

Internet Engineering Task Force (IETF)  
Request for Comments: 8431  
Category: Standards Track  
ISSN: 2070-1721

L. Wang  
Individual  
M. Chen  
Huawei  
A. Dass  
Ericsson  
H. Ananthakrishnan  
Netflix  
S. Kini  
Individual  
N. Bahadur  
Uber  
September 2018

## A YANG Data Model for the Routing Information Base (RIB)

Модель данных YANG для базы маршрутной информации (RIB)

### Аннотация

Документ определяет модель данных YANG для базы маршрутной информации (Routing Information Base или RIB), согласованную с информационной моделью RIB интерфейса в систему маршрутизации (Routing System или I2RS).

### Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF<sup>1</sup> и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG<sup>2</sup>. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <https://www.rfc-editor.org/info/rfc8431>.

Авторские права

Авторские права (Copyright (c) 2018) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с упрощённой лицензией BSD, как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

## Оглавление

1. Введение.....	1
1.1. Уровни требований.....	2
1.2. Определения и сокращения.....	2
1.3. Диаграммы деревьев.....	2
2. Структура модели.....	2
2.1. Возможности RIB.....	4
2.2. Экземпляр маршрутизации и RIB.....	4
2.3. Маршрут.....	5
2.4. Следующий узел.....	5
2.5. Операции RPC.....	7
2.6. Уведомления.....	9
3. Модуль YANG.....	10
4. Взаимодействие с IANA.....	34
5. Вопросы безопасности.....	34
6. Литература.....	34
6.1. Нормативные документы.....	34
6.2. Дополнительная литература.....	35
Благодарности.....	35
Участники работы.....	35
Адреса авторов.....	35

## 1. Введение

Интерфейс в систему маршрутизации (I2RS) [RFC7921] обеспечивает доступ к чтению и записи данных и состояний в процессах маршрутизации, имеющихся внутри элементов маршрутизации. Это обеспечивается обменом сообщениями

<sup>1</sup>Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

<sup>2</sup>Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

протокола между клиентами I2RS и агентами I2RS, связанными с системой маршрутизации. Одними из функций I2RS являются чтение и запись данных базы маршрутной информации (RIB). В [I2RS-REQS] даны примеры применения RIB. Информационная модель RIB определена в [RFC8430].

Этот документ задаёт модель данных YANG [RFC7950] [RFC6991] для RIB, которая соответствует применению RIB и согласована с информационной моделью RIB.

## 1.1. Уровни требований

Ключевые слова **должно** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

## 1.2. Определения и сокращения

### RIB

Routing Information Base - база маршрутной информации.

### FIB

Forwarding Information Base - база информации для пересылки.

### RPC

Remote Procedure Call - удалённый вызов процедуры.

### IM

Information Model - информационная модель. Абстрактная модель концептуальной области, независимая от конкретной реализации или представления данных.

## 1.3. Диаграммы деревьев

Диаграммы деревьев в этом документе приведены в нотации [RFC8340].

## 2. Структура модели

На рисунке 1 обзорно показана структура дерева модуля ietf-i2rs-rib. Для краткости некоторые детали опущены, их можно найти в последующих параграфах.

```

module: ietf-i2rs-rib
  +---rw routing-instance
    +---rw name string
    +---rw interface-list* [name]
      | +---rw name if:interface-ref
    +---rw router-id? yang:dotted-quad
    +---rw lookup-limit? uint8
    +---rw rib-list* [name]
      +---rw name string
      +---rw address-family address-family-definition
      +---rw ip-rpf-check? boolean
      +---rw route-list* [route-index]
        | +---rw route-index uint64
        | +---rw match
        | | +---rw (route-type)?
        | | | +---:(ipv4)
        | | | | ...
        | | | +---:(ipv6)
        | | | | ...
        | | | +---:(mpls-route)
        | | | | ...
        | | | +---:(mac-route)
        | | | | ...
        | | | +---:(interface-route)
        | | | | ...
        | | +---rw nexthop
        | | | +---rw nexthop-id? uint32
        | | | +---rw sharing-flag? boolean
        | | | +---rw (nexthop-type)?
        | | | | +---:(nexthop-base)
        | | | | | ...
        | | | | +---:(nexthop-chain) {nexthop-chain}?
        | | | | | ...
        | | | | +---:(nexthop-replicate) {nexthop-replicate}?
        | | | | | ...
        | | | | +---:(nexthop-protection) {nexthop-protection}?
        | | | | | ...
        | | | | +---:(nexthop-load-balance) {nexthop-load-balance}?
        | | | | | ...
        | | +---rw route-status
        | | | ...
        | +---rw route-attributes
        | | ...
        +---rw route-vendor-attributes
    +---rw nexthop-list* [nexthop-member-id]
      +---rw nexthop-member-id uint32

rpcs:
  +---x rib-add
    | +---w input
    | | +---w name string
  
```

```

| | +---w address-family      address-family-definition
| | +---w ip-rpf-check?      boolean
| | +--ro output
| |   +--ro result boolean
| |   +--ro reason? string
+---x rib-delete
| +---w input
| | +---w name string
| | +--ro output
| |   +--ro result boolean
| |   +--ro reason? string
+---x route-add
| +---w input
| | +---w return-failure-detail?  boolean
| | +---w rib-name                string
| | +---w routes
| |   +---w route-list* [route-index]
| |   ...
| | +--ro output
| |   +--ro success-count        uint32
| |   +--ro failed-count        uint32
| |   +--ro failure-detail
| |     +--ro failed-routes* [route-index]
| |     +--ro route-index uint32
| |     +--ro error-code? uint32
+---x route-delete
| +---w input
| | +---w return-failure-detail?  boolean
| | +---w rib-name                string
| | +---w routes
| |   +---w route-list* [route-index]
| |   ...
| | +--ro output
| |   +--ro success-count        uint32
| |   +--ro failed-count        uint32
| |   +--ro failure-detail
| |     +--ro failed-routes* [route-index]
| |     +--ro route-index uint32
| |     +--ro error-code? uint32
+---x route-update
| +---w input
| | +---w return-failure-detail?  boolean
| | +---w rib-name                string
| | +---w (match-options)?
| |   +---:(match-route-prefix)
| |   | ...
| |   +---:(match-route-attributes)
| |   | ...
| |   +---:(match-route-vendor-attributes) {...}?
| |   | ...
| |   +---:(match-nexthop)
| |   ...
| | +--ro output
| |   +--ro success-count uint32
| |   +--ro failed-count uint32
| |   +--ro failure-detail
| |     +--ro failed-routes* [route-index]
| |     +--ro route-index uint32
| |     +--ro error-code? uint32
+---x nh-add
| +---w input
| | +---w rib-name                string
| | +---w nexthop-id?            uint32
| | +---w sharing-flag?         boolean
| | +---w (nexthop-type)?
| |   +---:(nexthop-base)
| |   | ...
| |   +---:(nexthop-chain) {nexthop-chain}?
| |   | ...
| |   +---:(nexthop-replicate) {nexthop-replicate}?
| |   | ...
| |   +---:(nexthop-protection) {nexthop-protection}?
| |   | ...
| |   +---:(nexthop-load-balance) {nexthop-load-balance}?
| |   ...
| | +--ro output
| |   +--ro result              boolean
| |   +--ro reason?            string
| |   +--ro nexthop-id?       uint32
+---x nh-delete
| +---w input
| | +---w rib-name                string
| | +---w nexthop-id?            uint32
| | +---w sharing-flag?         boolean
| | +---w (nexthop-type)?
| |   +---:(nexthop-base)

```

```

| | | ...
| | | +--:(nexthop-chain) {nexthop-chain}?
| | | | ...
| | | +--:(nexthop-replicate) {nexthop-replicate}?
| | | | ...
| | | +--:(nexthop-protection) {nexthop-protection}?
| | | | ...
| | | +--:(nexthop-load-balance) {nexthop-load-balance}?
| | | | ...
+--ro output
  +--ro result boolean
  +--ro reason? string
notifications:
+--n nexthop-resolution-status-change
| +--ro nexthop
| | +--ro nexthop-id?          uint32
| | +--ro sharing-flag?      boolean
| | +--ro (nexthop-type)?
| |   +--:(nexthop-base)
| |   | ...
| |   +--:(nexthop-chain) {nexthop-chain}?
| |   | ...
| |   +--:(nexthop-replicate) {nexthop-replicate}?
| |   | ...
| |   +--:(nexthop-protection) {nexthop-protection}?
| |   | ...
| |   +--:(nexthop-load-balance) {nexthop-load-balance}?
| |   | ...
| | +--ro nexthop-state nexthop-state-definition
+--n route-change
  +--ro rib-name              string
  +--ro address-family        address-family-definition
  +--ro route-index          uint64
  +--ro match
  | +--ro (route-type)?
  |   +--:(ipv4)
  |   | ...
  |   +--:(ipv6)
  |   | ...
  |   +--:(mpls-route)
  |   | ...
  |   +--:(mac-route)
  |   | ...
  |   +--:(interface-route)
  |   | ...
  +--ro route-installed-state route-installed-state-definition
  +--ro route-state           route-state-definition
  +--ro route-change-reasons* [route-change-reason]
  +--ro route-change-reason   route-change-reason-definition

```

Рисунок 1. Структура модуля I2RS RIB.

## 2.1. Возможности RIB

Согласование возможностей RIB очень важно, поскольку не всё оборудование поддерживает все виды nexthop (следующий узел) и могут быть ограничения на число фактически выполняемых уровней поиска. Поэтому модель данных RIB должна указывать способ, с помощью которого внешний объект может узнать функциональные возможности сетевого устройства.

В то же время могут применяться цепочки nexthop для задания нескольких заголовков пакета перед пересылкой конкретного пакета. Не каждое сетевое устройство способно поддерживать все виды цепочек nexthop вместе с произвольным числом связанных в цепочку заголовков. Модели данных RIB нужен способ раскрытия возможностей поддержки цепочек nexthop на данном сетевом устройстве.

Этот модуль использует операторы feature и if-feature для анонсирования возможностей.

