

Internet Engineering Task Force (IETF)  
Request for Comments: 8349  
Obsoletes: 8022  
Category: Standards Track  
ISSN: 2070-1721

L. Lhotka  
CZ.NIC  
A. Lindem  
Cisco Systems  
Y. Qu  
Huawei  
March 2018

## A YANG Data Model for Routing Management (NMDA Version)

Модель данных YANG для управления маршрутизацией (версия NMDA)

### Аннотация

Документ задаёт 3 модуля YANG и 1 submodule, которые совместно формируют модель данных ядра маршрутизации, которая служит основой для настройки и управления подсистемой маршрутизации. Предполагается, что эти модули будут дополнены модулями YANG, определяющими модели данных для протоколов плоскости управления, маршрутных фильтров и других функций. Модель данных ядра маршрутизации обеспечивает базовые блоки для таких расширений - маршруты, базы маршрутных данных (Routing Information Base или RIB) и протоколы плоскости управления.

Модули YANG в этом документе соответствуют архитектуре хранилищ данных управления сетью (Network Management Datastore Architecture или NMDA). Этот документ отменяет RFC 8022.

### 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/rfc8349>.

### 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.....	2
2. Термины и обозначения.....	2
2.1. Новые термины.....	2
2.2. Диаграммы деревьев.....	3
2.3. Префиксы имён узлов данных.....	3
3. Цели.....	3
4. Устройство ядра модели данных маршрутизации.....	3
4.1. Контролируемые системой и пользователем записи списков.....	4
5. Базовые элементы.....	4
5.1. Маршруты.....	4
5.2. Базы RIB.....	4
5.3. Протоколы плоскости управления.....	4
5.3.1. Псевдопротоколы маршрутизации.....	5
5.3.2. Определение новых протоколов плоскости управления.....	5
5.4. Параметры анонсов маршрутизаторов IPv6.....	5
6. Взаимодействия с другими модулями YANG.....	6
6.1. Модуль ietf-interfaces.....	6
6.2. Модуль ietf-ip.....	6
7. Модуль управления маршрутизацией.....	6
8. Модуль управления маршрутизацией IPv4.....	14
9. Модуль управления маршрутизацией IPv6.....	18
9.1. Submodule IPv6 RA.....	23
10. Взаимодействие с IANA.....	28
11. Вопросы безопасности.....	29
12. Литература.....	29

<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.

12.1. Нормативные документы.....	29
12.2. Дополнительная литература.....	30
Приложение А. Полное дерево схемы.....	30
Приложение В. Минимальная реализация.....	33
Приложение С. Пример добавления нового протокола управления.....	33
Приложение D. Пример дерева данных.....	34
Приложение E. Пример отклика на NETCONF Get Data.....	38
Благодарности.....	40
Адреса авторов.....	40

## 1. Введение

Этот документ задаёт перечисленные ниже модули YANG.

- ietf-routing содержит базовые компоненты модели данных маршрутизации.
- ietf-ipv4-unicast-routing дополняет модуль ietf-routing данными для индивидуальной адресации IPv4.
- ietf-ipv6-unicast-routing дополняет модуль ietf-routing данными для индивидуальной адресации IPv6 unicast. Его submodule ietf-ipv6-router-advertisements дополняет модули ietf-interfaces [RFC8343] и ietf-ip [RFC8344] переменными настройки маршрутизаторов IPv6, требуемыми [RFC4861].

Эти модули совместно определяют ядро модели данных маршрутизации, предназначенное в качестве основы для будущих моделей более изощрённых систем маршрутизации. Хотя эти три модуля можно напрямую применять для простых устройств IP со статической маршрутизацией (см. Приложение В), их основная задача состоит в предоставлении блоков для создания более сложных моделей данных, включающих несколько протоколов плоскости управления, групповую маршрутизацию, дополнительные семейства адресов и расширенные функции, такие как фильтрация маршрутов или маршрутизация на основе правил. Поэтому предполагается, что базовая модель будет дополнена многочисленными модулями, определёнными различными рабочими группами IETF.

Модули YANG в этом документе соответствуют архитектуре NMDA [RFC8342]. Документ отменяет RFC 8022 [RFC8022].

## 2. Термины и обозначения

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.

Ниже перечислены термины, определённые в [RFC8342]:

- client - клиент;
- server - сервер;
- configuration - конфигурация;
- system state - состояние системы;
- operational state - рабочее состояние;
- intended configuration - предполагаемая конфигурация.

Ряд терминов определён в [RFC7950]:

- action - действие;
- augment - дополнение;
- container - контейнер;
- data model - модель данных;
- data node - узел данных;
- feature - свойство, функция;
- leaf - лист;
- list - список;
- mandatory node - обязательный узел;
- module - модуль;
- presence container - контейнер присутствия;
- schema tree - дерево схемы;
- RPC (Remote Procedure Call) operation - вызов удалённой процедуры.

### 2.1. Новые термины

#### **core routing data model - ядро модели данных**

Модель данных YANG, включающая модули ietf-routing, ietf-ipv4-unicast-routing, ietf-ipv6-unicast-routing.

#### **direct route - прямой маршрут**

Маршрут в подключённую напрямую сеть.

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

Объект, содержащий список маршрутов вместе с другой информацией (см. параграф 5.2).

**system-controlled entry - управляемый системой объект (сущность)**

Запись в списке операционного состояния (config false), создаваемая системой независимо от того, что было настроено явно (см. параграф 4.1).

**user-controlled entry - управляемый пользователем объект (сущность)**

Запись в списке операционного состояния (config false), которая создаётся и удаляется как прямое следствие некоторых конфигурационных изменений (см. параграф 4.1).

## 2.2. Диаграммы деревьев

Диаграммы деревьев в этом документе используют обозначения, определённые в [RFC8340].

## 2.3. Префиксы имён узлов данных

В этом документе имена узлов данных, действий и других объектов модели данных используются без префиксов, если из контекста очевидно, в каком модуле YANG они определены. В остальных случаях имена указываются со стандартным префиксом соответствующего модуля YANG (Таблица 1).

Таблица 1. Префиксы и модули YANG.  
Документ

Префикс	Модуль YANG	
if	ietf-interfaces	[RFC8343]
ip	ietf-ip	[RFC8344]
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

При разработке ядра модели данных маршрутизации преследовались указанные ниже цели.

- Модели данных следует поддерживать работу с основными семействами адресов (в частности, IPv4 и IPv6) для индивидуальной (unicast) и групповой (multicast) маршрутизации, а также многопротокольной коммутации по меткам (Multiprotocol Label Switching или MPLS).
- Простым системам маршрутизации IP (например, применяющим лишь статические маршруты) следует быть настраиваемыми в идеальном случае без необходимости разрабатывать дополнительные модули YANG.
- Схема ядра маршрутизации должна допускать сложные реализации, включающие множество RIB и протоколов плоскости управления, а также управляемое распространение маршрутной информации.
- Поскольку производитель устройств захотят сопоставить модели данных, основанные на базовой схеме, с фирменными моделями данных и интерфейсами настройки, схеме следует быть достаточно гибкой, чтобы облегчить такое сопоставление и воспринимать модели данных с иной логикой.

## 4. Устройство ядра модели данных маршрутизации

Модель данных ядра маршрутизации состоит из трёх модулей YANG и одного submodule. Модуль ietf-routing определяет базовые компоненты системы маршрутизации. Два других модуля - ietf-ipv4-unicast-routing и ietf-ipv6-unicast-routing - дополняют ietf-routing узлами данных для индивидуальной маршрутизации IPv4 и IPv6, соответственно. Модуль ietf-ipv6-unicast-routing имеет submodule ietf-ipv6-router-advertisements, дополняющий модули ietf-interfaces [RFC8343] и ietf-ip [RFC8344] переменными конфигурации для анонсов IPv6 RA, требуемыми [RFC4861].

```

+--rw routing
  +--rw router-id?                yang:dotted-quad
  +--ro interfaces
  | +--ro interface* if:interface-ref
  +--rw control-plane-protocols
  | +--rw control-plane-protocol* [type name]
  |   +--rw type                identityref
  |   +--rw name                string
  |   +--rw description?       string
  |   +--rw static-routes
  |     +--rw v4ur:ipv4
  |     | ...
  |     +--rw v6ur:ipv6
  |     | ...
  +--rw ribs
  +--rw rib* [name]
  | +--rw name                  string
  | +--rw address-family?      identityref
  | +--ro default-rib?         boolean {multiple-ribs}?
  | +--ro routes
  | | +--ro route*
  | | | ...
  | +--x active-route
  | | +--w input
  | | | +--w v4ur:destination-address? inet:ipv4-address
  | | | +--w v6ur:destination-address? inet:ipv6-address
  | | +--ro output
  | | | ...
  | +--rw description?        string

```

Рисунок 1. Иерархия данных.

