

## A YANG Data Model for Routing Management

Модель данных YANG для управления маршрутизацией

### Аннотация

В этом документе приведены спецификации трёх модулей YANG и одного submodule, которые совместно формируют модель данных маршрутизации, служащую основой для настройки и управления подсистемой маршрутизации. Предполагается, что эти модули будут дополнены другими модулями YANG, определяющими модели данных для протоколов плоскости управления, фильтров маршрутов и других функций. Ядро модели данных маршрутизации даёт основу для таких расширений - маршруты, базы маршрутных сведений (Routing Information Base или RIB), протоколы плоскости управления.

### 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 <http://www.rfc-editor.org/info/rfc8022>.

### Авторские права

Copyright (c) 2016. Авторские права 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.....	2
2. Термины и обозначения.....	2
2.1. Глоссарий новых терминов.....	2
2.2. Диаграммы деревьев.....	2
2.3. Префиксы в именах узлов данных.....	3
3. Цели.....	3
4. Устройство модели данных ядра маршрутизации.....	3
4.1. Управляемые системой и пользователем записи списков.....	4
5. Базовые блоки.....	4
5.1. Маршрут.....	4
5.2. База маршрутных данных (RIB).....	4
5.3. Протокол плоскости управления.....	4
5.3.1. Псевдопротоколы маршрутизации.....	4
5.3.2. Определение новых протоколов плоскости управления.....	5
5.4. Параметры анонсов маршрутизаторов IPv6.....	5
6. Взаимодействие с другими модулями YANG.....	5
6.1. ietf-interfaces.....	5
6.2. ietf-ip.....	5
7. Модуль YANG для управления маршрутизацией.....	6
8. Модуль YANG для управления маршрутизацией IPv4 Unicast.....	12
9. Модуль YANG для управления маршрутизацией IPv6 Unicast.....	15
9.1. Submodule для анонсов маршрутизаторов IPv6.....	18
10. Взаимодействие с IANA.....	23
11. Вопросы безопасности.....	25
12. Литература.....	25
12.1. Нормативные документы.....	25
12.2. Дополнительная литература.....	25
Приложение А. Полные деревья данных.....	25
А.1. Данные конфигурации.....	26
А.2. Данные состояния.....	26

<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.

Приложение В. Минимальная реализация.....	27
Приложение С. Пример добавления протокола плоскости управления.....	27
Приложение D. Пример дерева данных.....	28
Благодарности.....	33
Адреса авторов.....	33

## 1. Введение

Этот документ содержит спецификации трёх модулей YANG.

- Модуль `ietf-routing` содержит базовые компоненты модели данных маршрутизации.
- Модуль `ietf-ipv4-unicast-routing` дополняет `ietf-routing` данными для индивидуального трафика IPv4.
- Модуль `ietf-ipv6-unicast-routing` дополняет `ietf-routing` данными для индивидуального трафика IPv6, а его submodule `ietf-ipv6-router-advertisements` дополняет модули `ietf-interfaces` [RFC7223] и `ietf-ip` [RFC7277] переменными конфигурации маршрутизатора IPv6, требуемыми [RFC4861].

Вместе эти модули определяют так называемую модель данных ядра маршрутизации, которая предназначена стать основой для будущих разработок, охватывающих более сложные системы маршрутизации. Хотя эти три модуля можно использовать напрямую для простых устройств IP со статической маршрутизацией (Приложение В. Минимальная реализация), их основным назначением является предоставление базовых блоков для более сложных моделей с множеством протоколов плоскости управления, групповой адресацией, дополнительными семействами адресов и расширенными функциями, такими как фильтрация адресов и маршрутизация на основе правил. Поэтому предполагается дополнение модели данных ядра маршрутизации модулями, созданными рабочими группами IETF.

## 2. Термины и обозначения

Ключевые слова **необходимо** (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 [RFC2119].

Ниже указаны термины, определённые в [RFC6241]:

- `client` - клиент;
- `message` - сообщение;
- `protocol operation` - протокольная операция;
- `server` - сервер.

Ниже указаны термины, определённые в [RFC7950]:

- `action` - действие;
- `augment` - дополнение;
- `configuration data` - данные конфигурации;
- `container` - контейнер;
- `container with presence` - контейнер с присутствием;
- `data model` - модель данных;
- `data node` - узел данных;
- `feature` - свойство (функция);
- `leaf` - лист;
- `list` - список;
- `mandatory node` - обязательный узел;
- `module` - модуль;
- `schema tree` - дерево схемы;
- `state data` - данные состояния.
- `RPC (Remote Procedure Call) operation` - операция вызова удалённой процедуры.

### 2.1. Глоссарий новых терминов

***core routing data model*** - модель данных ядра маршрутизации

Модель данных YANG, состоящая из модулей `ietf-routing`, `ietf-ipv4-unicast-routing` и `ietf-ipv6-unicast-routing`.

***direct route*** - прямой маршрут

Маршрут в подключенную непосредственно сеть.

***Routing Information Base (RIB)*** - база маршрутной информации

Объект, содержащий список маршрутов и другие сведения (см. 5.2. База маршрутных данных (RIB)).

***system-controlled entry*** - контролируемая системой сущность

Запись из списка данных состояния (`config false`), создаваемая системой независимо от того, что было настроено явно (см. 4.1. Управляемые системой и пользователем записи списков).

***user-controlled entry*** - контролируемая пользователем сущность

Запись из списка данных состояния (`config false`), создаваемая и удаляемая в результате прямого воздействия операций настройки (см. 4.1. Управляемые системой и пользователем записи списков).

### 2.2. Диаграммы деревьев

Упрощённое графическое представление полного дерева данных дано в Приложении А, а фрагменты дерева приведены в тексте документа.

- Квадратные скобки [ и ] содержат в себе ключи.
- Фигурные скобки { и } содержат имена необязательных функций, делающие соответствующий узел условным.
- В сокращениях перед именами узлов `rw` указывает данные конфигурации (`read-write`), `ro` - данные состояния (`read-only`), `-x` - операции RPC или действия, `-n` - уведомления.
- После имени узла символ ? указывает необязательный узел, ! - контейнер с присутствием, \* - list или leaf-list.

- Круглые скобки включают узлы choice и case, а узлы case помечаются также двоеточием (:).
- Три точки (...) указывают пропущенное содержимое субдерева (ветви).

## 2.3. Префиксы в именах узлов данных

В этом документе имена узлов данных, действий и других объектов модели часто указываются без префиксов, если из контекста понятно, в каком модуле YANG они определены. В остальных случаях применяется префикс соответствующего модуля YANG, как показано в таблице 1.

Таблица 1. Префиксы и соответствующие модули YANG.

Префикс	Модуль YANG	Документ
if	ietf-interfaces	[RFC7223]
ip	ietf-ip	[RFC7277]
rt	ietf-routing	Раздел 7
v4ur	ietf-ipv4-unicast-routing	Раздел 8
v6ur	ietf-ipv6-unicast-routing	Раздел 9
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

## 3. Цели

Исходные цели разработки разработки модуля данных ядра маршрутизации указаны ниже.

- Модели следует подходить для базовых семейств адресов, в частности, IPv4 и IPv6, а также для индивидуальной и групповой (multicast) маршрутизации и многопротокольной коммутации по меткам (Multiprotocol Label Switching или MPLS).
- Следует обеспечивать простоту настройки простых систем маршрутизации IP (например, с применением лишь статических маршрутов), в идеале без создания дополнительных модулей YANG.
- Схема ядра маршрутизации должна допускать сложные реализации с множеством RIB и протоколов плоскости управления, а также контролируемым распространением маршрутной информации.
- Поскольку производители захотят сопоставить модели данных на основе базовой схемы со своими фирменными моделями и интерфейсами настройки, схеме следует обеспечивать гибкость, достаточную для облегчения таких сопоставлений и приспособления моделей данных с иной логикой.

