka, administrative distance) of the candidate routes installed different protocols.

The module defined in this document extends the RIB to support more route attributes, such as multiple next-hops, route metrics, and administrative tags.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) [RFC8342].

2. Terminology and Notation

The following terms are defined in [RFC8342]:

- *configuration
- *system state
- *operational state

The following terms are defined in [RFC7950]:

- *action
- *augment
- *container
- *container with presence
- *data model
- *data node
- *leaf
- *list
- *mandatory node
- *module
- *schema tree
- *RPC (Remote Procedure Call) operation

The following terms are defined in [RFC8349] Section 5.2:

- *RIB

2.1. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

2.2. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise,

names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
v4ur	ietf-ipv4-unicast-routing	[RFC8349]
v6ur	ietf-ipv6-unicast-routing	[RFC8349]
inet	ietf-inet-types	[RFC6991]
ospf	ietf-ospf	[RFC9129]
isis	ietf-isis	[RFC9130]

Table 1: Prefixes and Corresponding YANG Modules

3. Design of the Model

The YANG module defined in this document augments the ietf-routing YANG modules defined in [RFC8349], which provide a basis for routing system data model development. Together with YANG modules defined in [RFC8349], a generic RIB YANG model is defined to implement and monitor a RIB.

The modules in [RFC8349] also define the basic configuration and operational state for both IPv4 and IPv6 static routes. This document provides augmentations for static routes to support multiple next-hops and more next-hop attributes.

3.1. Tags and Preference

Individual route tags are supported at both the route and next-hop level. A preference per next-hop is also supported for selection of the most preferred reachable static route.

The following tree snapshot shows tag and preference which augment static IPv4 unicast routes and IPv6 unicast routes next-hop.

```
augment /rt:routing/rt:control-plane-protocols
        /rt:control-plane-protocol/rt:static-routes/v4ur:ipv4
        /v4ur:route/v4ur:next-hop/v4ur:next-hop-options
        /v4ur:simple-next-hop:
  +--rw preference?   uint32
  +--rw tag?          uint32
augment /rt:routing/rt:control-plane-protocols
        /rt:control-plane-protocol/rt:static-routes/v4ur:ipv4
        /v4ur:route/v4ur:next-hop/v4ur:next-hop-options
        /v4ur:next-hop-list/v4ur:next-hop-list/v4ur:next-hop:
  +--rw preference?   uint32
  +--rw tag?          uint32
augment /rt:routing/rt:control-plane-protocols
        /rt:control-plane-protocol/rt:static-routes/v6ur:ipv6
        /v6ur:route/v6ur:next-hop/v6ur:next-hop-options
        /v6ur:simple-next-hop:
  +--rw preference?   uint32
  +--rw tag?          uint32
augment /rt:routing/rt:control-plane-protocols
        /rt:control-plane-protocol/rt:static-routes/v6ur:ipv6
        /v6ur:route/v6ur:next-hop/v6ur:next-hop-options
        /v6ur:next-hop-list/v6ur:next-hop-list/v6ur:next-hop:
  +--rw preference?   uint32
  +--rw tag?          uint32
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
  +--ro metric?            uint32
  +--ro tag*               uint32
  +--ro application-tag?   uint32
```

3.2. Repair Path

The IP Fast Reroute (IPFRR) calculation by routing protocol pre-computes repair paths [RFC5714], and the repair paths are installed in the RIB.

Each route next-hop in the RIB is augmented with a repair path, and is shown in the following tree snapshot.

```
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route
        /rt:next-hop/rt:next-hop-options/rt:simple-next-hop:
  +--ro repair-path
     +--ro outgoing-interface?   if:interface-state-ref
     +--ro next-hop-address?     inet:ip-address-no-zone
     +--ro metric?               uint32
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route
        /rt:next-hop/rt:next-hop-options/rt:next-hop-list
        /rt:next-hop-list/rt:next-hop:
  +--ro repair-path
     +--ro outgoing-interface?   if:interface-state-ref
     +--ro next-hop-address?     inet:ip-address-no-zone
     +--ro metric?               uint32
```

4. RIB Model Tree

The ietf-routing.yang tree with the augmentations herein is included in Appendix A. The meaning of the symbols can be found in [RFC8340].

