

## RIB Information Model

Информационная модель RIB

### Аннотация

Маршрутизация и её функции в сети предприятия или оператора обычно выполняются сетевыми устройствами (маршрутизаторы и коммутаторы), использующими базы маршрутной информации (Routing Information Base или RIB). Протоколы и конфигурации выталкивают данные в RIB, а менеджер RIB устанавливает состояние в оборудовании для пересылки пакетов. Этот документ задаёт информационную модель RIB, позволяющую определить стандартизованную модель данных. Рабочая группа IETF I2RS использовала этот документ для разработки модели данных I2RS RIB. Этот документ публикуется для решений с информационными моделями RIB более высокого уровня, чтобы другие разработчики RIB могли воспользоваться преимуществами концепций проектирования.

### Статус документа

Документ не относится к категории Internet Standards Track и публикуется лишь для информации.

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>. Не все документы, одобренные IESG, претендуют на статус стандартов. Дополнительную информацию о стандартах Internet можно найти в разделе 2 RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <https://www.rfc-editor.org/info/rfc8430>.

### Авторские права

Copyright (c) 2018. 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
1.1. Уровни требований.....	3
2. Данные RIB.....	3
2.1. Определение RIB.....	3
2.2. Экземпляр маршрутизации.....	3
2.3. Маршрут.....	4
2.4. Следующий узел.....	4
2.4.1. Базовые nexthop.....	5
2.4.1.1. Специальные nexthop.....	6
2.4.2. Производные nexthop.....	6
2.4.2.1. Атрибуты списка nexthop.....	6
2.4.3. Косвенность nexthop.....	6
3. Чтение из RIB.....	7
4. Запись в RIB.....	7
5. Уведомления.....	7
6. Грамматика RIB.....	7
6.1. Разъяснение грамматики nexthop.....	8
7. Применение грамматики RIB.....	9
7.1. Использование предпочтений маршрутов.....	9
7.2. Использование различных типов nexthop.....	9
7.2.1. Туннельные nexthop.....	9
7.2.2. Списки репликации.....	9
7.2.3. Взвешенные списки.....	9
7.2.4. Защита.....	9
7.2.5. Цепочки nexthop.....	10
7.2.6. Списки списков.....	10

<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.

7.3. Групповая передача.....	10
8. Масштабные операции RIB.....	11
8.1. Чтение RIB.....	11
8.2. Запись в RIB.....	11
8.3. События и уведомления RIB.....	11
9. Вопросы безопасности.....	11
10. Взаимодействие с IANA.....	11
11. Литература.....	11
11.1. Нормативные документы.....	11
11.2. Дополнительная литература.....	12
Благодарности.....	12
Адреса авторов.....	12

## 1. Введение

Маршрутизация и её функции в сети предприятия или оператора обычно выполняются сетевыми устройствами. Маршрутизаторы обычно применяют протоколы маршрутизации, которые (вместе со статической конфигурацией) заполняют базы маршрутных данных (RIB) на маршрутизаторе. Базы RIB управляются менеджером RIB, а тот обеспечивает северный интерфейс с клиентами (т. е. протоколами маршрутизации) для вставки маршрутов в RIB. Менеджер RIB обращается к RIB для программирования базы пересылки (Forwarding Information Base или FIB) в оборудовании, взаимодействуя с менеджером FIB. Связи между этими элементами показаны на рисунке 1.

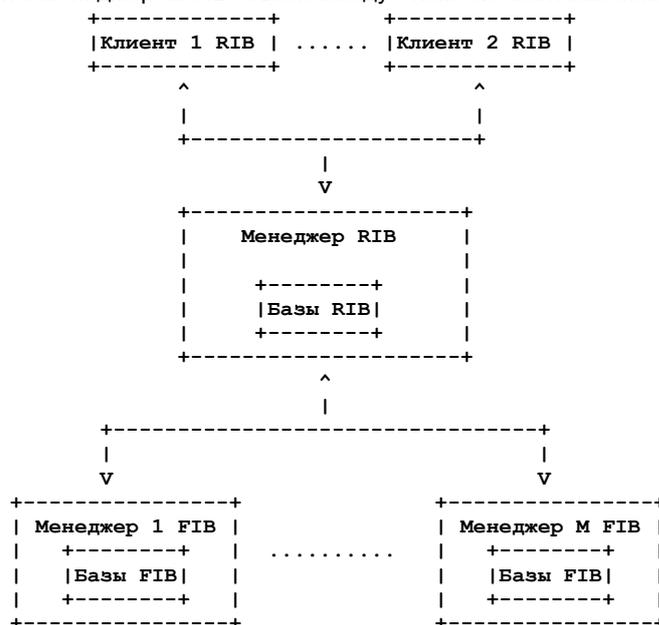


Рисунок 1. Менеджер RIB, клиенты RIB и менеджеры FIB.

Протоколы маршрутизации являются распределёнными по своей природе и каждый маршрутизатор независимо принимает решения на основе маршрутных данных, полученных от партнёров. С появлением новых парадигм развёртывания и потребностью в специализированных приложениях, возникает потребность в управлении функциями маршрутизации на маршрутизаторах [RFC7920]. Традиционное заполнение RIB сетевых устройств на основе протоколов достаточно для большинства применений, где используется распределённое управление сетью. Однако есть варианты применения, где операторы в настоящее время решают задачу путём настройки статических маршрутов, политики и правил импорта-экспорта RIB на маршрутизаторах. Существует также расширяющийся набор вариантов применения, где операторы имеют желание программировать RIB на основе данных, связанных не только с маршрутизацией (внутри сетевого домена). Программирование RIB может базироваться на иных сведениях (таких, как маршрутные данные смежного домена или нагрузка на хранилище и процессоры). Это может быть также просто программный способ создания по запросам динамических наложений (например, туннелей GRE) между вычислительными хостами (без запуска на них традиционных протоколов маршрутизации). Наличие стандартизованного программного интерфейса в RIB с общедоступной документацией позволило бы использовать дополнительные сетевые приложения для разных вариантов применения [RFC7920].

Программный интерфейс RIB включает операции чтения из RIB и записи (добавление, изменение, удаление) в RIB.

Для понимания содержимого RIB в маршрутизаторах применяются методы, похожие на SNMP MIB для протоколов и «скобление» экрана (screen scraping). Эти методы не расширяемы, поскольку являются клиентскими механизмами вытягивания, а не упреждающего выталкивания (из маршрутизатора). Скобление экрана подвержено ошибкам (поскольку формат может меняться) и зависит от производителя. Создание RIB из протокольных MIB подвержено ошибкам, поскольку MIB представляют данные протокола, а не точные сведения, вошедшие в RIB. Таким образом, простое получение доступных лишь для чтения сведений RIB от маршрутизатора является сложной задачей.

