Internet Engineering Task Force (IETF)
Request for Comments: 7921
Category: Informational
ISSN: 2070-1721

A. Atlas
Juniper Networks
J. Halpern
Ericsson
S. Hares
Huawei
D. Ward
Cisco Systems
T. Nadeau
Brocade

June 2016

An Architecture for the Interface to the Routing System

Архитектура интерфейса с системой маршрутизации

Аннотация

Этот документ описывает архитектуру IETF[1] для стандартного программного интерфейса для переноса состояний в систему маршрутизации Internet и из неё. Документ описывает архитектуру, её блоки и интерфейсы на высоком уровне, уделяя особое внимание элементам, которые должны быть стандартизованы как часть интерфейса с системой маршрутизации (Interface to the Routing System или I2RS).

Статус документа

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

Документ является результатом работы IETF и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. 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 http://www.rfc-editor.org/info/rfc7921.

Оглавление

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

Маршрутизаторы, формирующие инфраструктуру маршрутизации Internet, поддерживают состояние с различными уровнями детализации и функциями. Например, типичный маршрутизатор поддерживает базу маршрутных данных (Routing Information Base или RIB) и реализует протоколы маршрутизации, такие как OSPF, IS-IS, BGP, для обмена сведениями о доступности, топологии, состояния протоколов и другой информацией о состоянии сети с другими маршрутизаторами. Маршрутизаторы преобразуют все эти данные в записи пересылки, служащие для передачи пакетов и потоков между элементами сети. Плоскость пересылки и записи пересылки содержат сведения об активном состоянии, описывающие ожидаемое и наблюдаемое поведение маршрутизатора, которые нужны сетевым приложениям. Ориентированным на сеть приложениям нужен простой доступ к этой информации, чтобы знать топологию сети, проверять установку запрограммированного состояния в плоскости пересылки, измерять поведение различных потоков, маршрутов и элементов пересылки, а также понимать настроенные и активные состояния маршрутизаторов. Таким приложениям также нужен простой доступ к интерфейсам, позволяющим программировать и контролировать состояния, связанные с пересылкой.

Этот документ задаёт архитектуру базового, основанного на стандартах интерфейса для этой информации. Интерфейс с системой маршрутизации (I2RS) облегчает контроль и наблюдение за связанным с маршрутизацией состоянием (например, состоянием менеджера RIB в элементе маршрутизации), а также позволяет создавать ориентированные на сеть приложения для современных маршрутизируемых сетей. I2RS - это программный асинхронный интерфейс для обмена состояниями в системе маршрутизации Internet. Эта архитектура I2RS признает, что система маршрутизации и операционная система (Operating System или OS) маршрутизатора предоставляют полезные сведения, которые приложения могут использовать для достижения своих целей. Эти ориентированные на сети приложения могут применять программный интерфейс I2RS для создания новых способов комбинированного извлечения данных маршрутизации Internet, анализа этих данных и установки состояний маршрутизаторов.

Основой I2RS являются чёткие модели данных, определяющие семантику информации, которая может быть записана и прочитана. I2RS обеспечивает приложениям возможности настраивать поведение сети, одновременно используя имеющуюся системы маршрутизации. I2RS предоставляет приложениям (включая контроллеры) платформу для регистрации и запросов соответствующих каждому конкретному приложению сведений.

Хотя архитектура I2RS является достаточно общей для поддержки информационных моделей и моделей различных данных, а аспекты решения I2RS могут быть полезны не только в сфере маршрутизации, I2RS и этот документ сосредоточены на интерфейсе для данных маршрутизации.

Безопасность важна для нового интерфейса I2RS, поэтому в разделе 4 приведён обзор вопросов безопасности для архитектуры I2RS. Детальные требования к безопасности протокола I2RS заданы в [I2RS-PROT-SEC], а требования для сред, в которых работает протокол I2RS, в [I2RS-ENV-SEC].

1.1. Мотивы создания архитектуры I2RS

Имеется 4 основных мотива создания архитектуры I2RS. Во-первых, это потребность в программно, асинхронном интерфейсе, обеспечивающем быстрый интерактивный доступ к элементарным (atomic) операциям. Во-вторых, это доступ к структурированным сведениям и состоянию, которые часто невозможно напрямую настроить или промоделировать в имеющихся реализациях и протоколах настройки. В-третьих, это возможность подписки на структурируемые, фильтруемые уведомления о событиях от маршрутизаторов. В-четвёртых, работа I2RS должна основываться на моделях данных для возможности расширения и предоставления стандартных моделей данных, которые будут применять сетевые приложения.

I2RS описывается как асинхронный программный интерфейс, основные свойства которого представлены в разделе 5 [RFC7920].

Архитектура I2RS облегчает получение сведений от маршрутизаторов и обеспечивает возможность не только читать конкретные данные, но и подписываться на целевые информационные потоки, фильтруемые события и заданные порогами события.

Такой интерфейс также облегчает внедрение эфемерных состояний в систему маршрутизации. Эфемерным состоянием маршрутизатора является состояние, которое не сохраняется при перезагрузке устройства маршрутизации или программы, обслуживающей I2RS на таком устройстве. Не относящиеся к маршрутизации протоколы или приложения могут внедрять состояние в элемент маршрутизации через соответствующие функции I2RS и это состояние будет распространяться в протоколы маршрутизации и сигнализации и/или применяться локально (например, программами, размещёнными в плоскости пересылки). I2RS разрешает лишь изменения состояния, которые можно реализовать путём локальной настройки, а прямое воздействие на внутренние детали протокола, динамически определяемые данными, не допускается.

1.2. Обзор архитектуры

На рисунке 1 показана базовая архитектура I2RS между приложениями, использующими I2RS, их связанными клиентами I2RS и агентами I2RS. Приложения обращаются к службам I2RS через клиентов I2RS и один клиент может предоставлять доступ нескольким приложениям. На рисунке также показаны типы моделей данных, связанных с системой маршрутизации (динамическая и статическая конфигурация, локальная конфигурация, настройки маршрутизации и сигнализации), к которым модели данных агента I2RS могут обращаться или дополнять их.

Рисунок 1 поход на рисунок 1 из [RFC7920], но здесь показаны дополнительные детали использования приложениями клиентов I2RS для взаимодействия с агентами I2RS. Дано также логическое представления моделей данных, связанных с системой маршрутизации, а не функциональное представление (RIB, FIB[1], топологии, политике, протоколах маршрутизации и сигнализации и т. п.).

На рисунке 1 каждый из клиентов A и B обеспечивает доступ к одному приложению (A и B, соответственно), а клиент P предоставляет доступ к нескольким приложениям.

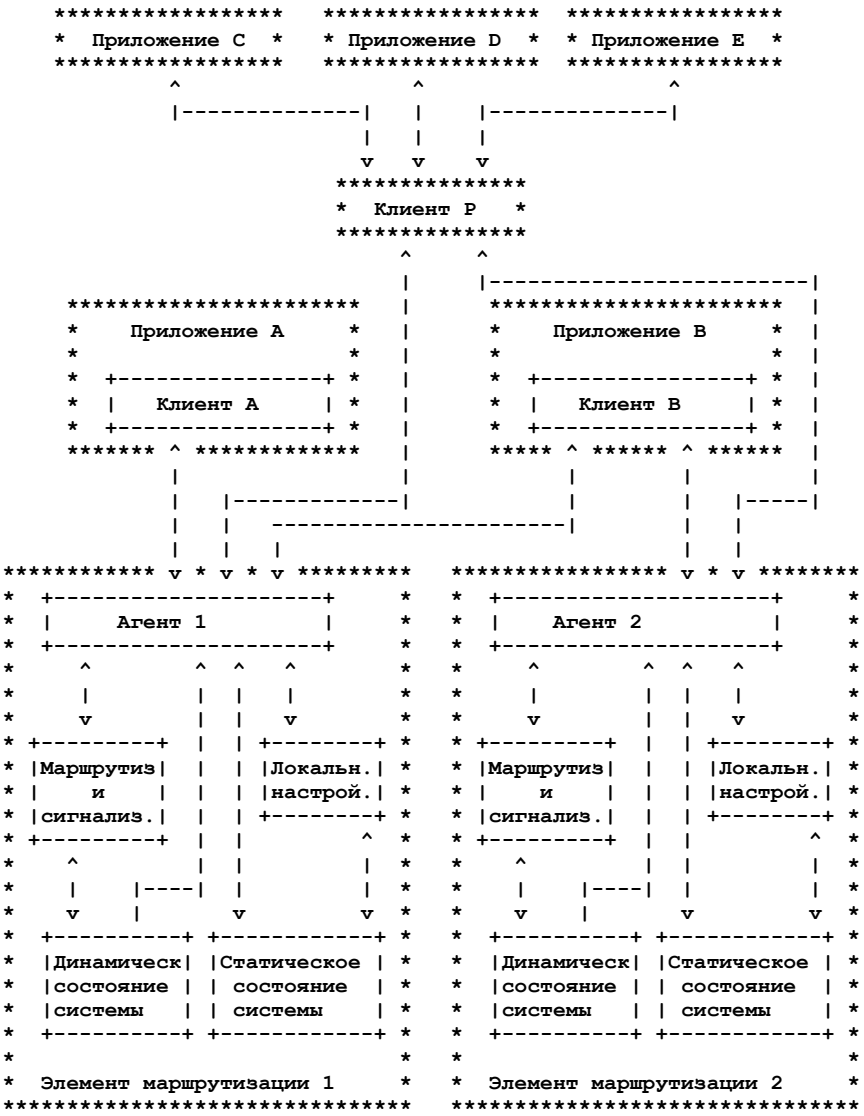

Рисунок 1. Архитектура клиентов и агентов I2RS.