## 2.2. Экземпляр маршрутизации и RIB

Экземпляром маршрутизации в контексте информационной модели RIB является набор RIB, интерфейсов и параметров протоколов маршрутизации. Экземпляр маршрутизации создаёт логический срез (slice) маршрутизатора и может позволять множеству таких логических срезов в наборе маршрутизаторов взаимодействовать между собой. Параметры протоколов маршрутизации управляют сведениями, доступными в RIB. Более подробное описание экземпляров маршрутизации приведено в параграфе 2.2 [RFC8430].

Экземпляр маршрутизации может иметь несколько RIB, поэтому в модели для их представления используется список (list). Структура дерева для экземпляра маршрутизации представлена на рисунке 2

```

+--rw routing-instance
  +--rw name                string
  +--rw interface-list* [name]
  | +--rw name if:interface-ref
  +--rw router-id?         yang:dotted-quad
  +--rw lookup-limit?     uint8
  +--rw rib-list* [name]
  +--rw name                string
  +--rw address-family      address-family-definition

```

```

+--rw ip-rpf-check?   boolean
+--rw route-list* [route-index]
  ... // См. параграф 2.3

```

Рисунок 2. Структура экземпляра маршрутизации.

## 2.3. Маршрут

Маршрут - это, по сути, условие сопоставления и действие, выполняемое по этому условию. Условие задаёт тип маршрута (например, IPv4, MPLS, MAC<sup>1</sup>, Interface, etc.) and the set of fields to match on.

Маршрут **должен** содержать атрибут ROUTE\_PREFERENCE (см. параграф 2.3 в [RFC8430]). Кроме того, в откликах на операции чтения или записи RIB с маршрутом **должен** быть связан один из указанных ниже атрибутов состояния.

- Active (активен) указывает, имеется ли у маршрута хотя бы один полностью распознанный nexthop, который, следовательно подходит для включения в FIB.
- Installed (установлен), указывает, что маршрут включён в FIB.
- Reason (причина) указывает конкретную причину отказа, например, Not authorized (не разрешено).

Дополнительно маршрут может иметь один или несколько атрибутов (например, route-vendor-attributes).

RIB включает множество маршрутов, которые указываются в форме списка в дереве соответствующей базы RIB. Каждая база RIB имеет свой список маршрутов.

```

+--rw route-list* [route-index]
  +--rw route-index          uint64
  +--rw match
    | +--rw (route-type)?
    |   +--:(ipv4)
    |   | +--rw ipv4
    |   |   +--rw (ip-route-match-type)?
    |   |   +--:(dest-ipv4-address)
    |   |   | ...
    |   |   +--:(src-ipv4-address)
    |   |   | ...
    |   |   +--:(dest-src-ipv4-address)
    |   |   | ...
    |   +--:(ipv6)
    |   | +--rw ipv6
    |   |   +--rw (ip-route-match-type)?
    |   |   +--:(dest-ipv6-address)
    |   |   | ...
    |   |   +--:(src-ipv6-address)
    |   |   | ...
    |   |   +--:(dest-src-ipv6-address)
    |   |   | ...
    |   +--:(mpls-route)
    |   | +--rw mpls-label          uint32
    |   +--:(mac-route)
    |   | +--rw mac-address          uint32
    |   +--:(interface-route)
    |   | +--rw interface-identifier if:interface-ref
  +--rw nexthop
  | ... (refer to Section 2.4)

```

Рисунок 3. Структура маршрута.

## 2.4. Следующий узел

Объект nexthop является результатом поиска маршрута. Как показано на рисунке 4 в [RFC8430], для поддержки вариантов применения (например, распределение нагрузки, защита, групповая передача или их сочетание) nexthop моделируется многоуровневой структурой и поддерживает рекурсию. Первый уровень nexthop включает 4 типа.

- Базовый (base) служит основой для остальных типов и включает:
  - nexthop-id;
  - адрес IPv4;
  - адрес IPv6;
  - egress-interface
  - egress-interface с адресом IPv4;
  - egress-interface с адресом IPv6;
  - egress-interface с адресом MAC;
  - logical-tunnel;
  - tunnel-encapsulation;
  - tunnel-decapsulation;
  - rib-name.
- Цепочка (chain) обеспечивает способ выполнить над пакетом несколько операций, объединяя их логически.
- Балансировка нагрузки (load-balance) предназначена для распределения трафика, обычно с применением нескольких взвешенных nexthop.
- Защитный (protection) служит для вариантов защиты, обычно включающих основной и резервный nexthop.
- Репликация (replicate) служит для пересылки нескольким адресатам.

<sup>1</sup>Media Access Control - управление доступом к среде.

Дерево nexthop показано на рисунках 4 и 5.

```

+--rw nexthop
| +--rw nexthop-id?          uint32
| +--rw sharing-flag?      boolean
| +--rw (nexthop-type)?
| | +--:(nexthop-base)
| | | ... (refer to Figure 5)
| | +--:(nexthop-chain) {nexthop-chain}?
| | | +--rw nexthop-chain
| | | | +--rw nexthop-list* [nexthop-member-id]
| | | | +--rw nexthop-member-id uint32
| | +--:(nexthop-replicate) {nexthop-replicate}?
| | | +--rw nexthop-replicate
| | | | +--rw nexthop-list* [nexthop-member-id]
| | | | +--rw nexthop-member-id uint32
| | +--:(nexthop-protection) {nexthop-protection}?
| | | +--rw nexthop-protection
| | | | +--rw nexthop-list* [nexthop-member-id]
| | | | +--rw nexthop-member-id uint32
| | | | +--rw nexthop-preference nexthop-preference-definition
| +--:(nexthop-load-balance) {nexthop-load-balance}?
| | +--rw nexthop-lb
| | | +--rw nexthop-list* [nexthop-member-id]
| | | +--rw nexthop-member-id uint32
| | | +--rw nexthop-lb-weight nexthop-lb-weight-definition

```

Рисунок 4. Структура Nexthop.

На рисунке 5 приведён фрагмент дерева nexthop, содержащий структуру базового nexthop.

```

+--:(nexthop-base)
| +--rw nexthop-base
| | +--rw (nexthop-base-type)?
| | | +--:(special-nexthop)
| | | | +--rw special? special-nexthop-definition
| | +--:(egress-interface-nexthop)
| | | +--rw outgoing-interface if:interface-ref
| | +--:(ipv4-address-nexthop)
| | | +--rw ipv4-address inet:ipv4-address
| | +--:(ipv6-address-nexthop)
| | | +--rw ipv6-address inet:ipv6-address
| | +--:(egress-interface-ipv4-nexthop)
| | | +--rw egress-interface-ipv4-address
| | | | +--rw outgoing-interface if:interface-ref
| | | | +--rw ipv4-address      inet:ipv4-address
| | +--:(egress-interface-ipv6-nexthop)
| | | +--rw egress-interface-ipv6-address
| | | | +--rw outgoing-interface if:interface-ref
| | | | +--rw ipv6-address      inet:ipv6-address
| | +--:(egress-interface-mac-nexthop)
| | | +--rw egress-interface-mac-address
| | | | +--rw outgoing-interface if:interface-ref
| | | | +--rw ieee-mac-address yang:mac-address
| | +--:(tunnel-encapsulation-nexthop) {nexthop-tunnel}?
| | | +--rw tunnel-encapsulation
| | | | +--rw (tunnel-type)?
| | | | | +--:(ipv4) {ipv4-tunnel}?
| | | | | | +--rw ipv4-header
| | | | | | | +--rw src-ipv4-address inet:ipv4-address
| | | | | | | +--rw dest-ipv4-address inet:ipv4-address
| | | | | | | +--rw protocol          uint8
| | | | | | | +--rw ttl?             uint8
| | | | | | | +--rw dscp?           uint8
| | | | | +--:(ipv6) {ipv6-tunnel}?
| | | | | | +--rw ipv6-header
| | | | | | | +--rw src-ipv6-address inet:ipv6-address
| | | | | | | +--rw dest-ipv6-address inet:ipv6-address
| | | | | | | +--rw next-header      uint8
| | | | | | | +--rw traffic-class? uint8
| | | | | | | +--rw flow-label?
| | | | | | | | inet:ipv6-flow-label
| | | | | | | +--rw hop-limit?      uint8
| | | | +--:(mpls) {mpls-tunnel}?
| | | | | +--rw mpls-header
| | | | | | +--rw label-operations* [label-oper-id]
| | | | | | +--rw label-oper-id uint32
| | | | | | +--rw (label-actions)?
| | | | | | | +--:(label-push)
| | | | | | | | +--rw label-push
| | | | | | | | | +--rw label          uint32
| | | | | | | | | +--rw s-bit?         boolean
| | | | | | | | | +--rw tc-value?    uint8
| | | | | | | | | +--rw ttl-value?   uint8
| | | | | | +--:(label-swap)
| | | | | | | +--rw label-swap
| | | | | | | +--rw out-label      uint32