5. RIB Extension YANG Model

```
<CODE BEGINS> file "ietf-rib-extension@2023-10-16.yang"

 module ietf-rib-extension {
  yang-version "1.1";
  namespace "urn:ietf:params:xml:ns:yang:ietf-rib-extension";

  prefix rib-ext;

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-interfaces {
    prefix "if";
    reference "RFC 8343: A YANG Data Model for Interface
                         Management (NMDA Version)";
  }

  import ietf-routing {
    prefix "rt";
    reference "RFC 8349: A YANG Data Model for Routing
                         Management (NMDA Version)";
  }

  import ietf-ipv4-unicast-routing {
    prefix "v4ur";
    reference "RFC 8349: A YANG Data Model for Routing
                         Management (NMDA Version)";
  }

  import ietf-ipv6-unicast-routing {
    prefix "v6ur";
    reference "RFC 8349: A YANG Data Model for Routing
                         Management (NMDA Version)";
  }

  import ietf-ospf {
    prefix "ospf";
    reference "RFC 9129: A YANG Data Model for the OSPF Protocol";
  }

  import ietf-isis {
    prefix "isis";
    reference "RFC 9130: A YANG Data Model for the IS-IS Protocol";
  }

  organization
    "IETF RTGWG - Routing Working Group";
```

```
contact
  "WG Web:   <https://datatracker.ietf.org/group/rtgwg/>
   WG List:  <mailto:rtgwg@ietf.org>

   Author:   Acee Lindem
             <mailto:acee.ietf@gmail.com>
   Author:   Yingzhen Qu
             <mailto:yingzhen.qu@futurewei.com>";

description
  "This YANG module extends the RIB defined in the ietf-routing
   YANG module with additional route attributes.

   This YANG module conforms to the Network Management
   Datastore Architecture (NDMA) as described in RFC 8342.

   Copyright (c) 2023 IETF Trust and the persons identified as
   authors of the code.  All rights reserved.

   Redistribution and use in source and binary forms, with or
   without modification, is permitted pursuant to, and subject to
   the license terms contained in, the Revised BSD License set
   forth in Section 4.c of the IETF Trust's Legal Provisions
   Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC XXXX;
   see the RFC itself for full legal notices.";

revision 2023-10-16 {
  description
    "Initial Version";
  reference
    "RFC XXXX: A YANG Data Model for RIB Extensions.";
}

/* Groupings */
grouping rib-statistics {
  description
    "Statistics grouping used for RIB augmentation.";
  container statistics {
    config false;
    description
      "Container for RIB statistics.";
    leaf total-routes {
      type uint32;
      description
        "Total number of routes in the RIB";
    }
    leaf total-active-routes {
```

```
      type uint32;
      description
        "Total number of active routes in the RIB. An active
         route is the route that is preferred over other routes
         to the same destination prefix.";
    }
    leaf total-route-memory {
      type uint64;
      units "bytes";
      description
        "Total memory for all routes in the RIB.";
    }
    list protocol-statistics {
      description
        "RIB statistics for routing protocols installing
         routes in the RIB.";
      leaf protocol {
        type identityref {
          base rt:routing-protocol;
        }
        description
          "Routing protocol installing routes in the RIB.";
      }
      leaf routes {
        type uint32;
        description
          "Total number of routes in the RIB for the routing
           protocol identified by the 'protocol'.";
      }
      leaf active-routes {
        type uint32;
        description
          "Total number of active routes in the RIB for the routing
           protocol for the routing protocol identified by the
           'protocol'. An active route is preferred over other
           routes to the same destination prefix.";
      }
      leaf route-memory {
        type uint64;
        units "bytes";
        description
          "Total memory for all routes in the RIB for the routing
           protocol identified by the 'protocol'.";
      }
    }
  }
}

grouping repair-path {
```

```
  description
    "Grouping for IP Fast Reroute repair path.";
  container repair-path {
    description
      "IP Fast Reroute next-hop repair path.";
    leaf outgoing-interface {
      type if:interface-state-ref;
      description
        "Name of the outgoing interface.";
    }
    leaf next-hop-address {
      type inet:ip-address-no-zone;
      description
        "IP address of the next hop.";
    }
    leaf metric {
      type uint32;
      description
        "The metric for the repair path. While the IP Fast
         Reroute re-route repair is local and the metric is
         not advertised externally, the metric for repair path
         is useful for troubleshooting purposes.";
    }
    reference
      "RFC 5714: IP Fast Reroute Framework.";
  }
}

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rt:static-routes/v4ur:ipv4/"
  + "v4ur:route/v4ur:next-hop/v4ur:next-hop-options/"
  + "v4ur:simple-next-hop"
{
  description
    "Augment 'simple-next-hop' case in IPv4 unicast route.";
  leaf preference {
    type uint32;
    default "1";
    description
      "The preference is used to select among multiple static
       routes. Routes with a lower preference next-hop are
       preferred and equal preference routes result in
       Equal-Cost-Multi-Path (ECMP) static routes.";