На рисунке 1 дано краткое представление иерархий, а полные деревья данных приведены в Приложении А. Как можно видеть на рисунке 1, модель данных ядра маршрутизации вводит несколько базовых элементов схемы маршрутизации

- маршруты, RIB со списками маршрутов, протоколы плоскости управления. Эти элементы более подробно описаны в разделе 5.

## 4.1. Контролируемые системой и пользователем записи списков

Модель данных ядра маршрутизации определяет несколько списков в дереве схемы (таких как `rib`), которые в корректно работающем устройстве должны включать как минимум одну запись, а клиент может добавлять записи. В таком списке сервер создаёт требуемый элемент как контролируруемую системой запись рабочего состояния, т. е. доступных лишь для чтения списков в контейнере `routing`. Например, в Приложении D список `/routing/ribs/rib` имеет 2 контролируемых системой записи - `ipv4-master` и `ipv6-master`.

Дополнительные записи, контролируемые пользователем, могут быть созданы в конфигурации клиентом, например, с помощью протокола NETCONF (Network Configuration Protocol). Если сервер воспринимает контролируемую пользователем запись, она тоже появляется в рабочей версии списка.

Соответствующие записи обеих версий списка (предполагаемая конфигурация и рабочее состояние) [RFC8342] имеют одно значение ключа списка.

Клиент может также представить дополнительную конфигурацию контролируемых системой записей. Для этого клиент создаёт в конфигурации новую запись с желаемым содержимым. Для привязки этой записи к соответствующей записи рабочего состояния ключ записи в конфигурации должен совпадать с ключом записи рабочего состояния.

Удаление контролируемой пользователем записи из предполагаемой конфигурации ведёт к удалению соответствующей записи из списка рабочего состояния. Если клиент удаляет контролируемую системой запись из предполагаемой конфигурации, удаляется лишь дополнительная конфигурация, заданная удаляемой записью, а соответствующая запись рабочего состояния остаётся.

## 5. Базовые элементы

В этом разделе описаны важные компоненты модели данных ядра маршрутизации.

### 5.1. Маршруты

Маршруты являются базовыми элементами системы маршрутизации. Модель данных ядра маршрутизации определяет лишь минимальный набор атрибутов маршрута, перечисленных ниже.

#### *destination-prefix - префикс адресата*

Адресный префикс, задающий набор адресатов, для которых может применяться маршрут. Обязательный атрибут.

#### *route-preference - предпочтительность маршрута*

Целое число, называемое также административной дистанцией (*administrative distance*), которое служит для выбора предпочтительного маршрута из числа ведущих к префиксу. Меньшее значение указывает более предпочтительный маршрут.

#### *next-hop - следующий узел пересылки*

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

Маршруты в первую очередь являются состояниями системы и отображаются в записях таблиц RIB (параграф 5.2), но могут присутствовать и в данных конфигурации, например, как заданные вручную статические маршруты. В последнем случае настраиваемые атрибуты маршрутов обычно являются подмножеством атрибутов, определённых для RIB.

### 5.2. Базы RIB

Каждая реализация модели ядра маршрутизации поддерживает хотя бы одну базу RIB, представляющую собой список маршрутов, дополненных административными данными. Каждая база RIB содержит маршруты лишь для одного семейства адресов, представляемого идентификатором, выведенным из базового отождествления `rt:address-family`.

В модели данных ядра маршрутизации базы RIB представляются записями в списке `/routing/ribs/rib` рабочего состояния. Содержимое RIB контролируется и управляется операциями протокола плоскости управления, которые могут приводить к добавлению, удалению и изменению маршрутов. Включаются также операции псевдопротоколов `static` и/или `direct`, описанные в параграфе 5.3.1.

Для каждого поддерживаемого семейства адресов одна база RIB **должна** быть помечена как принятая по умолчанию (`default RIB`), куда протоколы плоскости управления помещают маршруты по умолчанию.

Простые реализации маршрутизаторов, не анонсирующие свойство `multiple-ribs`, обычно создают одну контролируемую системой базу RIB для семейства адресов и помечают её как RIB. Более сложные реализации, анонсирующие `multiple-ribs`, поддерживают несколько RIB на семейство адресов, которые могут служить для маршрутизации на основе правил и других целей.

Для списка `rib` определено указанное ниже действие (см. параграф 7.15 в [RFC7950]).

- `active-route` возвращает активный маршрут RIB для адреса получателя, заданного входным параметром действия.

### 5.3. Протоколы плоскости управления

Модель данных ядра маршрутизации предоставляет открытую структуру для определения множества экземпляров протоколов плоскости управления, например для протоколов маршрутизации L3. Каждому экземпляру протокола плоскости управления **должен** назначаться тип, который выводится из базового отождествления `rt:control-plane-protocol`. Модель данных ядра маршрутизации определяет два отождествления для псевдопротоколов - `direct` и `static` (параграф 5.3.1).

**Можно** настроить несколько экземпляров протоколов плоскости управления одного типа.

### 5.3.1. Псевдопротоколы маршрутизации

Модель данных ядра маршрутизации определяет два особых типа протоколов маршрутизации - `direct` и `static`. Оба являются псевдопротоколами, что означает привязку к локальному устройству и отсутствие обмена маршрутными данными со смежными маршрутизаторами.

Каждая реализация модели данных ядра маршрутизации должна обеспечивать один экземпляр псевдопротокола типа `direct`, являющийся источником прямых маршрутов для всех настроенных семейств адресов. Прямые маршруты обычно предоставляет ядро операционной системы на основе адресов сетевых интерфейсов (параграф 6.2).

Псевдопротокол типа `static` позволяет задать маршруты вручную. Его можно (но не требуется) настроить в нескольких экземплярах, хотя в типичных конфигурациях применяется лишь один экземпляр.

### 5.3.2. Определение новых протоколов плоскости управления

Предполагается, что будут разработаны модули YANG с моделями данных для дополнительных типов протоколов плоскости управления. Такие модули будут определять узлы данных для конкретных протоколов и интегрировать их с моделью ядра маршрутизации, как описано ниже.

- **Должны** определяться новые отождествления для протоколов плоскости управления, а в качестве базового для них **должен** устанавливаться идентификатор `rt:control-plane-protocol` или производный от него.
- **Могут** определяться дополнительные атрибуты маршрутов, предпочтительно в одном месте, путём задания группировок YANG. Новые атрибуты должны задаваться путём дополнения определений узлов `/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route`, а также могут помещаться в другие места дерева схемы.
- Узлы данных для новых протоколов могут определяться дополнением узлов данных `control-plane-protocol` в ветви `/routing`.

Дополненные узлы данных для нового протокола **следует** делать условными с помощью оператора `when` и они будут действительны лишь при совпадении `rt:type` или `rt:source-protocol` (или производного от него) с отождествлением нового протокола.

**Рекомендуется** также инкапсулировать связанные с протоколом узлы данных в контейнер присутствия с подходящим именем. Такой контейнер может содержать обязательные узлы данных, которые в ином случае будут запрещены на верхнем уровне дополнения.

Указанные выше шаги реализованы в примере модуля YANG для протокола RIP (Routing Information Protocol), приведённом в Приложении С.

## 5.4. Параметры анонсов маршрутизаторов IPv6

Модуль YANG `ietf-ipv6-router-advertisements` (параграф 9.1) с submodule `ietf-ipv6-unicast-routing` дополняет дерево схемы интерфейсов IPv6 определениями приведённых ниже переменных, требуемых параграфом 6.2.1 [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 префиксов для анонсирования).

С каждым префиксом в списке связаны приведённые ниже параметры.

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

Примечания.