```

```

+---rw ttl-action?
|         ttl-action-definition
+---: (gre) {gre-tunnel}?
| +---rw gre-header
| | +---rw (dest-address-type)?
| | | +---: (ipv4)
| | | | +---rw ipv4-dest inet:ipv4-address
| | | +---: (ipv6)
| | | | +---rw ipv6-dest inet:ipv6-address
| | +---rw protocol-type uint16
| | +---rw key? uint64
+---: (nvgre) {nvgre-tunnel}?
| +---rw nvgre-header
| | +---rw (nvgre-type)?
| | | +---: (ipv4)
| | | | +---rw src-ipv4-address inet:ipv4-address
| | | | +---rw dest-ipv4-address inet:ipv4-address
| | | | +---rw protocol uint8
| | | | +---rw ttl? uint8
| | | | +---rw dscp? uint8
| | | +---: (ipv6)
| | | | +---rw src-ipv6-address inet:ipv6-address
| | | | +---rw dest-ipv6-address inet:ipv6-address
| | | | +---rw next-header uint8
| | | | +---rw traffic-class? uint8
| | | | +---rw flow-label?
| | | | | inet:ipv6-flow-label
| | | | +---rw hop-limit? uint8
| | +---rw virtual-subnet-id uint32
| | +---rw flow-id? uint8
+---: (vxlan) {vxlan-tunnel}?
+---rw vxlan-header
+---rw (vxlan-type)?
| +---: (ipv4)
| | +---rw src-ipv4-address inet:ipv4-address
| | | +---rw dest-ipv4-address inet:ipv4-address
| | | +---rw protocol uint8
| | | +---rw ttl? uint8
| | | +---rw dscp? uint8
| | +---: (ipv6)
| | | +---rw src-ipv6-address inet:ipv6-address
| | | | +---rw dest-ipv6-address inet:ipv6-address
| | | | +---rw next-header uint8
| | | | +---rw traffic-class? uint8
| | | | +---rw flow-label? inet:ipv6-flow-label
| | | | +---rw hop-limit? uint8
+---rw vxlan-identifier uint32
+---: (tunnel-decapsulation-nexthop) {nexthop-tunnel}?
+---rw tunnel-decapsulation
+---rw (tunnel-type)?
+---: (ipv4) {ipv4-tunnel}?
| +---rw ipv4-decapsulation
| | +---rw ipv4-decapsulation
| | | tunnel-decapsulation-action-definition
| | | +---rw ttl-action? ttl-action-definition
+---: (ipv6) {ipv6-tunnel}?
| +---rw ipv6-decapsulation
| | +---rw ipv6-decapsulation
| | | tunnel-decapsulation-action-definition
| | | +---rw hop-limit-action?
| | | | hop-limit-action-definition
+---: (mpls) {mpls-tunnel}?
+---rw label-pop
+---rw label-pop mpls-label-action-definition
+---rw ttl-action? ttl-action-definition
+---: (logical-tunnel-nexthop) {nexthop-tunnel}?
+---rw logical-tunnel
+---rw tunnel-type tunnel-type-definition
+---rw tunnel-name string
+---: (rib-name-nexthop)
+---rw rib-name? string
+---: (nexthop-identifier)
+---rw nexthop-ref nexthop-ref

```

Рисунок 5. Структура базового Nexthop.

## 2.5. Операции RPC

Этот модуль определяет указанные ниже операции RPC

- `rib-add` добавляет RIB в экземпляр маршрутизации, принимая на входе имя RIB, семейство адресов RIB и (необязательно) режим проверки RPF. На выходе возвращается значение `true` при успехе или `false` при отказе (в случае отказа агент I2RS может возвращать конкретную причину ошибки).

- `rib-delete` удаляет RIB из экземпляра маршрутизации. При удалении RIB удаляются все маршруты, установленные в RIB. Входным параметром является имя (`rib-name`), а на выходе возвращается значение `true` при успехе или `false` при отказе (в случае отказа агент I2RS может возвращать конкретную причину ошибки).
- `route-add` добавляет маршрут или набор маршрутов в RIB, принимая на входе имя RIB, префиксы маршрутов, `route-attributes`, `route-vendor-attributes`, `nexthop` и флаг предоставления деталей при отказах. Перед вызовом `route-add` rpc нужно вызвать `the nh-add` для создания и/или возврата идентификатора `nexthop`. Если `nexthop` уже существует и `nexthop-id` известен, это можно не делать. На выходе возвращается комбинация состояний операции для соответствующего узла в дереве данных:
  - `success-count` - число добавленных маршрутов;
  - `failed-count` - число маршрутов, при добавлении которых возникли отказы;
  - `failure-detail` - конкретные маршруты, которые не удалось добавить.
- `route-delete` удаляет маршрут или набор маршрутов из RIB, принимая на входе имя RIB, префиксы маршрутов и флаг предоставления деталей при отказах. На выходе возвращается комбинация состояний операции:
  - `success-count` - число удалённых маршрутов;
  - `failed-count` - число маршрутов, при удалении которых возникли отказы;
  - `failure-detail` - конкретные маршруты, которые не удалось удалить.
- `route-update` обновляет маршрут или набор маршрутов в RIB, принимая на входе имя RIB, префиксы маршрутов, `route-attributes`, `route-vendor-attributes` или `nexthop`. Сопоставление может выполняться по префиксам, `route-attributes`, `route-vendor-attributes` или `nexthop`. Обновления могут применяться к `nexthop`, `route-attributes` и `route-vendor-attributes`. На выходе возвращается комбинация состояний операции:
  - `success-count` - число обновлённых маршрутов;
  - `failed-count` - число маршрутов, при обновлении которых возникли отказы;
  - `failure-detail` - конкретные маршруты, которые не удалось обновить.
- `nh-add` добавляет `nexthop` в RIB, принимая на входе имя RIB и `nexthop`. Сетевому узлу требуется назначить идентификатор для `nexthop`. На выходе возвращается значение `true` (клиент I2RS возвращает идентификатор `identifier`) при успехе или `false` при отказе (в случае отказа агент I2RS может возвращать конкретную причину ошибки).
- `nh-delete` удаляет `nexthop` из RIB, принимая на входе имя RIB и `nexthop` или идентификатор `nexthop`. На выходе возвращается значение `true` при успехе или `false` при отказе (в случае отказа агент I2RS может возвращать конкретную причину ошибки).

Дерево вызовов удалённых процедур показано на рисунке 6.

```

rpcs:
+---x rib-add
| +---w input
| | +---w rib-name      string
| | +---w address-family  address-family-definition
| | +---w ip-rpf-check?  boolean
| +--ro output
|   +--ro result uint32
|   +--ro reason? string
+---x rib-delete
| +---w input
| | +---w rib-name string
| +--ro output
|   +--ro result uint32
|   +--ro reason? string
+---x route-add
| +---w input
| | +---w return-failure-detail?  boolean
| | +---w rib-name                string
| | +---w routes
| |   +---w route-list* [route-index]
| |   ...
| +--ro output
|   +--ro success-count      uint32
|   +--ro failed-count      uint32
|   +--ro failure-detail
|   +---ro failed-routes* [route-index]
|   +--ro route-index uint32
|   +--ro error-code? uint32
+---x route-delete
| +---w input
| | +---w return-failure-detail?  boolean
| | +---w rib-name                string
| | +---w routes
| |   +---w route-list* [route-index]
| |   ...
| +--ro output
|   +--ro success-count      uint32
|   +--ro failed-count      uint32
|   +--ro failure-detail
|   +---ro failed-routes* [route-index]

```



```

|         +---ro route-index uint32
|         +---ro error-code? uint32
+---x route-update
|   +---w input
|   |   +---w return-failure-detail?      boolean
|   |   +---w rib-name                    string
|   |   +---w (match-options)?
|   |   |   +---:(match-route-prefix)
|   |   |   |   ...
|   |   |   +---:(match-route-attributes)
|   |   |   |   ...
|   |   |   +---:(match-route-vendor-attributes) {...}?
|   |   |   |   ...
|   |   |   +---:(match-nexthop)
|   |   |   |   ...
|   |   |   ...
|   +---ro output
|   |   +---ro success-count uint32
|   |   +---ro failed-count uint32
|   |   +---ro failure-detail
|   |   |   +---ro failed-routes* [route-index]
|   |   |   |   +---ro route-index uint32
|   |   |   |   |   +---ro error-code? uint32
+---x nh-add
|   +---w input
|   |   +---w rib-name                    string
|   |   +---w nexthop-id?                 uint32
|   |   +---w sharing-flag?              boolean
|   |   +---w (nexthop-type)?
|   |   |   ...
|   +---ro output
|   |   +---ro result                    uint32
|   |   +---ro reason?                   string
|   |   +---ro nexthop-id?              uint32
+---x nh-delete
|   +---w input
|   |   +---w rib-name                    string
|   |   +---w nexthop-id?                 uint32
|   |   +---w sharing-flag?              boolean
|   |   +---w (nexthop-type)?
|   |   |   ...
|   +---ro output
|   |   +---ro result uint32
|   |   +---ro reason? string

```

Рисунок 6. Структура RPC.

## 2.6. Уведомления

Менеджер RIB сетевого устройства передаёт асинхронные уведомления внешнему объекту при запуске на сетевом устройстве того или иного события. Реализация этой модели данных RIB должна поддерживать отправку двух типов асинхронных уведомлений.

- Изменение маршрута:
  - Installed (установлен) указывает, что маршрут помещён в FIB;
  - Active (активен) указывает, есть ли у маршрута хотя бы один полностью распознанный nexthop и, следовательно, является ли маршрут кандидатом на включение в FIB;
  - Reason (причина), например, Not authorized (не разрешено)
- Статус распознавания nexthop (полностью распознан или не распознан)

В распознанном nexthop имеется адекватный уровень сведений для передачи исходящих пакетов в направлении получателя путём пересылки в интерфейс, напрямую соединённый с соседом. Для нераспознанного nexthop от менеджера RIB требуется определить окончательный распознанный nexthop. Например, это может быть адрес IP и менеджер RIB будет выяснять, как достичь этого адреса, например, проверяя его доступность через обычную пересылку IP, туннель MPLS или обоими способами. Если менеджер RIB не может распознать (преобразовать) nexthop, сохраняется нераспознанное состояние nexthop и он не является кандидатом на включение в FIB.

Реализация этой модели данных RIB **должна** поддерживать отправку уведомлений об изменении маршрута при каждой из указанных ниже смен состояния:

- из активного в неактивное.
- из неактивного в активно;
- из установленного в неустановленный;
- из неустановленного в установленный.

**Можно** использовать одно уведомление при смене неактивного и неустановленного состояния на активное и установленное или наоборот.