  }
  leaf tag {
    type uint32;
    default "0";
    description
      "The tag is a 32-bit opaque value associated with the
```

```
        route that can be used for policy decisions such as
        advertisement and filtering of the route.";
    }
  }

  augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rt:static-routes/v4ur:ipv4/"
    + "v4ur:route/v4ur:next-hop/v4ur:next-hop-options/"
    + "v4ur:next-hop-list/v4ur:next-hop-list/v4ur:next-hop"
  {
    description
      "Augment static route configuration 'next-hop-list'.";

    leaf preference {
      type uint32;
      default "1";
      description
        "The preference is used to select among multiple static
         routes. Routes with a lower preference next-hop are
         preferred and equal preference routes result in
         Equal-Cost-Multi-Path (ECMP) static routes.";
    }
    leaf tag {
      type uint32;
      default "0";
      description
        "The tag is a 32-bit opaque value associated with the
         route that can be used for policy decisions such as
         advertisement and filtering of the route.";
    }
  }

  augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rt:static-routes/v6ur:ipv6/"
    + "v6ur:route/v6ur:next-hop/v6ur:next-hop-options/"
    + "v6ur:simple-next-hop"
  {
    description
      "Augment 'simple-next-hop' case in IPv6 unicast route.";
    leaf preference {
      type uint32;
      default "1";
      description
        "The preference is used to select among multiple static
         routes. Routes with a lower preference next-hop are
         preferred and equal preference routes result in
         Equal-Cost-Multi-Path (ECMP) static routes.";
    }
    leaf tag {
```

```
    type uint32;
    default "0";
    description
      "The tag is a 32-bit opaque value associated with the
       route that can be used for policy decisions such as
       advertisement and filtering of the route.";
  }
}

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rt:static-routes/v6ur:ipv6/"
  + "v6ur:route/v6ur:next-hop/v6ur:next-hop-options/"
  + "v6ur:next-hop-list/v6ur:next-hop-list/v6ur:next-hop"
{
  description
    "Augment static route configuration 'next-hop-list'.";

  leaf preference {
    type uint32;
    default "1";
    description
      "The preference is used to select among multiple static
       routes. Routes with a lower preference next-hop are
       preferred and equal preference routes result in
       Equal-Cost-Multi-Path (ECMP) static routes.";
  }
  leaf tag {
    type uint32;
    default "0";
    description
      "The tag is a 32-bit opaque value associated with the
       route that can be used for policy decisions such as
       advertisement and filtering of the route.";
  }
}

augment "/rt:routing/rt:ribs/rt:rib"
{
  description
    "Augment a RIB with statistics.";
  uses rib-statistics;
}

augment "/rt:routing/rt:ribs/rt:rib/"
  + "rt:routes/rt:route" {
    description
    "Augment a route in RIB with common attributes.";
  leaf metric {
    when "not(derived-from("
```

```
      + "../rt:source-protocol, 'ospf:ospf')) "
      + "and not(derived-from( "
      + "../rt:source-protocol, 'isis:isis'))" {
      description
        "This augmentation is only valid for routes whose
         source protocol is not OSPF or IS-IS since their YANG
         models already include a 'metric' augmentation for
         routes.";
    }
    type uint32;
    description
      "The metric is a numeric value indicating the cost
       of the route from the perspective of the routing
       protocol installing the route. In general, routes with
       a lower metric installed by the same routing protocol
       are lower cost to reach and are preferable to routes
       with a higher metric. However, metrics from different
       routing protocols are not comparable.";
  }
  leaf-list tag {
    when "not(derived-from("
      + "../rt:source-protocol, 'ospf:ospf')) "
      + "and not(derived-from( "
      + "../rt:source-protocol, 'isis:isis'))" {
      description
        "This augmentation is only valid for routes whose
         source protocol is not OSPF or IS-IS since their YANG
         models already include a 'tag' augmentation for
         routes.";
    }
    type uint32;
    description
      "A tag is a 32-bit opaque value associated with the
       route that can be used for policy decisions such as
       advertisement and filtering of the route.";
  }
  leaf application-tag {
    type uint32;
    description
      "The application-specific tag is an additional tag that
       can be used by applications that require semantics and/or
       policy different from that of the tag. For example,
       the tag is usually automatically advertised in OSPF
       AS-External Link State Advertisements (LSAs) while this
       application-specific tag is not advertised implicitly.";
  }
}

augment "/rt:routing/rt:ribs/rt:rib/"
```

```
    + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
    + "rt:simple-next-hop"
  {
    description
      "Augment simple-next-hop with repair-path.";
    uses repair-path;
  }

  augment "/rt:routing/rt:ribs/rt:rib/"
    + "rt:routes/rt:route/rt:next-hop/rt:next-hop-options/"
    + "rt:next-hop-list/rt:next-hop-list/rt:next-hop"
  {
    description
      "Augment the next-hop with a repair path.";
    uses repair-path;
  }
}

<CODE ENDS>
```

6. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in ietf-rib-extensions.yang module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/v4ur:next-hop-options/v4ur:simple-next-hop/rib-ext:preference

/v4ur:next-hop-options/v4ur:simple-next-hop/rib-ext:tag

/v4ur:next-hop-options/v4ur:next-hop-list/v4ur:next-hop-list /
v4ur:next-hop/rib-ext:preference

/v4ur:next-hop-options/v4ur:next-hop-list/v4ur:next-hop-list / v4ur:next-hop/rib-ext:tag

/v6ur:next-hop-options/v6ur:simple-next-hop/rib-ext:preference

/v6ur:next-hop-options/v6ur:simple-next-hop/rib-ext:tag

/v6ur:next-hop-options/v6ur:next-hop-list/v6ur:next-hop-list / v6ur:next-hop/rib-ext:preference

/v6ur:next-hop-options/v6ur:next-hop-list/v6ur:next-hop-list / v6ur:next-hop/rib-ext:tag

For these augmentations to ietf-routing.yang, the ability to delete, add, and modify IPv4 and IPv6 static route preference and tag would allow traffic to be misrouted.

Some of the readable data nodes in the ietf-rib-extensions.yang module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:ribs/rt:rib/rib-ext:statistics

/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rib-ext:metric

/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rib-ext:tag

/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route /rib-ext:application-tag

/rt:route/rt:next-hop/rt:next-hop-options/rt:simple-next-hop / rib-ext:repair-path

/rt:routes/rt:route/rt:next-hop/rt:next-hop-options /rt:next-hop-list/rt:next-hop-list/rt:next-hop/rib-ext:repair-path

The exposure of the Routing Information Base (RIB) will expose the routing topology of the network. This may be undesirable due to the fact that exposure may facilitate other attacks. Additionally, network operators may consider their topologies to be sensitive confidential data.

All the security considerations for [RFC8349] writable and readable data nodes apply to the augmentations described herein.

7. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-rib-extension
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-rib-extension
namespace: urn:ietf:params:xml:ns:yang:ietf-rib-extension
prefix: rib-ext
reference: RFC XXXX

8. References

8.1. Normative References

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol

(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

8.2. Informative References

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <https://www.rfc-editor.org/info/rfc5714>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[RFC9129] Yeung, D., Qu, Y., Zhang, J., Chen, I., and A. Lindem, "YANG Data Model for the OSPF Protocol", RFC 9129, DOI 10.17487/RFC9129, October 2022, <https://www.rfc-editor.org/info/rfc9129>.

[RFC9130] Litkowski, S., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for IS-IS Protocol", RFC 9130, DOI 10.17487/RFC9130, October 2022, <https://www.rfc-editor.org/info/rfc9130>.

Appendix A. Combined Tree Diagram

This appendix includes the combined ietf-routing.yang, ietf-ipv4-unicast-routing.yang, ietf-ipv6-unicast-routing.yang and ietf-rib-extensions.yang tree diagram.

```
module: ietf-routing
+--rw routing
+--rw router-id?                yang:dotted-quad {router-id}?
+--ro interfaces
|  +--ro interface*   if:interface-ref
+--rw control-plane-protocols
|  +--rw control-plane-protocol* [type name]
|     +--rw type             identityref
|     +--rw name             string
|     +--rw description?     string
|     +--rw static-routes
|        +--rw v4ur:ipv4
|        |  +--rw v4ur:route* [destination-prefix]
|        |     +--rw v4ur:destination-prefix    inet:ipv4-prefix
|        |     +--rw v4ur:description?          string
|        |     +--rw v4ur:next-hop
|        |        +--rw (v4ur:next-hop-options)
|        |           +--:(v4ur:simple-next-hop)
|        |           |  +--rw v4ur:outgoing-interface?
|        |           |  |   if:interface-ref
|        |           |  +--rw v4ur:next-hop-address?
|        |           |  |   inet:ipv4-address
|        |           |  +--rw rib-ext:preference?        uint32
|        |           |  +--rw rib-ext:tag?               uint32
|        |           +--:(v4ur:special-next-hop)
|        |           |  +--rw v4ur:special-next-hop?     enumeration
|        |           +--:(v4ur:next-hop-list)
|        |              +--rw v4ur:next-hop-list
|        |                 +--rw v4ur:next-hop* [index]
|        |                    +--rw v4ur:index                string
|        |                    +--rw v4ur:outgoing-interface?
|        |                    |   if:interface-ref
|        |                    +--rw v4ur:next-hop-address?
|        |                    |   inet:ipv4-address
|        |                    +--rw rib-ext:preference?       uint32
|        |                    +--rw rib-ext:tag?              uint32
|        +--rw v6ur:ipv6
|           +--rw v6ur:route* [destination-prefix]
|              +--rw v6ur:destination-prefix    inet:ipv6-prefix
|              +--rw v6ur:description?          string
|              +--rw v6ur:next-hop
|                 +--rw (v6ur:next-hop-options)
|                    +--:(v6ur:simple-next-hop)
|                    |  +--rw v6ur:outgoing-interface?
|                    |  |   if:interface-ref
|                    |  +--rw v6ur:next-hop-address?
|                    |  |   inet:ipv6-address
|                    |  +--rw rib-ext:preference?       uint32
|                    |  +--rw rib-ext:tag?              uint32
```

```
|                    +--:(v6ur:special-next-hop)
|                    |  +--rw v6ur:special-next-hop?     enumeration
|                    +--:(v6ur:next-hop-list)
|                       +--rw v6ur:next-hop-list
|                          +--rw v6ur:next-hop* [index]
|                             +--rw v6ur:index                 string
|                             +--rw v6ur:outgoing-interface?
|                             |   if:interface-ref
|                             +--rw v6ur:next-hop-address?
|                             |   inet:ipv6-address
|                             +--rw rib-ext:preference?        uint32