Добавление в RIB содержимого от клиента RIB сегодня можно выполнить с использованием механизмов статической настройки, обеспечиваемых производителями маршрутизаторов. Однако набор того, что можно изменить в значениях RIB, меняется от производителя к производителю, а методы настройки также зависят от производителя. Это осложняет клиентам RIB программирование сети в оборудовании разных производителей согласованным и независимым от производителя способом.

Целью этого документа является задание информационной модели для RIB. Используя такую модель можно создать подробную модель данных для RIB. Такая модель данных может применяться клиентами RIB для программирования сетевых устройств. Одной из основанных на этом документе является модель данных I2RS RIB [RFC8431].

В разделе 2 подробно рассказано, что входит в RIB и что можно запрограммировать там. Разделы 3 и 4 содержат рекомендации по чтению и записи в RIB. В разделе 5 дано высокоуровневое представление событий и уведомлений от сетевых устройств клиенту RIB для обновления клиента по асинхронным событиям. Грамматика RIB описана в разделе 6, а примеры её использования - в разделе 7. В разделе 8 рассматривается выполнение операций RIB.

## 1.1. Уровни требований

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

## 2. Данные RIB

В этом разделе описаны детали RIB. Раздел ссылается на грамматику RIB, описанную в разделе 6. Грамматика RIB. Высокоуровневое представление содержимого RIB представлено на рисунке 2. Для простоты на рисунке показан 1 экземпляр маршрутизации, 1 RIB и 1 маршрут. В параграфах этого раздела описаны логические узлы данных, которым следует присутствовать в RIB. В разделах 3 и 4 приведены высокоуровневые описания операций чтения и записи.

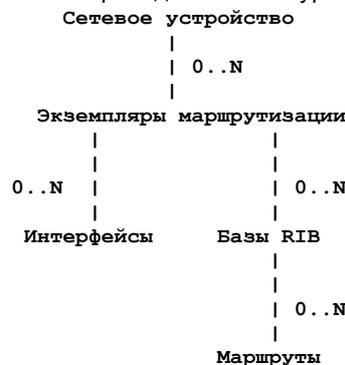


Рисунок 2. Информационная модель RIB.

### 2.1. Определение RIB

В контексте информационной модели RIB является объектом, содержащим маршруты. Таблица указывается именем и содержится внутри экземпляра маршрутизации (параграф 2.2). Сетевое устройство **может** содержать экземпляры маршрутизации, каждый из которых **может** содержать таблицы RIB. Имя **должно** быть уникальным в рамках экземпляра маршрутизации. Все маршруты в данной RIB **должны** относиться к одному семейству адресов (например, IPv4). Каждая таблица RIB **должна** относиться к экземпляру маршрутизации. Каждый экземпляр маршрутизации может содержать несколько RIB для одного семейства адресов (например, IPv6). В типичном случае это может служить для маршрутизации с несколькими топологиями [RFC4915] [RFC5120].

С каждой RIB **может** быть связан атрибут ENABLE\_IP\_RPF\_CHECK, разрешающий проверку пересылки по обратному пути (Reverse Path Forwarding или RPF) на всех маршрутах IP этой RIB. Проверка RPF применяется для предотвращения подмены адресов (spoofing) и ограничения вредоносного трафика. Для пакетов IP отыскивается адрес отправителя и интерфейсы RPF, связанные с маршрутом, для которого найден IP-адрес отправителя. Если интерфейс для входящего пакета IP соответствует одному из интерфейсов RPF, пакет IP пересылается по IP-адресу получателя, в иных случаях - отбрасывается.

### 2.2. Экземпляр маршрутизации

В контексте информационной модели RIB экземпляром маршрутизации является набор RIB, интерфейсов и параметров маршрутизации. Экземпляр маршрутизации создаёт логический «срез» (slice) маршрутизатора. Это позволяет таким срезам на разных маршрутизаторах взаимодействовать между собой. Виртуальные сети L3 VPN, L2 VPN (L2VPN) и VPLS<sup>1</sup> можно моделировать как экземпляры маршрутизации. Отметим, что моделирование L2VPN с применением экземпляра маршрутизации представляет лишь аспект L3 (RIB), не представляя сведений L2 (например, ARP), которые могут быть связаны с L2VPN. Набор интерфейсов указывает, какие интерфейсы связаны с экземпляром маршрутизации. Таблицы RIB указывают, как будет пересылаться входящий трафик, а параметры маршрутизации контролируют информацию в RIB. Пересечение наборов интерфейсов двух экземпляров маршрутизации **должно** быть пустым, т. е. интерфейсу **недопустимо** присутствовать в двух экземплярах маршрутизации. Таким образом, экземпляр маршрутизации описывает данные и параметры маршрутизации через набор интерфейсов. Экземпляр маршрутизации **должен** включать указанные ниже поля.

#### INSTANCE\_NAME

Имя, служащее для идентификации данного экземпляра маршрутизации, уникальное для каждого экземпляра в данном сетевом устройстве.

#### rib-list

Список RIB, связанных с экземпляром маршрутизации. Каждый экземпляр может иметь несколько RIB для представления маршрутов разных типов. Например, маршруты IPv4 могут помещаться в одну RIB, а маршруты MPLS - в другую. Список RIB может быть пустым.

Экземпляр маршрутизации **может** включать указанные ниже поля.

#### interface-list

Список интерфейсов, связанных с экземпляром маршрутизации, помогающий ограничить область пересылки пакетов. Пакеты, приходящие через эти интерфейсы, напрямую связаны с данным экземпляром маршрутизации. Список состоит из идентификаторов, однозначно указывающих интерфейсы.

<sup>1</sup>Virtual Private LAN Service - служба виртуальных частных ЛВС.

**ROUTER\_ID**

Это поле идентифицирует сетевое устройство по взаимодействиям плоскости управления с другими сетевыми устройствами. Поле требуется в случае создания на физическом маршрутизаторе нескольких виртуальных маршрутизаторов, каждый из которых **должен** иметь уникальное значение ROUTER\_ID в рамках данного домена.

Экземпляр маршрутизации может создаваться исключительно для обработки пакетов и не иметь каких-либо связанных с ним интерфейсов. Например, входящие пакеты экземпляра маршрутизации А могут иметь в качестве nexthop экземпляр В, а после обработки в В значение nexthop может указывать экземпляр С, при этом экземпляр В не будет связан с каким-либо интерфейсом. С учётом того, что этот экземпляр не взаимодействует в плоскости управления с другими устройствами, ROUTER\_ID для него не требуется.

**2.3. Маршрут**

Маршрут - это, по сути, условие сопоставления и действие, выполняемое по этому условию. Условие задаёт тип маршрута (IPv4, MPLS и т. п.) и набор полей для сопоставления. Содержимое маршрута показано на рисунке 3. Для простоты на рисунке показан лишь 1 экземпляр атрибутов маршрута, флагов сопоставления и nexthop.