Структура дерева уведомлений показана на рисунке 7.

notifications:

```

+---n nexthop-resolution-status-change
| +--ro nexthop
| | +--ro nexthop-id          uint32
| | +--ro sharing-flag      boolean
| | +--ro (nexthop-type)?
| |   +---:(nexthop-base)
| |     | ...
| |     +---:(nexthop-chain) {nexthop-chain}?
| |     | ...
| |     +---:(nexthop-replicate) {nexthop-replicate}?
| |     | ...
| |     +---:(nexthop-protection) {nexthop-protection}?
| |     | ...
| |     +---:(nexthop-load-balance) {nexthop-load-balance}?
| |     | ...
| |     ...
| +--ro nexthop-state nexthop-state-definition
+---n route-change
+--ro rib-name          string
+--ro address-family   address-family-definition
+--ro route-index      uint64
+--ro match
| +--ro (route-type)?
|   +---:(ipv4)
|   | ...
|   +---:(ipv6)
|   | ...
|   +---:(mpls-route)
|   | ...
|   +---:(mac-route)
|   | ...
|   +---:(interface-route)
|   | ...
|   ...
+--ro route-installed-state route-installed-state-definition
+--ro route-state          route-state-definition
+--ro route-change-reason  route-change-reason-definition

```

Рисунок 7. Структура уведомления.

### 3. Модуль YANG

Этот модуль YANG ссылается на [RFC2784], [RFC7348], [RFC7637], [RFC8344].

```

<CODE BEGINS> file "ietf-i2rs-rib@2018-09-13.yang"

module ietf-i2rs-rib {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-i2rs-rib";
  prefix iir;

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991";
  }
  import ietf-interfaces {
    prefix if;
    reference "RFC 8344";
  }
  import ietf-yang-types {
    prefix yang;
    reference "RFC 6991";
  }

  organization
    "IETF I2RS (Interface to Routing System) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/i2rs/>
    WG List: <mailto:i2rs@ietf.org>

    Editor: Lixing Wang
            <mailto:wang\_little\_star@sina.com>

    Editor: Mach(Guoyi) Chen
            <mailto:mach.chen@huawei.com>

    Editor: Amit Dass
            <mailto:dass.amit@gmail.com>

    Editor: Hariharan Ananthkrishnan
            <mailto:hari@netflix.com>

    Editor: Sriganesh Kini
            <mailto:sriganeshkini@gmail.com>

    Editor: Nitin Bahadur
            <mailto:nitin\_bahadur@yahoo.com>";

```

```
description
"Этот модуль определяет модель данных YANG для базы RIB, которая
соответствует информационной модели I2RS RIB.

Авторские права (Copyright (c) 2018) принадлежат IETF Trust и
лицам, указанным как авторы. Все права защищены.

Распространение и применение модуля в исходной или двоичной
форме с изменениями или без таковых разрешено в соответствии с
лицензией Simplified BSD License, изложенной в параграфе 4.c
IETF Trust's Legal Provisions Relating to IETF Documents
(https://trustee.ietf.org/license-info).

Эта версия модуля YANG является частью RFC 8341, где правовые
аспекты приведены более полно.";
revision 2018-09-13 {
  description
  "Исходный выпуск";
  reference "RFC 8431";
}

//Свойства (функции)

feature nexthop-tunnel {
  description
  "Указывает поддержку узлом туннельных nexthop.";
}

feature nexthop-chain {
  description
  "Указывает поддержку узлом цепочек nexthop.";
}

feature nexthop-protection {
  description
  "Указывает поддержку узлом защитных nexthop.";
}

feature nexthop-replicate {
  description
  "Указывает поддержку узлом nexthop для репликации.";
}

feature nexthop-load-balance {
  description
  "Указывает поддержку узлом nexthop для распределения нагрузки";
}

feature ipv4-tunnel {
  description
  "Указывает поддержку узлом туннельной инкапсуляции IPv4.";
}

feature ipv6-tunnel {
  description
  "Указывает поддержку узлом туннельной инкапсуляции IPv6.";
}

feature mpls-tunnel {
  description
  "Указывает поддержку узлом туннельной инкапсуляции MPLS.";
}

feature vxlan-tunnel {
  description
  "Указывает поддержку узлом туннельной инкапсуляции VXLAN.";
  reference "RFC 7348";
}

feature gre-tunnel {
  description
  "Указывает поддержку узлом туннельной инкапсуляции GRE.";
  reference "RFC 2784";
}

feature nvgre-tunnel {
  description
  "Указывает поддержку узлом туннельной инкапсуляции NVGRE.";
  reference "RFC 7637";
}

feature route-vendor-attributes {
  description
  "Указывает поддержку узлом фирменных атрибутов маршрута.";
}
```

```
// Определения идентификаторов и типов
identity mpls-label-action {
    description
    "Базовый идентификатор для вывода операций с метками MPLS
    push для добавления новой метки в стек
    pop для выталкивания верхней метки из стека
    swap для замены верхней метки стека на новую метку";
}

identity label-push {
    base mpls-label-action;
    description
    "Операция push для добавления новой метки в стек MPLS.";
}

identity label-pop {
    base mpls-label-action;
    description
    "Операция pop для выталкивания верхней метки из стека MPLS.";
}

identity label-swap {
    base mpls-label-action;
    description
    "Операция swap для замены верхней метки в стеке MPLS.";
}

typedef mpls-label-action-definition {
    type identityref {
        base mpls-label-action;
    }
    description
    "Определение действий с метками MPLS.";
}

identity tunnel-decapsulation-action {
    description
    "Базовый идентификатор для действий туннельной декапсуляции
    ipv4-decapsulation (декапсуляция туннеля IPv4)
    ipv6-decapsulation (декапсуляция туннеля IPv6)";
}

identity ipv4-decapsulation {
    base tunnel-decapsulation-action;
    description
    "Декапсуляция туннеля IPv4.";
}

identity ipv6-decapsulation {
    base tunnel-decapsulation-action;
    description
    "Декапсуляция туннеля IPv6.";
}

typedef tunnel-decapsulation-action-definition {
    type identityref {
        base tunnel-decapsulation-action;
    }
    description
    "Определение туннельной декапсуляции.";
}

identity ttl-action {
    description
    "Базовый идентификатор для действий с TTL.";
}

identity no-action {
    base ttl-action;
    description
    "Ничего не делать с TTL.";
}

identity copy-to-inner {
    base ttl-action;
    description
    "Копировать TTL из внешнего заголовка во внутренний.";
}

identity decrease-and-copy-to-inner {
    base ttl-action;
    description
    "Уменьшить TTL на 1 и скопировать во внутренний заголовок.";
}

identity decrease-and-copy-to-next {
```

```
base ttl-action;
description
  "Уменьшить TTL на 1 и скопировать в следующий заголовок,
  например, при смене метки MPLS уменьшается TTL в
  in_label и копируется в out_label.";
}

typedef ttl-action-definition {
  type identityref {
    base ttl-action;
  }
  description
    "Определение действий с TTL.";
}

identity hop-limit-action {
  description
    "Базовый идентификатор для действий с hop limit.";
}

identity hop-limit-no-action {
  base hop-limit-action;
  description
    "Ничего не делать с hop limit.";
}

identity hop-limit-copy-to-inner {
  base hop-limit-action;
  description
    "Копировать hop limit из внешнего заголовка во внутренний.";
}

typedef hop-limit-action-definition {
  type identityref {
    base hop-limit-action;
  }
  description
    "Определение действий с IPv6 hop limit.";
}

identity special-nexthop {
  description
    "Базовый идентификатор для вывода специальных nexthops.";
}

identity discard {
  base special-nexthop;
  description
    "Указывает, что сетевому устройству следует отбросить пакет и
    инкрементировать счётчик отбрасывания.";
}

identity discard-with-error {
  base special-nexthop;
  description
    "Указывает, что сетевому устройству следует отбросить пакет,
    инкрементировать счётчик отбрасывания и передать подходящее
    сообщение об ошибке (например, ICMP).";
}

identity receive {
  base special-nexthop;
  description
    "Указывает, что трафик адресов сетевому устройству, например
    является пакетом протокола или OAM. Весь трафик для локального
    устройства следует дросселировать для предотвращения атак на
    службы маршрутизатора в плоскости управления. Можно задать
    ограничитель скорости для указания способа дросселирования
    трафика, адресованного в плоскость управления.";
}

identity cos-value {
  base special-nexthop;
  description
    "Специальный nexthop по значению CoS.";
}

typedef special-nexthop-definition {
  type identityref {
    base special-nexthop;
  }
  description
    "Определение специального nexthop.";
}

identity ip-route-match-type {
```

```
description
  "Базовый идентификатор для вывода сопоставлений маршрутов -
  по источнику, получателю или обоим.";
}

identity match-ip-src {
  base ip-route-match-type;
  description
    "Сопоставление по источнику.";
}

identity match-ip-dest {
  base ip-route-match-type;
  description
    "Сопоставление по получателю.";
}

identity match-ip-src-dest {
  base ip-route-match-type;
  description
    "Сопоставление по источнику и получателю.";
}

typedef ip-route-match-type-definition {
  type identityref {
    base ip-route-match-type;
  }
  description
    "Определение типа сопоставления для маршрутов IP.";
}

identity address-family {
  description
    "Базовый идентификатор для вывода семейств адресов RIB.";
}

identity ipv4-address-family {
  base address-family;
  description
    "Семейство адресов IPv4 RIB.";
}

identity ipv6-address-family {
  base address-family;
  description
    "Семейство адресов IPv6 RIB.";
}

identity mpls-address-family {
  base address-family;
  description
    "Семейство адресов MPLS RIB.";
}

identity ieee-mac-address-family {
  base address-family;
  description
    "Семейство адресов MAC RIB.";
}

typedef address-family-definition {
  type identityref {
    base address-family;
  }
  description
    "Определение семейств адресов RIB.";
}

identity route-type {
  description
    "Базовый идентификатор для вывода типов маршрутов.";
}

identity ipv4-route {
  base route-type;
  description
    "Маршрут IPv4.";
}

identity ipv6-route {
  base route-type;
  description
    "Маршрут IPv6.";
}

identity mpls-route {
```

```
base route-type;
description
  "Маршрут MPLS.";
}

identity ieee-mac {
  base route-type;
  description
    "Маршрут MAC.";
}

identity interface {
  base route-type;
  description
    "Маршрут через интерфейс.";
}

typedef route-type-definition {
  type identityref {
    base route-type;
  }
  description
    "Определение типа маршрута.";
}

identity tunnel-type {
  description
    "Базовый идентификатор для вывода типов туннелей.";
}

identity ipv4-tunnel {
  base tunnel-type;
  description
    "Туннель IPv4";
}

identity ipv6-tunnel {
  base tunnel-type;
  description
    "Туннель IPv6";
}

identity mpls-tunnel {
  base tunnel-type;
  description
    "Туннель MPLS";
}

identity gre-tunnel {
  base tunnel-type;
  description
    "Туннель GRE";
}

identity vxlan-tunnel {
  base tunnel-type;
  description
    "Туннель VXLAN";
}

identity nvgre-tunnel {
  base tunnel-type;
  description
    "Туннель NVGRE";
}

typedef tunnel-type-definition {
  type identityref {
    base tunnel-type;
  }
  description
    "Определение типа туннеля.";
}

identity route-state {
  description
    "Базовый идентификатор для вывода состояний маршрутов.";
}

identity active {
  base route-state;
  description
    "Активное состояние.";
}

identity inactive {
```

```
base route-state;
description
  "Неаутивное состояние.";
}

typedef route-state-definition {
  type identityref {
    base route-state;
  }
  description
    "Определение состояния маршрута.";
}

identity nexthop-state {
  description
    "Базовый идентификатор для вывода состояний nexthop.";
}

identity resolved {
  base nexthop-state;
  description
    "Распознанный nexthop.";
}

identity unresolved {
  base nexthop-state;
  description
    "Нераспознанный nexthop.";
}

typedef nexthop-state-definition {
  type identityref {
    base nexthop-state;
  }
  description
    "Определение состояния nexthop.";
}

identity route-installed-state {
  description
    "Базовый идентификатор для вывода статуса установки маршрута.";
}

identity uninstalled {
  base route-installed-state;
  description
    "Не установлен.";
}

identity installed {
  base route-installed-state;
  description
    "Установлен.";
}

typedef route-installed-state-definition {
  type identityref {
    base route-installed-state;
  }
  description
    "Определение состояния установки маршрута.";
}

// Идентификаторы причин изменения маршрута

identity route-change-reason {
  description
    "Базовый идентификатор для вывода причин изменения маршрута.";
}

identity lower-route-preference {
  base route-change-reason;
  description
    "Маршрут установлен в FIB, поскольку он имеет меньшее значение
    preference (больший приоритет), чем у прежнего маршрута.";
}

identity higher-route-preference {
  base route-change-reason;
  description
    "Маршрут удалён из FIB, поскольку он имеет большее значение
    preference (меньший приоритет), чем у прежнего маршрута.";
}

identity resolved-nexthop {
  base route-change-reason;
```



```

description
  "Маршрут активирован, поскольку имеет хотя бы 1
  распознанный nexthop.";
}

identity unresolved-nexthop {
  base route-change-reason;
  description
    "Маршрут деактивирован по отсутствию распознанных nexthop.";
}

typedef route-change-reason-definition {
  type identityref {
    base route-change-reason;
  }
  description
    "Определение причины смены маршрута.";
}

typedef nexthop-preference-definition {
  type uint8 {
    range "1..99";
  }
  description
    "Значение nexthop-preference применяется в схемах защиты. Это
    целое число от 1 до 99, меньшее значение предпочтительней.
    Для загрузки N nexthop в FIB выбираются N nexthop с
    наименьшими значениями. При наличии большего числа nexthop
    с одинаковыми предпочтениями реализации клиента I2RS следует
    самостоятельно выбрать N nexthop и загрузить их.";
}

typedef nexthop-lb-weight-definition {
  type uint8 {
    range "1..99";
  }
  description
    "Значение nexthop-lb-weight служит для распределения нагрузки.
    Каждому элементу списка СЛЕДУЕТ назначать вес от 1 до 99. Вес
    определяет долю трафика через nexthop при пересылке как
    отношение веса этого nexthop к сумме весов других nexthop
    этого маршрута, применяемых для пересылки. Для равного
    распределения МОЖНО задать значение 0 для всех nexthop. Это
    значение зарезервировано для равномерного распределения и при
    его использовании ДОЛЖНО задаваться для всех nexthop.
    Это значение выбрано по историческим причинам и применяется
    обычно для ESMR.";
}

typedef nexthop-ref {
  type leafref {
    path "/iir:routing-instance" +
        "/iir:rib-list" +
        "/iir:route-list" +
        "/iir:nexthop" +
        "/iir:nexthop-id";
  }
  description
    "Ссылка для косвенного nexthop.";
}

//Группировки

grouping route-prefix {
  description
    "Базовые атрибуты для всех типов префиксов маршрута.";
  leaf route-index {
    type uint64;
    mandatory true;
    description
      "Индекс маршрута.";
  }
  container match {
    description
      "Условие задаёт тип маршрута (IPv4, MPLS и т. п.) и
      набор полей для сопоставления.";
    choice route-type {
      description
        "Типы маршрутов: IPv4, IPv6, MPLS, MAC и т. п.";
      case ipv4 {
        description
          "Маршрут IPv4.";
        container ipv4 {
          description
            "Сопоставление для маршрута IPv4.";
          choice ip-route-match-type {