|                             +--rw rib-ext:tag?               uint32
+--rw ribs
   +--rw rib* [name]
      +--rw name                              string
      +--rw address-family                    identityref
      +--ro default-rib?                      boolean {multiple-ribs}?
      +--ro routes
      |  +--ro route* []
      |     +--ro route-preference?          route-preference
      |     +--ro next-hop
      |     |  +--ro (next-hop-options)
      |     |     +--:(simple-next-hop)
      |     |     |  +--ro outgoing-interface?      if:interface-ref
      |     |     |  +--ro v4ur:next-hop-address?   inet:ipv4-address
      |     |     |  +--ro v6ur:next-hop-address?   inet:ipv6-address
      |     |     |  +--ro rib-ext:repair-path
      |     |     |     +--ro rib-ext:outgoing-interface?
      |     |     |     |   if:interface-state-ref
      |     |     |     +--ro rib-ext:next-hop-address?
      |     |     |     |   inet:ip-address-no-zone
      |     |     |     +--ro rib-ext:metric?              uint32
      |     |     +--:(special-next-hop)
      |     |     |  +--ro special-next-hop?        enumeration
      |     |     +--:(next-hop-list)
      |     |        +--ro next-hop-list
      |     |           +--ro next-hop* []
      |     |              +--ro outgoing-interface?
      |     |              |   if:interface-ref
      |     |              +--ro v4ur:address?
      |     |              |   inet:ipv4-address
      |     |              +--ro v6ur:address?
      |     |              |   inet:ipv6-address
      |     |              +--ro rib-ext:repair-path
      |     |                 +--ro rib-ext:outgoing-interface?
      |     |                 |   if:interface-state-ref
      |     |                 +--ro rib-ext:next-hop-address?
      |     |                 |   inet:ip-address-no-zone
      |     |                 +--ro rib-ext:metric?           uint32
```

```
|     +--ro source-protocol            identityref
|     +--ro active?                    empty
|     +--ro last-updated?              yang:date-and-time
|     +--ro v4ur:destination-prefix?   inet:ipv4-prefix
|     +--ro v6ur:destination-prefix?   inet:ipv6-prefix
|     +--ro rib-ext:metric?            uint32
|     +--ro rib-ext:tag*               uint32
|     +--ro rib-ext:application-tag?   uint32
+---x active-route
|  +---w input
|  |  +---w v4ur:destination-address?   inet:ipv4-address
|  |  +---w v6ur:destination-address?   inet:ipv6-address
|  +--ro output
|     +--ro route
|        +--ro next-hop
|        |  +--ro (next-hop-options)
|        |     +--:(simple-next-hop)
|        |     |  +--ro outgoing-interface?    if:interface-ref
|        |     |  +--ro v4ur:next-hop-address?
|        |     |  |   inet:ipv4-address
|        |     |  +--ro v6ur:next-hop-address?
|        |     |  |   inet:ipv6-address
|        |     +--:(special-next-hop)
|        |     |  +--ro special-next-hop?        enumeration
|        |     +--:(next-hop-list)
|        |        +--ro next-hop-list
|        |           +--ro next-hop* []
|        |              +--ro outgoing-interface?
|        |              |   if:interface-ref
|        |              +--ro v4ur:next-hop-address?
|        |              |   inet:ipv4-address
|        |              +--ro v6ur:next-hop-address?
|        |              |   inet:ipv6-address
|        +--ro source-protocol            identityref
|        +--ro active?                    empty
|        +--ro last-updated?              yang:date-and-time
|        +--ro v4ur:destination-prefix?   inet:ipv4-prefix
|        +--ro v6ur:destination-prefix?   inet:ipv6-prefix
+--rw description?                      string
+--ro rib-ext:statistics
   +--ro rib-ext:total-routes?              uint32
   +--ro rib-ext:total-active-routes?       uint32
   +--ro rib-ext:total-route-memory?        uint64
   +--ro rib-ext:protocol-statistics* []
      +--ro rib-ext:protocol?             identityref
      +--ro rib-ext:routes?    uint32
      +--ro rib-ext:active-routes?   uint32
      +--ro rib-ext:route-memory?    uint64
```

Appendix B. ietf-rib-extension.yang example

The following is an XML example using the RIB extension module and RFC 8349.