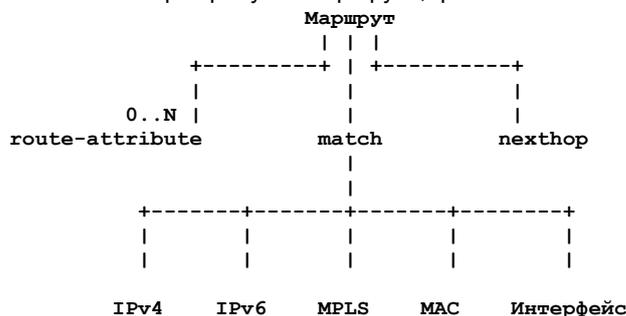


Рисунок 3. Модель маршрута.

Ниже указаны типы сопоставлений, задаваемые этим документом.

- IPv4 - сопоставление с IP-адресом получателя и/или отправителя в заголовке IPv4.
- IPv6 - сопоставление с IP-адресом получателя и/или отправителя в заголовке IPv6.
- MPLS - сопоставление с меткой MPLS на вершине стека меток MPLS.
- MAC - сопоставление с MAC<sup>1</sup>-адресом получателя в заголовке Ethernet.
- Interface - сопоставление с интерфейсом, принявшим пакет.

Маршрут **может** сопоставляться с одним или несколькими типами в соответствии с политикой, задающей ограничение числа маршрутов (AND) или объединение двух фильтров (OR).

С каждым маршрутом **должен** быть связан обязательный атрибут, указанный ниже.

**ROUTE\_PREFERENCE**

Численное значение, позволяющее сравнивать маршруты от разных протоколов (статическая настройка тоже считается протоколом). Этот атрибут называют также административной дистанцией (administrative distance). Меньшее значение указывает более предпочтительный маршрут. Например, может быть маршрут OSPF для 192.0.2.1/32 (или IPv6 2001:DB8::1/128) с приоритетом 5. Если контроллер программирует маршрут для 192.0.2.1/32 (или IPv6 2001:DB8::1/128) с приоритетом 2, менеджер RIB предпочтёт маршрут от контроллера. Предпочтения следует использовать для управления поведением. Дополнительные примеры даны в параграфе 7.1.

С каждым маршрутом могут быть связаны необязательные атрибуты, указанные ниже.

**route-vendor-attributes**

Производители могут задавать свои атрибуты, используя это поле. Детали таких атрибутов не рассматриваются здесь.

С каждым маршрутом связан следующий узел пересылки (nexthop). Описание параметра приведено в параграфе 2.4.

Были рассмотрены дополнительные свойства для сопоставления групповых пакетов (например, значение TTL для ограничения протяжённости multicast-групп), но они не включены в эту информационную модель. В будущих информационных моделях RIB следует исследовать такие групповые свойства.

**2.4. Следующий узел**

Объект nexthop является результатом поиска маршрута. Например, если поиск указывает передачу пакета через данный интерфейс, nexthop будет представлять этот интерфейс.

Значения nexthop могут быть распознанными полностью или нераспознанными. В полностью распознанном nexthop имеются адекватные сведения для отправки исходящего пакета получателю путём его пересылки на интерфейс, напрямую подключённый к соседу. Например, nexthop с интерфейсом «точка-точка» или IP-адресом интерфейса Ethernet является распознанным полностью. При нераспознанном nexthop менеджер RIB должен определить финальное распознанное значение nexthop. Например, это может быть адрес IP. Менеджер RIB будет выяснять, как достичь этот адрес IP, например, он может быть доступен путём обычной пересылки IP, через туннель MPLS или обоими способами. Если менеджер RIB не может преобразовать nexthop, сохраняется нераспознанное состояние и nexthop **недопустимо** считать кандидатом на установку в FIB. Последующие события RIB могут привести к распознаванию (например, адрес IP будет анонсирован соседом IGP). И наоборот, распознанные полностью nexthop могут стать нераспознанными (например, в результате отключения туннеля) и перестанут быть кандидатами на включение в FIB.

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

Если хотя бы 1 из nexthop маршрута удалось распознать, маршрут можно применять для пересылки пакетов. Такие маршруты считаются подходящими для включения в FIB и далее называются FIB-eligible. И наоборот, если ни один из nexthop маршрута распознать не удалось, маршрут больше не применяется для пересылки пакетов. Такой маршрут считается неподходящим для установки в FIB и далее называется FIB-ineligible. Сведения информационной модели RIB позволяют клиентам RIB программировать маршруты, в которых nexthop изначально могут быть нераспознанными. При распознавании nexthop менеджер RIB передаёт уведомление об этом (см. 5. Уведомления).

Общая структура и использование nexthop показаны на рисунке 4, где для простоты указано лишь по одному экземпляру каждого элемента.

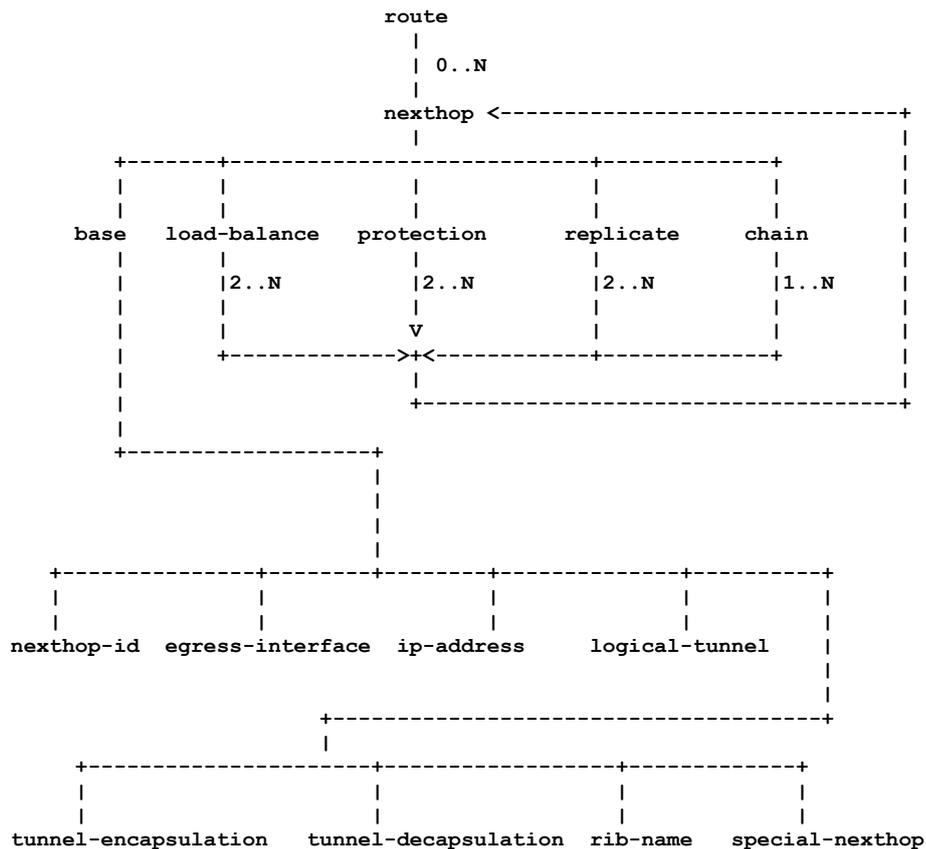


Рисунок 4. Модель nexthop.