1. Флаг `IsRouter`, требуемый [RFC4861], реализован в модуле `ietf-ip` [RFC8344] (лист `ip:forwarding`).
2. Спецификация обнаружения соседей (ND) [RFC4861] позволяет реализациям выбирать для параметров `valid-lifetime` и `preferred-lifetime` сохранение в последующих анонсах или их уменьшение в реальном масштабе времени. Однако второй вариант проблематичен, поскольку значения могут быть сброшены в (большие) настроенные значения после перезагрузки конфигурации. Кроме того, не известно ни одной реализации, использующей декрементирование. Поэтому в submodule `ietf-ipv6-router-advertisements` применяется первый вариант с постоянными значениями.

## 6. Взаимодействия с другими модулями YANG

Семантика модели данных ядра маршрутизации зависит также от нескольких параметров конфигурации, определённых в других модулях YANG.

### 6.1. Модуль ietf-interfaces

Приведённый ниже логический переключатель определён в модуле YANG ietf-interfaces [RFC8343].

#### */if:interfaces/if:interface/if:enabled*

При установке значения false для интерфейса сетевого уровня на этом интерфейсе **должны** отключаться все функции маршрутизации и пересылки пакетов.

### 6.2. Модуль ietf-ip

Приведённые ниже логические переключатели определены в модуле YANG ietf-ip [RFC8344].

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

При установке значения false для интерфейса сетевого уровня на этом интерфейсе **должны** отключаться все функции маршрутизации и пересылки пакетов IPv4.

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

При установке значения false для интерфейса сетевого уровня пересылка дейтаграмм IPv4 через этот интерфейс должна быть отключена. Однако интерфейс **может** участвовать в других функциях маршрутизации IPv4, например, в протоколах маршрутизации.

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

При установке значения false для интерфейса сетевого уровня на этом интерфейсе **должны** отключаться все функции маршрутизации и пересылки пакетов IPv6.

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

При установке значения false для интерфейса сетевого уровня пересылка дейтаграмм IPv6 через этот интерфейс **должна** быть отключена. Однако интерфейс **может** участвовать в других функциях маршрутизации IPv6, например, в протоколах маршрутизации.

В дополнение к этому модуль ietf-ip позволяет настраивать адреса и префиксы или маски IPv4 и IPv6 на интерфейсах сетевого уровня. Настройка этих параметров на включённом интерфейсе **должна** приводить к немедленному созданию соответствующего прямого маршрута. Префикс адресата для этого маршрута устанавливается в соответствии с настроенным адресом IP и префиксом или маской сети, а интерфейс устанавливается как выходной для маршрута.

## 7. Модуль управления маршрутизацией

```
<CODE BEGINS> file "ietf-routing@2018-03-13.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";
    description
      "Версия модуля ietf-interfaces, совместимая с архитектурой
      хранилища данных сетевого управления (NMDA), нужна для работы";
  }

  organization
    "IETF NETMOD (Network Modeling) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:rtgw@ietf.org>

    Editor: Ladislav Lhotka
           <mailto:lhotka@nic.cz>
           Acee Lindem
           <mailto:acee@cisco.com>
           Yingzhen Qu
           <mailto:yingzhen.qu@huawei.com>";

  description
    "Этот модуль YANG определяет важные компоненты для управления
    подсистемой маршрутизации. Модель полностью соответствует
    архитектуре хранилища данных управления сетью (NMDA).

    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 8349, где правовые
    аспекты приведены более полно.";
```

```
revision 2018-03-13 {
  description
    "Версия для архитектуры хранилища данных управления сетью (NMDA).";
  reference
    "RFC 8349: A YANG Data Model for Routing Management
      (NMDA Version)";
}

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

/* Свойства */
feature multiple-ribs {
  description
    "Это свойство показывает, что сервер поддерживает определённые
    пользователем базы RIB.

    Серверам, не анонсирующим это свойство, СЛЕДУЕТ предоставлять
    в точности 1 контролируруемую системой RIB на поддерживаемое
    семейство адресов и делать её default RIB. Эта RIB появляется
    как запись в списке /routing/ribs/rib.";
}

feature router-id {
  description
    "Показывает, что сервер поддерживает явный 32-битовый идентификатор
    router ID, используемый некоторыми протоколами маршрутизации.

    Серверы, не поддерживающие это свойство, устанавливают router ID
    алгоритмически, обычно выбирая один из настроенных адресов IPv4.
    Однако алгоритм зависит от реализации.";
}

/* Отождествления */

identity address-family {
  description
    "Базовое отождествление из которого выводятся отождествления
    семейств 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
    "Базовое отождествление из которого выводятся отождествления
    протоколов плоскости управления.";
}

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

identity direct {
  base routing-protocol;
  description
    "Псевдопротокол маршрутизации, обеспечивающий маршруты в
    непосредственно подключённые сети.";
}

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

/* Определения типов */

typedef route-preference {
```

```
type uint32;
description
  "Применяется для предпочтений маршрута.";
}

/* Группировки */

grouping address-family {
  description
    "Идентификация листа семейства адресов.";
  leaf address-family {
    type identityref {
      base address-family;
    }
    mandatory true;
    description
      "Семейство адресов.";
  }
}

grouping router-id {
  description
    "Обеспечивает значение router ID.";
  leaf router-id {
    type yang:dotted-quad;
    description
      "32-битовое число в формате с разделением байтов точками,
      служащее некоторым протоколам маршрутизации для идентификации
      маршрутизатора.";
    reference
      "RFC 2328: OSPF Version 2";
  }
}

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

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

      Предполагается добавление других случаев с помощью операторов
      augment в других модулях.";
    case simple-next-hop {
      description
        "Представляет простой вариант next hop, состоящий из адреса
        и/или выходного интерфейса.

        Модули для семейств адресов ДОЛЖНЫ дополнять этот вариант
        листом с адресом next-hop из данного 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 нескольких next hop.";
    list next-hop {
      key "index";
      description
        "Запись в списке next-hop.

        Модули для семейств адресов ДОЛЖНЫ дополнять этот
        список листом, с адресом next-hop своего семейства.";
      leaf index {
        type string;
        description
          "Заданный пользователем идентификатор, применяемый
          для однозначного указания записи next-hop в списке.
          Значение этого индекса не имеет семантического смысла
          кроме ссылки на запись списка.";
      }
      leaf outgoing-interface {
        type if:interface-ref;
        description
          "Имя выходного интерфейса.";
      }
    }
  }
}
}
}
}

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

      Предполагается добавление других опций с помощью операторов
      augment из других модулей, например, для рекурсии next hop.";
    case simple-next-hop {
      description
        "Это представляет единственную запись next hop с адресом
        next-hop и/или выходным интерфейсом.

        Модули для семейств адресов ДОЛЖНЫ дополнять этот список
        адресом next-hop из своего 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 нескольких next hop.";
        list next-hop {
          description
            "Запись в списке next-hop.

            Модули для семейств адресов ДОЛЖНЫ дополнять этот
            список адресом next-hop из своего семейства.";
          leaf outgoing-interface {
            type if:interface-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 предпочтительней
    других маршрутов в данному префиксу в той же базе RIB.";
}
leaf last-updated {
  type yang:date-and-time;
  description
    "Время последнего изменения маршрута. Если маршрут никогда не
    менялся, это будет время вставки маршрута в RIB.";
}
}
/* Узлы данных */

container routing {
  description
    "Параметры конфигурации подсистемы маршрутизации.";
  uses router-id {
    if-feature "router-id";
    description
      "Поддержка глобального router ID. Протоколы маршрутизации,
      применяющие router ID, могут использовать или переопределить
      этот параметр.";
  }
  container interfaces {
    config false;
    description
      "Интерфейсы сетевого уровня, применяемые для маршрутизации.";
    leaf-list interface {
      type if:interface-ref;
      description
        "Каждая запись указывает имя настроенного интерфейса
        сетевого уровня.";
    }
  }
  container control-plane-protocols {
    description
      "Поддержка экземпляров протокола плоскости управления.";
    list control-plane-protocol {
      key "type name";
      description
        "Каждая запись содержит экземпляр протокола плоскости управления.";
      leaf type {
        type identityref {
          base control-plane-protocol;
        }
        description
          "Тип протокола плоскости управления - отождествления,
          выведенное из базового отождествления control-plane-protocol.";
      }
      leaf name {
        type string;
        description
          "Произвольное имя экземпляра протокола плоскости управления.";
      }
      leaf description {
        type string;
        description
          "Текстовое описание экземпляра протокола плоскости управления.";
      }
      container static-routes {
        when "derived-from-or-self(..type, 'rt:static')" {
          description
            "Этот контейнер действителен лишь для протокола static.";
        }
        description
          "Поддержка псевдопротокола static.

          Модули для семейств адресов дополняют этот узел
          своими списками маршрутов.";
      }
    }
  }
  container ribs {
    description
      "Поддержка для баз RIB.";
    list rib {