```



```

    }
  }
}
case mpls-route {
  description
    "Маршрут MPLS.";
  leaf mpls-label {
    type uint32;
    mandatory true;
    description
      "Метка для сопоставления.";
  }
}
case mac-route {
  description
    "Маршрут MAC.";
  leaf mac-address {
    type yang:mac-address;
    mandatory true;
    description
      "MAC-адрес для сопоставления.";
  }
}
case interface-route {
  description
    "Маршрут через интерфейс.";
  leaf interface-identifier {
    type if:interface-ref;
    mandatory true;
    description
      "Интерфейс для сопоставления.";
  }
}
}
}
}
}

grouping route {
  description
    "Базовые атрибуты для всех типов маршрутов.";
  uses route-prefix;
  container nexthop {
    description
      "Значение nexthop для маршрута.";
    uses nexthop;
  }
  // В информационной модели это называется статистикой маршрута
  container route-status {
    description
      "Сведения о статусе маршрута.";
    leaf route-state {
      type route-state-definition;
      config false;
      description
        "Статус маршрута - активен или не активен.";
    }
    leaf route-installed-state {
      type route-installed-state-definition;
      config false;
      description
        "Статус установки маршрута - установлен или не установлен";
    }
    leaf route-reason {
      type route-change-reason-definition;
      config false;
      description
        "Причина изменения состояния маршрута.";
    }
  }
  container route-attributes {
    description
      "Атрибуты маршрута.";
    uses route-attributes;
  }
  container route-vendor-attributes {
    description
      "Фирменные (vendor) атрибуты маршрута.";
    uses route-vendor-attributes;
  }
}

grouping nexthop-list {
  description
    "Базовый список nexthop.";
  list nexthop-list {
    key "nexthop-member-id";

```

```
description
  "Список nexthop.";
leaf nexthop-member-id {
  type uint32;
  mandatory true;
  description
    "Идентификатор nexthop для элемента списка nexthop.";
}
}
}

grouping nexthop-list-p {
  description
    "Список nexthop с параметром preference.";
  list nexthop-list {
    key "nexthop-member-id";
    description
      "Список nexthop.";
    leaf nexthop-member-id {
      type uint32;
      mandatory true;
      description
        "Идентификатор nexthop для элемента списка nexthop.";
    }
    leaf nexthop-preference {
      type nexthop-preference-definition;
      mandatory true;
      description
        "Значение nexthop-preference применяется в схемах защиты.
        Это целое число от 1 и 99 и меньшее значение указывает
        более высокий приоритет. Для загрузки группы «основной-
        резервный-третий» в FIB выбираются распознанные nexthop с
        большим предпочтением (меньшими значениями).";
    }
  }
}

grouping nexthop-list-w {
  description
    "Список nexthop с весовыми параметрами.";
  list nexthop-list {
    key "nexthop-member-id";
    description
      "Список nexthop.";
    leaf nexthop-member-id {
      type uint32;
      mandatory true;
      description
        "Идентификатор nexthop для элемента списка nexthop.";
    }
    leaf nexthop-lb-weight {
      type nexthop-lb-weight-definition;
      mandatory true;
      description
        "Вес nexthop среди выбранных для распределения нагрузки.";
    }
  }
}

grouping nexthop {
  description
    "Структура nexthop.";
  leaf nexthop-id {
    type uint32;
    description
      "Идентификатор, указывающий nexthop.";
  }
  leaf sharing-flag {
    type boolean;
    description
      "Указывает, является ли nexthop обобщаемым:
      true - обобщаемый (может применяться в других маршрутах)
      false - необобщаемый (непригоден для других маршрутов).";
  }
  choice nexthop-type {
    description
      "Варианты типов nexthop.";
    case nexthop-base {
      container nexthop-base {
        description
          "Базовый nexthop.";
        uses nexthop-base;
      }
    }
    case nexthop-chain {
      if-feature "nexthop-chain";
    }
  }
}
```

```

    container nexthop-chain {
      description
        "Цепочка nexthop.";
      uses nexthop-list;
    }
  }
  case nexthop-replicate {
    if-feature "nexthop-replicate";
    container nexthop-replicate {
      description
        "Реплицирующий nexthop.";
      uses nexthop-list;
    }
  }
  case nexthop-protection {
    if-feature "nexthop-protection";
    container nexthop-protection {
      description
        "Защитный nexthop.";
      uses nexthop-list-p;
    }
  }
  case nexthop-load-balance {
    if-feature "nexthop-load-balance";
    container nexthop-lb {
      description
        "Балансирующий нагрузку nexthop.";
      uses nexthop-list-w;
    }
  }
}

grouping nexthop-base {
  description
    "Базовый nexthop.";
  choice nexthop-base-type {
    description
      "Варианты типа базового nexthop.";
    case special-nexthop {
      leaf special {
        type special-nexthop-definition;
        description
          "Специальный nexthop.";
      }
    }
    case egress-interface-nexthop {
      leaf outgoing-interface {
        type if:interface-ref;
        mandatory true;
        description
          "Выходной интерфейс как nexthop.";
      }
    }
    case ipv4-address-nexthop {
      leaf ipv4-address {
        type inet:ipv4-address;
        mandatory true;
        description
          "Адрес IPv4 как nexthop.";
      }
    }
    case ipv6-address-nexthop {
      leaf ipv6-address {
        type inet:ipv6-address;
        mandatory true;
        description
          "Адрес IPv6 как nexthop.";
      }
    }
    case egress-interface-ipv4-nexthop {
      container egress-interface-ipv4-address {
        leaf outgoing-interface {
          type if:interface-ref;
          mandatory true;
          description
            "Имя выходного интерфейса.";
        }
        leaf ipv4-address {
          type inet:ipv4-address;
          mandatory true;
          description
            "nexthop, указывающий интерфейс с адресом IPv4.";
        }
      }
      description
        "nexthop является egress-interface и адресом IP. Это