Этот документ определяет лишь базовую, расширяемую и рекурсивную грамматику для nexthop, которые могут быть базовыми или производными. В параграфе 2.4.1 рассмотрены базовые nexthop, а параграф 2.4.2 разъясняет разные типы производных nexthop. Имеются также специальные виды nexthop, описанные в параграфе 2.4.1.1. В параграфе 2.4.3 рассматривается косвенность nexthop и её применение. Примеры использования туннельных и производных nexthop приведены в параграфе 7.2.

### 2.4.1. Базовые nexthop

Ниже перечислены типы nexthop на самом низком уровне.

#### Идентификатор

Идентификатор, возвращаемый сетевым устройством, представляющим nexthop. Может служить для многократного применения nexthop при программировании производных nexthop.

#### Интерфейсные nexthop

Имеются nexthop, указывающие интерфейс. С ними связаны указанные ниже атрибуты.

##### Egress-interface

Представляет физический, логический или виртуальный интерфейс сетевого устройства. Для такого интерфейса недопустимо требовать распознавания адресов. Интерфейс может относиться к любому экземпляру маршрутизации.

##### IP address

Выполняется поиск маршрута по этому для определения egress-interface. Распознавание адреса может потребоваться в зависимости от интерфейса.

- Необязательное поле rib-name может задаваться для указания RIB, в которой будет выполняться поиск по адресу IP. Можно использовать rib-name для передачи пакета из одного домена в другой. По умолчанию используется RIB, к которой относится маршрут.

Эти атрибуты могут сочетаться с указанными ниже.

##### Egress-interface u IP address

Может применяться в случаях, где, например, IP-адрес относится к локальному каналу (link-local).

##### Egress-interface u MAC address

Выходной интерфейс (egress-interface) должен быть интерфейсом Ethernet. Распознавание адреса не требуется для этого nexthop.

#### Туннельные nexthop

Некоторые nexthop указывают туннель. Типы туннельных nexthop приведены ниже.

##### tunnel-encapsulation

Это может быть инкапсуляция, представляющая туннель IP, MPLS или иной, как описано в этом документе. С tunnel-encapsulation может быть связано значение egress-interface для указания интерфейса, передающего

пакеты. Это полезно, когда сетевое устройство имеет интерфейсы Ethernet и нужно распознавать адрес для пакетов IP.

#### **tunnel-decapsulation**

Служит для декапсуляции туннельного заголовка. После декапсуляции для пакета может выполняться новый поиск для связывания с другим nexthop или пакет может передаваться напрямую через egress-interface.

#### **logical-tunnel**

Это может быть путь с коммутацией по меткам MPLS (Label Switched Path или LSP) или туннель GRE (а также другие, определённые в этом документе), представленный уникальным идентификатором (например, именем).

#### **rib-name**

Значение nexthop, указывающее RIB для продолжения поиска маршрута (способ организации цепочек поиска).

Туннельные nexthop позволяют клиентам RIB программировать статические туннельные заголовки. Могут быть случаи, когда удалённая конечная точка туннеля не поддерживает динамическую сигнализацию (например, хост не поддерживает LDP) и в таких случаях клиенту RIB может потребоваться программировать туннельные заголовки на обеих сторонах туннеля. Туннельный nexthop сохраняется базовым в соответствии со спецификациями для часто применяемых туннелей. Предполагается, что модель данных будет точно представлять такие типы туннелей.

### **2.4.1.1. Специальные nexthop**

Специальные nexthop предназначены для выполнения особых чётко заданных функций (например, DISCARD).

#### **DISCARD**

Указывает, что сетевому устройству следует отбросить пакет и увеличить значение счётчика отбрасывания.

#### **DISCARD\_WITH\_ERROR**

Указывает, что сетевому устройству следует отбросить пакет, увеличить значение счётчика и передать обратно подходящее сообщение об ошибке (такое как ICMP).

#### **RECEIVE**

Указывает, что трафик предназначен для сетевого устройства, например, пакеты протокола или OAM<sup>1</sup>. Весь трафик для локального получателя **следует** дросселировать (throttle) во избежание DoS<sup>2</sup>-атак на плоскость управления маршрутизатора. Можно задавать дополнительный ограничитель скорости, указывающий, как ограничить трафик, адресованный плоскости управления. Описание таких ограничителей выходит за рамки документа.

### **2.4.2. Производные nexthop**

Производными nexthop могут быть:

- взвешенные списки, применяемые для распределения нагрузки;
- списки предпочтений, служащие для защиты с использованием основного и резервного пути;
- списки репликации - списки nexthop, в которые передаются копии пакета;
- цепочки nexthop для связывания нескольких операций или добавления нескольких заголовков;
- списки списков для рекурсивного применения указанного выше.

Цепочки nexthop (см. параграф 7.2.5) обеспечивают способ связывания нескольких операций над пакетом путём их логического объединения. Например, можно сцепить операции декапсуляции заголовка MPLS и передачи пакета через заданный интерфейс egress-interface. Цепочки могут служить для указания в пакете нескольких заголовков перед его пересылкой. Одним из простых примеров является работа MPLS через GRE, где пакет имеет внутренний заголовок MPLS, за которым следует заголовок GRE, затем - заголовок IP. Внешний заголовок IP определяется сетевым устройством, а заголовки MPLS или GRE - контроллером. Не каждое сетевое устройство способно поддерживать все типы цепочек nexthop и связывание произвольного числа заголовков. Модели данных RIB **следует** обеспечивать способ раскрытия возможностей цепочек nexthop, поддерживаемых данным сетевым устройством.

Предполагается, что у всех сетевых устройств будет ограничение глубины (числа уровней) поиска и не все устройства смогут поддерживать все типы цепочек nexthop. Поэтому согласование возможностей RIB становится очень важным и модель данных RIB **должна** указывать для клиентов RIB способ получения сведений о возможностях сетевого устройства.