Note: '\' line wrapping per [RFC8792].

```
<routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
  <control-plane-protocols>
    <control-plane-protocol>
      <type>static</type>
      <name>static-routing-protocol</name>
      <static-routes>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv4-unicast-routing">
          <route>
            <destination-prefix>0.0.0.0/0</destination-prefix>
            <next-hop>
              <next-hop-address>192.0.2.2</next-hop-address>
              <preference xmlns="urn:ietf:params:xml:ns:yang:\
                ietf-rib-extension">30</preference>
              <tag xmlns="urn:ietf:params:xml:ns:yang:\
                ietf-rib-extension">99</tag>
            </next-hop>
          </route>
        </ipv4>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv6-unicast-routing">
          <route>
            <destination-prefix>::/0</destination-prefix>
            <next-hop>
             <next-hop-address>2001:db8:aaaa::1111</next-hop-address>
             <preference xmlns="urn:ietf:params:xml:ns:yang:\
               ietf-rib-extension">30</preference>
             <tag xmlns="urn:ietf:params:xml:ns:yang:\
               ietf-rib-extension">66</tag>
            </next-hop>
          </route>
        </ipv6>
      </static-routes>
    </control-plane-protocol>
  </control-plane-protocols>
  <ribs>
    <rib>
      <name>ipv4-primary</name>
      <address-family xmlns:v4ur="urn:ietf:params:xml:ns:yang:\
        ietf-ipv4-unicast-routing">v4ur:ipv4-unicast</address-family>
      <default-rib>true</default-rib>
      <routes>
        <route>
          <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:\
            ietf-ipv4-unicast-routing">0.0.0.0/0</destination-prefix>
          <next-hop>
            <next-hop-address xmlns="urn:ietf:params:xml:ns:yang:\
              ietf-ipv4-unicast-routing">192.0.2.2</next-hop-address>
          </next-hop>
```

```
      <route-preference>5</route-preference>
      <source-protocol>static</source-protocol>
      <last-updated>2015-10-24T18:02:45+02:00</last-updated>
    </route>
    <route>
      <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:\
        ietf-ipv4-unicast-routing">198.51.100.0/24\
      </destination-prefix>
      <next-hop>
        <next-hop-address xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv4-unicast-routing">192.0.2.2</next-hop-address>
        <repair-path xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-rib-extension">
          <next-hop-address>203.0.113.1</next-hop-address>
          <metric>200</metric>
        </repair-path>
      </next-hop>
      <route-preference>110</route-preference>
      <source-protocol xmlns:ospf="urn:ietf:params:xml:ns:yang:\
        ietf-ospf">ospf:ospf</source-protocol>
      <last-updated>2015-10-24T18:02:45+02:00</last-updated>
    </route>
  </routes>
</rib>
<rib>
  <name>ipv6-primary</name>
  <address-family xmlns:v6ur="urn:ietf:params:xml:ns:yang:\
    ietf-ipv6-unicast-routing">v6ur:ipv6-unicast</address-family>
  <default-rib>true</default-rib>
  <routes>
    <route>
      <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:\
        ietf-ipv6-unicast-routing">0::/0</destination-prefix>
      <next-hop>
        <next-hop-address xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv6-unicast-routing">2001:db8:aaaa::1111\
        </next-hop-address>
      </next-hop>
      <route-preference>5</route-preference>
      <source-protocol>static</source-protocol>
      <last-updated>2015-10-24T18:02:45+02:00</last-updated>
    </route>
    <route>
      <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:\
        ietf-ipv6-unicast-routing">2001:db8:bbbb::/64\
      </destination-prefix>
      <next-hop>
        <next-hop-address xmlns="urn:ietf:params:xml:ns:yang:\
          ietf-ipv6-unicast-routing">2001:db8:aaaa::1111\
```

```
            </next-hop-address>
            <repair-path xmlns="urn:ietf:params:xml:ns:yang:\
             ietf-rib-extension">
             <next-hop-address>2001:db8:cccc::2222</next-hop-address>
             <metric>200</metric>
            </repair-path>
          </next-hop>
          <route-preference>110</route-preference>
          <source-protocol xmlns:ospf="urn:ietf:params:xml:ns:yang:\
            ietf-ospf">ospf:ospf</source-protocol>
          <last-updated>2015-10-24T18:02:45+02:00</last-updated>
        </route>
      </routes>
    </rib>
  </ribs>
</routing>
```

