Internet Engineering Task Force (IETF)
Request for Comments: 6244
Category: Informational
ISSN: 2070-1721

P. Shafer
Juniper Networks

June 2011

An Architecture for Network Management Using NETCONF and YANG

Архитектура управления сетью с использованием NETCONF и YANG

Аннотация

Протокол настройки сети (Network Configuration Protocol или NETCONF) предоставляет доступ к естественным возможностям устройств в сети, задавая методы манипулирования базами данных конфигурации, извлечения рабочих данных и вызова конкретных операций. YANG обеспечивает способы задания содержимого, передаваемого через NETCONF (данные и операции). Объединив эти технологии, можно определить стандартные модули обеспечивающие функциональную совместимость и унификацию устройств, позволяя тем выражать свои уникальные возможности.

Этот документ описывает применение NETCONF и YANG для создания управляющих приложений, соответствующих потребностям операторов сетей.

Статус документа

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

Документ является результатом работы IETF[1] 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 Internet, см. . Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 5741.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке http://www.rfc-editor.org/info/rfc6244.

Оглавление

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

1. Истоки NETCONF и YANG

Сети становятся все более сложными и ёмкими, а также растёт плотность развёрнутых служб. Требования к времени безотказной работы (uptime), надёжности и предсказуемой задержке вызывают необходимость автоматизации. Задача управления сетью не так проста, является комплексной и запутанной. Однако проблемы нужно решать, чтобы удовлетворить потребности в стабильности существующих служб одновременно с добавлением новых услуг при расутщей нехватке квалифицированных сетевых инженеров.

В июне 2002 г. Совет по архитектуре Internet (Internet Architecture Board или IAB) провёл семинар по управлению сетями [RFC3535]. Участники этого семинара внесли ряд замечаний и рекомендация для рассмотрения в IETF проблем, с которыми операторы сталкиваются в своей работе, направляя усилия IETF на эту область.

Выводы этого семинара были сфокусированы на текущих задачах. Наблюдения были разумными и откровенными, включая необходимость транзакций, откат (rollback), низкую стоимость реализации и возможность сохранять и восстанавливать данные конфигурации устройств. Многие наблюдения дают представление о проблемах операторов при использовании имеющихся решений для управления сетями, таких как отсутствие полного охвата возможностей устройств и способности отличать данные конфигурации от других типов данных.

На основе этих указания была создана рабочая группа NETCONF и протокол настройки сети (Network Configuration или NETCONF). Этот протокол определяет простой механизм, где сетевые приложения, выступающие как клиенты, могут вызывать операции на устройстве, действующем как сервер. Спецификация NETCONF [RFC4741] задаёт небольшой набор операций и старается избегать требований к передаваемым в этих операциях данным, предпочитая разрешать протоколу передачу любых данных. Такой подход, не зависящий от данных, позволяет независимо создавать модели данных.

Из-за отсутствия средств для определения моделей данных протокол NETCONF невозможно было использовать для работы на основе стандартов. Были рассмотрены имевшиеся языки моделирования данных, такие как определение схемы XML (XML Schema Definition или XSD) [W3CXSD] и языки определения схемы документа (Document Schema Definition Languages или DSDL) [ISODSDL], но они были отвергнуты как неподходящие для решения задач. Задание новой модели данных или протокола в XML отличается от создания документа XML. Использование операций NETCONF вносит требования к содержимому данных, которое не применяется совместно с предметной областью статических документов, для которых предназначены языки схем вроде XSD или RELAX NG.

В 2007 и 2008 гг. вопрос моделирования данных для NETCONF обсуждался в рамках OPS и APP на конференциях IETF 70 и 71 и команде разработчиков было поручено подготовить документ с требованиями [RCDML]. После обсуждения доступных вариантов в рамках CANMOD BoF на IETF 71 сообщество подготовило устав для рабочей группы NETMOD. Отличное описание этого этапа доступно по ссылке <http://www.ietf.org/mail-archive/web/ietf/current/msg51644.html>.

В 2008 и 2009 гг. рабочая группа NETMOD подготовила спецификацию YANG [RFC6020] как средство определения моделей данных для NETCONF, позволяющее публиковать стандартные и фирменные (proprietary) модели данных в форме, которая легко воспринимается человеком и удовлетворяет многим требованиям, внесенным на семинаре IAB NM. Это позволило использовать NETCONF для разработки стандартных моделей данных в рамках IETF.