```

```
    может применяться, например, при адресе IP link-local.";
}
}
case egress-interface-ipv6-nexthop {
  container egress-interface-ipv6-address {
    leaf outgoing-interface {
      type if:interface-ref;
      mandatory true;
      description
        "Имя выходного интерфейса.";
    }
    leaf ipv6-address {
      type inet:ipv6-address;
      mandatory true;
      description
        "nexthop, указывающий интерфейс с адресом IPv6.";
    }
    description
      "nexthop является egress-interface и адресом IP. Это
      может применяться, например, при адресе IP link-local.";
  }
}
case egress-interface-mac-nexthop {
  container egress-interface-mac-address {
    leaf outgoing-interface {
      type if:interface-ref;
      mandatory true;
      description
        "Имя выходного интерфейса.";
    }
    leaf ieee-mac-address {
      type yang:mac-address;
      mandatory true;
      description
        "nexthop указывает интерфейс с конкретным адресом MAC";
    }
    description
      "Выходной интерфейс должен быть интерфейсом Ethernet.
      Распознавание адреса не требуется для такого nexthop.";
  }
}
case tunnel-encapsulation-nexthop {
  if-feature "nexthop-tunnel";
  container tunnel-encapsulation {
    uses tunnel-encapsulation;
    description
      "Инкапсуляция, представляющая туннель IP, MPLS или иной
      туннель, определённый в информационной модели.
      Можно связать egress-interface с инкапсуляцией туннеля
      для указания передающего интерфейса. Это полезно в
      случаях, когда сетевое устройство имеет интерфейсы
      Ethernet и нужно распознавание адреса для пакетов IP.";
  }
}
case tunnel-decapsulation-nexthop {
  if-feature "nexthop-tunnel";
  container tunnel-decapsulation {
    uses tunnel-decapsulation;
    description
      "Задаёт декапсуляцию туннельного заголовка.";
  }
}
case logical-tunnel-nexthop {
  if-feature "nexthop-tunnel";
  container logical-tunnel {
    uses logical-tunnel;
    description
      "MPLS LSP или туннель GRE (или иной, заданный в этом
      документе), указанный уникальным идентификатором
      (например, именем).";
  }
}
case rib-name-nexthop {
  leaf rib-name {
    type string;
    description
      "Значение nexthop, указывающее на RIB, задаёт продолжение
      поиска в этой RIB (цепочка поиска).";
  }
}
case nexthop-identifier {
  leaf nexthop-ref {
    type nexthop-ref;
    mandatory true;
    description
      "Идентификатор, указывающий nexthop.";
  }
}
```

```
    }
  }
}

grouping route-vendor-attributes {
  description
    "Фирменные (vendor) атрибуты маршрута.";
}

grouping logical-tunnel {
  description
    "Логический туннель, указанный типом и именем.";
  leaf tunnel-type {
    type tunnel-type-definition;
    mandatory true;
    description
      "Тип туннеля.";
  }
  leaf tunnel-name {
    type string;
    mandatory true;
    description
      "Имя, указывающее логический туннель.";
  }
}

grouping ipv4-header {
  description
    "Сведения из заголовка инкапсуляции IPv4.";
  leaf src-ipv4-address {
    type inet:ipv4-address;
    mandatory true;
    description
      "IP-адрес отправителя в заголовке.";
  }
  leaf dest-ipv4-address {
    type inet:ipv4-address;
    mandatory true;
    description
      "IP-адрес получателя в заголовке.";
  }
  leaf protocol {
    type uint8;
    mandatory true;
    description
      "Идентификатор протокола в заголовке.";
  }
  leaf ttl {
    type uint8;
    description
      "Значение TTL в заголовке.";
  }
  leaf dscp {
    type uint8;
    description
      "Значение DSCP в заголовке.";
  }
}

grouping ipv6-header {
  description
    "Сведения из заголовка инкапсуляции IPv6.";
  leaf src-ipv6-address {
    type inet:ipv6-address;
    mandatory true;
    description
      "IP-адрес отправителя в заголовке.";
  }
  leaf dest-ipv6-address {
    type inet:ipv6-address;
    mandatory true;
    description
      "IP-адрес получателя в заголовке.";
  }
  leaf next-header {
    type uint8;
    mandatory true;
    description
      "Поле next header в заголовке IPv6.";
  }
  leaf traffic-class {
    type uint8;
    description
      "Значение класса трафика в заголовке.";
  }
}
```

```
leaf flow-label {
  type inet:ipv6-flow-label;
  description
    "Метка потока в заголовке.";
}
leaf hop-limit {
  type uint8 {
    range "1..255";
  }
  description
    "Значение hop limit в заголовке.";
}
}

grouping nvgre-header {
  description
    "Сведения из заголовка инкапсуляции NVGRE.";
  choice nvgre-type {
    description
      "NVGRE может применять заголовок инкапсуляции IPv4 или IPv6";
    case ipv4 {
      uses ipv4-header;
    }
    case ipv6 {
      uses ipv6-header;
    }
  }
  leaf virtual-subnet-id {
    type uint32;
    mandatory true;
    description
      "Идентификатор подсети в заголовке NVGRE.";
  }
  leaf flow-id {
    type uint8;
    description
      "Идентификатор потока в заголовке NVGRE.";
  }
}

grouping vxlan-header {
  description
    "Сведения из заголовка инкапсуляции VXLAN.";
  choice vxlan-type {
    description
      "VXLAN может применять заголовок инкапсуляции IPv4 или IPv6";
    case ipv4 {
      uses ipv4-header;
    }
    case ipv6 {
      uses ipv6-header;
    }
  }
  leaf vxlan-identifier {
    type uint32;
    mandatory true;
    description
      "Идентификатор VXLAN в заголовке VXLAN.";
  }
}

grouping gre-header {
  description
    "Сведения из заголовка инкапсуляции GRE.";
  choice dest-address-type {
    description
      "Варианты GRE - IPv4 или IPv6";
    case ipv4 {
      leaf ipv4-dest {
        type inet:ipv4-address;
        mandatory true;
        description
          "IP-адрес получателя в заголовке GRE.";
      }
    }
    case ipv6 {
      leaf ipv6-dest {
        type inet:ipv6-address;
        mandatory true;
        description
          "IP-адрес получателя в заголовке GRE.";
      }
    }
  }
  leaf protocol-type {
    type uint16;
  }
}
```



```

mandatory true;
description
  "Тип протокола в заголовке GRE.";
}
leaf key {
  type uint64;
  description
    "Ключ GRE в заголовке GRE.";
}
}

grouping mpls-header {
  description
    "Сведения из заголовка инкапсуляции MPLS.";
  list label-operations {
    key "label-oper-id";
    description
      "Label operations.";
    leaf label-oper-id {
      type uint32;
      description
        "Необязательный идентификатор операции с меткой.";
    }
    choice label-actions {
      description
        "Операции с метками.";
      case label-push {
        container label-push {
          description
            "Операция вталкивания метки (push).";
          leaf label {
            type uint32;
            mandatory true;
            description
              "Метка для вталкивания.";
          }
          leaf s-bit {
            type boolean;
            description
              "Бит вершины стека (s) для вталкиваемой метки.";
          }
          leaf tc-value {
            type uint8;
            description
              "Класс трафика для вталкиваемой метки.";
          }
          leaf ttl-value {
            type uint8;
            description
              "Значение TTL для вталкиваемой метки.";
          }
        }
      }
    }
  }
  case label-swap {
    container label-swap {
      description
        "Операция смены метки (swap).";
      leaf in-label {
        type uint32;
        mandatory true;
        description
          "Метка для замены.";
      }
      leaf out-label {
        type uint32;
        mandatory true;
        description
          "Устанавливаемая взамен метка MPLS.";
      }
      leaf ttl-action {
        type ttl-action-definition;
        description
          "Действие с TTL для метки
          - ничего не делать
          - копировать по внутренней метку
          - уменьшить в in-label на 1 и поместить vout-label";
      }
    }
  }
}
}

grouping tunnel-encapsulation {
  description
    "Сведения туннельной инкапсуляции.";
}

```

```
choice tunnel-type {
  description
  "Варианты туннеля для nexthop.";
  case ipv4 {
    if-feature "ipv4-tunnel";
    container ipv4-header {
      uses ipv4-header;
      description
        "Заголовок IPv4.";
    }
  }
  case ipv6 {
    if-feature "ipv6-tunnel";
    container ipv6-header {
      uses ipv6-header;
      description
        "Заголовок IPv6.";
    }
  }
  case mpls {
    if-feature "mpls-tunnel";
    container mpls-header {
      uses mpls-header;
      description
        "Заголовок MPLS.";
    }
  }
  case gre {
    if-feature "gre-tunnel";
    container gre-header {
      uses gre-header;
      description
        "Заголовок GRE.";
    }
  }
  case nvgre {
    if-feature "nvgre-tunnel";
    container nvgre-header {
      uses nvgre-header;
      description
        "Заголовок NVGRE.";
    }
  }
  case vxlan {
    if-feature "vxlan-tunnel";
    container vxlan-header {
      uses vxlan-header;
      description
        "Заголовок VXLAN.";
    }
  }
}

grouping tunnel-decapsulation {
  description
  "Сведения туннельной декапсуляции.";
  choice tunnel-type {
    description
    "Варианты туннелей nexthop.";
    case ipv4 {
      if-feature "ipv4-tunnel";
      container ipv4-decapsulation {
        description
        "Декапсуляция IPv4.";
        leaf ipv4-decapsulation {
          type tunnel-decapsulation-action-definition;
          mandatory true;
          description
            "Операции декапсуляции IPv4.";
        }
        leaf ttl-action {
          type ttl-action-definition;
          description
            "Действие для TTL - нет или копирование во внутренних
            заголовков.";
        }
      }
    }
    case ipv6 {
      if-feature "ipv6-tunnel";
      container ipv6-decapsulation {
        description
        "Декапсуляция IPv6.";
        leaf ipv6-decapsulation {
          type tunnel-decapsulation-action-definition;