Приложения могут обращаться к службам I2RS через локальных или удалённых клиентов. Локальный клиент работает на одном устройстве с системой маршрутизации, удалённый - через сеть. На рисунке приложения A и B получают услуги I2RS через локальных клиентов, C, D, E - через удалённого. Детали взаимодействия приложений с удаленным клиентом выходят за рамки I2RS.

[1]Forwarding Information Base - база сведений о пересылке.

Клиент I2RS может иметь доступ к 1 или нескольким агентам I2RS. На рисунке 1 клиенты B и P обращаются к агентам I2RS 1 и 2. Агент I2RS может обслуживать 1 или множество клиентов. На рисунке 1 агент I2RS 1 предоставляет услуги клиентам A, B и P, а агент 2 - клиентам B и P.

Агенты и клиенты I2RS взаимодействуют по асинхронному протоколу, поэтому один клиент может передавать несколько одновременных запросов одному или нескольким агентам. Агент может обрабатывать одновременно множество запросов от одного или множества клиентов.

Агент I2RS обеспечивает доступ для чтения и записи выбранных данных на элементе маршрутизации, организованных в службы I2RS. В разделе 4 описано управление доступом через механизмы аутентификации и контроля доступа. На рисунке 1 показано, как агенты I2RS могут записывать статическое эфемерное состояние (например, записи RIB) и считывать данные динамического состояния (например, MPLS LSP-ID[1] или число активных партнёров BGP).

В дополнение к доступу для чтения и записи агент I2RS позволяет клиентам подписываться на различные типы уведомлений о событиях, влияющих на разные экземпляры объектов. Одним из примеров уведомления о таком событии (не связанном с созданием, изменением или удалением объекта) является распознавание nexthop в RIB, позволяющее менеджеру RIB установить его на уровне пересылки как часть конкретного маршрута. Детали этого описаны в параграфах 7.6. Уведомления и 7.7. Сбор сведений.

Областью действия I2RS является определение взаимодействий между агентом и клиентом I2RS и подобающего их поведения.

Routing Element - элемент маршрутизации

Routing Element реализует то или иное подмножество системы маршрутизации. Элемент не обязательно связан с плоскостью пересылки. Примеры Routing Element включают:

- маршрутизаторы с плоскостью пересылки и менеджером RIB, работающим с протоколом IS-IS, OSPF, BGP, PIM и т. п.;
- узлы BGP, действующие как рефлекторы маршрутов (Route Reflector);
- маршрутизаторы с коммутацией по меткам (Label Switching Router или LSR), реализующие RSVP-TE, OSPF-TE, протокол обмена с элементами расчёта пути (Path Computation Element или PCE) и имеет плоскость пересылки и связанный менеджер RIB;
- серверы, на которых работают протоколы IS-IS, OSPF, BGP и применяется протокол разделения элементов управления и пересылки (Forwarding and Control Element Separation или ForCES) для управления удалённой плоскостью пересылки.

Элементом маршрутизации можно управлять локально через командный интерфейс (command-line interface или CLI), по протоколу SNMP или протокола управления сетью (Network Configuration Protocol или NETCONF).

Routing and Signaling - маршрутизация и сигнализация

Этот блок представляет часть элемента маршрутизации, участвующую в системе маршрутизации Internet. Это включает не только стандартизованные протоколы (например, IS-IS, OSPF, BGP, PIM, RSVP-TE, LDP и т. п.), но и уровень менеджера RIB.

Local Configuration - локальная конфигурация

Поведение «черного ящика» во взаимодействии между эфемерным состоянием, которое I2RS устанавливает в элементе маршрутизации. Локальная конфигурация определяется этим документом, а поведение задаёт протокол I2RS.

Dynamic System State - динамическое состояние системы

Агенту I2RS нужен доступ к состоянию элемента маршрутизации сверх того, что содержится в подсистеме маршрутизации. Такие состояния могут включать различные счётчики, статистику, данные потоков и локальные события. Это подмножество рабочего состояния, которое требуется сетевым приложениям на основе I2RS, не включённое в сведения маршрутизации и сигнализации. Способы предоставления этих сведений агенту I2RS выходят за рамки документа, но стандартизованные сведения и модели раскрываемых данных являются частью I2RS.

Static System State - статическое состояние системы

Агенту I2RS нужен доступ к статическому состоянию элемента маршрутизации сверх того, что содержится в подсистеме маршрутизации. Примером такого состояния является задание поведения очередей для интерфейса или трафика. Изменение и получение агентом I2RS этих сведений выходит за рамки документа но стандартизованные сведения и модели раскрываемых данных являются частью I2RS.

I2RS agent - агент I2RS

См. определение в разделе 2.

Application - приложение

Сетевое приложение, которому нужно наблюдать за сетью или манипулировать ею для достижения своих требований к обслуживанию.

I2RS client - клиент I2RS

См. определение в разделе 2.

Как показано на рисунке 1, клиент I2RS может взаимодействовать с несколькими агентами I2RS. Агент I2RS тоже может взаимодействовать с несколькими клиентам, отвечать на их запросы, отправлять уведомления и т. п. Своевременные уведомления важны, поскольку позволяют одновременно работающим приложениям получать актуальные сведения о состоянии сети.

Каждый клиент может отправить агенту I2RS множество операций записи. Для сохранения простоты протокола двум клиентам не следует записывать (изменять) одну и ту же часть информации у агента I2RS, это будет ошибкой. Однако такие конфликты могут возникать и в параграфе 7.8. Управление из нескольких мест описано, как агент I2RS разрешает конфликты, используя значения приоритета и обслуживая запросы в порядке поступления. Архитектура I2RS задаёт поведение в таких случаях просто для предсказуемости. Такая предсказуемость упрощает обработку ошибок и предотвращает флуктуации. Для других типов ошибок следует предусматривать меры в сетевых приложениях и системах управления.

Хотя нескольким клиентам I2RS может потребоваться предоставить данные в один список (например, список префиксов или фильтров), это не является ошибкой и должно корректно обрабатываться. Нюансы обработки конфликтов при записи следует обрабатывать в информационной модели.

[1]Label Switched Path Identifier - идентификатор пути с коммутацией по меткам.

Целью архитектуры I2RS является обеспечение предсказуемого поведения при таких ошибках и предоставление отчётов заинтересованным клиентам. Детали соответствующей политики рассмотрены в параграфе 7.8. Управление из нескольких мест. Такой же механизм правил (приоритет для клиента I2RS) применяется к взаимодействиям агента I2RS с CLI/SNMP/NETCONF, как описано в параграфе 6.3. Взаимодействие с локальной конфигурацией.

Кроме того, следует отметить возможность непрямых опосредованной зависимости между операциями записи. Основным примером является запись двух разных, но перекрывающихся префиксов с разным поведением пересылки. Обнаружение и предотвращение таких взаимодействий выходит за рамки I2RS и остаётся за разработчиками агентов.

2. Терминология

Ниже приведены определения используемых в документе терминов.

agent или I2RS agent - агент (I2RS)

Агент I2RS поддерживает предоставляемые I2RS услуги от подсистемы маршрутизации локальной системы путём взаимодействия с элементом маршрутизации для обеспечения заданного поведения. Агент I2RS понимает протокол I2RS и модет взаимодействовать с клиентами I2RS.

client или I2RS client - клиент (I2RS)

Клиент реализует протокол I2RS, применяет его для взаимодействия с агентами I2RS и использует службы I2RS для выполнения задач. Он взаимодействует с другими элементами политики, обеспечения и конфигурации системы с помощью средств, выходящих за рамки I2RS. Клиент взаимодействует с агентами I2RS для сбора сведений от систем маршрутизации и пересылки. На основе информации и ориентированных на правила взаимодействий клиент I2RS может взаимодействовать с агентами I2RS для изменения состояния связанных систем маршрутизации для достижения эксплуатационных целей. Клиента I2RS можно рассматривать как часть приложения, которое использует и поддерживает I2RS и может быть программной библиотекой.

service или I2RS service - служба (I2RS)

В I2RS службой называется набор функций, связанных с доступом к состояниям, вместе с правилами, контролирующими их применение. Предполагается, что службы будут представлены в моделях данных. Например, служба RIB может быть примером службы, предоставляющей доступ вк поля состояния в RIB устройства.

read scope - область чтения

Областью чтения клиента I2RS для агента I2RS считается набор сведений, которые клиент I2RS уполномочен считывать с агента I2RS. Область чтения задаёт ограничения доступа для просмотра наличия данных и считывания их значений.

notification scope - область уведомлений

Областью уведомления является набор событий и связанных с ними сведений, для которых клиент I2RS может запросить выталкивание (push) агентом I2RS. Клиенты I2RS имеют возможность регистрироваться для конкретных событий и потоков информации, но должны ограничиваться в соответствии с правами доступа, связанными с их областью уведомлений.

write scope - область записи

Областью записи является набор значений полей, для которых клиенту I2RS разрешена запись (добавление, изменение, удаление). Этот доступ может быть ограничивать набор изменяемых или создаваемых данных, а также наборы устанавливаемых значений.

scope - область действия

Когда не указан тип области (чтение, запись, уведомления), термин «область действия» относится ко всем.

resources - ресурсы

Ресурсами считаются связанные с I2RS расход памяти, хранилища или исполнения, которые разрешено клиенту для выполнения операций I2RS. Размер каждого ресурса, доступного клиенту в контексте конкретного агента, может ограничиваться на основе роли клиента в плане безопасности. Примером ресурса является число уведомлений, на которые клиент может зарегистрироваться. Имеются ресурсы, относящиеся к протоколу и сети.

role или security role - роль (роль в безопасности)