```

```

key "name";
description
  "Каждая запись содержит конфигурацию для базы RIB,
  указанной ключом name.

  Записи с тем же ключом, что и у управляемой системой
  записи в списке /routing/ribs/rib, применяются для
  настройки параметров этой записи. Остальные определяют
  дополнительные RIB, управляемые пользователем.";
leaf name {
  type string;
  description
    "Имя RIB.

    Для управляемых системой записей значение этого листа
    должно совпадать с именем соответствующей записи в
    рабочем состоянии.

    Для контролируемых пользователем записей можно задавать
    любое имя.";
}
uses address-family {
  description
    "Семейство адресов управляемой системой RIB.";
}

leaf default-rib {
  if-feature "multiple-ribs";
  type boolean;
  default "true";
  config false;
  description
    "Флаг, имеющий значение true лишь для принятой по
    умолчанию RIB в данном семействе адресов.

    По умолчанию протоколы плоскости управления
    помещают свои маршруты в принятые по умолчанию RIB.";
}
container routes {
  config false;
  description
    "Текущее содержимое RIB.";
  list route {
    description
      "Маршрутная запись RIB. Этот узел данных ДОЛЖЕН
      дополняться сведениями для маршрутов каждого
      семейства адресов.";
    leaf route-preference {
      type route-preference;
      description
        "Этот атрибут маршрута, называемый также administrative
        distance, позволяет выбрать предпочтительный маршрут
        из числа ведущих к одному префиксу. Меньшее значение
        указывает более предпочтительный маршрут.";
    }
    container next-hop {
      description
        "Атрибут next-hop для маршрута.";
      uses next-hop-state-content;
    }
    uses route-metadata;
  }
}
action active-route {
  description
    "Возвращает активный маршрут RIB, используемый для
    адреса получателя.

    Модули для семейств адресов ДОЛЖНЫ дополнять входные
    параметры листом destination-address.";
  output {
    container route {
      description
        "Активный маршрут RIB для заданного адресата.

        Если в RIB нет маршрута к адресату, вывод будет пустым.

        Модули для семейств адресов ДОЛЖНЫ дополнять этот
        контейнер соответствующим содержимым маршрута.";
      container next-hop {
        description
          "Атрибут next-hop для маршрута .";
        uses next-hop-state-content;
      }
      uses route-metadata;
    }
  }
}

```

```
    }
  }
  leaf description {
    type string;
    description
      "Текстовое описание RIB.";
  }
}
}
/*
 * Перечисленные ниже узлы данных признаны устаревшими и отменены
 * архитектурой хранилищ данных управления сетью NMDA, описанной
 * в RFC 8342.
 */
container routing-state {
  config false;
  status obsolete;
  description
    "Данные состояния для подсистемы маршрутизации.";
  uses router-id {
    status obsolete;
    description
      "Глобальный идентификатор router ID.

      Может настраиваться или автоматически выбираться реализацией.";
  }
}
container interfaces {
  status obsolete;
  description
    "Интерфейсы сетевого уровня, применяемые для маршрутизации.";
  leaf-list interface {
    type if:interface-state-ref;
    status obsolete;
    description
      "Каждая запись указывает имя интерфейса сетевого уровня.";
  }
}
container control-plane-protocols {
  status obsolete;
  description
    "Контейнер для списка экземпляров протоколов маршрутизации.";
  list control-plane-protocol {
    key "type name";
    status obsolete;
    description
      "Данные состояния для экземпляра протокола плоскости управления.

      Реализация ДОЛЖНА предоставлять в точности 1 управляемый
      системой экземпляр псевдопротокола direct. Конфигурация МОЖЕТ
      создавать другие экземпляры протоколов плоскости управления.";
    leaf type {
      type identityref {
        base control-plane-protocol;
      }
      status obsolete;
      description
        "Тип протокола плоскости управления.";
    }
    leaf name {
      type string;
      status obsolete;
      description
        "Имя экземпляра протокола плоскости управления.

        Для управляемых системой экземпляров это имя является
        постоянным, т. е. его НЕ СЛЕДУЕТ менять при перезагрузке.";
    }
  }
}
}
container ribs {
  status obsolete;
  description
    "Контейнер для таблиц RIB.";
  list rib {
    key "name";
    min-elements 1;
    status obsolete;
    description
      "Каждая запись представляет RIB, указанную ключом name. Все
      маршруты в RIB ДОЛЖНЫ относиться к одному семейству адресов.

      Реализации СЛЕДУЕТ обеспечивать одну управляемую системой
      базу RIB для каждого поддерживаемого семейства адресов.";
    leaf name {
```

```

    type string;
    status obsolete;
    description
        "Имя RIB.";
}
uses address-family {
    status obsolete;
    description
        "Семейство адресов RIB.";
}
leaf default-rib {
    if-feature "multiple-ribs";
    type boolean;
    default "true";
    status obsolete;
    description
        "Флаг, имеющий значение true лишь в случае, когда RIB
        является default RIB для данного семейства адресов.

        По умолчанию протоколы плоскости управления помещают
        свои маршруты в default RIB.";
}
container routes {
    status obsolete;
    description
        "Текущее содержимое RIB.";
    list route {
        status obsolete;
        description
            "Запись маршрута в RIB. Этот узел данных ДОЛЖЕН
            дополняться данными для маршрутов каждого семейства
            адресов.";
        leaf route-preference {
            type route-preference;
            status obsolete;
            description
                "Этот атрибут маршрута, называемый также administrative
                distance, позволяет выбрать предпочтительный маршрут
                из числа ведущих к одному префиксу. Меньшее значение
                указывает более предпочтительный маршрут.";
        }
        container next-hop {
            status obsolete;
            description
                "Атрибут next-hop для маршрута.";
            uses next-hop-state-content {
                status obsolete;
                description
                    "Рабочее состояние атрибута next-hop для маршрута.";
            }
        }
        uses route-metadata {
            status obsolete;
            description
                "Метаданные маршрута.";
        }
    }
}
action active-route {
    status obsolete;
    description
        "Возвращает активный маршрут RIB, используемый для
        адреса получателя.

        Модули для семейств адресов ДОЛЖНЫ дополнять входные
        параметры листом destination-address.";
    output {
        container route {
            status obsolete;
            description
                "Активный маршрут RIB для указанного адресата.

                При отсутствии в RIB маршрута к адресату ничего
                не выводится.

                Модули для семейств адресов ДОЛЖНЫ дополнять этот
                контейнер соответствующим содержимым маршрутов.";
        }
        container next-hop {
            status obsolete;
            description
                "Атрибут next-hop для маршрута.";
            uses next-hop-state-content {
                status obsolete;
                description
                    "Данные состояния активного маршрута.";
            }
        }
    }
}

```



```

description
  "Отождествление семейства индивидуальных адресов IPv4.";
}

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

augment "/rt:routing/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
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  description
    "Дополнение к simple-next-hop для маршрутов IPv4 unicast.";
  leaf next-hop-address {
    type inet:ipv4-address;
    description
      "Адрес IPv4 для следующего маршрутизатора (next hop).";
  }
}

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
  + "rt:next-hop-list/rt:next-hop" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  description
    "Дополнение к next-hop-list для маршрутов IPv4 unicast.";
  leaf address {
    type inet:ipv4-address;
    description
      "Адрес IPv4 для следующего маршрутизатора (next hop).";
  }
}

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

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
  + "rt:output/rt:route" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  description
    "Добавляет префикс адресата к отклику действия active-route.";
  leaf destination-prefix {
    type inet:ipv4-prefix;
    description
      "Префикс адресата IPv4.";
  }
}

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
  + "rt:output/rt:route/rt:next-hop/rt:next-hop-options/"