## 4. Устройство модели данных ядра маршрутизации

Модель данных ядра маршрутизации состоит из 3 модулей и 1 субмодуля. Первый модуль, ietf-routing, задаёт базовые компоненты системы маршрутизации, а два других, ietf-ipv4-unicast-routing и ietf-ipv6-unicast-routing, дополняют его узлами данных для индивидуальной маршрутизации IPv4 и IPv6. Субмодуль, ietf-ipv6-router-advertisements, дополняет модули ietf-interfaces [RFC7223] и ietf-ip [RFC7277] конфигурационными переменными для анонсов маршрутизаторов IPv6 в соответствии с [RFC4861]. На рисунках 1 и 2 приведены сокращённые формы иерархий данных конфигурации и состояния, а полные деревья представлены в Приложении А.

```
+--rw routing
  +--rw router-id?
  +--rw control-plane-protocols
  | +--rw control-plane-protocol* [type name]
  |   +--rw type
  |   +--rw name
  |   +--rw description?
  |   +--rw static-routes
  |     +--rw v6ur:ipv6
  |     |   ...
  |     +--rw v4ur:ipv4
  |     |   ...
  +--rw ribs
    +--rw rib* [name]
      +--rw name
      +--rw address-family?
      +--rw description?
```

Рисунок 1. Иерархия данных конфигурации.

На рисунках показаны несколько базовых компонентов, добавляемых в модель маршрутизации, - маршруты, базы RIB со списками маршрутов и протоколы плоскости управления (см. 5. Базовые блоки).

```
+--ro routing-state
  +--ro router-id?
  +--ro interfaces
  | +--ro interface*
  +--ro control-plane-protocols
  | +--ro control-plane-protocol* [type name]
  |   +--ro type
  |   +--ro name
  +--ro ribs
    +--ro rib* [name]
      +--ro name
      +--ro address-family
      +--ro default-rib?
      +--ro routes
      | +--ro route*
      |   ...
```

Рисунок 2. Иерархия данных состояния.

## 4.1. Управляемые системой и пользователем записи списков

Модель данных ядра маршрутизации определяет в дереве схемы несколько списков, таких как `rib`, которые в корректно работающем устройстве должны включать хотя бы по одной записи, а дополнительные записи может добавлять клиент. В этих списках сервер создаёт обязательный элемент - управляемую системой запись в данных состояния - в контейнере `routing-state`. В примере Приложения D список `/routing-state/ribs/rib` имеет две управляемые системой записи - `ipv4-master` и `ipv6-master`.

Дополнительные записи может вносить в конфигурацию клиент, например, по протоколу NETCONF. Такие записи называют управляемыми клиентом. Если сервер воспринимает запись от клиента, она появляется в списке версии данных состояния.

Соответствующие записи в списках данных конфигурации и состояния имеют одинаковые ключи списка.

Клиент может также предоставить дополнительную конфигурацию управляемых системой записей, создавая для этого новую запись с желаемым содержимым в конфигурации. Для привязки этой записи к соответствующей записи списка данных состояния ключу конфигурационной записи присваивается значение ключа из записи состояния.

Удаление управляемой пользователем записи из списка конфигурации ведёт к удалению соответствующей записи в списке состояний. Если же из списка конфигурации удаляется управляемая системой запись, это ведёт лишь к удалению дополнительной конфигурации, а соответствующая запись данных состояния остаётся в списке.

## 5. Базовые блоки

В этом разделе описаны важные компоненты модели данных ядра маршрутизации.

### 5.1. Маршрут

Маршруты являются базовыми элементами системы маршрутизации. Модель данных ядра маршрутизации определяет лишь минимальный набор маршрутных атрибутов.

#### *destination-prefix*

Адресный префикс, задающий адреса получателей, для которых можно использовать маршрут. Обязательный атрибут.

#### *route-preference*

Целое число (называется также административной дистанцией), применяемое при выборе предпочтительного маршрута среди имеющихся для одного префикса. Меньшее значение указывает более предпочтительный маршрут.

#### *next-hop*

Определяет выходной интерфейс или адрес(а) следующего узла пересылки (`next-hop`) или выполняемую для пакета специальную операцию.

Маршруты - это в первую очередь данные состояния, присутствующие в форме записей таблиц RIB (параграф 5.2), но они могут быть и конфигурационными данными, например, в форме заданных вручную статических маршрутов. В этом случае настраиваемые атрибуты маршрута обычно являются подмножеством атрибутов, заданных для маршрутов RIB.

### 5.2. База маршрутных данных (RIB)

Каждая реализация модели данных ядра маршрутизации управляет одной или множеством таблиц RIB, которые являются списками маршрутов, дополненных административными данными. Каждая таблица RIB содержит маршруты лишь для одного семейства адресов, которое представляется отождествлением, выведенным из `rt:address-family`.

В модели данных ядра маршрутизации таблицы RIB служат данными состояния, представленными записями списка `/routing-state/ribs/rib`. Содержимым RIB управляют и манипулируют операции протоколов плоскости управления, что может приводить к добавлению, изменению и удалению маршрутов. Это также включает манипуляции, выполняемые через псевдопротоколы `static` и `direct` (5.3.1. Псевдопротоколы маршрутизации).

Для каждого поддерживаемого семейства адресов **должна** присутствовать в точности 1 таблица RIB, называемая `default RIB`, в которую протоколы плоскости управления помещают принятые по умолчанию маршруты.

Простые реализации маршрутизаторов, не анонсирующие свойство `multiple-ribs`, обычно создают управляемую системой RIB для каждого поддерживаемого семейства адресов и помечают её как `default RIB`. Более сложные маршрутизаторы с поддержкой `multiple-ribs` создают множество RIB для каждого семейства адресов, которые могут применяться для маршрутизации на основе правил и других целей. Для списка `rib` задано 1 действие (см. параграф 7.15 в [RFC7950]).

#### *active-route*

Возвращает активный маршрут RIB для адреса получателя, указанного входным параметром действия.

### 5.3. Протокол плоскости управления

Модель данных ядра маршрутизации обеспечивает открытую схему для задания нескольких экземпляров протоколов плоскости управления, например, протоколов маршрутизации L3. Каждому экземпляру протокола плоскости управления **должен** быть назначен тип, идентификатор которого выводится из `rt:control-plane-protocol`. Модель данных ядра маршрутизации определяет идентификаторы для псевдопротоколов `direct` и `static` (параграф 5.3.1). **Можно** настроить несколько экземпляров одного протокола плоскости управления.

#### 5.3.1. Псевдопротоколы маршрутизации

Модель данных ядра маршрутизации задаёт два специальных типа протоколов маршрутизации - `direct` (прямое подключение) и `static` (статический маршрут). Оба фактически являются псевдопротоколами, т. е. ограничены локальным устройством и не обмениваются маршрутными данными с соседними маршрутизаторами. Каждая реализация модели данных ядра маршрутизации **должна** предоставлять в точности 1 экземпляр псевдопротокола `direct`, который является источником прямых маршрутов ко всем настроенным семействам адресов. Прямые маршруты обычно поддерживаются ядром ОС на основе адресов сетевых интерфейсов (6.2. `ietf-ip`).

Псевдопротокол `static` служит для задания маршрутов вручную. **Можно** (но не требуется) настроить множество экземпляров протокола, но обычно применяется один.

### 5.3.2. Определение новых протоколов плоскости управления

Предполагается, что будущие модули YANG будут создавать модели данных для дополнительных типов протоколов плоскости управления. Такие новые модули будут определять зависящую от протокола данные конфигурации и состояния, а также интегрировать их в схему ядра маршрутизации.