Роль в безопасности задаёт область действия, ресурсы, приоритеты и т. п., которые имеет клиент или агент. Если с идентификатором связано несколько ролей в системе безопасности, разрешены все операции, разрешённые в какой-либо из этих ролей. Несколько идентификаторов могут использовать одну роль в безопасности.

identity - идентификатор (отождествление)

С клиентом связан единственный конкретный идентификатор. Состояние можно отнести к конкретному идентификатору. Возможно использование нескольких коммуникационных каналов для одного идентификатора, в этом случае предполагается координация таких коммуникаций соответствующим клиентом.

identity and scope - идентификатор и область действия

С идентификатором может быть связано несколько ролей, каждая из которых имеет свою область действия, и такой идентификатор может использовать комбинацию этих областей. Каждый идентификатор имеет:

- область чтения - логическое объединение (OR - или) областей чтения всех ролей;
- область записи - логическое объединение (OR - или) областей записи всех ролей;
- область уведомления - логическое объединение (OR - или) областей уведомления всех ролей.

secondary identity - вторичный идентификатор

Клиент I2RS может предоставлять вторичный необрабатываемый (opaque) идентификатор, который агент I2RS не интерпретирует. Примером применения такого идентификатора является случай, когда клиент I2RS является посредником для нескольких приложений и нужно отслеживать, какое из них запросило конкретную операцию.

ephemeral data - эфемерные данные

Эфемерными считаются данные, не сохраняющиеся при перезапуске (программы или устройства) или отключении питания. Это могут быть настраиваемые данные или данные, записанные в результате операций маршрутизатора. Система не способна (самостоятельно) вернуться к прежнему эфемерному состоянию конфигурации.

group - группа

В модели управления доступом NETCONF [RFC6536] группой называется административная группа, в которой чётко разделяются учётная запись root и пользовательские учётные записи с меньшими привилегиями. Группой считается также одно отождествление (например, root), применяемое группой пользователей.

routing system/subsystem - система (подсистема) маршрутизации

Системой или подсистемой маршрутизации является набор программ и/или оборудования, определяющего, куда пересылать пакеты. Агент I2RS является частью системы маршрутизации. Пакетами могут называться кадры L1 и

L2, пакеты L3. В системе маршрутизации Internet пакетами считаются пакеты IP на уровне L3. Подсистемой маршрутизации называют программы и/или оборудования, являющиеся частью более крупной системы.

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 [RFC2119].

3. Основные свойства архитектуры

Ниже разъясняются несколько основных свойств архитектуры I2RS (простота, расширяемость, управляемые моделями программные интерфейсы). Такие свойства, как производительность и масштабирование не рассматриваются здесь, поскольку они описаны в [RFC7920] и могут различаться в зависимости от варианта применения.

3.1. Простота

За прошедшие годы было предпринято много усилий по улучшению доступа к сведениям, доступным из системы маршрутизации и пересылки. Удобное представление таких сведений системе управления сетью и приложениям обеспечивает много очевидных преимуществ, однако возникают две взаимосвязанные проблемы. Во-первых, объем и разнообразие потенциально доступных данных очень велико. Во-вторых, различия в структуре данных и типах требуемых операций, как правило, усложняют протокол.

Хотя рассматриваемые здесь типы операций сложны по своей природе, важна простота развёртывания и отказоустойчивость I2RS. Добавление сложности сверх необходимого для удовлетворения хорошо известных и понятных потребностей будет препятствовать простоте реализации, отказоустойчивости протокола и его внедрению. Чрезмерно сложные модели данных ведут, как правило, к окостенению наборов информации, пытаясь охватить и описать все возможные варианты, что затрудняет расширяемость.

Поэтому одной из важнейших целей I2RS является сохранение простоты протокола и архитектуры моделирования.

3.2. Расширяемость

Расширяемость протокола и модели данных очень важна. В частности, с учётом ограничений объёма первоначальной работы очень важно, чтобы начальная разработка включала надёжную поддержку расширяемости.

Работы по I2RS выполнялись поэтапно, чтобы обеспечить на каждом этапе пригодные к развёртыванию результаты. У каждого этапа имеется свой набор требований, которым будет соответствовать протокол I2RS и модели данных. Поэтому для моделей данных I2RS желательна простая расширяемость для представления дополнительных аспектов элементов сети и сетевых систем. Следует обеспечивать для моделей данных I2RS простоту интеграции с другими данными. Это повышает важность разработки моделей данных с должной расширяемостью, желательно в привычном и простом виде.

Рабочая группа I2RS задаёт операции для протокола I2RS. Было бы слишком оптимистично предполагать, что не потребуется что-либо ещё по мере расширения масштабов I2RS. Таким образом, важно учитывать расширяемость не только моделей, но и примитивов, а также операций протокола.

3.3. Программные интерфейсы на основе моделей

Важнейшей частью I2RS являются стандартные модели информации и данных со связанной семантикой. Хотя многие компоненты систем маршрутизации стандартизованы, соответствующие модели данных для них ещё недоступны. Каждый маршрутизатор применяет свою информацию, механизмы и интерфейсы, а также разные CLI, что делает стандартный интерфейс для использования приложениями чрезвычайно громоздким для разработки и поддержки. Есть общеизвестные языки моделирования данных, которые можно применить для создания моделей данных I2RS.

Применение управляемой на основе модели архитектуры и протоколов обеспечит важные преимущества для I2RS. Во-первых, это позволит обрабатывать данные управления на основе модели данных, что обеспечит модульную реализацию клиентов и агентов I2RS. Клиенту I2RS нужно реализовать лишь модели данных, к которым тот имеет доступ, а агенту I2RS - лишь те модели, которые он поддерживает. Во-вторых, инструменты могут автоматизировать проверку и манипуляции с данными, что особо ценно как для расширяемости, так и для простоты манипулирования и проверки фирменных (proprietary) моделей данных.

Различным службам, поддерживаемым I2RS, могут соответствовать разные модели данных. Агент I2RS может указывать поддерживаемые им модели.

Цель модели данных заключается в предоставлении определений относящихся к системе маршрутизации сведений, которые могут применяться в работающих сетях. Если данные маршрутизации моделируются впервые, логическая информационная модель может быть стандартизована до создания модели данных.

4. Вопросы безопасности

Эта архитектура I2RS описывает интерфейсы, которые явно требуют серьёзного внимания к безопасности. Архитектура I2RS была разработана для использования имеющихся протоколов обмена данными управления сетью. Двумя протоколами, используемыми в I2RS версии 1, являются NETCONF [RFC6241] и RESTCONF [RESTCONF]. В будущих версиях I2RS могут применяться и другие протоколы.

Процесс разработки протокола I2RS будет включать задание дополнительных требований (включая безопасность) к существующим протоколам для поддержки архитектуры I2RS. После изменения имеющихся протоколов (например, NETCONF или RESTCONF) в соответствии с требованиями I2RS будет проверяться соответствие этим требованиям. При обзоре изменений в имеющихся протоколах для обслуживания архитектуры I2RS необходимо выполнить глубокий анализ безопасности пересмотренных протоколов.

Из-за стратегии применения имеющихся протоколов в архитектуре I2RS этот раздел описывает предполагаемую среду защиты для I2RS с дополнительным рассмотрением a) отождествления и проверки подлинности, b) проверки

полномочий и c) резервирования клиентов. Каждый протокол, предложенный для включения как протокол I2RS, должен оцениваться в плане ограничений, связанных с безопасностью. Детальные требования к протоколу I2RS и среде защиты I2RS будут заданы в этих глобальных средах защиты.

Требования безопасности для протокола I2RS версии 1 заданы в [I2RS-PROT-SEC], а глобальные требования к среде защиты I2RS - в [I2RS-ENV-SEC].

Далее приведено краткое описание предполагаемой среды защиты для I2RS. Агент I2RS, связанный с элементом маршрутизации является доверенной частью этого элемента. Например, это может быть часть поставляемого производителем подписанного программного образа для всего элемента маршрутизации или доверенное и подписанное приложение, установленное оператором. У агента I2RS предполагается отдельный канал проверки подлинности и полномочий, через который он может проверить отождествление и полномочия клиента I2RS. Для поддержки многочисленных и быстрых взаимодействий между агентом и клиентом I2RS предполагается, что агент I2RS может кэшировать доверие в конкретным клиентам I2RS и связанные с ними области полномочий. Это предполагает, что сведения о разрешениях могут устареть в модели вытягивания (pull), пока агент I2RS снова не запросит их, или в модели выталкивания (push), пока канал аутентификации и предоставления полномочий не сможет уведомить агента I2RS об изменениях.

Требуется взаимная проверка подлинности между клиентом и агентом I2RS. Клиент I2RS должен быть способен доверять тому, что агент I2RS подключён к соответствующему элементу маршрутизации для корректного применения операций записи и изменения, а также тому, что он может доверять сведениям, полученным от агента I2RS.

Клиент I2RS не считается доверенным автоматически. Каждый клиент I2RS имеет идентификатор с набором ограничений для области действия. Приложениям, использующим клиентов I2RS следует понимать, что ограничения для клиента I2RS определяются его идентификатором (см. параграф 4.1) и назначенной для этого идентификатора ролью. Роль устанавливает конкретные пределы полномочий для действий, которые клиент I2RS может запросить у агента I2RS (параграф 4.2). Например, один клиент I2RS может быть способен лишь читать таблицу статических маршрутов, а другому может разрешено добавлять эфемерные маршруты в статическую таблицу.

Если клиент I2RS служит посредником для нескольких приложений, поддержка защиты, проверки подлинности и полномочий для таких взаимодействий выходит за пределы области действия, но ничто не мешает посреднику использовать протокол I2RS и отдельный канал проверки подлинности и полномочий. Независимо от механизма, служащий посредником клиент I2RS отвечает за проверку доверия к приложениям и их правам делать конкретные запросы.