The following is the same example using JSON format.

```
{
 "ietf-routing:routing": {
   "control-plane-protocols": {
     "control-plane-protocol": [
       {
         "type": "static",
         "name": "static-routing-protocol",
         "static-routes": {
           "ietf-ipv4-unicast-routing:ipv4": {
             "route": [
               {
                 "destination-prefix": "0.0.0.0/0",
                 "next-hop": {
                   "next-hop-address": "192.0.2.2",
                   "ietf-rib-extension:preference": 30,
                   "ietf-rib-extension:tag": 99
                 }
               }
             ]
           },
           "ietf-ipv6-unicast-routing:ipv6": {
             "route": [
               {
                 "destination-prefix": "::/0",
                 "next-hop": {
                   "next-hop-address": "2001:db8:aaaa::1111",
                   "ietf-rib-extension:preference": 30,
                   "ietf-rib-extension:tag": 66
                 }
               }
             ]
           }
         }
       }
     ]
   },
   "ribs": {
     "rib": [
       {
         "name": "ipv4-primary",
         "address-family": "ietf-ipv4-unicast-routing:ipv4-unicast",
         "default-rib": true,
         "routes": {
           "route": [
             {
               "next-hop": {
                 "ietf-ipv4-unicast-routing:next-hop-address": \
                 "192.0.2.2"
               },
```

```
        "route-preference": 5,
        "source-protocol": "static",
        "last-updated": "2015-10-24T18:02:45+02:00",
        "ietf-ipv4-unicast-routing:destination-prefix": \
        "0.0.0.0/0"
      },
      {
        "next-hop": {
          "ietf-rib-extension:repair-path": {
            "next-hop-address": "203.0.113.1",
            "metric": 200
          },
          "ietf-ipv4-unicast-routing:next-hop-address": \
          "192.0.2.2"
        },
        "route-preference": 110,
        "source-protocol": "ietf-ospf:ospf",
        "last-updated": "2015-10-24T18:02:45+02:00",
        "ietf-ipv4-unicast-routing:destination-prefix": \
        "198.51.100.0/24"
      }
    ]
  }
},
{
  "name": "ipv6-primary",
  "address-family": "ietf-ipv6-unicast-routing:ipv6-unicast",
  "default-rib": true,
  "routes": {
    "route": [
      {
        "next-hop": {
          "ietf-ipv6-unicast-routing:next-hop-address": \
          "2001:db8:aaaa::1111"
        },
        "route-preference": 5,
        "source-protocol": "static",
        "last-updated": "2015-10-24T18:02:45+02:00",
        "ietf-ipv6-unicast-routing:destination-prefix": "::/0"
      },
      {
        "next-hop": {
          "ietf-rib-extension:repair-path": {
            "next-hop-address": "2001:db8:cccc::2222",
            "metric": 200
          },
          "ietf-ipv6-unicast-routing:next-hop-address": \
          "2001:db8:aaaa::1111"
        },
```

```
              "route-preference": 110,
              "source-protocol": "ietf-ospf:ospf",
              "last-updated": "2015-10-24T18:02:45+02:00",
              "ietf-ipv6-unicast-routing:destination-prefix": \
              "2001:db8:bbbb::/64"
            }
          ]
        }
      }
    ]
  }
 }
}
```

Appendix C. Acknowledgments

The RFC text was produced using Marshall Rose's xml2rfc tool.

The authors wish to thank Les Ginsberg, Krishna Deevi, and Suyoung Yoon for their helpful comments and suggestions.

The authors wish to thank Tom Petch, Rob Wilton, Chris Hopps, Martin Bjorklund, Jeffrey Zhang, Eric Vyncke, Lars Eggert, and Bo Wu for their reviews and comments.

Authors' Addresses

Acee Lindem
LabN Consulting LLC
301 Midenhall Way
Cary, NC 27513
United States of America

Email: acee.ietf@gmail.com

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
United States of America

Email: yingzhen.qu@futurewei.com