- **Должны** определяться новые идентификаторы для протоколов плоскости управления, а из базовым идентификатором **должен** быть `rt:control-plane-protocol` или идентификатор, выведенный из него.
- **Могут** определяться дополнительные атрибуты маршрутов, желательно в одном месте с использованием группировки YANG. Новые атрибуты добавляются путём дополнения узлов `/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route` и `/rt:routing-state/rt:ribs/rt:rib/rt:output/rt:route` и, возможно других узлов в данных конфигурации и состояния, уведомлений и параметров ввода-вывода действий или операций RPC.
- Параметры конфигурации и/или данные состояния для нового протокола могут определяться путём дополнения узла данных `control-plane-protocol` в `/routing` и `/routing-state`.

С помощью оператора `when` дополненные параметры конфигурации и данные состояния, связанные с новым протоколом, **следует** делать условными и действительными лишь при совпадении `rt:type` или `rt:source-protocol` с идентификатором нового протокола (или производным от него).

**Рекомендуется** инкапсулировать специфические для протокола узлы данных в подбающим образом именованный контейнер с присутствием. Такой контейнер может содержать обязательные узлы данных, которые иначе были бы запрещены на верхнем уровне дополнения.

Описанные выше шаги реализованы в примере модуля YANG для протокола (RIP) в Приложении С.

## 5.4. Параметры анонсов маршрутизаторов IPv6

Модуль YANG `ietf-ipv6-router-advertisements` (параграф 9.1), являющийся submodule `ietf-ipv6-unicast-routing`, дополняет данные конфигурации и состояния интерфейсов IPv6 определениями указанных ниже переменных в соответствии с требованиями параграфа 6.2.1 в [RFC4861].

- `send-advertisements`;
- `max-rtr-adv-interval`;
- `min-rtr-adv-interval`;
- `managed-flag`;
- `other-config-flag`;
- `link-mtu`;
- `reachable-time`;
- `retrans-timer`;
- `cur-hop-limit`;
- `default-lifetime`;
- `prefix-list` (список анонсируемых префиксов).

С каждым префиксом списка связан ряд параметров:

- `valid-lifetime`;
- `on-link-flag`;
- `preferred-lifetime`;
- `autonomous-flag`.

Примечания.

1. Флаг `IsRouter`, который требует [RFC4861], реализован в модуле `ietf-ip` [RFC7277] (лист `ip:forwarding`).
2. Исходная спецификация [RFC4861] позволяет реализациям решать, сохраняются параметры `valid-lifetime` и `preferred-lifetime` неизменными в последовательных анонсах или декрементируются в соответствии с временем. Однако декрементирование представляется проблематичным, поскольку значения могут снова быть сброшены к заданным конфигурацией (большим) после перезагрузки конфигурации. Более того, неизвестно ни одной реализации с декрементированием. Поэтому в submodule `ietf-ipv6-router-advertisements` сохранено прежнее поведение с постоянными значениями.

## 6. Взаимодействие с другими модулями YANG

Семантика модели данных ядра маршрутизации зависит от нескольких параметров конфигурации из других модулей YANG.

### 6.1. ietf-interfaces

В модуле YANG `ietf-interfaces` [RFC7223] определён логический параметр

**`/if:interfaces/if:interface/if:enabled`**

При установке значения `false` для интерфейса сетевого уровня на нем **должны** быть отключены все функции маршрутизации и пересылки.

### 6.2. ietf-ip

В модуле YANG `ietf-ip` [RFC7277] определены логические параметры, указанные ниже.

**`/if:interfaces/if:interface/ip:ipv4/ip:enabled`**

При установке значения `false` для интерфейса сетевого уровня на нем **должны** быть отключены все функции маршрутизации и пересылки IPv4.

***/if:interfaces/if:interface/ip:ipv4/ip:forwarding***

При установке значения false для интерфейса сетевого уровня на нем **должны** быть отключены все функции пересылки дейтаграмм IPv4. Однако интерфейс **может** участвовать в других функциях маршрутизации IPv4, таких как протоколы маршрутизации.

***/if:interfaces/if:interface/ip:ipv6/ip:enabled***

При установке значения false для интерфейса сетевого уровня на нем **должны** быть отключены все функции маршрутизации и пересылки IPv6.

***/if:interfaces/if:interface/ip:ipv6/ip:forwarding***

При установке значения false для интерфейса сетевого уровня на нем **должны** быть отключены все функции пересылки дейтаграмм IPv6. Однако интерфейс **может** участвовать в других функциях маршрутизации IPv6, таких как протоколы маршрутизации.

Кроме того, модуль ietf-ip позволяет настраивать на интерфейсах сетевого уровня адреса IPv4 и IPv6, префиксы и маски сетей. Настройка этих параметров на включённом (enabled) интерфейсе **должна** незамедлительно приводить к созданию соответствующего прямого (direct) маршрута. Префикс получателей для маршрута устанавливается в соответствии с настроенным адресом IP, префиксом и маской сети, а интерфейс становится выходным для маршрута.