#### **2.4.2.1. Атрибуты списка nexthop**

В nexthop в форме списка с каждым элементом списка могут быть связаны атрибуты, указывающие его роль.

#### **NEXTHOP\_PREFERENCE**

Применяется в схемах защиты и задаётся целым числом от 1 до 99 (меньшее значение указывает более высокий приоритет). Для выгрузки пары основной-резервный в FIB nexthop преобразуются и выбираются два наиболее высоких приоритета. Каждому <NEXTHOP\_PREFERENCE> следует иметь уникальное значение в <nexthop-protection> (см. 6. Грамматика RIB).

#### **NEXTHOP\_LB\_WEIGHT**

Служит для распределения нагрузки. Каждому элементу списка **должен** назначаться вес от 1 до 99. Вес определяет долю трафика, передаваемого через nexthop при пересылке отношением веса данного nexthop к сумме весов всех прочих nexthop этого маршрута, применяемых для пересылки. Для равномерного распределения нагрузки **можно** задать вес в всем nexthop. Это значение зарезервировано для равномерного распределения и при его применении **должно** устанавливаться для всех nexthop. Отметим, что значение 0 сохранилось по историческим причинам.

### **2.4.3. Косвенность nexthop**

Можно применять идентификаторы nexthop для задания косвенности. Идентификаторы устанавливаются менеджером RIB и возвращаются клиенту RIB по запросу. Одним из примеров применения косвенности служит nexthop с указанием на другое сетевое устройство (например, партнер BGP). Возвращаемый идентификатор nexthop можно использовать

<sup>1</sup>Operations, Administration, and Maintenance - операции, администрирования, управление (поддержка).

<sup>2</sup>Denial-of-service - отказ в обслуживании.

для программирования маршрутов, указывающих этот nexthop. Учитывая, что менеджер RIB организовал косвенность с использованием идентификатора nexthop, изменение транспортного пути к сетевому устройству (партнёр BGP) будет незаметно для клиента RIB и все маршруты, указывающие на это сетевое устройство, автоматически пойдут по новому транспортному пути. Косвенность nexthop с использованием идентификаторов может применяться не только к индивидуальному (unicast) nexthop, но и к nexthop, содержащим цепочки и вложенные nexthop. Примеры представлены в параграфе 2.4.2. Производные nexthop.

### 3. Чтение из RIB

Модель данных RIB **должна** позволять клиенту RIB читать записи RIB, созданных этим объектом. Администратор сетевого устройства **может** разрешить клиенту RIB чтение других RIB через списки доступа на сетевом устройстве. Детали списков доступа выходят за рамки этого документа. Модель данных **должна** поддерживать полное считывание RIB и последующее инкрементное считывание изменений в RIB. При отправке данных клиенту RIB менеджеру RIB **следует** пытаться передать все зависимости объекта до отправки самого объекта.

### 4. Запись в RIB

Модель данных RIB **должна** позволять клиенту RIB вносить записи в RIB, созданные этим объектом. Администратор сетевого устройства **может** разрешить клиенту RIB запись в другие RIB через списки доступа на сетевом устройстве. Детали списков доступа выходят за рамки этого документа. При записи объекта в RIB клиенту RIB **следует** пытаться записать все зависимости объекта до отправки самого объекта. Модели данных **следует** поддерживать запросы идентификаторов для nexthop и сбор этих идентификаторов в отклике.

Программирование маршрутов в RIB **должно** приводить к коду возврата, содержащему указанные ниже атрибуты.

- Installed (установлен) - Yes/No (установлен ли маршрут в RIB).
- Active (активен) - Yes/No (был ли маршрут распознан полностью и является ли кандидатом для выбора).
- Reason (причина) - например, Not authorized (не разрешено).

Модель данных **должна** указывать, какие объекты можно изменять. Изменяемым считается объект, содержимое которого можно поменять без влияния на пересылку данных. Для изменения неизменяемого объекта требуется создать новый объект и удалить имеющийся. Например, маршрутам, использующим nexthop, указанный идентификатором nexthop, следует быть неизменными при смене содержимого nexthop.

### 5. Уведомления

Менеджер RIB на сетевом устройстве отправляет клиенту RIB асинхронные уведомления при тех или иных событиях на устройстве. Модель данных RIB **должна** поддерживать передачу асинхронных уведомлений. Краткий список предлагаемых уведомлений приведён ниже.

- Изменение маршрута (с возвратом кода, указанного в разделе 4. Запись в RIB).
- Статус распознавания nexthop (распознан или не распознан).

### 6. Грамматика RIB

В этом разделе задана информационная модель RIB в форме Бэкуса-Наура для маршрутизации (Routing Backus-Naur Form или rBNF) [RFC5511]. Эта грамматика поможет читателю лучше понять раздел 2 для вывода модели данных.

```

<routing-instance> ::= <INSTANCE_NAME>
                        [<interface-list>] <rib-list>
                        [<ROUTER_ID>]

<interface-list> ::= (<INTERFACE_IDENTIFIER> ...)

<rib-list> ::= (<rib> ...)
<rib> ::= <rib-name> <address-family>
          [<route> ... ]
          [ENABLE_IP_RPF_CHECK]
<address-family> ::= <IPV4_ADDRESS_FAMILY> | <IPV6_ADDRESS_FAMILY> |
                    <MPLS_ADDRESS_FAMILY> | <IEEE_MAC_ADDRESS_FAMILY>

<route> ::= <match> <nexthop>
            [<route-attributes>]
            [<route-vendor-attributes>]

<match> ::= <IPV4> <ipv4-route> | <IPV6> <ipv6-route> |
            <MPLS> <MPLS_LABEL> | <IEEE_MAC> <MAC_ADDRESS> |
            <INTERFACE> <INTERFACE_IDENTIFIER>
<route-type> ::= <IPV4> | <IPV6> | <MPLS> | <IEEE_MAC> | <INTERFACE>

<ipv4-route> ::= <ip-route-type>
                (<destination-ipv4-address> | <source-ipv4-address> |
                 (<destination-ipv4-address> <source-ipv4-address>))
<destination-ipv4-address> ::= <ipv4-prefix>
<source-ipv4-address> ::= <ipv4-prefix>
<ipv4-prefix> ::= <IPV4_ADDRESS> <IPV4_PREFIX_LENGTH>

<ipv6-route> ::= <ip-route-type>
                (<destination-ipv6-address> | <source-ipv6-address> |
                 (<destination-ipv6-address> <source-ipv6-address>))
<destination-ipv6-address> ::= <ipv6-prefix>
<source-ipv6-address> ::= <ipv6-prefix>