К разным аспектам I2RS относятся различные уровни защиты целостности и конфиденциальности, а также защиты от повторного использования (replay). Основной коммуникационный канал, применяемый для аутентификации клиента, который клиент позднее использует для записи данных, требует защиты целостности и конфиденциальности, а также защиты от повторного использования. Соответствующий выбор требуемого по умолчанию транспортного протокола является предпочтительным способом выполнения этих требований.

Другие взаимодействия через I2RS могут не требовать защиты целостности и конфиденциальности, а также защиты от повторного использования. Например, если клиент I2RS подписывается на поток сведений об анонсах префиксов от OSPF, ему может быть нужна защита целостности, но скорей всего не потребуется защита конфиденциальности и защита от повторного использования. Точно так же для потока сведений о статистике интерфейса может не требоваться даже гарантий доставки. В параграфе 7.2 рассматриваются дополнительные коммуникационные каналы и их использование. С точки зрения безопасности важно понимать, что агент I2RS может создать новый канал связи на основе сведений от клиента I2RS (как описано в параграфе 7.2). Например, клиент I2RS может запросить уведомления о некоторых событиях и агент будет создавать канал для передачи таких уведомлений. Поэтому для предотвращения косвенных атак такие запросы должны выполняться в контексте клиента, подлинность и полномочия которого проверены и сообщения этого клиента не могут быть изменены.

4.1. Отождествление и проверка подлинности

Как отмечено выше, все управляющие обмены между клиентом и агентом I2RS должны обеспечиваться аутентификацией и защитой целостности (чтобы содержимое нельзя было изменить незаметно). Кроме того, манипуляции с системой должны быть чётко прописываемыми. В идеальной архитектуре следует защищать даже сбор сведений и уведомления, это может быть инженерным компромиссом при разработке.

Клиенты I2RS могут работать от имени других приложений. Хотя для идентификаторов этих приложений не требуется проверка подлинности полномочий, каждому приложению следует иметь уникальный неинтерпретируемый (opaque) идентификатор, который может быть предоставлен клиентом I2RS агенту I2RS для отслеживания полномочий приложения при операциях (или информирования по идентификаторам). Такое отслеживание операций поддерживает функциональность I2RS по трассировке действий (для поиска неполадок в маршрутизаторах) и регистрации изменений в сети.

4.2. Предоставление полномочий

Все операции с использованием I2RS, как для наблюдения, так и для манипуляций, должны проходить соответствующий контроль полномочий. Этот контроль основывается на идентификаторе и назначенной роли клиента I2RS, выполняющего операции, и агента I2RS в элементе сети. Для одной роли может применяться несколько идентификаторов. Как отмечено выше в определениях терминов «идентификатор» и «роль», при связывании с одним идентификатором нескольких ролей этот идентификатор даёт полномочия на выполнении любой операции, разрешённой для какой-либо из этих ролей.

Агенты I2RS, собирающие сведения и манипулирующие ими, будут действовать от имени клиентов I2RS. Поэтому полномочия для каждой операции будут основаны на более ограниченном из разрешений самого агента и аутентифицированного клиента. Механизм с помощью которого полномочия реализуются в устройстве, выходит за рамки I2RS.

Подходящий или требуемый уровень детализации для области действия может зависеть от конкретной службы I2RS и детализации в реализации. Подход к аналогичной задаче управления доступом определён в модели контроля доступа

NETCONF (NETCONF Access Control Model или NACM) [RFC6536] и позволяет задать произвольные права доступа к экземпляру узла данных при определении значимых изменяемых установок, принятых по умолчанию. Идентификатор в NACM [RFC6536] может быть задан для имени пользователя или группы имён пользователей (например, Root) и это имя связывается с политикой области действия, которая содержится в наборе правил управления доступом. Аналогично ожидается, что идентификатор I2RS свяжет роль с политикой области действия, заданной набором правил контроля доступа. Эта политика может быть указана через локальную конфигурацию, раскрываемую как служба I2RS для манипуляция со стороны уполномоченных клиентов или иным способом (например, службой AAA[1])

Хотя агент I2RS разрешает доступ на основе политики области действия клиента I2RS, это не означает, что требуется доступ для прибытия по конкретному транспортному протоколу или от конкретного клиента I2RS в соответствии с архитектурой I2RS. Применяемая оператором политика области действия может (но не обязана) ограничивать транспортные соединения или идентификаторы, которые могут иметь доступ к локальному агенту I2RS.

Когда клиент I2RS аутентифицирован, его идентификатор предоставляется агенту I2RS и связывает клиента с ролью, с которой связана политика области действия. С одной ролью может быть связано несколько идентификаторов. Такой ролью может быть, например, мониторинг внутренних маршрутов (Internal-Routes-Monitor), которому разрешено считывать часть I2RS RIB, связанную с префиксами IP, применяемыми для внутренних адресов устройства в AS.

4.3. Резервирование клиента

В I2RS должно поддерживаться резервирование (избыточность) клиентов. В простейшем случае это может быть реализовано наличием основного и резервного сетевого приложения, использующих один идентификатор клиента, позволяющий проверить подлинность обоих. Поскольку I2RS не требует постоянного транспортного соединения и поддерживает множество транспортных сессий, это позволяет обеспечить некую базовую избыточность. Однако это не избавляет от необходимости устранения неполадок и регистрации изменений в сети, чтобы получить сведения о фактически активном сетевом приложении. Может регистрироваться по меньшей мере базовая транспортная информация о каждом соединении и времени вместе с идентификатором.

4.4. I2RS в персональных устройствах

Если агент или клиент I2RS тесно связан с человеком (например, агент I2RS работает на телефоне для управления подключением), это может вызывать проблемы приватности в дополнение к обычным проблемам безопасности в I2RS. Одним из примеров взаимодействия I2RS, которое может вызывать проблемы приватности, является возможность отслеживания местоположения телефона. Протоколу I2RS и модели данных следует учитывать возможные проблемы приватности при отмеченном использовании клиентов или агентов.

5. Сетевые приложения и клиент I2RS

Предполагается использование I2RS ориентированными на сеть приложениями в разной архитектуре. Хотя интерфейс между ориентированными на сеть приложениями и клиентом I2RS выходит за рамки I2RS, рассмотрение различных архитектур важно для спецификации I2RS.

В простейшей архитектуре с прямым доступом ориентированное на сеть приложение имеет клиент I2RS как библиотеку или драйвер для взаимодействия с элементом маршрутизации. В архитектуре с посредником (broker) несколько сетевых приложений взаимодействуют незаданными способами с приложением-посредником, содержащим клиент I2RS. Такое приложение требует дополнительной функциональности для проверки подлинности и полномочий ориентированных на сеть приложений. Эти функции выходят за рамки I2RS, но к ним применимы соображения, аналогичные указанным в параграфе 4.2. Предоставление полномочий. Как указано в параграфе 4.1. Отождествление и проверка подлинности, клиенту I2RS у посредника следует задавать разные неинтерпретируемые (opaque) идентификаторы для каждого ориентированного на сеть приложения, которое использует его. Клиент I2RS у посредника может передавать соответствующее значение в качестве идентификатора, который может применяться для отслеживания операций.

В другой архитектуре элемент маршрутизации или ориентированное на сеть приложение, использующие клиент I2RS для доступа к службам на других элементах маршрутизации, может включать агент I2RS для предоставления услуг другим ориентированным на сеть приложениям. Однако там, где потребные для этих услуг сведения и модели данных отличаются от применяемых традиционными элементами маршрутизации, такие модели (по крайней мере изначально) выходят за рамки I2RS. В следующем параграфе рассматривается пример такого сетевого приложения.

5.1. Пример сетевого приложения - менеджер топологии

Менеджер топологии включает клиент I2RS, который использует модели данных и протокол I2RS для сбора сведений о состоянии сети, взаимодействуя напрямую с одним или множеством агентов I2RS. От этих агентов I2RS менеджер топологии собирает данные настройки маршрутизации и рабочего состояния, такие как интерфейс и сведения о путях LSP. Кроме того, менеджер топологии может несколькими способами собирать сведения о состоянии каналов - от моделей I2RS, партнёров BGP-LS [RFC7752], через прослушивание IGP.

Набор функция и собираемой менеджером информации может быть встроен как часть приложения, такого как приложение для расчёта путей. Как автономное приложение, менеджер топологии может быть полезен другим сетевым приложениям, предоставляя согласованную картину состояния пути, доступного через другой интерфейс. Этот интерфейс может использовать тот же протокол I2RS и может предоставлять услуги топологии, используя расширения моделей данных I2RS.

6. Роль и функциональность агента I2RS

Агент I2RS является частью элемента маршрутизации, поэтому он взаимодействует с этим элементом в целом и его различными компонентами.

[1]Authentication, Authorization, and Accounting - проверка подлинности и полномочий, а также учёт.

6.1. Взаимоотношения с элементом маршрутизации

Элемент маршрутизации может быть реализован в самых разных вариантах архитектуры - интегрированный маршрутизатор, расщепленная или распределенная архитектура и т. п. Это не должно влиять на базовое поведение агента I2RS.

Для универсальности и рвсширяемости агент I2RS может отвечать за сбор и доставку больших объёмов данных от разных частей элемента маршрутизации, которые могут размещаться даже в разных физических устройствах. Поэтому для расширяемости и отказоустойчивости важно, чтобы архитектура позволяла использовать распределенный набор компонентов отчётности, передающих собранные сведения от агента I2RS соответствующим клиентам I2RS. В одном маршрутизаторе может быть несколько агентов I2RS и в этом случае они должны имет непересекающиеся наборы информации, которыми они манипулируют.