```

```
        mandatory true;
        description
            "Операции декапсуляции IPv6.";
    }
    leaf hop-limit-action {
        type hop-limit-action-definition;
        description
            "Действия для hop limit - нет или копирование
            во внутренний заголовок.";
    }
}
}
case mpls {
    if-feature "mpls-tunnel";
    container label-pop {
        description
            "Декапсуляция MPLS.";
        leaf label-pop {
            type mpls-label-action-definition;
            mandatory true;
            description
                "Выталкивания (pop) метки и стека.";
        }
        leaf ttl-action {
            type ttl-action-definition;
            description
                "Действие с TTL для метки.";
        }
    }
}
}
}
}

grouping route-attributes {
    description
        "Атрибуты маршрута.";
    leaf route-preference {
        type uint32;
        mandatory true;
        description
            "ROUTE_PREFERENCE - численное значение для сравнения
            маршрутов от разных протоколов (статические маршруты также
            считаются протоколом). Это называют также административной
            дистанцией. Меньшее значение указывает больший приоритет.";
    }
    leaf local-only {
        type boolean;
        mandatory true;
        description
            "Указывает, является ли атрибут лишь локальным.";
    }
}
container address-family-route-attributes {
    description
        "Атрибуты маршрута, связанные с семейством адресов.";
    choice route-type {
        description
            "Связанные с семейством адресов атрибуты маршрута. В
            будущих документах следует задавать такие атрибуты
            дополнением вариантов (case) в этот выбор (choice).";
        case ip-route-attributes {
        }
        case mpls-route-attributes {
        }
        case ethernet-route-attributes {
        }
    }
}
}

container routing-instance {
    description
        "Экземпляр маршрутизации в контексте информационной модели
        RIB служит набор RIB, интерфейсов и параметров маршрутизации";
    leaf name {
        type string;
        description
            "Имя экземпляра маршрутизации, которое ДОЛЖНО быть уникальным
            в рамках данного сетевого устройства.";
    }
    list interface-list {
        key "name";
        description
            "Список интерфейсов, связанных с экземпляром маршрутизации.
            Этот список помогает задать границы пересылки пакетов.
            Пакеты, приходящие с интерфейсов из списка, напрямую связаны
            с данным экземпляром маршрутизации. Список содержит
```

```
    идентификаторы, однозначно указывающие интерфейсы."
  leaf name {
    type if:interface-ref;
    description
      "Имя интерфейса сетевого уровня."
  }
}
leaf router-id {
  type yang:dotted-quad;
  description
    "Router ID - 32-битовое значение с разделением точками."
}
leaf lookup-limit {
  type uint8;
  description
    "Предельное число выполняемых уровней поиска."
}
list rib-list {
  key "name";
  description
    "Список RIB, связанных с экземпляром маршрутизации."
  leaf name {
    type string;
    mandatory true;
    description
      "Имя каждой RIB."
  }
  leaf address-family {
    type address-family-definition;
    mandatory true;
    description
      "Семейство адресов для RIB."
  }
  leaf ip-rpf-check {
    type boolean;
    description
      "С каждой RIB можно связать атрибут ENABLE_IP_RPF_CHECK,
      управляющий проверкой обратного пути (RPF) для всех
      маршрутов IP в RIB, служащей для предотвращения подмены
      адресов (spoofing) и ограничения вредного трафика"
  }
  list route-list {
    key "route-index";
    description
      "Список маршрутов RIB."
    uses route;
  }
  // Это список, поддерживающий nexthop, добавленные в RIB.
  uses nexthop-list;
}
}

// Операции RPC
rpc rib-add {
  description
    "Добавление RIB в экземпляр маршрутизации";
  input {
    leaf name {
      type string;
      mandatory true;
      description
        "Имя добавляемой RIB."
    }
    leaf address-family {
      type address-family-definition;
      mandatory true;
      description
        "Семейство адресов в RIB."
    }
    leaf ip-rpf-check {
      type boolean;
      description
        "С каждой RIB можно связать атрибут ENABLE_IP_RPF_CHECK,
        управляющий проверкой обратного пути (RPF) для всех
        маршрутов IP в RIB, служащей для предотвращения подмены
        адресов (spoofing) и ограничения вредного трафика"
    }
  }
  output {
    leaf result {
      type boolean;
      mandatory true;
      description
        "Результат операции rib-add - true - успех, false - отказ";
    }
    leaf reason {
```

```

    type string;
    description
      "Конкретная причина отказа.";
  }
}

rpc rib-delete {
  description
    "Удаление RIB из экземпляра маршрутизации с удалением всех
    маршрутов, установленных из этой RIB.";
  input {
    leaf name {
      type string;
      mandatory true;
      description
        "Имя удаляемой RIB.";
    }
  }
  output {
    leaf result {
      type boolean;
      mandatory true;
      description
        "Результат операции rib-delete - true - успех,
        false - отказ";
    }
    leaf reason {
      type string;
      description
        "Конкретная причина отказа.";
    }
  }
}

grouping route-operation-state {
  description
    "Статус операции для маршрутов.";
  leaf success-count {
    type uint32;
    mandatory true;
    description
      "Число добавленных, удалённых или изменённых маршрутов.";
  }
  leaf failed-count {
    type uint32;
    mandatory true;
    description
      "Число маршрутов с отказом добавления, удаления, изменения";
  }
  container failure-detail {
    description
      "Причины отказа операции с маршрутами. Это массив, содержащий
      индексы маршрутов и коды отказы.";
    list failed-routes {
      key "route-index";
      description
        "Список маршрутов с отказами.";
      leaf route-index {
        type uint32;
        description
          "Индекс маршрута с отказом.";
      }
      leaf error-code {
        type uint32;
        description
          "Код, указывающий причину отказа
          0 - резерв
          1 - попытка добавить имеющийся маршрут
          2 - попытка удалить отсутствующий маршрут
          3 - некорректные атрибуты маршрута.";
      }
    }
  }
}

rpc route-add {
  description
    "Добавление маршрута или набора маршрутов в RIB";
  input {
    leaf return-failure-detail {
      type boolean;
      default "false";
      description
        "Управляет возвратом сведений о причине отказа - true задаёт
        возврат причины, false отменяет возврат причины отказа.";
    }
  }
}

```

```
    }
    leaf rib-name {
      type string;
      mandatory true;
      description
        "Имя RIB.";
    }
  container routes {
    description
      "Маршруты, добавляемые в RIB.";
    list route-list {
      key "route-index";
      description
        "Список добавляемых маршрутов.";
      uses route-prefix;
      container route-attributes {
        uses route-attributes;
        description
          "Атрибуты маршрутов.";
      }
      container route-vendor-attributes {
        if-feature "route-vendor-attributes";
        uses route-vendor-attributes;
        description
          "Фирменные (vendor) атрибуты маршрутов.";
      }
      container nexthop {
        uses nexthop;
        description
          "Значение nexthop в добавляемом маршруте.";
      }
    }
  }
}
output {
  uses route-operation-state;
}

rpc route-delete {
  description
    "Удаление маршрута или набора маршрутов из RIB";
  input {
    leaf return-failure-detail {
      type boolean;
      default "false";
      description
        "Управляет возвратом сведений о причине отказа - true задаёт
        возврат причины, false отменяет возврат причины отказа.";
    }
    leaf rib-name {
      type string;
      mandatory true;
      description
        "Имя RIB.";
    }
  }
  container routes {
    description
      "Маршруты, удаляемые из RIB.";
    list route-list {
      key "route-index";
      description
        "Список удаляемых маршрутов.";
      uses route-prefix;
    }
  }
}
output {
  uses route-operation-state;
}

grouping route-update-options {
  description
    "Варианты обновления:
    1. обновить nexthop
    2. обновить атрибуты маршрута
    3. обновить фирменные атрибуты (route-vendor-attributes).";
  choice update-options {
    description
      "Варианты обновления:
      1. обновить nexthop
      2. обновить атрибуты маршрута
      3. обновить фирменные атрибуты (route-vendor-attributes).";
    case update-nexthop {
      container updated-nexthop {
```

```

    uses nexthop;
    description
      "Значение nexthop для обновления.";
  }
}
case update-route-attributes {
  container updated-route-attr {
    uses route-attributes;
    description
      "Атрибуты маршрута для обновления.";
  }
}
case update-route-vendor-attributes {
  container updated-route-vendor-attr {
    uses route-vendor-attributes;
    description
      "Фирменные (vendor) атрибуты для обновления.";
  }
}
}
}
}

rpc route-update {
  description
    "Обновление маршрута или набора маршрутов в RIB.
    Входные данные:
    1. Условия сопоставления, которые могут включать:
      а. префикс маршрута;
      б. атрибуты маршрута;
      в. nexthop.
    2. Параметры, используемые для обновления:
      а. новый nexthop;
      б. новые атрибуты маршрута.
    Действия:
    1. обновление nexthop;
    2. обновление атрибутов маршрута.
    Результат:
    success-count - число обновлённых маршрутов;
    failed-count - число маршрутов с отказами обновления;
    failure-detail - сведения о причинах отказа.";
  input {
    leaf return-failure-detail {
      type boolean;
      default "false";
      description
        "Управляет возвратом сведений о причине отказа - true задаёт
        возврат причины, false отменяет возврат причины отказа.";
    }
    leaf rib-name {
      type string;
      mandatory true;
      description
        "Имя RIB.";
    }
  }
  choice match-options {
    description
      "Опции сопоставления.";
    case match-route-prefix {
      description
        "Обновлять маршруты с совпадающими префиксами.";
      container input-routes {
        description
          "Маршруты для обновления.";
        list route-list {
          key "route-index";
          description
            "Список обновляемых маршрутов.";
          uses route-prefix;
          uses route-update-options;
        }
      }
    }
  }
  case match-route-attributes {
    description
      "Обновлять маршруты с совпадающими атрибутами.";
    container input-route-attributes {
      description
        "Атрибуты для сопоставления.";
      uses route-attributes;
    }
    container update-parameters {
      description
        "Варианты обновления:
        1. обновить nexthop
        2. обновить атрибуты маршрута
        3. обновить фирменные атрибуты.";
    }
  }
}
}
}