```



```

}
}

/*
 * Перечисленные ниже узлы данных признаны устаревшими и отменены
 * архитектурой хранилищ данных управления сетью NMDA, описанной
 * в RFC 8342.
 */
augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v4ur:ipv4-unicast')" {
    description
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  status obsolete;
  description
    "Этот лист дополняет индивидуальный маршрут IPv4.";
  leaf destination-prefix {
    type inet:ipv4-prefix;
    status obsolete;
    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
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  status obsolete;
  description
    "Дополняет вариант simple-next-hop в маршрутах IPv4 unicast.";
  leaf next-hop-address {
    type inet:ipv4-address;
    status obsolete;
    description
      "Адрес IPv4 для следующего маршрутизатора (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
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  status obsolete;
  description
    "Дополняет вариант next-hop-list в маршрутах IPv4 unicast.";
  leaf address {
    type inet:ipv4-address;
    status obsolete;
    description
      "Адрес IPv4 для следующего маршрутизатора (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
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  status obsolete;
  description
    "Добавляет входной параметр для действия active-route.";
  leaf destination-address {
    type inet:ipv4-address;
    status obsolete;
    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
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  status obsolete;
  description
    "Добавляет префикс адресата к действию active-route.";
}

```

```

leaf destination-prefix {
  type inet:ipv4-prefix;
  status obsolete;
  description
    "Префикс адресата 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
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  status obsolete;
  description
    "Дополняет вариант simple-next-hop в отклике на действие
    active-route.";
  leaf next-hop-address {
    type inet:ipv4-address;
    status obsolete;
    description
      "Адрес IPv4 для следующего маршрутизатора (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
      "Это дополнение действительно лишь для IPv4 unicast.";
  }
  status obsolete;
  description
    "Дополняет вариант next-hop-list в отклике на действие
    active-route.";
  leaf next-hop-address {
    type inet:ipv4-address;
    status obsolete;
    description
      "Адрес IPv4 для следующего маршрутизатора (next hop).";
  }
}
}
}
}

```

<CODE ENDS>

## 9. Модуль управления маршрутизацией IPv6

<CODE BEGINS> file "ietf-ipv6-unicast-routing@2018-03-13.yang"

```

module ietf-ipv6-unicast-routing {
  yang-version "1.1";
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing";
  prefix "v6ur";

  import ietf-routing {
    prefix "rt";
    description
      "Для работы требуется версия модуля ietf-routing, совместимая
      с архитектурой хранилищ данных управления сетью (NMDA).";
  }

  import ietf-inet-types {
    prefix "inet";
    description
      "Для работы требуется версия модуля ietf-interfaces, совместимая
      с архитектурой хранилищ данных управления сетью (NMDA).";
  }

  include ietf-ipv6-router-advertisements {
    revision-date 2018-03-13;
  }

  organization
    "IETF NETMOD (Network Modeling) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:rtgw@ietf.org>

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

```

```
<mailto:acee@cisco.com>
Yingzhen Qu
<mailto:yingzhen.qu@huawei.com>;
```

```
description
"Этот модуль YANG дополняет модуль ietf-routing базовыми параметрами
для индивидуальной маршрутизации IPv6. Модель полностью соответствует
архитектуре хранилищ данных управления сетью (NMDA).

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 8349, где правовые
аспекты приведены более полно."

revision 2018-03-13 {
  description
  "Версия для архитектуры хранилища данных управления сетью (NMDA).";
  reference
  "RFC 8349: A YANG Data Model for Routing Management
  (NMDA Version)";
}

/* Отождествления */

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

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

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

augment "/rt:routing/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
    "Это дополнение действительно лишь для IPv6 unicast.";
  }
  description
  "Дополняет случай simple-next-hop в индивидуальном маршруте IPv6.";
  leaf next-hop-address {
    type inet:ipv6-address;
    description
    "Адрес IPv6 следующего интервала.";
  }
}

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/"
+ "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
+ "rt:next-hop-list/rt:next-hop" {
  when "derived-from-or-self(..../rt:address-family, "
  + "'v6ur:ipv6-unicast')" {
    description
    "Это дополнение действительно лишь для IPv6 unicast.";
  }
  description
  "Этот лист дополняет случай next-hop-list индивидуальных
маршрутов IPv6.";
```

```

leaf address {
  type inet:ipv6-address;
  description
    "Адрес IPv6 следующего интервала.";
}
}

augment
"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input" {
  when "derived-from-or-self(..../rt:address-family, "
    + "'v6ur:ipv6-unicast')" {
    description
      "Это дополнение действительно лишь для IPv6 unicast RIB.";
  }
  description
    "Добавляет входной параметр действия active-route.";
  leaf destination-address {
    type inet:ipv6-address;
    description
      "Адрес получателя IPv6.";
  }
}

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

augment "/rt:routing/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
      "Это дополнение действительно лишь для IPv6 unicast.";
  }
  description
    "Дополняет случай simple-next-hop в отклике на действие active-route.";
  leaf next-hop-address {
    type inet:ipv6-address;
    description
      "Адрес IPv6 следующего интервала.";
  }
}

augment "/rt:routing/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
      "Это дополнение действительно лишь для IPv6 unicast.";
  }
  description
    "Дополняет случай next-hop-list в отклике на действие active-route.";
  leaf next-hop-address {
    type inet:ipv6-address;
    description
      "Адрес IPv6 следующего интервала.";
  }
}

/* Дополнения узлов данных */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rt:static-routes" {
  description
    "Определяет псевдопротокол static данными, относящимися к IPv6 unicast.";
  container ipv6 {
    description
      "Поддержка экземпляра псевдопротокола static включает список маршрутов.";
    list route {
      key "destination-prefix";
      description
        "Список статических маршрутов.";
    }
  }
}

```



```

status obsolete;
description
  "Дополняет случай next-hop-list в индивидуальном маршруте IPv6.";
leaf address {
  type inet:ipv6-address;
  status obsolete;
  description
    "Адрес IPv6 следующего интервала.";
}
}
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
      "Это дополнение действительно лишь для IPv6 unicast.";
  }
  status obsolete;
  description
    "Добавляет входной параметр действия active-route.";
  leaf destination-address {
    type inet:ipv6-address;
    status obsolete;
    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
      "Это дополнение действительно лишь для IPv6 unicast.";
  }
  status obsolete;
  description
    "Добавляет префикс получателя в отклик на действие active-route.";
  leaf destination-prefix {
    type inet:ipv6-prefix;
    status obsolete;
    description
      "Префикс получателя 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
      "Это дополнение действительно лишь для IPv6 unicast.";
  }
  status obsolete;
  description
    "Добавляет случай simple-next-hop в отклике на active-route.";
  leaf next-hop-address {
    type inet:ipv6-address;
    status obsolete;
    description
      "Адрес IPv6 следующего интервала.";
  }
}
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
      "Это дополнение действительно лишь для IPv6 unicast.";
  }
  status obsolete;
  description
    "Добавляет случай next-hop-list в отклике на active-route.";
  leaf next-hop-address {
    type inet:ipv6-address;
    status obsolete;
    description
      "Адрес IPv6 следующего интервала.";
  }
}
}
}
}
<CODE ENDS>

```

## 9.1. Субмодуль IPv6 RA

```

<CODE BEGINS> file "ietf-ipv6-router-advertisements@2018-03-13.yang"

submodule ietf-ipv6-router-advertisements {
  yang-version "1.1";

  belongs-to ietf-ipv6-unicast-routing {
    prefix "v6ur";
  }

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

  import ietf-interfaces {
    prefix "if";
    description
      "Нужна версия модуля ietf-interfaces, совместимая с архитектурой
      сетевых хранилищ данных (NMDA).";
  }

  import ietf-ip {
    prefix "ip";
    description
      "Нужна версия модуля ietf-ip, совместимая с архитектурой
      сетевых хранилищ данных (NMDA).";
  }

  organization
    "IETF NETMOD (Network Modeling) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:rtgwg@ietf.org>

    Editor: Ladislav Lhotka
            <mailto:lhotka@nic.cz>
            Acee Lindem
            <mailto:acee@cisco.com>
            Yingzhen Qu
            <mailto:yingzhen.qu@huawei.com>";

  description
    "Этот модуль YANG дополняет модуль ietf-ip параметрами для анонсов
    маршрутизаторов IPv6 (RA). Модель полностью соответствует архитектуре
    сетевых хранилищ данных (NMDA).

    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 8349, где правовые
    аспекты приведены более полно.";

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

  revision 2018-03-13 {
    description
      "Выпуск для архитектуры сетевых хранилищ данных (NMDA).";
    reference
      "RFC 8349: A YANG Data Model for Routing Management
      (NMDA Version)";
  }

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

  augment "/if:interfaces/if:interface/ip:ipv6" {
    description
      "Добавляет в конфигурацию интерфейсов параметры IPv6 RA.";
    container ipv6-router-advertisements {
      description
        "Поддержка анонсов маршрутизаторов IPv6.";
      leaf send-advertisements {
        type boolean;
      }
    }
  }
}