Для упрощения работы, развёртывания и устранения неполадок важно поддерживать отслеживание запросов, полученных агентом I2RS, и предпринятых действий с помощью общей модели данных.

6.2. Хранилище состояния I2RS

Запросы на изменение состояния передаются агенту I2RS в элементе маршрутизации клиентами I2RS. Агент I2RS отвечает за применение этих изменений в системе с учётом имеющихся полномочий, как отмечено выше. Агент I2RS будет хранить сведения о внесённых изменениях и клиентах, от чьего имени эти изменения внесены. Агент I2RS будет также хранить активные подписки. Эти наборы данных формируют хранилище I2RS. Данные сохраняются агентом, пока клиент не удалит состояние, оно не будет переопределено какой-либо иной операцией, такой как CLI, или устройство не будет перезагружено. Рекомендуется вести содержательный журнал применения и удаления изменений. Применяемые I2RS изменения в элементе маршрутизации не будут сохраняться при перезагрузке этого элемента. Хранилище I2RS не сохраняется при перезагузке элемента маршрутизации, поэтому агент I2RS не будет пытаться повторить изменения после перезагрузки.

6.2.1. Отказ агента I2RS

Предполагается, что у агента I2RS могут происходить отказы независимо от связанного элемента маршрутизации. Это может произойти из-за отключения I2RS в элементе маршрутизации или сбоя самого агента I2RS, который может быть отдельным процессом и даже работать на другом процессоре. Точно так же, как удаляется эфемерное состояние, полученное от отказавшего источника, эфемерное состояние I2RS обычно будет удаляться вскоре после обнаружения отказа или при аккуратном отключении процесса. Для обработки таких отказов агент I2RS **должен** поддерживать два разных уведомления - об аккуратном завершении агента I2RS и о запуске агента I2RS после неожиданного отказа. Эти уведомления описаны ниже, а затем рассмотрены случаи неожиданных отказов и аккуратного отключения.

NOTIFICATION_I2RS_AGENT_TERMINATING

Это уведомление сообщает, что связанный агент I2RS аккуратно отключается и эфемерное состояние I2RS будет удалено. Уведомление может включать временную метку момента отключения агента I2RS. Использование метки предполагает синхронизацию часов и детализацию значения метки не лучше 1 секунды, поскольку большая точность не гарантируется.

NOTIFICATION_I2RS_AGENT_STARTING

Это уведомление сообщает клиентам I2RS, что связанный агент I2RS был запущен. Уведомление включает счётчик загрузок (agent-boot-count), показывающий число перезапусков агента с момента перезагрузки соответствующего элемента маршрутизации. Счётчик позволяет клиенту I2RS определить, был ли агент I2RS перезапущен (отметим, что это уведомление агент I2RS передаёт клиентам I2RS, известным агенту после перезагрузки; способ сохранения агентом I2RS сведений о клиентах I2RS выходит за рамки этой архитектуры).

Возможны два типа отказов, которые различаются своим поведением.

Неожиданный отказ

Агент I2RS даёт неожиданный сбой и не может ни о чем уведомить клиентов. Поскольку I2RS не требует постоянного соединения между клиентом и агентом I2RS, агенту I2RS требуется иметь механизм уведомления клиентов, у которых есть подписка или записанное эфемерное состояние. Таких клиентов I2RS следует кэшировать системе агента I2RS в постоянном хранилище. При запуске агента I2RS ему следует передать уведомление NOTIFICATION_I2RS_AGENT_STARTING каждому кэшированному клиенту.

Аккуратное отключение

В этом случае агент I2RS может выполнять определённую ограниченную работу в рамках отключаемого процесса. Агент I2RS должен передать уведомление NOTIFICATION_I2RS_AGENT_TERMINATING всем своим кэшированным клиентам I2RS. Если агент I2RS перезапускается после аккуратного отключения, он будет передавать уведомление NOTIFICATION_I2RS_AGENT_STARTING каждому кэшированному клиенту.

6.2.2. Начало и завершение

Когда клиент I2RS вносит изменения через протокол I2RS, эти изменения применяются и остаются, пока не будут удалены или не перезагрузится элемент маршрутизации. Сетевое приложение может принимать решения о том, что запрашивать через I2RS, на основе разных условий, которые подразумевают разное время начала и завершения. Эти сложности обрабатывает сетевое приложение, а не I2RS.

6.2.3. Возврат к прежнему состоянию

Агент I2RS может решить, что некое состояние больше не следует применять, а клиент I2RS может указывать агенту удалить состояние, которое клиент применил. В таких случаях будет возвращаться состояние, которое было до взаимодействия между клиентом и агентом I2RS. Обычно это состояние определяется через CLI, NETCONF, SNMP и т. п., агенты I2RS не пытаются сохранить множество альтернативных состояний или определить, к какому из них следует вернуться. Таким образом, модель не похожа на RIB, где хранится множество маршрутов с разным приоритетом (сведения о состоянии I2RS при наличии двух клиентов I2RS приведены в параграфах 1.2 и 7.8)

Клиент I2RS может зарегистрироваться на уведомления (в зависимости от своей области действия для уведомлений) о смене состояния или удалении конкретного клиента I2RS.

6.3. Взаимодействие с локальной конфигурацией

Изменения могут исходить от локальной настройки или I2RS. Изменения и данные, хранимые I2RS, отделены от конфигурации локального устройства, но конфликты между ними должны разрешаться детерминированным способом с соблюдением применяемой оператором политики. Детерминированое изменение является результатом применения базовых правил I2RS, правил системы, элементов управления, заданных политикой оператора, и правил, заданных моделью данных YANG (зачастую в MUST и WHEN для зависимостей).

Элементы управления политики оператора могут задавать, переопределяет ли локальная конфигурация запросы конкретного клиента I2RS и наоборот. Обычно большинство устройств имеет применяемую оператором политику, которая отдаёт приоритет эфемерным изменениям конфигурации клиента I2RS, чтобы те переопределяли локальную конфигурацию.

Элементы управления политики оператора могут быть реализованы многими способами. Одним из них является настройка элементом маршрутизации приоритета для локальной конфигурации и записей эфемерной конфигурации от клиента I2RS. Механизм I2RS будет сравнивать эти значения и выбирать более приоритетную конфигурацию.

Чтобы убедиться в соответствии запросов клиентов I2RS пожеланиям оператора, в модулях данных I2RS имеется общее правило, по которому локальная конфигурация по умолчанию имеет более высокий приоритет, чем эфемерная конфигурация I2RS. Это обусловлено тем, что при отсутствии политки оператора по переопределению эфемерной конфигурацией I2RS локальных настроек будут применяться последние. Это общее правило позволяет устанавилвать агенты I2RS в системах маршрутизации и проверять связь между клиентами и агентами I2RS без переопределения состояния конфигурации агентом I2RS. Если модель данных всегда отдаёт приоритет эфемерному состоянию I2RS, это состояние устанавливается вместо локальной конфигурации. Сведения о локальной конфигурации сохраняются и могут быть восстановлены, когда клиент I2RS удалит эфемерное состояние I2RS. Если приоритет всегда отдаётся локальной конфигурации, требуются некоторые взаимодействия между этой подсистемой и агентом I2RS. Поскольку агент I2RS подключается к подсистеме маршрутизации, он должен также взаимодействовать с локальной конфигурацией для обмена данными модели, чтобы агент I2RS знал детали каждого изменения конфигурации конкретного устройства, которое ему разрешено вносить. Кроме того, когда система определяет, что состояние от клиента I2RS вытеснено, агент I2RS должен уведомить затрагиваемых клиентов I2RS. Исполнение этого системой зависит от реализации.

Очень важно, чтобы применялась политика, основанная на источнике, поскольку разрешение не может основываться на времени. Если просто отдать предпочтение последнему состоянию, это может привести к состязанию и финальное состояние может стать неопределенным.

6.3.1. Примеры локальной конфигурации и эфемерной конфигурации I2RS

Для иллюстрации принципов архитектуры полезен набор примеров. Предположим, что имеется 3 маршрутизатора Router A, Router B, Router C. Есть также 2 элемента применяемой оператором политики, которые должны быть у этих маршрутизаторов применительно к эфемерному состоянию:

- Knob 1 - эфемерная конфигурация переопределяет локальную;
- Knob 2 - обновление локальной конфигураци замещает и переопределяет эфемерную конфигурацию.

Для Knob 1 маршрутизаторы с агентом I2RS получающим эфемерную запись в модель данных должны понять:

1. разрешает ли полритика оператора эфемерным изменениям конфигурации переопределять имеющуюся локальную конфигурацию?
2. есть ли в модели данных YANG какие-либо правила, связанные с эфемерной конфигурацией (такие как правила MUST или WHEN)?

В рассматриваемом примере нет правил MUST или WHEN для записываемых данных. Тогда установки политики будут иметь вид

	Knob 1	Knob 2
Router A	Эфемерное состояние имеет приоритет	Эфемерное состояние имеет приоритет
Router B	Локальная конфигурация имеет приоритет	Локальная конфигурация имеет приоритет
Router C	Эфемерное состояние имеет приоритет	Локальная конфигурация имеет приоритет

Нормальной политикой оператор для Router A является установка элементами Knob 1 и Knob 2 приоритета эфемерной конфигурации в агенте I2RS. Пусть клиент I2RS передаёт значение для записи в эфемерную конфигурацию через агент I2RS в Router A. Агент I2RS переопределяет значение в соответствующей конфигурации и возвращает подтверждение записи. Если значение изменяется в локальной конфигурации, Router A остаётся в эфемерной конфигурации, записанной клиентом I2RS.