**7. Модуль YANG для управления маршрутизацией**

```
<CODE BEGINS> file "ietf-routing@2016-11-04.yang"
```

```
module ietf-routing {
  yang-version "1.1";

  namespace "urn:ietf:params:xml:ns:yang:ietf-routing";

  prefix "rt";

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-interfaces {
    prefix "if";
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    WG Chair: Lou Berger
              <mailto:lberger@labn.net>

    WG Chair: Kent Watsen
              <mailto:kwatsen@juniper.net>

    Editor: Ladislav Lhotka
            <mailto:lhotka@nic.cz>

    Editor: Acee Lindem
            <mailto:acee@cisco.com>";

  description
    "Этот модуль YANG задаёт важные компоненты управления подсистемой
    маршрутизации.

    Авторские права (Copyright (c) 2016) принадлежат IETF Trust
    и лицам, указанным в качестве авторов кода. Все права защищены.

    Распространение и использование в исходной или двоичной форме с
    изменениями или без таковых разрешено в соответствии с лицензией
    Simplified BSD, изложенной в разделе 4 IETF Trust's Legal
    Provisions применительно к документам IETF
    (http://trustee.ietf.org/license-info).

    Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
    СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
    НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
    RFC 2119.

    Эта версия данного модуля YANG является частью RFC 8022, где
    правовые вопросы рассмотрены более полно.";

  revision 2016-11-04 {
    description
      "Исходный выпуск.";
    reference
      "RFC 8022: A YANG Data Model for Routing Management";
  }

  /* Свойства (функции) */
}
```

```
feature multiple-ribs {
  description
    "Указывает поддержку сервером пользовательских таблиц RIB.

    Серверам, не анонсирующим это свойство, СЛЕДУЕТ поддерживать
    в точности 1 контролируемую системой таблицу RIB на
    поддерживаемое семейство адресов и делать её default RIB. Эта
    таблица будет записью в списке /routing-state/ribs/rib.";
}

feature router-id {
  description
    "Указывает поддержку сервером настройки явных 32-битовых
    Router ID, применяемых некоторыми протоколами маршрутизации.

    Серверы, не анонсирующие это свойство, задают Router ID
    алгоритмически, обычно по настроенным адресам IPv4. Однако
    алгоритм зависит от реализации.";
}

/* Отождествления (идентификаторы) */

identity address-family {
  description
    "Базовый идентификатор, из которого выводятся идентификаторы,
    описывающие семейства адресов.";
}

identity ipv4 {
  base address-family;
  description
    "Представляет семейство адресов IPv4.";
}

identity ipv6 {
  base address-family;
  description
    "Представляет семейство адресов IPv6.";
}

identity control-plane-protocol {
  description
    "Базовый идентификатор, из которого выводятся идентификаторы
    протоколов плоскости управления.";
}

identity routing-protocol {
  base control-plane-protocol;
  description
    "Идентификатор, из которого выводятся идентификаторы протоколов
    маршрутизации L3.";
}

identity direct {
  base routing-protocol;
  description
    "Псевдопротокол для маршрутов в подключённые напрямую сети.";
}

identity static {
  base routing-protocol;
  description
    "Псевдопротокол статической маршрутизации.";
}

/* Определение типа */

typedef route-preference {
  type uint32;
  description
    "Тип для предпочтений маршрутов.";
}

/* Группировки */

grouping address-family {
  description
    "Поддерживает листья, указывающие семейства адресов.";
  leaf address-family {
    type identityref {
      base address-family;
    }
    mandatory "true";
    description
      "Семейство адресов.";
  }
}
```

```
}
}

grouping router-id {
  description
  "Группировка для Router ID.";
  leaf router-id {
    type yang:dotted-quad;
    description
    "32-битовое число с разделением точками, применяемое в
    некоторых протоколах маршрутизации как идентификатор.";
    reference
    "RFC 2328: OSPF Version 2.";
  }
}

grouping special-next-hop {
  description
  "Обеспечивает листья с перечислением специальных next-hop.";
  leaf special-next-hop {
    type enumeration {
      enum blackhole {
        description
        "Отбрасывание пакетов без уведомления.";
      }
      enum unreachable {
        description
        "Отбрасывание пакетов с передачей отправителю сообщения
        об ошибке, указывающего недоступность хоста-адресата.";
      }
      enum prohibit {
        description
        "Отбрасывание пакетов с передачей отправителю сообщения
        об ошибке, указывающего административный запрет связи.";
      }
      enum receive {
        description
        "Пакет был получен локальной системой.";
      }
    }
  }
  description
  "Опции для специальных next-hop.";
}

grouping next-hop-content {
  description
  "Базовые параметры next-hop в статических маршрутах.";
  choice next-hop-options {
    mandatory "true";
    description
    "Опции next hop в статических маршрутах.

    Предполагается добавление case в будущих модулях.";
    case simple-next-hop {
      description
      "Простое указание next-hop адресом следующего узла и/или
      выходным интерфейсом.

      Модули для семейств адресов ДОЛЖНЫ дополнять этот вариант
      листом с адресом next-hop из этого семейства.";
      leaf outgoing-interface {
        type if:interface-ref;
        description
        "Имя выходного интерфейса.";
      }
    }
    case special-next-hop {
      uses special-next-hop;
    }
    case next-hop-list {
      container next-hop-list {
        description
        "Контейнер для множества next-hop.";
        list next-hop {
          key "index";
          description
          "Запись списка next-hop. Модули для семейств адресов
          ДОЛЖНЫ дополнять этот список листом с адресом
          next-hop из этого семейства.";
          leaf index {
            type string;
            description
            "Пользовательский идентификатор для однозначного
            указания записи в списке next-hop. Значение не имеет
            какой-либо дополнительной семантики.";
          }
        }
      }
    }
  }
}
```



```

    }
    leaf outgoing-interface {
        type if:interface-ref;
        description
            "Имя выходного интерфейса.";
    }
}
}
}
}
}

grouping next-hop-state-content {
    description
        "Базовые параметры next-hop в данных состояния.";
    choice next-hop-options {
        mandatory "true";
        description
            "Опции для next-hop в данных состояния.

            Предполагается добавление case в других модулях,
            например, для рекурсивных next-hop.";
        case simple-next-hop {
            description
                "Простое указание next-hop адресом следующего узла и/или
                выходным интерфейсом.

                Модули для семейств адресов ДОЛЖНЫ дополнять этот вариант
                листом с адресом next-hop из этого семейства.";
            leaf outgoing-interface {
                type if:interface-state-ref;
                description
                    "Имя выходного интерфейса.";
            }
        }
        case special-next-hop {
            uses special-next-hop;
        }
        case next-hop-list {
            container next-hop-list {
                description
                    "Контейнер для множества next-hop.";
                list next-hop {
                    description
                        "Запись списка next-hop. Модули для семейств адресов
                        ДОЛЖНЫ дополнять этот список листом с адресом
                        next-hop из этого семейства.";
                    leaf outgoing-interface {
                        type if:interface-state-ref;
                        description
                            "Имя выходного интерфейса.";
                    }
                }
            }
        }
    }
}

grouping route-metadata {
    description
        "Базовые метаданные маршрута.";
    leaf source-protocol {
        type identityref {
            base routing-protocol;
        }
        mandatory "true";
        description
            "Тип протокола маршрутизации, создавшего маршрут.";
    }
    leaf active {
        type empty;
        description
            "Наличие этого листа указывает, что маршрут является
            предпочтительным перед другими маршрутами таблицы RIB
            с тем же префиксом адресата.";
    }
    leaf last-updated {
        type yang:date-and-time;
        description
            "Метка времени последнего изменения маршрута. Если маршрут не
            менялся, это будет время вставки маршрута в таблицу RIB.";
    }
}

/* Данные состояния */

```

```
container routing-state {
  config "false";
  description
    "Данные состояния подсистемы маршрутизации.";
  uses router-id {
    description
      "Глобальное значение Router ID. Может настраиваться или
      алгоритмически определяться реализацией.";
  }
  container interfaces {
    description
      "Интерфейсы сетевого уровня, служащие для маршрутизации.";
    leaf-list interface {
      type if:interface-state-ref;
      description
        "Каждая запись указывает имя настроенного интерфейса
        сетевого уровня.";
    }
  }
}
container control-plane-protocols {
  description
    "Контейнер для списка экземпляров протоколов маршрутизации.";
  list control-plane-protocol {
    key "type name";
    description
      "Данные состояния протокола плоскости управления.

      Реализация ДОЛЖНА обеспечивать в точности 1 управляемый
      системой экземпляр псевдопротокола direct. Конфигурация
      МОЖЕТ создавать другие экземпляры протоколов плоскости
      управления.";
    leaf type {
      type identityref {
        base control-plane-protocol;
      }
      description
        "Тип протокола плоскости управления.";
    }
    leaf name {
      type string;
      description
        "Имя экземпляра протокола плоскости управления. Для
        управляемых системой экземпляров это имя постоянно,
        т. е. его НЕ СЛЕДУЕТ менять при перезагрузке.";
    }
  }
}
container ribs {
  description
    "Контейнер для таблиц RIB.";
  list rib {
    key "name";
    min-elements "1";
    description
      "Каждая запись представляет RIB, указываемую ключом name.
      Все маршруты в таблице RIB ДОЛЖНЫ относиться к одному
      семейству адресов. Реализации СЛЕДУЕТ обеспечивать 1
      управляемую системой default RIB для каждого
      поддерживаемого семейства адресов.";
    leaf name {
      type string;
      description
        "Имя таблицы RIB.";
    }
    uses address-family;
    leaf default-rib {
      if-feature "multiple-ribs";
      type boolean;
      default "true";
      description
        "Значение true устанавливается для таблицы, служащей
        default RIB для этого семейства адресов. По умолчанию
        протоколы плоскости управления помещают свои маршруты
        в таблицы default RIB.";
    }
  }
  container routes {
    description
      "Текущее содержимое RIB.";
    list route {
      description
        "Маршрутная запись RIB. Этот узел данных ДОЛЖЕН
        дополняться сведениями о маршрутах для каждого
        семейства адресов.";
      leaf route-preference {
        type route-preference;
        description

```





```

description
  "Этот модуль YANG дополняет модуль ietf-routing базовыми данными
  конфигурации и состояния для индивидуальной маршрутизации IPv4.

  Авторские права (Copyright (c) 2016) принадлежат IETF Trust
  и лицам, указанным в качестве авторов кода. Все права защищены.

  Распространение и использование в исходной или двоичной форме с
  изменениями или без таковых разрешено в соответствии с лицензией
  Simplified BSD, изложенной в разделе 4 IETF Trust's Legal
  Provisions применительно к документам IETF
  (http://trustee.ietf.org/license-info).

  Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
  СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
  НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
  RFC 2119.

  Эта версия данного модуля YANG является частью RFC 8022, где
  правовые вопросы рассмотрены более полно."

revision 2016-11-04 {
  description
    "Исходный выпуск.";
  reference
    "RFC 8022: A YANG Data Model for Routing Management";
}

/* Идентификаторы (отождествления) */

identity ipv4-unicast {
  base rt:ipv4;
  description
    "Представляет семейство индивидуальных адресов IPv4."
}

/* State data */

augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Дополнение, действительное лишь для IPv4 unicast."
  }
  description
    "Этот лист дополняет маршрут IPv4 unicast."
  leaf destination-prefix {
    type inet:ipv4-prefix;
    description
      "Префикс получателя IPv4."
  }
}

augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Дополнение, действительное лишь для IPv4 unicast."
  }
  description
    "Дополняет вариант simple-next-hop в маршрутах IPv4 unicast."
  leaf next-hop-address {
    type inet:ipv4-address;
    description
      "Адрес IPv4 следующего узла (next-hop)."
  }
}

augment "/rt:routing-state/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" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Дополнение, действительное лишь для IPv4 unicast."
  }
  description
    "Дополнение варианта next-hop-list в маршрутах IPv4 unicast."
  leaf address {
    type inet:ipv4-address;
    description
      "Адрес IPv4 следующего узла (next-hop)."
  }
}