```

```

<ipv6-prefix> ::= <IPV6_ADDRESS> <IPV6_PREFIX_LENGTH>
<ip-route-type> ::= <SRC> | <DEST> | <DEST_SRC>

<route-attributes> ::= <ROUTE_PREFERENCE> [<LOCAL_ONLY>]
                    [<address-family-route-attributes>]

<address-family-route-attributes> ::= <ip-route-attributes> |
                                       <mpls-route-attributes> |
                                       <ethernet-route-attributes>

<ip-route-attributes> ::= <>
<mpls-route-attributes> ::= <>
<ethernet-route-attributes> ::= <>
<route-vendor-attributes> ::= <>

<nexthop> ::= <nexthop-base> |
              (<NEXTHOP_LOAD_BALANCE> <nexthop-lb>) |
              (<NEXTHOP_PROTECTION> <nexthop-protection>) |
              (<NEXTHOP_REPLICATE> <nexthop-rotate>) |
              <nexthop-chain>

<nexthop-base> ::= <NEXTHOP_ID> |
                  <nexthop-special> |
                  <egress-interface> |
                  <ipv4-address> | <ipv6-address> |
                  (<egress-interface>
                   (<ipv4-address> | <ipv6-address>)) |
                  (<egress-interface> <IEEE_MAC_ADDRESS>) |
                  <tunnel-encapsulation> | <tunnel-decapsulation> |
                  <logical-tunnel> |
                  <rib-name>

<egress-interface> ::= <INTERFACE_IDENTIFIER>

<nexthop-special> ::= <DISCARD> | <DISCARD_WITH_ERROR> |
                    (<RECEIVE> [<COS_VALUE>])

<nexthop-lb> ::= <NEXTHOP_LB_WEIGHT> <nexthop>
                (<NEXTHOP_LB_WEIGHT> <nexthop>) ...

<nexthop-protection> = <NEXTHOP_PREFERENCE> <nexthop>
                      (<NEXTHOP_PREFERENCE> <nexthop>)...

<nexthop-rotate> ::= <nexthop> <nexthop> ...

<nexthop-chain> ::= <nexthop> ...

<logical-tunnel> ::= <tunnel-type> <TUNNEL_NAME>
<tunnel-type> ::= <IPV4> | <IPV6> | <MPLS> | <GRE> | <VxLAN> | <NVGRE>

<tunnel-encapsulation> ::= (<IPV4> <ipv4-header>) |
                           (<IPV6> <ipv6-header>) |
                           (<MPLS> <mpls-header>) |
                           (<GRE> <gre-header>) |
                           (<VXLAN> <vxlan-header>) |
                           (<NVGRE> <nvgre-header>)

<ipv4-header> ::= <SOURCE_IPv4_ADDRESS> <DESTINATION_IPv4_ADDRESS>
                 <PROTOCOL> [<TTL>] [<DSCP>]

<ipv6-header> ::= <SOURCE_IPV6_ADDRESS> <DESTINATION_IPV6_ADDRESS>
                 <NEXT_HEADER> [<TRAFFIC_CLASS>]
                 [<FLOW_LABEL>] [<HOP_LIMIT>]

<mpls-header> ::= (<mpls-label-operation> ...)
<mpls-label-operation> ::= (<MPLS_PUSH> <MPLS_LABEL> [<S_BIT>]
                           [<TOS_VALUE>] [<TTL_VALUE>]) |
                           (<MPLS_SWAP> <IN_LABEL> <OUT_LABEL>
                           [<TTL_ACTION>])

<gre-header> ::= <GRE_IP_DESTINATION> <GRE_PROTOCOL_TYPE> [<GRE_KEY>]
<vxlan-header> ::= (<ipv4-header> | <ipv6-header>)
                  [<VXLAN_IDENTIFIER>]
<nvgre-header> ::= (<ipv4-header> | <ipv6-header>)
                  <VIRTUAL_SUBNET_ID>
                  [<FLOW_ID>]

<tunnel-decapsulation> ::= ((<IPV4> <IPV4_DECAP> [<TTL_ACTION>]) |
                           (<IPV6> <IPV6_DECAP> [<HOP_LIMIT_ACTION>]) |
                           (<MPLS> <MPLS_POP> [<TTL_ACTION>]))

```

Рисунок 5. Грамматика rBNF для RIB.

## 6.1. Разъяснение грамматики nexthop

Значение nexthop указывает следующий элемент сети для пересылки трафика, а также служит для распределения нагрузки, защиты с использованием приоритета и групповой передачи с применением репликации. Это явно указано в

грамматике. В nexthop имеется рекурсия для применения в сложных случаях, подобных описанному в параграфе 7.2.6. Списки списков.

## 7. Применение грамматики RIB

Грамматика RIB является очень обобщённой и охватывает множество функций и свойств. В этом разделе приведены примеры использования объектов в грамматике RIB и некоторых вариантов применения.

### 7.1. Использование предпочтений маршрутов

С помощью приоритета маршрута клиент может заранее установить альтернативные пути в сети. Например, если в OSPF задан приоритет маршрута 10, другой клиент может организовать маршрут к тому же получателю с приоритетом 20. В результате маршрут OSPF будет помещён в FIB, но при отзыве маршрута OSPF в FIB будет помещён альтернативный путь.

Предпочтение маршрута может также служить для предотвращения атак на службы путём установки предпочтительных маршрутов, которые отбрасывают трафик атаки или направляют его на станцию мониторинга и анализа. Поскольку маршрут устанавливается с высоким приоритетом, он заменит любые маршруты, заданные другими протоколами.

### 7.2. Использование различных типов nexthop

Грамматика RIB позволяет создавать nexthop разных типов, часть которых рассмотрена в последующих параграфах.

#### 7.2.1. Туннельные nexthop

Туннельный nexthop указывает тот или иной туннель. Туннельный трафик инкапсулируется в зависимости от типа туннеля. Туннельные nexthop полезны для абстрагирования от деталей сети, поскольку трафик беспрепятственно пересекает границы сетей. В конце туннеля трафик декапсулируется. Таким образом, грамматика поддерживает два типа операций - инкапсуляцию и декапсуляцию.