```

```

default "false";
description
  "Флаг, указывающий передаёт ли маршрутизатор периодические
  сообщения RA и отвечает ли на сообщения RS.";
reference
  "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
  - AdvSendAdvertisements";
}
leaf max-rtr-adv-interval {
  type uint16 {
    range "4..65535";
  }
  units "seconds";
  default "600";
  description
    "Максимальное время, разрешённое между отправкой групповых
    незапрошенных анонсов RA 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
      "НЕДОПУСТИМО значение больше 75% max-rtr-adv-interval.";
  }
  description
    "Минимальное время, разрешённое между отправкой групповых
    незапрошенных анонсов RA с интерфейса.

    Если этот лист не задан, используется принятое по
    умолчанию значение:

    - если max-rtr-adv-interval >= 9 секунд, принято значение
      0,33 * max-rtr-adv-interval;

    - иначе используется значение 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 анонсов RA.";
  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
    анонсов RA.";
  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.
    Значение 0 говорит, что опции MTU не передаются.";
  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 анонсов RA,
    передаваемых маршрутизатором. Значение 0 говорит о
    незаданном (этим маршрутизатором) времени.";
  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 анонсов RA,
      передаваемых маршрутизатором. Значение 0 говорит о
      незаданном (этим маршрутизатором) времени.";
    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 анонсов RA,
      передаваемых маршрутизатором. Значение 0 говорит о
      незаданном (этим маршрутизатором) счётчике.

      Если этот параметр не задан, устройству СЛЕДУЕТ применять
      заданное IANA значение принятого по умолчанию параметра IPv4
      TTL, которое действовало на момент реализации.";
    reference
      "RFC 3232: Assigned Numbers: RFC 1700 is Replaced by
        an On-line Database
      RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
        - AdvCurHopLimit
      IANA: IP Parameters
        (https://www.iana.org/assignments/ip-parameters)";
  }
  leaf default-lifetime {
    type uint16 {
      range "0..65535";
    }
    units "seconds";
    description
      "Значение, помещаемое в поле Router Lifetime анонсов RA,
      передаваемых с интерфейса (в секундах). Лист ДОЛЖЕН иметь
      значение 0 или число из интервала max-rtr-adv-interval -
      9000 (секунд). Значение 0 указывает, что маршрутизатор не
      используется по умолчанию. Ограничения могут быть
      переопределены документами, описывающими работу IPv6 на
      разных канальных уровнях.

      Если параметр не задан, устройству СЛЕДУЕТ применять
      значение 3 * max-rtr-adv-interval.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
        - AdvDefaultLifetime";
  }
  container prefix-list {
    description
      "Поддержка размещения префиксов в опциях Prefix
      Information анонсов RA, передаваемых через интерфейс.

      Префиксы, анонсируемые по умолчанию, но не имеющие
      записей в дочернем списке prefix. Анонсируются с
      принятыми по умолчанию значениями 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-spec";
      description
        "Поддержка записей для анонсируемых префиксов.";
      leaf prefix-spec {
        type inet:ipv6-prefix;
        description
          "Адресный префикс IPv6.";
      }
    }
    choice control-adv-prefixes {
      default "advertise";
      description
        "(1) префикс, явно удаляемый из числа анонсируемых, или
        (2) параметры, с которыми анонсируется префикс (по
        умолчанию).";
      leaf no-advertise {
        type empty;
        description
          "Префикс, который не будет анонсироваться."
      }
    }
  }

```



```
"Максимальное разрешённое время между отправкой групповых
незапрошенных анонсов RA через интерфейс.";
}
leaf min-rtr-adv-interval {
  type uint16 {
    range "3..1350";
  }
  units "seconds";
  status obsolete;
  description
    "Минимальное разрешённое время между отправкой групповых
незапрошенных анонсов RA через интерфейс.";
}
leaf managed-flag {
  type boolean;
  status obsolete;
  description
    "Значение, помещаемое в поле флага Managed address
configuration в анонсах RA.";
}
leaf other-config-flag {
  type boolean;
  status obsolete;
  description
    "Значение, помещаемое в поле флага Other configuration
в анонсах RA.";
}
leaf link-mtu {
  type uint32;
  status obsolete;
  description
    "Значение, помещаемое в опции MTU, передаваемые маршрутизатором.
Значение 0 говорит, что опции MTU не передаются.";
}
leaf reachable-time {
  type uint32 {
    range "0..3600000";
  }
  units "milliseconds";
  status obsolete;
  description
    "Значение, помещаемое в поле Reachable Time анонсов RA,
передаваемых маршрутизатором. Значение 0 указывает, что
поле не задано (этим маршрутизатором).";
}
leaf retrans-timer {
  type uint32;
  units "milliseconds";
  status obsolete;
  description
    "Значение, помещаемое в поле Retrans Timer анонсов RA,
передаваемых маршрутизатором. Значение 0 указывает, что
поле не задано (этим маршрутизатором).";
}
leaf cur-hop-limit {
  type uint8;
  status obsolete;
  description
    " Значение, помещаемое в поле Cur Hop Limit анонсов RA,
передаваемых маршрутизатором. Значение 0 указывает, что
поле не задано (этим маршрутизатором).";
}
leaf default-lifetime {
  type uint16 {
    range "0..9000";
  }
  units "seconds";
  status obsolete;
  description
    " Значение, помещаемое в поле Router Lifetime анонсов RA,
передаваемых через интерфейс (в секундах). Значение 0
говорит, что маршрутизатор не используется по умолчанию.";
}
container prefix-list {
  status obsolete;
  description
    "Список префиксов, помещаемых в опции Prefix Information
анонсов RA, передаваемых через интерфейс.

По умолчанию это все префиксы, которые маршрутизатор
анонсирует через протоколы маршрутизации, как on-link для
интерфейса, передающего анонс.";
  list prefix {
    key "prefix-спец";
    status obsolete;
    description

```



Этот документ регистрирует указанный ниже submodule YANG в реестре YANG Module Names [RFC6020].

Name: ietf-ipv6-router-advertisements  
Module: ietf-ipv6-unicast-routing  
Reference: RFC 8349

## 11. Вопросы безопасности

Заданные этим документом модули YANG определяют схему данных, предназначенных для доступа по протоколам сетевого управления, таким как NETCONF [RFC6241] и RESTCONF [RFC8040]. Нижним уровнем NETCONF является защищённый транспортный уровень с обязательной реализацией защищённого транспорта Secure Shell (SSH) [RFC6242]. Нижним уровнем RESTCONF является HTTPS с обязательной реализацией защищённого транспорта TLS [RFC5246].

Модель управления доступом NETCONF [RFC8341] обеспечивает средства, позволяющие предоставить отдельным пользователям NETCONF или RESTCONF доступ к предопределённому подмножеству протокольных операций и содержимого NETCONF и RESTCONF.

В этих модулях YANG определено множество узлов данных с возможностью записи, создания, удаления (т. е. по умолчанию установлено config true). Такие узлы могут считаться чувствительными или уязвимыми в некоторых сетевых средах. Операции записи в такие узлы (например, edit-config) без подобающей защиты могут оказывать негативное влияние на работу сети. Ниже указаны поддеревья и узлы данных с возможными уязвимостями.

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

Этот список указывает настроенные на устройстве протоколы плоскости управления.

### **/routing/ribs/rib**

Этот список указывает настроенные на устройстве базы RIB.