```

```

augment
  "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/rt:input" {
  when "derived-from-or-self(..rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Дополнение, действительное лишь для IPv4 unicast RIB.";
  }
  description
    "Добавляет входной параметр действия active-route.";
  leaf destination-address {
    type inet:ipv4-address;
    description
      "Адрес получателя IPv4.";
  }
}

augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
  + "rt:output/rt:route" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Дополнение, действительное лишь для IPv4 unicast.";
  }
  description
    "Добавляет префикс адресата в ответ для действия active-route.";
  leaf destination-prefix {
    type inet:ipv4-prefix;
    description
      "Адрес получателя IPv4.";
  }
}

augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
  + "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"
  + "rt:simple-next-hop" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Дополнение, действительное лишь для IPv4 unicast.";
  }
  description
    "Дополняет вариант simple-next-hop в ответе на active-route.";
  leaf next-hop-address {
    type inet:ipv4-address;
    description
      "Адрес IPv4 следующего узла (next-hop).";
  }
}

augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
  + "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"
  + "rt:next-hop-list/rt:next-hop-list/rt:next-hop" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Дополнение, действительное лишь для IPv4 unicast.";
  }
  description
    "Дополняет вариант next-hop-list в ответе на active-route.";
  leaf next-hop-address {
    type inet:ipv4-address;
    description
      "Адрес IPv4 следующего узла (next-hop).";
  }
}

/* Данные конфигурации */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rt:static-routes" {
  description
    "Дополняет конфигурацию псевдопротокола static данными
    для IPv4 unicast.";
  container ipv4 {
    description
      "Конфигурация экземпляра псевдопротокола static состоит из
    списка маршрутов.";
    list route {
      key "destination-prefix";
      description
        "Список статических маршрутов.";
      leaf destination-prefix {
        type inet:ipv4-prefix;
        mandatory "true";
        description
          "Префикс адресата IPv4.";
      }
    }
  }
}

```



Авторские права (Copyright (c) 2016) принадлежат IETF Trust и лицам, указанным в качестве авторов кода. Все права защищены.

Распространение и использование в исходной или двоичной форме с изменениями или без таковых разрешено в соответствии с лицензией Simplified BSD, изложенной в разделе 4 IETF Trust's Legal Provisions применительно к документам IETF (<http://trustee.ietf.org/license-info>).

Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО, СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО, НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с RFC 2119.

Эта версия данного модуля YANG является частью RFC 8022, где правовые вопросы рассмотрены более полно.";

```

revision 2016-11-04 {
  description
    "Исходный выпуск.";
  reference
    "RFC 8022: A YANG Data Model for Routing Management";
}

/* Идентификаторы (отождествления) */

identity ipv6-unicast {
  base rt:ipv6;
  description
    "Представляет семейство индивидуальных адресов IPv6.";
}

/* Данные состояния */

augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v6ur:ipv6-unicast')" {
    description
      "Дополнение, действительное лишь для IPv6 unicast.";
  }
  description
    "Дополняет маршрут IPv6 unicast.";
  leaf destination-prefix {
    type inet:ipv6-prefix;
    description
      "Префикс адресата IPv6.";
  }
}

augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v6ur:ipv6-unicast')" {
    description
      "Дополнение, действительное лишь для IPv6 unicast.";
  }
  description
    "Augment 'simple-next-hop' case in IPv6 unicast routes.";
  leaf next-hop-address {
    type inet:ipv6-address;
    description
      "Адрес IPv6 следующего узла (next-hop).";
  }
}

augment "/rt:routing-state/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" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v6ur:ipv6-unicast')" {
    description
      "Дополнение, действительное лишь для IPv6 unicast.";
  }
  description
    "Дополняет вариант next-hop-list для маршрутов IPv6 unicast.";
  leaf address {
    type inet:ipv6-address;
    description
      "Адрес IPv6 следующего узла (next-hop).";
  }
}

augment
  "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/rt:input" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v6ur:ipv6-unicast')" {

```



```

    description
      "Дополнение, действительное лишь для IPv6 unicast.";
  }
  description
    "Добавляет входной параметр действия active-route.";
  leaf destination-address {
    type inet:ipv6-address;
    description
      "Адрес получателя IPv6.";
  }
}

augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
+ "rt:output/rt:route" {
  when "derived-from-or-self(..../rt:address-family, "
+ "'v6ur:ipv6-unicast')" {
    description
      "Дополнение, действительное лишь для IPv6 unicast.";
  }
  description
    "Добавляет префикс адресата в ответ на действие active-route.";
  leaf destination-prefix {
    type inet:ipv6-prefix;
    description
      "Префикс адресата IPv6.";
  }
}

augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
+ "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"
+ "rt:simple-next-hop" {
  when "derived-from-or-self(..../rt:address-family, "
+ "'v6ur:ipv6-unicast')" {
    description
      "Дополнение, действительное лишь для IPv6 unicast.";
  }
  description
    "Дополняет вариант simple-next-hop в ответе на active-route.";
  leaf next-hop-address {
    type inet:ipv6-address;
    description
      "Адрес IPv6 следующего узла (next-hop).";
  }
}

augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
+ "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"
+ "rt:next-hop-list/rt:next-hop-list/rt:next-hop" {
  when "derived-from-or-self(..../rt:address-family, "
+ "'v6ur:ipv6-unicast')" {
    description
      "Дополнение, действительное лишь для IPv6 unicast.";
  }
  description
    "Дополняет вариант next-hop-list в ответе на active-route.";
  leaf next-hop-address {
    type inet:ipv6-address;
    description
      "Адрес IPv6 следующего узла (next-hop).";
  }
}

/* Данные конфигурации */

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/rt:static-routes" {
  description
    "Дополняет конфигурацию псевдопротокола static данными
для IPv6 unicast.";
  container ipv6 {
    description
      "Конфигурация экземпляра псевдопротокола static состоит из
списка маршрутов.";
    list route {
      key "destination-prefix";
      description
        "Список статических маршрутов.";
      leaf destination-prefix {
        type inet:ipv6-prefix;
        mandatory "true";
        description
          "Префикс адресата IPv6.";
      }
    }
    leaf description {
      type string;
      description
    }
  }
}

```



изменениями или без таковых разрешено в соответствии с лицензией Simplified BSD, изложенной в разделе 4 IETF Trust's Legal Provisions применительно к документам IETF (<http://trustee.ietf.org/license-info>).

Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО, СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО, НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с RFC 2119.