```

```

    uses route-update-options;
  }
}
case match-route-vendor-attributes {
  if-feature "route-vendor-attributes";
  description
    "Обновить маршруты с совпадающими фирменными атрибутами";
  container input-route-vendor-attributes {
    description
      "Фирменные (vendor) атрибуты для сопоставления.";
    uses route-vendor-attributes;
  }
  container update-parameters-vendor {
    description
      "Варианты обновления:
      1. обновить nexthop
      2. обновить атрибуты маршрута
      3. обновить фирменные атрибуты.";
    uses route-update-options;
  }
}
case match-nexthop {
  description
    "Обновить маршруты с совпадающим nexthop.";
  container input-nexthop {
    description
      "Значение nexthop для сопоставления.";
    uses nexthop;
  }
  container update-parameters-nexthop {
    description
      "Варианты обновления:
      1. обновить nexthop
      2. обновить атрибуты маршрута
      3. обновить фирменные атрибуты.";
    uses route-update-options;
  }
}
}
}
output {
  uses route-operation-state;
}
}
rpc nh-add {
  description
    "Добавить nexthop в RIB.
    Входные параметры:
    1. rib-name
    2. nexthop
    Действие:
    Добавить nexthop в RIB.
    Результаты:
    1. результата операции (true - успех, false - отказ)
    2. идентификатор nexthop";
  input {
    leaf rib-name {
      type string;
      mandatory true;
      description
        "Имя RIB.";
    }
    uses nexthop;
  }
  output {
    leaf result {
      type boolean;
      mandatory true;
      description
        "Результат операции nh-add (true - успех, false - отказ)";
    }
    leaf reason {
      type string;
      description
        "Конкретная причина отказа.";
    }
    leaf nexthop-id {
      type uint32;
      description
        "Идентификатор, выделенный для nexthop.";
    }
  }
}
}
rpc nh-delete {
  description

```



```

"Удалить nexthop из RIB";
input {
  leaf rib-name {
    type string;
    mandatory true;
    description
      "Имя RIB.";
  }
  uses nexthop;
}
output {
  leaf result {
    type boolean;
    mandatory true;
    description
      "Результат nh-delete (true - успех, false - отказ)";
  }
  leaf reason {
    type string;
    description
      "Конкретная причина отказа.";
  }
}
}

// Уведомления
notification nexthop-resolution-status-change {
  description
    "Статус распознавания nexthop (распознан или не распознан).";
  container nexthop {
    description
      "The nexthop.";
    uses nexthop;
  }
  leaf nexthop-state {
    type nexthop-state-definition;
    mandatory true;
    description
      "Статус распознавания nexthop (распознан или не распознан).";
  }
}

notification route-change {
  description
    "Изменение маршрута.";
  leaf rib-name {
    type string;
    mandatory true;
    description
      "Имя RIB.";
  }
  leaf address-family {
    type address-family-definition;
    mandatory true;
    description
      "Семейство адресов в RIB.";
  }
  uses route-prefix;
  leaf route-installed-state {
    type route-installed-state-definition;
    mandatory true;
    description
      "Указывает, включён ли маршрут в FIB.";
  }
  leaf route-state {
    type route-state-definition;
    mandatory true;
    description
      "Указывает, активен ли маршрут.";
  }
  list route-change-reasons {
    key "route-change-reason";
    description
      "Причина изменения маршрута. Это может быть, например,
      результатом распознавания nexthop, активирующего маршрут А,
      который предпочтительней активного ранее маршрута В.";
    leaf route-change-reason {
      type route-change-reason-definition;
      mandatory true;
      description
        "Причина изменения маршрута.";
    }
  }
}
}
}
<CODE ENDS>

```

## 4. Взаимодействие с IANA

Этот документ регистрирует URI в субреестре ns реестра IETF XML Registry [RFC3688]

```
URI: urn:ietf:params:xml:ns:yang:ietf-i2rs-rib
Registrant Contact: The IESG.
XML: N/A, запрошенный URI является пространством имён XML.
```

Этот документ регистрирует модуль YANG в реестре YANG Module Names [RFC7950]

```
name: ietf-i2rs-rib
namespace: urn:ietf:params:xml:ns:yang:ietf-i2rs-rib
prefix: iir
reference: RFC 8431
```

## 5. Вопросы безопасности

Заданный этим документом модуль YANG определяет схему для данных, предназначенную для доступа через сеть с использованием протоколов управления, таких как NETCONF [RFC6241] или RESTCONF [RFC8040]. Нижним уровнем NETCONF служит защищённый транспорт с обязательной поддержкой SSH (Secure Shell) [RFC6242]. Нижним уровнем RESTCONF служит протокол HTTPS с обязательной поддержкой защиты на транспортном уровне (TLS) [RFC8446].

Модель доступа NETCONF [RFC8341] обеспечивает возможность разрешить доступ лишь указанных пользователей NETCONF или RESTCONF к заранее заданному подмножеству операций NETCONF или RESTCONF и содержимого.

Модуль YANG задаёт информацию, которая может быть настраиваемой в некоторых экземплярах, например, RIB, маршрут, nexthop могут создаваться и удаляться клиентскими приложениями. Модель YANG также задаёт RPC, которые клиенты могут применять для удаления и добавления RIB, маршрутов и nexthop. Вредоносный клиент может попытаться добавить, удалить или обновить RIB, маршруты или nexthop, создавая или удаляя соответствующие элементы в списках RIB, маршрутов, nexthop. Удаление RIB или маршрута может привести к отказам или снижению производительности служб, а изменение маршрута может привести к неоптимальной маршрутизации и снижению качества обслуживания и даже его прекращению. По этим причинам важно активно применять модель контроля доступа NETCONF для предотвращения изменения настроек неуполномоченными клиентами.

В этом модуле данных YANG определено множество узлов данных, которые разрешают запись, создание и удаление (т. е. config true, как принято по умолчанию). Эти узлы могут быть конфиденциальными или уязвимыми в некоторых сетевых средах. Запись в такие узлы (например, edit-config) без должной защиты может негативно влиять на работу сети. Ниже перечислены ветви и узлы, которые могут быть конфиденциальными или уязвимыми.

### RIB

Злонамеренный клиент может попытаться удалить RIB из экземпляра маршрутизации для нарушения работы служб, обеспечиваемых RIB, или добавить RIB в экземпляр маршрутизации (для внедрения несанкционированного трафика в nexthop).

### route

Вредоносный клиент может попытаться добавить или удалить маршрут RIB, например, для нарушения работы служб, обеспечиваемых RIB.

### nexthop

Вредоносный клиент может попытаться добавить или удалить nexthop, что может приводить к неоптимальным путям, ухудшать уровень обслуживания и даже прерывать его.

## 6. Литература

### 6.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", [RFC 8344](#), DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.
- [RFC8430] Bahadur, N., Ed., Kini, S., Ed., and J. Medved, "RIB Information Model", [RFC 8430](#), DOI 10.17487/RFC8430, September 2018, <<http://www.rfc-editor.org/info/rfc8430>>.

[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>>.

## 6.2. Дополнительная литература

- [I2RS-REQS] Hares, S. and M. Chen, "Summary of I2RS Use Case Requirements", Work in Progress, draft-ietf-i2rs-usecase-reqs-summary-03, November 2016.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", [RFC 2784](#), DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", [RFC 7348](#), DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", [RFC 7637](#), DOI 10.17487/RFC7637, September 2015, <<https://www.rfc-editor.org/info/rfc7637>>.
- [RFC7921] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", [RFC 7921](#), DOI 10.17487/RFC7921, June 2016, <<https://www.rfc-editor.org/info/rfc7921>>.
- [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>>.

## Благодарности

Авторы благодарны Chris Bowers, John Scudder, Tom Petch, Mike McBride, Ebben Aries за рецензии, предложения и комментарии к документу.

## Участники работы

Ниже указаны работы над документом.

- Zekun He, Tencent Holdings Ltd.
- Sujian Lu, Tencent Holdings Ltd.
- Jeffery Zhang, Juniper Networks

## Адреса авторов

**Lixing Wang**  
Individual  
Email: [wang\\_little\\_star@sina.com](mailto:wang_little_star@sina.com)

**Mach(Guoyi) Chen**  
Huawei  
Email: [mach.chen@huawei.com](mailto:mach.chen@huawei.com)

**Amit Dass**  
Ericsson  
Email: [dass.amit@gmail.com](mailto:dass.amit@gmail.com)

**Hariharan Ananthakrishnan**  
Netflix  
Email: [hari@netflix.com](mailto:hari@netflix.com)

**Sriganesh Kini**  
Individual  
Email: [sriganeshkini@gmail.com](mailto:sriganeshkini@gmail.com)

**Nitin Bahadur**  
Uber  
Email: [nitin\\_bahadur@yahoo.com](mailto:nitin_bahadur@yahoo.com)

## Перевод на русский язык

Николай Малых

[nmalykh@protokols.ru](mailto:nmalykh@protokols.ru)