Оператор Router B не желает записывать эфемерную конфигурацию для переопределения локальной, хотя в нем и установлен агент I2RS. Политика Router B отдаёт предпочтение локальной конфигурации над эфеменой. Когда агент I2RS на Router B получает запись от клиента I2RS, он проверяет Knob 1 и возвращает клиенту I2RS отклик, указывающий запрет переопределения локальной конфигурации. Пример с Router B показывает, почему в архитектуре I2RS по умолчанию принята локальная конфигурация. Поскольку функциональность I2RS является новой, оператор должен квключить её, иначе эфемерная конфигурация I2RS не будет работать. Оператор Router B может установить код I2RS и протестировать отклики, не применяя функцию перезаписи I2RS.

Оператор Router C устанавливает для Knob 1 переопределение клиентом I2RS имеющейся локальной конфигурации, а для Knob 2 задаёт обновление эфемерной конфигурации при изменении локальной. Для понимания причин этого представим, что Router C находится под контролем оператора, у которого есть система back-end, переписывающая локальную конфигурацию всех систем в 23 часа каждые сутки. Любые эфемерные изменения в сети сохраняются лишь до 23 часов, а затем back-end возвращает локальную конфигурацию. Пусть клиент I2RS записывает эфемерные состояния в течение для, а агент I2RS на Router C обновляет значения. В 23 часа back-end обновляет локальную конфигурацию через NETCONF и агент I2RS уведомляется об изменении. Агент I2RS информирует клиента I2RS о переопределении значения локальной настройкой. Клиент I2RS в этом случае является частью приложения, отслеживающего эфемерные изменения, чтобы убедиться в их включении при следующем запуске настройки.

6.4. Компоненты маршрутизации и связанные с ними службы I2RS

Для простоты каждый протокол или набор функций, который можно отдельно описать в информационной модели и модели данных, рассматривается как отдельная служба I2RS. Элемент маршрутизации не обязан реализовать все описанные компоненты маршрутизации или предоставлять соответствующие услуги I2RS. Службам I2RS следует включать модель возможностей, чтобы партнёры могли определить поддерживаемые части сервиса. Для каждой службы I2RS нужна информационная модель, описывающая хотя бы данные, доступные для чтения, данные, доступные для записи, уведомления, на которые можно подписаться, и отмеченную выше информационную модель.

Исходно включённые в архитектуру I2RS службы показаны на рисунке 2.

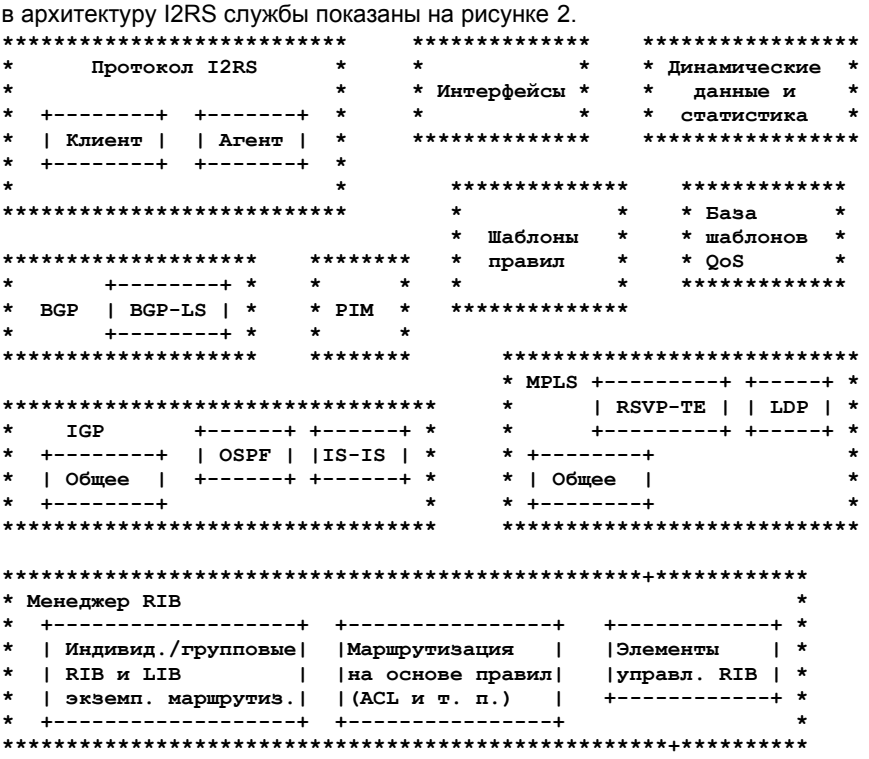

Рисунок 2. Ожидаемые службы I2RS.

Службы I2RS связаны между собой - RIB могут требоваться ссылки на конкретные экземпляры, указание общих составных типов (например, каналы, узлы, адреса IP) или возможность ссылаться на зависящие от реализации функции (например, предопределённые шаблоны для применения к интерфейсам или поведению QoS для трафика). В параграфе 6.4.5. Информационные модели, разные устройства и взаимосвязи информации рассматриваются конструкции моделирования информации и диапазоны применимых типов взаимоотношений.

6.4.1. Базы сведений о маршрутизации и метках

Элементы маршрутизации могут поддерживать одну или несколько информационных баз. Примеры включают RIB, такие как IPv4/IPv6 Unicast или IPv4/IPv6 Multicast. Другим примером служат базы сведений о метках MPLS (Label Information Base или LIB), базы на платформу, интерфейс или контекст. Эта функциональность, раскрываемая через службу I2RS, должна гладко взаимодействовать с теми же механизмами, которые элемент маршрутизации уже использует для обработки ввода в RIB из множества источников. Концептуально это можно сделать через агент I2RS, взаимодействующий с менеджером RIB, как отдельным источником маршрутизации.

Состояние «один с многими» (point-to-multipoint), добавляемое в RIB, не обязательно должно совпадать с установленным состоянием общеизвестного протокола групповой передачи. Агент I2RS может создавать произвольное состояние репликации в RIB в зависимости от анонсированных возможностей элемента маршрутизации.

6.4.2. IGP, BGP и групповые протоколы

Отдельная служба I2RS может представлять каждый протокол маршрутизации на устройстве. Такие службы I2RS могут включать ряд различных операций:

- чтение различных внутренних RIB протокола маршрутизации часто полезное для понимания состояния сети; прямая запись в связанные с протоколами RIB или базы данных выходит за рамки I2RS;
- чтение различных частей данных политики, которую конкретный экземпляр протокола применяет для управления своими операциями;
- запись сведений политики, таких как атрибуты интерфейса, связанные с протоколом маршрутизации, или политика BGP, которые могут косвенно манипулировать атрибутами маршрутов, передаваемых в BGP;
- запись маршрутов или префиксов, анонсируемых протоколом№
- присоединение и удаление интерфейсов для групповых деревьев;
- подписка на потоки информации об изменениях маршрутов;
- получение уведомлений о включении или отключении партнёров.

Например, взаимодействие с OSPF может включать изменение метрики каналов локального элемента маршрутизации, анонсирование локально подключённых префиксов или чтение части базы состояний каналов OSPF. Однако прямое изменение базы данных о состоянии каналов не разрешено, чтобы сохранить согласованность состояния сети.

6.4.3. MPLS

Службы I2RS нужны для раскрытия протоколов, создающих транспортные LSP (например, LDP и RSVP-TE) и протоколы (например, BGP, LDP), обеспечивающие услуги на основе MPLS (например, псевдопровода, L3VPN, L2VPN и т. п.). Это должно включать все локальные сведения о LSP, которые начинаются и завершаются в данном элементе маршрутизации, а также проходят через него (транзит).

6.4.4. Правила и механизмы QoS

Многие элементы сети имеют отдельные правила и механизмы QoS, включая регуляторы, влияющие на локальный расчёт пути и возможности управления очередями. Эти возможности существенно различаются в реализациях и I2RS не может моделировать весь диапазон сбора сведений или манипулирования этими атрибутами. Базовый набор должен включаться в информационные модели I2RS и поддерживаться на ожидаемых интерфейсах между агентом I2RS и элементом сети, чтобы обеспечить базовые возможности и заготовки для будущих расширений.

Используя преимущества расширяемости и создания субклассов, информационные модели могут задавать применение базовой модели, которая может быть заменена более детальной.

6.4.5. Информационные модели, разные устройства и взаимосвязи информации

I2RS сильно зависит от информационных моделей соответствующих аспектов элементов маршрутизации, которыми нужно манипулировать. Эти модели управляют моделями данных и протокольными операциями для I2RS. Важно, чтобы эти информационные модели хорошо работали с широким спектром фактических реализаций элементов маршрутизации, наблюдаемых в разных протоколах и у разных производителей. Имеется три способа учитывать такие вариации в информационных моделях - наследование классов или типов, необязательные функции и шаблоны.

6.4.5.1. Классы и типы объектов, наследование

Информация от элемента маршрутизации, смоделированная I2RS, может быть описана в терминах классов, типов или объектов, к которым могут применяться различные определения наследования. Архитектура не определяет их пригодность для I2RS и для простоты далее применяются термины класс и субкласс. Эта архитектура I2RS требует возможности устранения вариаций в элементах маршрутизации, позволяя информационным моделям определять родительские (базовые) классы и субклассы.

Базовый класс определяет общие аспекты, поддержка которых ожидается от всех элементов маршрутизации. Отдельные субклассы могут представлять вариации или дополнительные возможности. Когда это применимо, может задаваться несколько уровней уточнения. Затем протокол I2RS может предоставлять механизмы, позволяющие клиенту I2RS определить, какие классы доступны данному агенту I2RS. Клиенты I2RS, которым нужны лишь базовые возможности, могут работать исключительно с базовыми классами, а клиентам, нуждающимся в дополнительных деталях или функциях, могут работать с поддерживаемыми субклассами.