Эта версия данного модуля YANG является частью RFC 8022, где правовые вопросы рассмотрены более полно.";

```
reference
  "RFC 4861: Neighbor Discovery for IP version 6 (IPv6).";

revision 2016-11-04 {
  description
    "Исходный выпуск.";
  reference
    "RFC 8022: A YANG Data Model for Routing Management";
}

/* данные состояния */

augment "/if:interfaces-state/if:interface/ip:ipv6" {
  description
    "Дополняет данные состояния интерфейса параметрами анонсов
    маршрутизатора IPv6.";
  container ipv6-router-advertisements {
    description
      "Параметры анонсов маршрутизатора IPv6.";
    leaf send-advertisements {
      type boolean;
      description
        "Флаг, управляющий периодической отправкой сообщений
        Router Advertisement и откликов на Router Solicitation.";
    }
    leaf max-rtr-adv-interval {
      type uint16 {
        range "4..1800";
      }
      units "seconds";
      description
        "Максимальный интервал между отправкой незапрошенных
        групповых сообщений Router Advertisement с интерфейса.";
    }
    leaf min-rtr-adv-interval {
      type uint16 {
        range "3..1350";
      }
      units "seconds";
      description
        "Минимальный интервал между отправкой незапрошенных
        групповых сообщений Router Advertisement с интерфейса.";
    }
    leaf managed-flag {
      type boolean;
      description
        "Значение, помещаемое в поле флага Managed address
        configuration сообщений Router Advertisement.";
    }
    leaf other-config-flag {
      type boolean;
      description
        "Значение, помещаемое в поле флага Other configuration
        сообщений Router Advertisement.";
    }
    leaf link-mtu {
      type uint32;
      description
        "Значение, помещаемое в опции MTU, передаваемые
        маршрутизатором. 0 указывает отсутствие опций MTU.";
    }
    leaf reachable-time {
      type uint32 {
        range "0..3600000";
      }
      units "milliseconds";
      description
        "Значение, помещаемое в поле Reachable Time сообщений
        Router Advertisement, передаваемых маршрутизатором.
        0 говорит о незаданном (этим маршрутизатором) значении.";
    }
    leaf retrans-timer {
      type uint32;
    }
  }
}
```



```

"Дополняет конфигурацию интерфейса параметрами анонсов
маршрутизатора IPv6.";
container ipv6-router-advertisements {
  description
    "Конфигурация IPv6 Router Advertisement.";
  leaf send-advertisements {
    type boolean;
    default "false";
    description
      "Флаг, управляющий периодической отправкой сообщений
      Router Advertisement и откликов на Router Solicitation.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
      AdvSendAdvertisements.";
  }
  leaf max-rtr-adv-interval {
    type uint16 {
      range "4..1800";
    }
    units "seconds";
    default "600";
    description
      "Максимальный интервал между отправкой незапрошенных
      групповых сообщений Router Advertisement с интерфейса.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
      MaxRtrAdvInterval.";
  }
  leaf min-rtr-adv-interval {
    type uint16 {
      range "3..1350";
    }
    units "seconds";
    must ". <= 0.75 * ../max-rtr-adv-interval" {
      description
        "НЕДОПУСТИМЫ значения более 75% max-rtr-adv-interval.";
    }
    description
      "Минимальный интервал между отправкой незапрошенных
      групповых сообщений Router Advertisement с интерфейса.";

    Используемое по умолчанию значение определяется как
    указано ниже:

    - если max-rtr-adv-interval >= 9 секунд, по умолчанию
      принято 0.33 * max-rtr-adv-interval;

    - в иных случаях 0.75 * max-rtr-adv-interval.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
      MinRtrAdvInterval.";
  }
  leaf managed-flag {
    type boolean;
    default "false";
    description
      "Значение, помещаемое в поле флага Managed address
      configuration сообщений Router Advertisement.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
      AdvManagedFlag.";
  }
  leaf other-config-flag {
    type boolean;
    default "false";
    description
      "Значение, помещаемое в поле флага Other configuration
      сообщений Router Advertisement.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
      AdvOtherConfigFlag.";
  }
  leaf link-mtu {
    type uint32;
    default "0";
    description
      "Значение, помещаемое в опции MTU, передаваемые
      маршрутизатором. 0 указывает отсутствие опций MTU.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
      AdvLinkMTU.";
  }
  leaf reachable-time {
    type uint32 {
      range "0..3600000";
    }
  }
}

```

```
units "milliseconds";
default "0";
description
  "Значение, помещаемое в поле Reachable Time сообщений
  Router Advertisement, передаваемых маршрутизатором.
  0 говорит о незаданном (этим маршрутизатором) значении.";
reference
  "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
  AdvReachableTime.";
}
leaf retrans-timer {
  type uint32;
  units "milliseconds";
  default "0";
  description
    "Значение, помещаемое в поле Retrans Timer сообщений
    Router Advertisement, передаваемых маршрутизатором.
    0 говорит о незаданном (этим маршрутизатором) значении.";
  reference
    "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
    AdvRetransTimer.";
}
leaf cur-hop-limit {
  type uint8;
  description
    "Значение, помещаемое в поле Cur Hop Limit сообщений
    Router Advertisement, передаваемых маршрутизатором.
    0 говорит о незаданном (этим маршрутизатором) значении.";

    Если этот параметр не задан, устройству СЛЕДУЕТ применять
    значение из IANA Assigned Numbers на момент реализации.";
  reference
    "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
    AdvCurHopLimit.

    IANA: параметры IP,
    http://www.iana.org/assignments/ip-parameters";
}
leaf default-lifetime {
  type uint16 {
    range "0..9000";
  }
  units "seconds";
  description
    "Значение, помещаемое в поле Router Lifetime сообщений
    Router Advertisement, передаваемых с этого интерфейса
    (в секундах). Это ДОЛЖЕН быть 0 или значение из интервала
    max-rtr-adv-interval - 9000 секунд. 0 указывает, что
    маршрутизатор не применяется по умолчанию. Пределы могут
    быть переопределены документами, задающими работу IPv6
    на канальном уровне.

    По умолчанию устройству СЛЕДУЕТ использовать значение
    3 * max-rtr-adv-interval.";
  reference
    "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
    AdvDefaultLifeTime.";
}
container prefix-list {
  description
    "Конфигурация префиксов, помещаемых в опции Prefix
    Information сообщений Router Advertisement с этого
    интерфейса.

    Префиксы, анонсируемые по умолчанию, но не имеющие записей
    в списке дочерних префиксов, анонсируются с принятыми по
    умолчанию значениями всех параметров.

    Префикс link-local НЕ СЛЕДУЕТ включать в число
    анонсируемых.";
  reference
    "RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
    AdvPrefixList.";
  list prefix {
    key "prefix-спец";
    description
      "Конфигурация анонсируемой записи для префикса.";
    leaf prefix-спец {
      type inet:ipv6-prefix;
      description
        "Префикс IPv6.";
    }
  }
  choice control-adv-prefixes {
    default "advertise";
    description
      "Любой префикс явно удаляется из числа анонсируемых или
```



Reference: RFC 8022

Name: ietf-ipv4-unicast-routing



Reference: RFC 8022

Name: ietf-ipv6-unicast-routing  
 Namespace: urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing  
 Prefix: v6ur  
 Reference: RFC 8022

Этот документ регистрирует субмодуль YANG в реестре YANG Module Names [RFC6020].

Name: ietf-ipv6-router-advertisements  
 Module: ietf-ipv6-unicast-routing  
 Reference: RFC 8022

## 11. Вопросы безопасности

Данные конфигурации и состояния модели данных ядра маршрутизации (определена в этом документе) предназначены для доступа через протоколы управления сетью, такие как NETCONF [RFC6241]. Модель управления доступом NETCONF [RFC6536] обеспечивает средства, позволяющие предоставить доступ лишь конкретным пользователям NETCONF control model [RFC6536] к предопределённому подмножеству доступных в NETCONF протокольных операций и содержимого.

Многие узлы данных конфигурации в этом модуле YANG, относящиеся к модели данных ядра маршрутизации, доступны для записи, создания, удаления (т. е. с принятым по умолчанию config true). Эти узлы могут быть чувствительными или уязвимыми в некоторых сетевых средах. Операции записи (например, edit-config в NETCONF) в такие узлы без подобающей защиты могут оказывать негативное влияние на работу сети.

Ниже показаны уязвимые параметры и ветви config true.