#### 7.2.2. Списки репликации

Можно создать список репликации для отправки копий нескольким адресатам. Получатели, в свою очередь, могут быть производными nexthop (на уровне, поддерживаемом сетевым устройством). Примеры репликации включают передачу от одного к нескольким (point to multipoint) и широковещания. Список репликации (на простейшем уровне) имеет вид

```
<nexthop> ::= <NEXTHOP_REPLICATE> <nexthop> [ <nexthop> ... ]
```

Это можно вывести из грамматики, как показано ниже.

```
<nexthop> ::= <nexthop-replicate>
<nexthop> ::= <NEXTHOP_REPLICATE> <nexthop> <nexthop> ...
```

#### 7.2.3. Взвешенные списки

Взвешенные списки служат для распределения трафика между несколькими nexthop. С точки зрения моделирования взвешенный список очень похож на список репликации, но каждый элемент списка **должен** иметь весовой атрибут NEXTHOP\_LB\_WEIGHT. Взвешенный список (на простейшем уровне) имеет вид

```
<nexthop> ::= <NEXTHOP_LOAD_BALANCE> (<nexthop> <NEXTHOP_LB_WEIGHT>)
[ (<nexthop> <NEXTHOP_LB_WEIGHT>) ... ]
```

Это можно вывести из грамматики, как показано ниже.

```
<nexthop> ::= <nexthop-lb>
<nexthop> ::= <NEXTHOP_LOAD_BALANCE>
<NEXTHOP_LB_WEIGHT> <nexthop>
(<NEXTHOP_LB_WEIGHT> <nexthop>) ...
<nexthop> ::= <NEXTHOP_LOAD_BALANCE> (<NEXTHOP_LB_WEIGHT> <nexthop>)
(<NEXTHOP_LB_WEIGHT> <nexthop>) ...
```

#### 7.2.4. Защита

Защита «основной-резервный» может быть представлена в виде

```
<nexthop> ::= <NEXTHOP_PROTECTION> <1> <interface-primary>
<2> <interface-backup>
```

Это можно вывести из грамматики, как показано ниже.

```
<nexthop> ::= <nexthop-protection>
<nexthop> ::= <NEXTHOP_PROTECTION> (<NEXTHOP_PREFERENCE> <nexthop>
(<NEXTHOP_PREFERENCE> <nexthop>) ...)
<nexthop> ::= <NEXTHOP_PROTECTION> (<NEXTHOP_PREFERENCE> <nexthop>
(<NEXTHOP_PREFERENCE> <nexthop>))
<nexthop> ::= <NEXTHOP_PROTECTION> ((<NEXTHOP_PREFERENCE> <nexthop-base>
(<NEXTHOP_PREFERENCE> <nexthop-base>))
<nexthop> ::= <NEXTHOP_PROTECTION> (<1> <interface-primary>
<2> <interface-backup>))
```

Трафик можно распределить между несколькими основными nexthop с одним резервным, как показано ниже.

```
<nexthop> ::= <NEXTHOP_PROTECTION> (<1>
<NEXTHOP_LOAD_BALANCE>
(<NEXTHOP_LB_WEIGHT> <nexthop-base>
(<NEXTHOP_LB_WEIGHT> <nexthop-base>) ...)
<2> <nexthop-base>)
```

Резервный nexthop может иметь свой резерв, как показано ниже.

```
<nexthop> ::= <NEXTHOP_PROTECTION> (<1> <nexthop>
<2> <NEXTHOP_PROTECTION> (<1> <nexthop> <2> <nexthop>))
```

### 7.2.5. Цепочки nexthop

Цепочки nexthop позволяют выполнить над пакетом несколько операций путём их логического объединения. Например, при поступлении пакета VPN на интерфейс WAN для его пересылки в нужный интерфейс VPN нужно извлечь из пакета метку VPN перед его отправкой. Используя цепочку nexthop можно объединить операции выталкивания заголовка MPLS и передачи на определённый интерфейс egress-interface. Это можно вывести из грамматики, как показано ниже.

```
<nexthop-chain> ::= <nexthop> <nexthop>
<nexthop-chain> ::= <nexthop-base> <nexthop-base>
<nexthop-chain> ::= <tunnel-decapsulation> <egress-interface>
<nexthop-chain> ::= (<MPLS> <MPLS_POP>) <interface-outgoing>
```

Элементы цепочки nexthop обрабатываются слева направо.

Цепочки nexthop также позволяют поместить один или несколько заголовков в исходящий пакет. Примером может служить псевдопровод, обеспечиваемый MPLS через какой-либо транспорт (скажем, MPLS или GRE) или расширяемая виртуальная ЛВС (Virtual eXtensible Local Area Network или VXLAN) по протоколу IP. Цепочка nexthop позволяет клиенту RIB разбить программирование nexthop на независимые части (по одной на инкапсуляцию). Простой пример MPLS через GRE можно представить в виде

```
<nexthop-chain> ::= (<MPLS> <mpls-header>) (<GRE> <gre-header>)
<interface-outgoing>
```

Это можно вывести из грамматики, как показано ниже.

```
<nexthop-chain> ::= <nexthop> <nexthop> <nexthop>
<nexthop-chain> ::= <nexthop-base> <nexthop-base> <nexthop-base>
<nexthop-chain> ::= <tunnel-encapsulation> <tunnel-encapsulation>
<egress-interface>
<nexthop-chain> ::= (<MPLS> <mpls-header>) (<GRE> <gre-header>)
<interface-outgoing>
```

### 7.2.6. Списки списков

Список списков является производной конструкцией. Одним из примеров использования такой конструкции служит репликация трафика нескольким получателям с распределением нагрузки. Иными словами, для каждой ветви дерева репликации имеется несколько интерфейсов, между которыми нужно распределить трафик. Внешним списком является список репликации для группового трафика, а внутренним - взвешенный список для распределения нагрузки. Рассмотрим пример элемента сети, реплицирующего трафик двум другим сетевым элементам. Трафик к первому элементу нужно поровну разделить между двумя интерфейсами - outgoing-1-1 и outgoing-1-2. Трафик ко второму элементу следует разделить между тремя интерфейсами outgoing-2-1, outgoing-2-2, outgoing-2-3 в соотношении 20:20:60. Это можно вывести из грамматики, как показано ниже.

```
<nexthop> ::= <nexthop-replicate>
<nexthop> ::= <NEXTHOP_REPLICATE> (<nexthop> <nexthop>...)
<nexthop> ::= <NEXTHOP_REPLICATE> (<nexthop> <nexthop>)
<nexthop> ::= <NEXTHOP_REPLICATE> ((<NEXTHOP_LOAD_BALANCE> <nexthop-lb>)
(<NEXTHOP_LOAD_BALANCE> <nexthop-lb>))
<nexthop> ::= <NEXTHOP_REPLICATE> ((<NEXTHOP_LOAD_BALANCE>
(<NEXTHOP_LB_WEIGHT> <nexthop>
(<NEXTHOP_LB_WEIGHT> <nexthop>) ...))
((<NEXTHOP_LOAD_BALANCE>
(<NEXTHOP_LB_WEIGHT> <nexthop>
(<NEXTHOP_LB_WEIGHT> <nexthop>) ...))
<nexthop> ::= <NEXTHOP_REPLICATE> ((<NEXTHOP_LOAD_BALANCE>
(<NEXTHOP_LB_WEIGHT> <nexthop>
(<NEXTHOP_LB_WEIGHT> <nexthop>))
((<NEXTHOP_LOAD_BALANCE>
(<NEXTHOP_LB_WEIGHT> <nexthop>
(<NEXTHOP_LB_WEIGHT> <nexthop>)
(<NEXTHOP_LB_WEIGHT> <nexthop>))
<nexthop> ::= <NEXTHOP_REPLICATE> ((<NEXTHOP_LOAD_BALANCE>
(<NEXTHOP_LB_WEIGHT> <nexthop>
(<NEXTHOP_LB_WEIGHT> <nexthop>))
((<NEXTHOP_LOAD_BALANCE>
(<NEXTHOP_LB_WEIGHT> <nexthop>
(<NEXTHOP_LB_WEIGHT> <nexthop>)
(<NEXTHOP_LB_WEIGHT> <nexthop>))
<nexthop> ::= <NEXTHOP_REPLICATE>
((<NEXTHOP_LOAD_BALANCE>
(50 <outgoing-1-1>)
(50 <outgoing-1-2>)))
((<NEXTHOP_LOAD_BALANCE>
(20 <outgoing-2-1>)
(20 <outgoing-2-2>)
(60 <outgoing-2-3>)))
```