Некоторые из доступных для чтения узлов данных в этих модулях YANG могут считаться чувствительными или уязвимыми в некоторых сетевых средах. Важен правильный контроль операций чтения таких узлов данных (например, через get, get-config, notification). Ниже указаны поддеревья и узлы данных с возможными уязвимостями.

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

Эти списки содержат настроенные на устройстве протоколы плоскости управления. Чувствительные сведения из таких списков приведены в моделях плоскостей управления.

### **/routing/ribs/rib**

Этот список указывает базы RIB и их содержимое на устройстве. Доступ к этим сведениям может раскрывать топологию сети и другие данные.

Некоторые из операций RPC в этом модуле YANG могут быть чувствительными или уязвимыми в отдельных сетевых средах. Важно обеспечить подобающий контроль операций, перечисленных ниже вместе с уязвимостями.

### **/routing/ribs/rib/active-route**

Результат этих операций RPC возвращает маршруты к конкретным адресатам. Доступ к этим сведениям может раскрывать топологию сети и взаимоотношения (например, клиентов и серверов). Кроме того, используемые сетевым устройством маршруты могут служить для организации атак на проходящий через устройство трафик.

## 12. Литература

### 12.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](https://www.rfc-editor.org/info/rfc2119), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](https://www.rfc-editor.org/info/rfc3688), 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](https://www.rfc-editor.org/info/rfc4861), DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](https://www.rfc-editor.org/info/rfc5246), DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](https://www.rfc-editor.org/info/rfc6020), 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](https://www.rfc-editor.org/info/rfc6241), 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](https://www.rfc-editor.org/info/rfc6242), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](https://www.rfc-editor.org/info/rfc6991), 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](https://www.rfc-editor.org/info/rfc7950), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", [RFC 8022](https://www.rfc-editor.org/info/rfc8022), DOI 10.17487/RFC8022, November 2016, <<https://www.rfc-editor.org/info/rfc8022>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](https://www.rfc-editor.org/info/rfc8040), 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>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 8343](#), DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [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>>.
- [W3C.REC-xml-20081126] Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <<https://www.w3.org/TR/2008/REC-xml-20081126>>.

## 12.2. Дополнительная литература

- [RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", [RFC 7224](#), DOI 10.17487/RFC7224, May 2014, <<https://www.rfc-editor.org/info/rfc7224>>.
- [RFC7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", [RFC 7895](#), DOI 10.17487/RFC7895, June 2016, <<https://www.rfc-editor.org/info/rfc7895>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", [RFC 7951](#), DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [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>>.
- [YANG-Guidelines] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", Work in Progress<sup>1</sup>, draft-ietf-netmod-rfc6087bis-20, March 2018.

## Приложение А. Полное дерево схемы

В этом приложении представлено полное дерево модели данных ядра маршрутизации. Обозначения приведены в [RFC8340]. Тип данных каждого узла представлен справа от соответствующей строки.

```

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

```

<sup>1</sup>Опубликовано в [RFC 8407](#). Прим. перев.



```

o--ro router-id?                yang:dotted-quad
o--ro interfaces
| o--ro interface*             if:interface-state-ref
o--ro control-plane-protocols
| o--ro control-plane-protocol* [type name]
|   o--ro type                 identityref
|   o--ro name                 string
o--ro ribs
  o--ro rib* [name]
    o--ro name                 string
    o--ro address-family       identityref
    o--ro default-rib?         boolean {multiple-ribs}?
    o--ro routes
      | o--ro route*
      |   o--ro route-preference?          route-preference
      |   o--ro next-hop
      |     | o--ro (next-hop-options)
      |     |   o--:(simple-next-hop)
      |     |   | o--ro outgoing-interface?
      |     |   |   | if:interface-ref
      |     |   |   | o--ro v4ur:next-hop-address?
      |     |   |   |   | inet:ipv4-address
      |     |   |   |   | o--ro v6ur:next-hop-address?
      |     |   |   |   |   | inet:ipv6-address
      |     |   |   | o--:(special-next-hop)
      |     |   |   | | o--ro special-next-hop?          enumeration
      |     |   |   | o--:(next-hop-list)
      |     |   |   |   o--ro next-hop-list
      |     |   |   |     o--ro next-hop*
      |     |   |   |     | o--ro outgoing-interface?
      |     |   |   |     | | if:interface-ref
      |     |   |   |     | o--ro v4ur:address?
      |     |   |   |     | | inet:ipv4-address
      |     |   |   |     | o--ro v6ur:address?
      |     |   |   |     |   | inet:ipv6-address
      |     |   |   | o--ro source-protocol          identityref
      |     |   |   | o--ro active?                  empty
      |     |   |   | o--ro last-updated?           yang:date-and-time
      |     |   |   | o--ro v4ur:destination-prefix? inet:ipv4-prefix
      |     |   |   | o--ro v6ur:destination-prefix? inet:ipv6-prefix
    o---x active-route
      o---w input
      | o---w v4ur:destination-address?  inet:ipv4-address
      | o---w v6ur:destination-address?  inet:ipv6-address
    o--ro output
      o--ro route
        o--ro next-hop
          | o--ro (next-hop-options)
          |   o--:(simple-next-hop)
          |   | o--ro outgoing-interface?
          |   |   | if:interface-ref
          |   |   | o--ro v4ur:next-hop-address?
          |   |   |   | inet:ipv4-address
          |   |   |   | o--ro v6ur:next-hop-address?
          |   |   |   |   | inet:ipv6-address
          |   |   | o--:(special-next-hop)
          |   |   | | o--ro special-next-hop?
          |   |   | |   enumeration
          |   |   | o--:(next-hop-list)
          |   |   |   o--ro next-hop-list
          |   |   |     o--ro next-hop*
          |   |   |     | o--ro outgoing-interface?
          |   |   |     | | if:interface-ref
          |   |   |     | o--ro v4ur:next-hop-address?
          |   |   |     | | inet:ipv4-address
          |   |   |     | o--ro v6ur:next-hop-address?
          |   |   |     |   | inet:ipv6-address
          |   |   | o--ro source-protocol          identityref
          |   |   | o--ro active?                  empty
          |   |   | o--ro last-updated?           yang:date-and-time
          |   |   | o--ro v4ur:destination-prefix? inet:ipv4-prefix
          |   |   | | inet:ipv4-prefix
          |   |   | o--ro v6ur:destination-prefix? inet:ipv6-prefix
          |   |   | | inet:ipv6-prefix
module: ietf-ipv6-unicast-routing
augment /if:interfaces/if:interface/ip:ipv6:
+--rw ipv6-router-advertisements
+--rw send-advertisements?    boolean
+--rw max-rtr-adv-interval?   uint16
+--rw min-rtr-adv-interval?   uint16
+--rw managed-flag?          boolean
+--rw other-config-flag?     boolean
+--rw link-mtu?              uint32
+--rw reachable-time?        uint32
+--rw retrans-timer?         uint32

```

```

+--rw cur-hop-limit?          uint8
+--rw default-lifetime?      uint16
+--rw prefix-list
  +--rw prefix* [prefix-spec]
    +--rw prefix-spec          inet:ipv6-prefix
  +--rw (control-adv-prefixes)?
    +--:(no-advertise)
      | +--rw no-advertise?    empty
    +--:(advertise)
      +--rw valid-lifetime?    uint32
      +--rw on-link-flag?      boolean
      +--rw preferred-lifetime? uint32
      +--rw autonomous-flag?   boolean
augment /if:interfaces-state/if:interface/ip:ipv6:
o--ro ipv6-router-advertisements
  o--ro send-advertisements?  boolean
  o--ro max-rtr-adv-interval? uint16
  o--ro min-rtr-adv-interval? uint16
  o--ro managed-flag?         boolean
  o--ro other-config-flag?    boolean
  o--ro link-mtu?             uint32
  o--ro reachable-time?      uint32
  o--ro retrans-timer?        uint32
  o--ro cur-hop-limit?        uint8
  o--ro default-lifetime?     uint16
  o--ro prefix-list
    o--ro prefix* [prefix-spec]
      o--ro prefix-spec          inet:ipv6-prefix
      o--ro valid-lifetime?      uint32
      o--ro on-link-flag?        boolean
      o--ro preferred-lifetime?  uint32
      o--ro autonomous-flag?     boolean

```

## Приложение В. Минимальная реализация

Некоторые части и опции модели ядра маршрутизации, такие как пользовательские RIB, предназначены лишь для «продвинутых» маршрутизаторов. В этом приложении даны основные (ненормативные) рекомендации по реализации минимального набора доступных функций. Такие реализации подходят для хостов и простых маршрутизаторов.