### */routing/control-plane-protocols/control-plane-protocol*

Этот список задаёт настроенные на устройстве протоколы плоскости управления.

### */routing/ribs/rib*

Этот список задаёт настроенные на устройстве таблицы RIB.

Несанкционированный доступ к любому из узлов данных в этих ветвях может негативно влиять на подсистему маршрутизации в локальном устройстве и сети. Это может приводить к некорректной работе сети, доставке пакетов не по назначению и другим проблемам.

## 12. Литература

### 12.1. 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>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), DOI 10.17487/RFC4861, September 2007, <<http://www.rfc-editor.org/info/rfc4861>>.
- [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>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 7223](#), DOI 10.17487/RFC7223, May 2014, <<http://www.rfc-editor.org/info/rfc7223>>.
- [RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", [RFC 7277](#), DOI 10.17487/RFC7277, June 2014, <<http://www.rfc-editor.org/info/rfc7277>>.
- [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>>.

### 12.2. Дополнительная литература

- [RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <<http://www.rfc-editor.org/info/rfc6087>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", [RFC 6536](#), DOI 10.17487/RFC6536, March 2012, <<http://www.rfc-editor.org/info/rfc6536>>.
- [RFC7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", [RFC 7895](#), DOI 10.17487/RFC7895, June 2016, <<http://www.rfc-editor.org/info/rfc7895>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", [RFC 7951](#), DOI 10.17487/RFC7951, August 2016, <<http://www.rfc-editor.org/info/rfc7951>>.

## Приложение А. Полные деревья данных

В этом приложении представлены полные деревья данных конфигурации и состояния для модели данных ядра маршрутизации. Описание применяемых символов дано в параграфе 2.2. Тип данных каждого листа указан в правой части строки.

## A.1. Данные конфигурации

```

+--rw routing
+--rw router-id?                yang:dotted-quad
+--rw control-plane-protocols
| +--rw control-plane-protocol* [type name]
| | +--rw type                identityref
| | +--rw name                string
| | +--rw description?       string
| | +--rw static-routes
| | | +--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)
| | | | | | | +--rw (v6ur:simple-next-hop)
| | | | | | | | +--rw v6ur:outgoing-interface?
| | | | | | | | +--rw v6ur:next-hop-address?
| | | | | | | +--rw (v6ur:special-next-hop)
| | | | | | | | +--rw v6ur:special-next-hop?    enumeration
| | | | | | | +--rw (v6ur:next-hop-list)
| | | | | | | | +--rw v6ur:next-hop-list
| | | | | | | | | +--rw v6ur:next-hop* [index]
| | | | | | | | | +--rw v6ur:index                string
| | | | | | | | | +--rw v6ur:outgoing-interface?
| | | | | | | | | +--rw v6ur:next-hop-address?
| | | | +--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)
| | | | | | | +--rw (v4ur:simple-next-hop)
| | | | | | | | +--rw v4ur:outgoing-interface?
| | | | | | | | +--rw v4ur:next-hop-address?
| | | | | | | +--rw (v4ur:special-next-hop)
| | | | | | | | +--rw v4ur:special-next-hop?    enumeration
| | | | | | | +--rw (v4ur:next-hop-list)
| | | | | | | | +--rw v4ur:next-hop-list
| | | | | | | | | +--rw v4ur:next-hop* [index]
| | | | | | | | | +--rw v4ur:index                string
| | | | | | | | | +--rw v4ur:outgoing-interface?
| | | | | | | | | +--rw v4ur:next-hop-address?
| | +--rw ribs
| | | +--rw rib* [name]
| | | | +--rw name                string
| | | | +--rw address-family?     identityref
| | | | +--rw description?       string

```

## A.2. Данные состояния

```

+--ro routing-state
| +--ro router-id?                yang:dotted-quad
| +--ro interfaces
| | +--ro interface*    if:interface-state-ref
| +--ro control-plane-protocols
| | +--ro control-plane-protocol* [type name]
| | | +--ro type        identityref
| | | +--ro name        string
| +--ro ribs
| | +--ro rib* [name]
| | | +--ro name                string
| | | +--ro 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)
| | | | | | +--rw (simple-next-hop)
| | | | | | | +--ro outgoing-interface?
| | | | | | | +--ro v6ur:next-hop-address?
| | | | | | | +--ro v4ur:next-hop-address?
| | | | | | | +--rw (special-next-hop)
| | | | | | | | +--ro special-next-hop?    enumeration
| | | | | | | +--rw (next-hop-list)
| | | | | | | | +--ro next-hop-list
| | | | | | | | | +--ro next-hop*
| | | | | | | | | +--ro outgoing-interface?
| | | | | | | | | +--ro v6ur:address?
| | | | | | | | | +--ro v4ur:address?
| | | +--ro source-protocol        identityref
| | | +--ro active?                empty
| | | +--ro last-updated?          yang:date-and-time
| | | +--ro v6ur:destination-prefix?    inet:ipv6-prefix
| | | +--ro v4ur:destination-prefix?    inet:ipv4-prefix

```

```

+---x active-route
+---w input
| +---w v6ur:destination-address?  inet:ipv6-address
| +---w v4ur:destination-address?  inet:ipv4-address
+--ro output
+--ro route
+--ro next-hop
| +--ro (next-hop-options)
| | +--:(simple-next-hop)
| | | +--ro outgoing-interface?
| | | +--ro v6ur:next-hop-address?
| | | +--ro v4ur:next-hop-address?
| | +--:(special-next-hop)
| | | +--ro special-next-hop?      enumeration
| | +--:(next-hop-list)
| | | +--ro next-hop-list
| | | | +--ro next-hop*
| | | | +--ro outgoing-interface?
| | | | +--ro v6ur:next-hop-address?
| | | | +--ro v4ur:next-hop-address?
| +--ro source-protocol            identityref
+--ro active?                      empty
+--ro last-updated?                yang:date-and-time
+--ro v6ur:destination-prefix?    inet:ipv6-prefix
+--ro v4ur:destination-prefix?    inet:ipv4-prefix

```

## Приложение В. Минимальная реализация

Некоторые части и опции модели ядра маршрутизации, такие как определяемые пользователем таблицы RIB, предназначены лишь для маршрутизаторов старших классов. В этом приложении приведены базовые ненормативные рекомендации для реализации минимального набора функций. Такие реализации подойдут для хостов и простых маршрутизаторов.

От минимальной реализации не требуется поддержка свойства `multiple-ribs`. Это означает, что для каждого поддерживаемого семейства адресов (IPv4, IPv6) доступна одна управляемая системой таблица RIB. Такие таблицы называют также `default RIB`. Пользовательские таблицы RIB не разрешены.

В дополнение у обязательному экземпляру псевдопротокола `direct` минимальным реализациям следует поддерживать настройку экземпляров псевдопротокола `static`.

Для хостов, не предназначенных быть маршрутизаторами, следует исключить возможность отправки анонсов маршрутизатора IPv6 (параграф 5.4).

Платформы с сильно ограниченными ресурсами могут использовать отклонения (*deviation*) для ограничения модели данных (например, ограничивая число экземпляров протокола плоскости управления `static`).