## 7.3. Групповая передача

Групповая передача IP включает сопоставление пакетов по (S,G) или (\*,G), где S (Source) указывает источник, а G (Group) - префиксы IP. При совпадении пакет реплицируется одному или нескольким получателям. Подписка получателей на multicast-рассылки выходит за рамки этого документа.

При групповой передаче на основе PIM пакеты IP пересылаются по дереву групповой рассылки IP (multicast tree). Узлами нисходящего направления в каждой точке этого дерева являются адреса IP. Это можно представить как список репликации (параграф 7.2.2).

При групповой передаче на основе MPLS пакеты пересылаются в P2MP<sup>1</sup> LSP. Узел nexthop для P2MP LSP можно представить в грамматике nexthop как <logical-tunnel> (идентификатор P2MP LSP) или список репликации (параграф 7.2.2) <tunnel-encapsulation>, где каждый элемент tunnel-encapsulation представляется одним нисходящим MPLS nexthop.

## 8. Масштабные операции RIB

В этом разделе обсуждаются требования к расширяемости модели данных RIB. Модели RIB следует поддерживать обработку большого объема операций, чтобы приложения RIB можно было внедрять в больших сетях.

### 8.1. Чтение RIB

**Должна** поддерживаться группировка множества объектов в одном сообщении при передаче сетевым устройством данных RIB клиенту RIB. Модель данных **должна** позволять клиенту RIB запрашивать у сетевого устройства большой объем данных.

### 8.2. Запись в RIB

**Должна** поддерживаться группировка множества объектов в одном сообщении, когда клиент RIB хочет записать данные в RIB. Отклик от сетевого устройства **должен** включать код возврата для каждой операции записи из такого сообщения.

### 8.3. События и уведомления RIB

Возможны случаи, когда одно событие в сети вызывает множество событий и/или уведомлений, передаваемых элементом сети клиенту RIB. С другой стороны, синхронность нескольких одновременных событий может требовать от сетевого устройства отправки множества событий и/или уведомлений клиенту RIB. Создаваемые сетевым устройством события и уведомления **должны** поддерживать их группировку в одно сообщение.

## 9. Вопросы безопасности

Заданная в этом документе информационная модель определяет схему для моделей данных, предназначенных для доступа по протоколам управления сетью, таким как NETCONF [RFC6241] или RESTCONF [RFC8040]. Нижним уровнем NETCONF служит защищенный транспорт с обязательной поддержкой SSH (Secure Shell) [RFC6242]. Нижним уровнем RESTCONF служит протокол HTTPS с обязательной поддержкой защиты на транспортном уровне (TLS) [RFC8446].

Модель управления доступом NETCONF [RFC8341] обеспечивает возможность разрешить доступ лишь определённым пользователям NETCONF или RESTCONF к заранее заданному подмножеству операций NETCONF или RESTCONF и содержимого.

Информационная модель RIB задаёт операции чтения и записи для сетевых устройств. Эти устройства могут считаться чувствительными или уязвимыми в некоторых сетевых средах. Операции записи в эти сетевые устройства без надлежащей защиты могут оказывать негативное влияние на работу сети. По этой причине при разработке моделей данных рекомендуется учитывать указанные ниже факторы.

- Нужно требовать использования средств контроля подлинности и полномочий протокола NETCONF или RESTCONF.
- Следует расширить предельные объёмы данных, которые могут быть записаны или обновлены удаленным узлом, созданным с учётом требования защиты для модели данных RIB.
- Следует раскрывать конкретную модель данных RIB, реализованную через модели NETCONF/RESTCONF.

## 10. Взаимодействие с IANA

Этот документ не требует действий IANA.

## 11. Литература

### 11.1. 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>>.
- [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., «ping 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>>.
- [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](https://www.rfc-editor.org/info/rfc8174), 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](https://www.rfc-editor.org/info/rfc8341), DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8446] Rescorla, E., «The Transport Layer Security (TLS) Protocol Version 1.3», [RFC 8446](https://www.rfc-editor.org/info/rfc8446), DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

<sup>1</sup>Point-to-Multipoint - от одного ко многим.

## 11.2. Дополнительная литература

- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, «Multi-Topology (MT) Routing in OSPF», RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, «M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-Iss)», RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC5511] Farrel, A., «Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications», RFC 5511, DOI 10.17487/RFC5511, April 2009, <<https://www.rfc-editor.org/info/rfc5511>>.
- [RFC7920] Atlas, A., Ed., Nadeau, T., Ed., and O. Ward, «Problem Statement for the Interface to the Routing System», RFC 7920, DOI 10.17487/RFC7920, June 2016, <<https://www.rfc-editor.org/info/rfc7920>>.
- [RFC8431] Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini, S., and N. Bahadur, «A YANG Data Model for the Routing Information Base (RIB)», RFC 8431, DOI 10.17487/RFC8431, September 2018, <<http://www.rfc-editor.org/info/rfc8431>>.

## Благодарности

Авторы благодарны Ron Folkes, Jeffrey Zhang, сопредседателям рабочей группы и рецензентам за их комментарии и предложения к документу. На конференции I2RS Interim в апреле 2013 г. в разработку информационной модели RIB внесли свой вклад Wes George, Chris Liljenstolpe, Jeff Tantsura, Susan Hares, Fabian Schneider.

## Адреса авторов

**Nitin Bahadur** (editor)  
Uber  
900 Arastradero Rd  
Palo Alto, CA 94304  
United States of America  
Email: [nitin\\_bahadur@yahoo.com](mailto:nitin_bahadur@yahoo.com)

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

Jan Medved  
Cisco  
Email: [jmedved@cisco.com](mailto:jmedved@cisco.com)

## Перевод на русский язык

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