Минимальная реализация не поддерживает свойство `multiple-ribs`. Это означает доступность лишь одной контролируемой системой базы RIB для каждого поддерживаемого семейства адресов (IPv4, IPv6 или оба). Эти RIB используются по умолчанию. Управляемые пользователем базы RIB не разрешены.

В дополнение к обязательному экземпляру псевдопротокола `direct` минимальной реализации следует поддерживать настраиваемые экземпляры псевдопротокола `static`.

Для хостов, не предназначенных на роль маршрутизаторов, возможность включать передачу анонсов IPv6 RA (параграф 5.4) следует исключить.

Платформы с существенно ограниченными ресурсами могут применять отклонения (deviations) для ограничения модели данных, например, ограничивая число экземпляров контролируемых системой псевдопротоколов `static`.

## Приложение С. Пример добавления нового протокола управления

В этом приложении показано, как можно расширить модель данных ядра маршрутизации для поддержки новых протоколов плоскости управления. Приведённый ниже модуль YANG `example-rip` предназначен для иллюстрации, а не реального определения модели данных для протокола RIP (Routing Information Protocol). Для краткости в модуле не выполняются некоторые из рекомендаций [YANG-Guidelines] (см. также параграф 5.3.2).

```

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
      "Отождествление для Routing Information Protocol (RIP).";
  }

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

  grouping route-content {

```



```

"module-set-id": "c2e1f54169aa7f36e1a6e8d0865d441d3600f9c4",
"module": [
  {
    "name": "ietf-routing",
    "revision": "2018-03-13",
    "feature": [
      "multiple-ribs",
      "router-id"
    ],
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-ipv4-unicast-routing",
    "revision": "2018-03-13",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-ipv4-unicast-routing",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-ipv6-unicast-routing",
    "revision": "2018-03-13",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing",
    "conformance-type": "implement",
    "submodule": [
      {
        "name": "ietf-ipv6-router-advertisements",
        "revision": "2018-03-13"
      }
    ]
  },
  {
    "name": "ietf-interfaces",
    "revision": "2018-02-20",
    "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": "2014-05-08",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-ip",
    "revision": "2018-02-22",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
    "conformance-type": "implement"
  }
]
}
}

```

```

+-----+
|         |
| Маршрутизатор ISP |
|         |
+-----+-----+
|         |
|         |2001:db8:0:1::2
|         |192.0.2.2
|         |
|         |2001:db8:0:1::1
eth0|192.0.2.1
+-----+-----+
|         |
| Маршрутизатор A |
|         |
+-----+-----+
eth1|198.51.100.1
|2001:db8:0:2::1
|

```

Рисунок 2. Пример конфигурации сети.

В примере простой сети на рисунке 2 маршрутизатор A использует статические маршруты по умолчанию с маршрутизатором ISP в качестве следующего интервала. Анонсы IPv6 RA настроены лишь для интерфейса eth1 и отключены на eth0.

Экземпляр дерева данных может иметь приведённую ниже форму.

```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",
        "description": "Uplink to ISP.",
        "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
            }
          ]
        },
        "autoconf": {
          "create-global-addresses": false
        },
        "ietf-ipv6-unicast-routing:ipv6-router-advertisements": {
          "send-advertisements": false
        }
      }
    ],
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "description": "Interface to the internal network.",
      "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
          }
        ]
      },
      "autoconf": {
        "create-global-addresses": false
      },
      "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":
          "Static routing is used for the internal network.",
        "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"
                }
              }
            ]
          }
        ]
      }
    ]
  }
},
"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",

```



```

</address-family>
<default-rib>true</default-rib>
<routes>
  <route>
    <ietf-ipv4-unicast-routing:destination-prefix>
      192.0.2.1/24
    </ietf-ipv4-unicast-routing:destination-prefix>
    <next-hop>
      <outgoing-interface>eth0</outgoing-interface>
    </next-hop>
    <route-preference>0</route-preference>
    <source-protocol>ietf-routing:direct</source-protocol>
    <last-updated>2015-10-24T17:11:27+02:00</last-updated>
  </route>
  <route>
    <ietf-ipv4-unicast-routing:destination-prefix>
      198.51.100.0/24
    </ietf-ipv4-unicast-routing:destination-prefix>
    <next-hop>
      <outgoing-interface>eth1</outgoing-interface>
    </next-hop>
    <route-preference>0</route-preference>
    <source-protocol>ietf-routing:direct</source-protocol>
    <last-updated>2015-10-24T17:11:27+02:00</last-updated>
  </route>
  <route>
    <ietf-ipv4-unicast-routing:destination-prefix>0.0.0.0/0
    </ietf-ipv4-unicast-routing:destination-prefix>
    <next-hop>
      <ietf-ipv4-unicast-routing:next-hop-address>192.0.2.2
      </ietf-ipv4-unicast-routing:next-hop-address>
    </next-hop>
    <route-preference>5</route-preference>
    <source-protocol>ietf-routing:static</source-protocol>
    <last-updated>2015-10-24T18:02:45+02:00</last-updated>
  </route>
</routes>
</rib>
<rib or:origin="or:intended">
  <name>ipv6-master</name>
  <address-family>
    ietf-ipv6-unicast-routing:ipv6-unicast
  </address-family>
  <default-rib>true</default-rib>
  <routes>
    <route>
      <ietf-ipv6-unicast-routing:destination-prefix>
        2001:db8:0:1::/64
      </ietf-ipv6-unicast-routing:destination-prefix>
      <next-hop>
        <outgoing-interface>eth0</outgoing-interface>
      </next-hop>
      <route-preference>0</route-preference>
      <source-protocol>ietf-routing:direct</source-protocol>
      <last-updated>2015-10-24T17:11:27+02:00</last-updated>
    </route>
    <route>
      <ietf-ipv6-unicast-routing:destination-prefix>
        2001:db8:0:2::/64
      </ietf-ipv6-unicast-routing:destination-prefix>
      <next-hop>
        <outgoing-interface>eth1</outgoing-interface>
      </next-hop>
      <route-preference>0</route-preference>
      <source-protocol>ietf-routing:direct</source-protocol>
      <last-updated>2015-10-24T17:11:27+02:00</last-updated>
    </route>
    <route>
      <ietf-ipv6-unicast-routing:destination-prefix>::/0
      </ietf-ipv6-unicast-routing:destination-prefix>
      <next-hop>
        <ietf-ipv6-unicast-routing:next-hop-address>
          2001:db8:0:1::2
        </ietf-ipv6-unicast-routing:next-hop-address>
      </next-hop>
      <route-preference>5</route-preference>
      <source-protocol>ietf-routing:static</source-protocol>
      <last-updated>2015-10-24T18:02:45+02:00</last-updated>
    </route>
  </routes>
</rib>
</ribs>
</routing>
</data>
</rpc-reply>

```

## Благодарности

Авторы благодарны Nitin Bahadur, Martin Bjorklund, Dean Bogdanovic, Joe Clarke, Francis Dupont, Jeff Haas, Joel Halpern, Wes Hardaker, Jia He, Sriganesh Kini, Suresh Krishnan, David Lamparter, Xiang Li, Stephane Litkowski, Andrew McGregor, Jan Medved, Thomas Morin, Tom Petch, Bruno Rijsman, Juergen Schoenwaelder, Phil Shafer, Dave Thaler, Vladimir Vassilev, Rob Wilton, Yi Yang, Derek Man-Kit Yeung, Jeffrey Zhang за их полезные комментарии и предложения.

## Адреса авторов

**Ladislav Lhotka**

CZ.NIC

Email: [lhotka@nic.cz](mailto:lhotka@nic.cz)

**Acee Lindem**

Cisco Systems

Email: [acee@cisco.com](mailto:acee@cisco.com)

**Yingzhen Qu**

Huawei

2330 Central Expressway

Santa Clara, CA 95050

United States of America

Email: [yingzhen.qu@huawei.com](mailto:yingzhen.qu@huawei.com)

## Перевод на русский язык

Николай Малых

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