В рамках информационного моделирования I2RS следует задать чёткие правила взаимоотношений базовых классов и субклассов, например, возможность субкласса вносить изменения (и какие) в базовый класс. Описание таких правил следует делать так, чтобы правила можно было применять в инструментах моделирования данных, пока не будет выбран язык моделирования данных для I2RS.

6.4.5.2. Необязательные функции

Информационная модель I2RS должна чётко указывать необязательные аспекты. Например, должен ли экземпляр класса всегда включать конкретное поле X и, если так, должен ли клиент предоставить значение X при создании объекта или имеется общеизвестное значение, принятое по умолчанию? С точки зрения элемента маршрутизации в этом примере информационной модели следует предоставлять сведения, относящиеся к следующим аспектам:

- требуется ли X, чтобы поле данных было воспринято и применено?
- если X является необязательным, как X взаимодействует с обязательными аспектами поля данных?
- имеет ли поле данных принятые по умолчанию значения для обязательной и необязательной части?
- требуется ли X входить в определённый набор значений (например, диапазон, лина строки)?

Информационная модель должна чётко указывать, какие читаемые и записываемые значения устанавливаются клиентом и какие отклики или действия нужны агенту. Важно указать, что обязательно и необязательно в значениях клиента и откликах (действиях) агента.

6.4.5.3. Шаблоны

Шаблон - это набор сведений для решения проблемы, использующий понятия классов и экземпляров объектов. Шаблон предоставляет набор определённых значений для набора полей и может задавать набор значений, которые должны предоставляться для заполнения шаблона. Гибкая схема шаблона может разрешать перезаписывать некоторые из определённых значений. Например, назначение для трафика определённого класса обслуживания можно выполнить путём задания шаблона очередей с параметром для индикации класса (Gold, Silver, Best Effort). Способ реализации этого не моделируется. Это предполагает, что нужные шаблоны доступны на элементе маршрутизации через отличный от I2RS механизм. Идея состоит в том, что предоставляя подходящие шаблоны для задач, которые нужно выполнить, с шаблонами, по-разному реализованными для различных элементов маршрутизации, клиент легко может взаимодействовать с элементами маршрутизации, не заботясь о вариантах, которые могут обрабатываться включёнными в шаблон значениями.

Шаблоны могут не требоваться, если вариации реализаций можно раскрыть иначе. Однако шаблоны сами могут быть объектами ссылок в сообщениях протокола, при этом элементы маршрутизации нужно настраивать с подходящими шаблонами для выполнения операции. Этот вопрос требует дополнительного обсуждения.

6.4.5.4. Взаимосвязи объектов

Объекты (в элементе маршрутизации или вне его) не существуют изолированно и связаны между собой. Одним из важных аспектов определения класса является указания связей между экземплярами разных классов. Сложность таких связей может меняться и ниже представлены взаимоотношения, которые нужно поддерживать информационной модели.

6.4.5.4.1. Инициализация

Простейшей взаимосвязью является инициализация одного экземпляра объекта путём копирования другого. Например, может быть экземпляр, представляющий принятую по умолчанию настройку туннеля и все поля новых туннелей будут копироваться из него, если при организации туннеля эти поля не заданы. Это тесно связано с шаблонами, но не совпадает с ними полностью. Поскольку взаимосвязь является лишь временной, она зачастую не представляется формально при моделировании и фиксируется только в семантическом описании принятого по умолчанию объекта.

6.4.5.4.2. Указание корреляций

Зачастую достаточно указать в одном объекте, что он связан с другим, без задания строгой привязки между ними. Таким образом, для представления взаимоотношений применяется идентификатор. Это можно применять для обеспечения позднего связывания или слабой привязки, которая не обязана существовать. Имя правила в объекте может указывать, что при наличии этого правила оно применяется в определённых обстоятельствах. При моделировании это часто представляется типом значения.

6.4.5.4.3. Ссылки на объекты

Иногда связи между объектами сильнее. Например, действительная запись ARP должна указывать интерфейс, через который она получена. Это классическое использование ссылок на объекты в программировании и его можно применять для взаимоотношений, таких как сдерживание и зависимость. Обычно это представляется явной ссылкой в модели.

6.4.5.4.4. Активные ссылки

Имеются ещё более сильные связи между объектами, когда изменение одного из них всегда отражается в состоянии другого. Например, если при наличии у туннеля MTU (максимальный передаваемый блок) изменение MTU на канале должно незамедлительно отражаться в MTU туннеля, эти значения будут жёстко связаны. Этот вид связывания активных состояний предполагает жёсткое согласования, зачастую представляемое как модель подписки между объектами.

7. Интерфейс между клиентом и агентом I2RS

7.1. Протокол управления и обмена данными

Эта архитектура I2RS подразумевает управляемый на основе модели данных протокол, где модель данных задана в YANG 1.1 [YANG1.1] и связанных базовых документах YANG [RFC6991], [RFC7223], [RFC7224], [RFC7277], [RFC7317]. Двумя протоколами, расширяемыми для поддержки I2RS являются NETCONF [RFC6241] и RESTCONF [RESTCONF]. Это помогает обеспечить простоту и повышает возможности внедрения. Протоколу I2RS может потребоваться использование нескольких вариантов базового транспорта (TCP, SCTP[1], DCCP[2]) с подходящими механизмами проверки подлинности и защиты целостности. Этот различный транспорт может поддерживать разные типы взаимодействия (например, управление, чтение, уведомления, сбор информации) и разные наборы данных. Какой бы транспорт не применялся для обмена данными, он должен поддерживать подходящие механизмы контроля перегрузок. Выбранный транспорт должен быть удобен для оператора и разработчика, чтобы упростить внедрение.

Каждая версия протокола I2RS будет задавать a) транспорт, который может применяться протоколом I2RS, b) обязательный для реализации транспорт и c) необязательный для реализации транспорт.

7.2. Коммуникационные каналы

Требуется множество каналов связи нескольких типов. Может существовать ряд требований (например, к конфиденциальности и надёжности) и для поддержки расширяемости могут быть нужны каналы исходящие от нескольких субкомпонентов элемента маршрутизации и/или разных частей клиента I2RS. Все такие каналы будет использовать один протокол верхнего уровня I2RS (который сочетает защищённый транспорт и контекстную информацию I2RS). Использование дополнительных каналов будет согласовываться между клиентом и агентом I2RS с использованием этого протокола.

Коммуникации протокола I2RS могут обеспечиваться по основному каналу (in-band) плоскости данных системы маршрутизации, а также по отдельному каналу (out-of-band) через интерфейс управления. В зависимости от запрашиваемых операций взаимодействия I2RS могут прерывать работу по основному каналу, делая агент I2RS недоступным по этому каналу связи.

7.3. Согласование возможностей

Поддержка возможностей протокола и служб I2RS может сильно отличаться в разных клиентах I2RS и элементах маршрутизации, поддерживающих агент I2RS. Поскольку каждая служба I2RS должна включать модель возможностей (6.4. Компоненты маршрутизации и связанные с ними службы I2RS), согласование на уровне протокола может быть ограничено его спецификой и набором поддерживаемых услуг I2RS. Согласование возможностей (например, поддержка транспорта сверх обязательного минимума) явно потребуется. Важно сохранить простоту и отказоустойчивость такого согласования, поскольку эти механизмы могут стать причиной сложностей при реализации и внедрении.

[1]Stream Control Transport Protocol - протокол управления потоковой передачей.
[2]Datagram Congestion Control Protocol - протокол дейтаграмм с контролем перегрузок.

Согласование возможностей протокола можно разделить на базовое согласование версии (требуется для базового взаимодействия) и более сложный обмен возможностями, который может выполняться в рамках базового протокола. В частности, более сложный протокол и механизм согласования можно задать путём определения информационных моделей для агента и клиента I2RS, которые могут описывать различные варианты возможностей. Затем это можно представить и использовать при обмене важными сведениями об агенте и его возможностях.

7.4. Спецификации политики области действия

Как описано в параграфах 4.1 и 4.2, у каждого клиента I2RS имеется уникальный идентификатор и может быть второй идентификатор (см. 2. Терминология) для помощи при устранении неполадок. Как указано в разделе 4, все механизмы проверки подлинности и полномочий основаны на первичном идентификаторе, который связывает роль с областью действия политики для чтения и записи данных, а также для ограничения потребляемых ресурсов. Спецификации области данных политики (для чтения, записи, потребления ресурсов) должны указывать данные, которыми управляет политика и допустимые диапазоны диапазоны значений.

7.5. Связность

Клиент I2RS может (но не обязан) поддерживать активный канал связи с агентом I2RS, поэтому агенту может потребоваться организовать канал связи с клиентом для передачи ранее запрошенных сведений. Отсутствие активного канала связи не предполагает, что соответствующий клиент I2RS не функционирует. Когда канал связи нужен, клиент или агент I2RS могут организовать новый канал связи.

Удерживаемое агентом I2RS состояние, которое принадлежит клиенту I2RS, не может быть удалено или очищено, когда клиент прекратил обмен данными, даже если агенту не удаётся создать новый канал связи с клиентом.