## Приложение С. Пример добавления протокола плоскости управления

В этом приложении показано, как можно расширить модель данных ядра маршрутизации для поддержки нового протокола плоскости управления. Показанный ниже модуль YANG `example-rip` предназначен для иллюстрации, а не реального определения модели данных для протокола маршрутной информации (Routing Information Protocol или RIP). Для краткости в этом модуле не соблюдаются некоторые рекомендации [RFC6087]. См. также 5.3.2. Определение новых протоколов плоскости управления.

```

module example-rip {

  yang-version "1.1";

  namespace "http://example.com/rip";

  prefix "rip";

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing {
    prefix "rt";
  }

  identity rip {
    base rt:routing-protocol;
    description
      "Идентификатор для протокола RIP.";
  }

  typedef rip-metric {
    type uint8 {
      range "0..16";
    }
  }

  grouping route-content {
    description
      "Группировка определяет связанные с RIP атрибуты маршрутов.";
    leaf metric {

```



```

    "feature": [
      "multiple-ribs",
      "router-id"
    ],
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-ipv4-unicast-routing",
    "revision": "2016-11-04",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-ipv4-unicast-routing",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-ipv6-unicast-routing",
    "revision": "2016-11-04",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-interfaces",
    "revision": "2014-05-08",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-inet-types",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
    "revision": "2013-07-15",
    "conformance-type": "import"
  },
  {
    "name": "ietf-yang-types",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
    "revision": "2013-07-15",
    "conformance-type": "import"
  },
  {
    "name": "iana-if-type",
    "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
    "revision": "",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-ip",
    "revision": "2014-06-16",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
    "conformance-type": "implement"
  }
]
}

```

Предполагается простая сеть, показанная на рисунке 3. Маршрутизатор А использует статические маршруты по умолчанию с маршрутизатором ISP в качестве next hop. Анонсы маршрутизатора IPv6 настроены лишь на интерфейсе eth1 и запрещены на восходящем интерфейсе eth0.

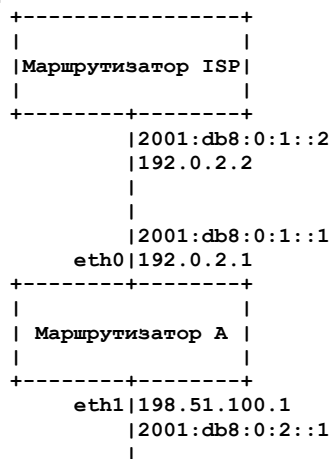


Рисунок 3. Пример конфигурации сети.

Ниже приведено возможное представление дерева данных.

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",

```

```

"description": "Uplink to ISP.",
"ietf-ip:ipv4": {
  "address": [
    {
      "ip": "192.0.2.1",
      "prefix-length": 24
    }
  ],
  "forwarding": true
},
"ietf-ip:ipv6": {
  "address": [
    {
      "ip": "2001:0db8:0:1::1",
      "prefix-length": 64
    }
  ],
  "forwarding": true,
  "autoconf": {
    "create-global-addresses": false
  }
}
},
{
  "name": "eth1",
  "type": "iana-if-type:ethernetCsmacd",
  "description": "Интерфейс во внутреннюю сеть.",
  "ietf-ip:ipv4": {
    "address": [
      {
        "ip": "198.51.100.1",
        "prefix-length": 24
      }
    ],
    "forwarding": true
  },
  "ietf-ip:ipv6": {
    "address": [
      {
        "ip": "2001:0db8:0:2::1",
        "prefix-length": 64
      }
    ],
    "forwarding": true,
    "autoconf": {
      "create-global-addresses": false
    },
    "ietf-ipv6-unicast-routing:ipv6-router-advertisements": {
      "send-advertisements": true
    }
  }
}
]
},
"ietf-interfaces:interfaces-state": {
  "interface": [
    {
      "name": "eth0",
      "type": "iana-if-type:ethernetCsmacd",
      "phys-address": "00:0C:42:E5:B1:E9",
      "oper-status": "up",
      "statistics": {
        "discontinuity-time": "2015-10-24T17:11:27+02:00"
      },
      "ietf-ip:ipv4": {
        "forwarding": true,
        "mtu": 1500,
        "address": [
          {
            "ip": "192.0.2.1",
            "prefix-length": 24
          }
        ]
      },
      "ietf-ip:ipv6": {
        "forwarding": true,
        "mtu": 1500,
        "address": [
          {
            "ip": "2001:0db8:0:1::1",
            "prefix-length": 64
          }
        ],
        "ietf-ipv6-unicast-routing:ipv6-router-advertisements": {
          "send-advertisements": false
        }
      }
    }
  ]
}

```

```

    }
  },
  {
    "name": "eth1",
    "type": "iana-if-type:ethernetCsmacd",
    "phys-address": "00:0C:42:E5:B1:EA",
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2015-10-24T17:11:29+02:00"
    },
    "ietf-ip:ipv4": {
      "forwarding": true,
      "mtu": 1500,
      "address": [
        {
          "ip": "198.51.100.1",
          "prefix-length": 24
        }
      ]
    },
    "ietf-ip:ipv6": {
      "forwarding": true,
      "mtu": 1500,
      "address": [
        {
          "ip": "2001:0db8:0:2::1",
          "prefix-length": 64
        }
      ],
      "ietf-ipv6-unicast-routing:ipv6-router-advertisements": {
        "send-advertisements": true,
        "prefix-list": {
          "prefix": [
            {
              "prefix-spec": "2001:db8:0:2::/64"
            }
          ]
        }
      }
    }
  }
],
},
"ietf-routing:routing": {
  "router-id": "192.0.2.1",
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-routing:static",
        "name": "st0",
        "description": "Для внутренней сети применяется статическая маршрутизация.",
        "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-ipv6-unicast-routing:ipv6": {
            "route": [
              {
                "destination-prefix": "::/0",
                "next-hop": {
                  "next-hop-address": "2001:db8:0:1::2"
                }
              }
            ]
          }
        }
      }
    ]
  }
},
"ietf-routing:routing-state": {
  "interfaces": {
    "interface": [
      "eth0",
      "eth1"
    ]
  }
},
"control-plane-protocols": {

```

```
"control-plane-protocol": [
  {
    "type": "ietf-routing:static",
    "name": "st0"
  }
],
"ribs": {
  "rib": [
    {
      "name": "ipv4-master",
      "address-family":
        "ietf-ipv4-unicast-routing:ipv4-unicast",
      "default-rib": true,
      "routes": {
        "route": [
          {
            "ietf-ipv4-unicast-routing:destination-prefix":
              "192.0.2.1/24",
            "next-hop": {
              "outgoing-interface": "eth0"
            },
            "route-preference": 0,
            "source-protocol": "ietf-routing:direct",
            "last-updated": "2015-10-24T17:11:27+02:00"
          },
          {
            "ietf-ipv4-unicast-routing:destination-prefix":
              "198.51.100.0/24",
            "next-hop": {
              "outgoing-interface": "eth1"
            },
            "source-protocol": "ietf-routing:direct",
            "route-preference": 0,
            "last-updated": "2015-10-24T17:11:27+02:00"
          },
          {
            "ietf-ipv4-unicast-routing:destination-prefix":
              "0.0.0.0/0",
            "source-protocol": "ietf-routing:static",
            "route-preference": 5,
            "next-hop": {
              "ietf-ipv4-unicast-routing:next-hop-address":
                "192.0.2.2"
            },
            "last-updated": "2015-10-24T18:02:45+02:00"
          }
        ]
      }
    }
  ],
  {
    "name": "ipv6-master",
    "address-family":
      "ietf-ipv6-unicast-routing:ipv6-unicast",
    "default-rib": true,
    "routes": {
      "route": [
        {
          "ietf-ipv6-unicast-routing:destination-prefix":
            "2001:db8:0:1::/64",
          "next-hop": {
            "outgoing-interface": "eth0"
          },
          "source-protocol": "ietf-routing:direct",
          "route-preference": 0,
          "last-updated": "2015-10-24T17:11:27+02:00"
        },
        {
          "ietf-ipv6-unicast-routing:destination-prefix":
            "2001:db8:0:2::/64",
          "next-hop": {
            "outgoing-interface": "eth1"
          },
          "source-protocol": "ietf-routing:direct",
          "route-preference": 0,
          "last-updated": "2015-10-24T17:11:27+02:00"
        },
        {
          "ietf-ipv6-unicast-routing:destination-prefix":
            "::/0",
          "next-hop": {
            "ietf-ipv6-unicast-routing:next-hop-address":
              "2001:db8:0:1::2"
          },
          "source-protocol": "ietf-routing:static",
          "route-preference": 5,

```