Для многих приложений может быть желательна очистка состояния, если сетевое приложение «умирает» до удаления созданного им состояния. Обычно это решается в части резервирования сетевого приложения. Если нужен более сильный механизм, внешние (не I2RS) механизмы могут позволить управляющему сетевому приложению отслеживать клиентов I2RS и на основе известных ему правил очищать состояние при умирании приложения. Реализация в агенте I2RS более сложных механизмов повысит сложность протокола I2RS, поэтому оставлена для будущей работы. Одним из таких механизмов является запрос клиентом очистки состояния при прерывании конкретной транспортной сессии. Другим является завершение состояния по сроку, заданному в политике, связанной с ролью клиента I2RS. Срок действия состояния может завершаться, когда не удалось создать канал связи или по инициативе клиента I2RS при заданной политикой продолжительности.

7.6. Уведомления

Как в любой системе правил, взаимодействующей с сетью, клиенту I2RS нужна возможность получать уведомления о смене состояний сети. Уведомления здесь относятся к непредвиденным изменениям, представляют события за пределами системы (например, отказы интерфейсов на контролируемых устройствах) или является достаточно редкими, чтобы считать их аномалиями. Уведомления могут применяться и для регулярных событий.

События могут быть интересны многим клиентам I2RS, контролирующим данные, обрабатываемые агентом I2RS, и другим клиентам I2RS, которые собирают сведения без управления. Поэтому архитектура требует для клиентов I2RS возможности регистрации на диапазон уведомлений, а для агентов I2RS - передавать уведомления множеству клиентов. Клиенту I2RS следует иметь возможность фильтровать получаемые уведомления, конкретные типы уведомлений и операций фильтрации могут зависеть от информационной модели и должны указываться в ней.

Информационная модель I2RS должна включать представление этих событий. Как отмечено выше, сведения о возможностях в модели позволят клиентам I2RS понимать, какие события может генерировать данный агент I2RS.

Для обеспечения производительности и расширяемости клиента I2RS и общей конфиденциальности информации клиентам I2RS нужна возможность регистрации только на интересующие события. Возможно также предоставление в I2RS потока уведомлений через механизмы публикации и подписки, которые не фильтруются агентом I2RS.

7.7. Сбор сведений

Ещё одним важным аспектом I2RS является упрощение сбора информации о состоянии сетевых элементов. Это включает как получение моментальных снимков (snapshot) большого объёма данных о текущем состоянии сети, так и подписку на происходящие изменения набора данных или его части. Это отделено в архитектуре от уведомлений из за различий в объёме и скорости передачи информации.

7.8. Управление из нескольких мест

Как описано выше, агент I2RS взаимодействует с несколькими клиентами I2RS, которые активно контролируют элемент сети. С точки зрения архитектуры и устройства предполагается, что с помощью средств извне системы данные, которыми нужно манипулировать в элементе сети, разделены должным образом, чтобы с любой данной частью информации работал лишь один клиент I2RS.

Тем не менее, возникают неожиданные взаимодействия и два (или более) клиента I2RS могут пытаться манипулировать одной и той же частью данных. Это считается ошибкой. Данная архитектура не пытается определить, каким должно быть корректное состояние данных при таких конфликтах. Вместо этого архитектура требует наличия средств, с помощью которых агенты I2RS могут обработать такие конфликты. Механизм обеспечения предсказуемости состоит в назначении каждому клиенту I2RS определённого приоритета и применении изменений, заданных более приоритетным клиентом. При совпадении приоритетов преимущество отдаётся клиенту, который первым запросил действия.

Чтобы такой подход к управлению из разных мест подходил для клиентов I2RS, нужно возможность регистрации клиентов I2RS для получения уведомлений об изменениях, вносимых в данные с разрешённой записью, состояние которых представляет для этого клиента конкретный интерес. Это включено в механизмы событий I2RS. Нужно также применять это к изменениям, вносимым CLI/NETCONF/SNMP в области действия записи агента I2RS, поскольку там

применяется тот же механизм приоритета (например, всегда отдаётся предпочтение CLI). После этого клиент I2RS может реагировать на ситуацию по своему усмотрению.

7.9. Транзакции

Для простоты архитектура I2RS не включает неделимость групп сообщений и механизмы отката. Вместо этого включена некоторая обработка ошибок для для набора операций, включённых в одно сообщение. Клиент I2RS может указать один из приведённых ниже методов обработки ошибок для данного сообщения с несколькими операциями, которое он передаёт агенту I2RS.

Все или ничего
: Эта традиционная семантика SNMP указывает, что клиент при обработке сообщения будет хранить достаточно состояний, чтобы вернуться к исходному состоянию при возникновении ошибки. Будут применены все операции или ни одной из них, а сообщение об ошибке будет указывать единственный отказ, из-за которого операции не были применены. Это полезно, например, при взаимозависимости операций из сообщения.

Выполнение до ошибки
: В этом случае операции из сообщения выполняются в заданном порядке и при возникновении ошибки обработка прекращается с возвратом сообщения, информирующего об ошибке. Это полезно, когда между операциями есть зависимости и их можно отсортировать топологически.

Выполнять все, записывая ошибки
: В этом случае агент I2RS пытается выполнить все операции из сообщения и возвращать сведения о каждой возникшей ошибке. Это полезно при отсутствии зависимостей между операциями или в случаях, когда клиент I2RS предпочитает сам разобраться с последствиями ошибок.

Для отказоустойчивости и чёткости состояния протокола включается явный отклик на операции изменения и записи даже при отсутствии отказов.

8. Вопросы эксплуатации и управляемости

Для облегчения устранения неполадок в элементах маршрутизации, реализующих агенты I2RS, таким элементам следует обеспечивать механизм для отображения активно предоставленного состояния I2RS и других внутренних сведений агента I2RS. Отметим, что эти сведения могут быть очень деликатными в части безопасности моделей данных, реализованных агентом, и поэтому должны защищаться в соответствии с требованиями моделей. Желательно для этого механизма применять привилегированное средство, отличное от простого подключения как клиент I2RS для получения данных. Применение другого механизма должно улучшить отслеживаемости и контроль отказов.

Управляемость является очень важным аспектом I2RS. Некоторые примере приведены ниже.

Ограничение ресурсов
: Применяя I2RS, приложения могут расходовать ресурсы, будь то операции во временных рамках, записи в RIB, сохраняемые операции для активирования и т. п. Важна возможность установить пределы ресурсов при предоставлении полномочий.

Взаимодействие конфигураций
: Необходимо чётко определять взаимодействие состояния, установленного через I2RS, и состояния настройки маршрутизатора. Как описано в этом документе, для обеспечения достаточно гибкости применяется простой механизм приоритетов.

Отслеживаемость взаимодействий
: Возможность отслеживать взаимодействия запросов, полученных агентом I2RS, и действий этих агентов необходима для того, чтобы агенты I2RS могли видеть операции в процессе развёртывания и устранять неполадки в программах или проблемы в сети.

Служба подписки на уведомления
: Возможность клиентов I2RS подписываться на поток уведомлений от агентов I2RS (вместо запроса их у агентов) обеспечивает более масштабируемую обработку уведомления для взаимодействий между клиентами и агентами I2RS.

9. Литература

9.1. Нормативные документы

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC7920] Atlas, A., Ed., Nadeau, T., Ed., and D. Ward, "Problem Statement for the Interface to the Routing System", RFC 7920, DOI 10.17487/RFC7920, June 2016, <http://www.rfc-editor.org/info/rfc7920>.

9.2. Дополнительная литература

[I2RS-ENV-SEC] Migault, D., Ed., Halpern, J., and S. Hares, "I2RS Environment Security Requirements", Work in Progress, draft-ietf-i2rs-security-environment-reqs-01, April 2016.

[I2RS-PROT-SEC] Hares, S., Migault, D., and J. Halpern, "I2RS Security Related Requirements", Work in Progress[1], draft-ietf-i2rs-protocol-security-requirements-06, May 2016.

[RESTCONF] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", Work in Progress[2], draft-ietf-netconf-restconf-14, June 2016.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[1]Опубликовано в RFC 8241. *Прим. перев.*
[2]Опубликовано в RFC 8040. *Прим. перев.*

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

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

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <http://www.rfc-editor.org/info/rfc7223>.

[RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", RFC 7224, DOI 10.17487/RFC7224, May 2014, <http://www.rfc-editor.org/info/rfc7224>.

[RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <http://www.rfc-editor.org/info/rfc7277>.

[RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <http://www.rfc-editor.org/info/rfc7317>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <http://www.rfc-editor.org/info/rfc7752>.

[YANG1.1] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", Work in Progress[1], draft-ietf-netmod-rfc6020bis-14, June 2016.

Благодарности

Значительная часть документа заимствована из Interface to the Routing System Framework (февраль 2013 г.) и A Policy Framework for the Interface to the Routing System (февраль 2013 г.).

Авторы благодарны Nitin Bahadur, Shane Amante, Ed Crabbe, Ken Gray, Carlos Pignataro, Wes George, Ron Bonica, Joe Clarke, Juergen Schoenwalder, Jeff Haas, Jamal Hadi Salim, Scott Brim, Thomas Narten, Dean Bogdanovic, Tom Petch, Robert Raszuk, Sriganesh Kini, John Mattsson, Nancy Cam-Winget, DaCheng Zhang, Qin Wu, Ahmed Abro, Salman Asadullah, Eric Yu, Deborah Brungard, Russ Housley, Russ White, Charlie Kaufman, Benoit Claise, Spencer Dawkins, Stephen Farrell за их предложения и рецензии.

Адреса авторов

Alia Atlas
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
United States
Email: akatlas@juniper.net

Joel Halpern
Ericsson
Email: Joel.Halpern@ericsson.com

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
United States
Phone: +1 734-604-0332
Email: shares@ndzh.com

Dave Ward
Cisco Systems
Tasman Drive
San Jose, CA 95134
United States
Email: wardd@cisco.com

Thomas D. Nadeau
Brocade
Email: tnadeau@lucidvision.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

[1]Опубликовано в RFC 7950. *Прим. перев.*