YANG позволяет разработчикам создавать модели, задавать организацию данных и вносить ограничения для данных. После публикации модуль YANG служит соглашением между клиентом и сервером, где обе стороны понимают, какого поведения ожидает от них партнёр. Клиент знает, как создать пригодные данные для сервера и какие данные сервер будет передавать ему. Сервер знает правила, управляющие данными, и как ему следует вести себя.

YANG также поддерживает расширяемость и гибкость, которых нет в других языках моделирования. Новые модули могут дополнять иерархии данных, определённые в других модулях, беспрепятственно добавляя данные в подходящие места имеющейся иерархии. YANG также позволяет определять новые операторы, что даёт возможность согласованного расширения самого языка.

Этот документ представляет архитектуру для YANG, описывая работу связанных с YANG технологий и построение решений на их основе для задач управления сетями.

2. Элементы архитектуры

2.1. NETCONF

NETCONF задаёт основанный на XML механизм удалённого вызова процедур (remote procedure call или RPC), использующий простоту и доступность высококачественных синтаксических анализаторов XML. Язык XML обеспечивает богатое, гибкое и иерархическое стандартное представление данных, соответствующее потребностям сетевых устройств. NETCONF доставляет данные конфигурации и операции в форме запросов и откликов с использованием кодирования RPC в формате XML и передачей по ориентированному на соединения транспорту.

Иерархическое представление данных XML позволяет естественным образом представлять сложные сетевые данные. Например, приведённая ниже конфигурация помещает интерфейсы в области OSPF. Элемент <ospf> содержит список элементов <area>, каждый из которых включает список элементов <interface>. Элемент <name> указывает конкретную область или интерфейс. Дополнительная конфгурация для каждой области или интерфейса указывается внутри соответствующего элемента.

```
<ospf xmlns="http://example.org/netconf/ospf">

  <area>
    <name>0.0.0.0</name>
    <interface>
      <name>ge-0/0/0.0</name>
      <!-- Приоритет для интерфейса -->
      <priority>30</priority>
      <metric>100</metric>
      <dead-interval>120</dead-interval>
    </interface>

    <interface>
      <name>ge-0/0/1.0</name>
      <metric>140</metric>
    </interface>
  </area>

  <area>
    <name>10.1.2.0</name>
    <interface>
      <name>ge-0/0/2.0</name>
      <metric>100</metric>
    </interface>

    <interface>
      <name>ge-0/0/3.0</name>
      <metric>140</metric>
      <dead-interval>120</dead-interval>
    </interface>
  </area>
</ospf>
```

NETCONF включает механизмы для контроля хранилищ данных конфигурации. Каждое хранилище является определённым набором данных конфигурации, который может быть источником или целью связанных с настройкой операций. Устройство может указать, есть ли у него отдельное хранилище для стартовой конфигурации (startup), возможна ли прямая запись в хранилище рабочей конфигурации (running), имеется ли хранилище подготовленной конфигурации (candidate), которая активируется вызовом операции <commit/>[1].

NETCONF задаёт операции, которые клиент (приложение) вызывает как RPC для выполнения на сервере, работающем на устройстве. В таблице приведён список этих операций.

Операция	***Описание***
commit	Представляет конфигурацию candidate в running
copy-config	Копирует одно хранилище данных в другое
delete-config	Удаляет хранилище данных конфигурации
edit-config	Меняет содержимое конфигурационного хранилища
get-config	Извлекает хранилище данных или его часть
lock	Предотвращает изменение хранилища данных другой стороной
unlock	Снимает блокировку хранилища данных

Механизм возможностей (capability) NETCONF позволяет устройству анонсировать набор поддерживаемых им возможностей (свойств, функций), включая операции протокола, хранилища и модели данных и т. п. Это анонсируется при организации сессии как часть сообщения <hello>. Клиент може проверить сообщение hello для определения возможностей устройства и способов взаимодействия с ним для выполнения нужных задач.

NETCONF также задает способы передачи клиенту асинхронных уведомлений от сервера, описанных в [RFC5277].

Кроме того, NETCONF может извлекать сведения о состоянии, получать уведомления и вызывать дополнительные методы RPC, определённые как часть возможности. Полные сведения о NETCONF представлены в [RFC4741].

2.1.1. Транспортное отображение NETCONF

NETCONF может работать по любому транспортному протоколу, удовлетворяющему требованиям RFC 4741, включая:

[1] В оригинале ошибочно сказано commit-configuration, см. https://www.rfc-editor.org/errata/eid5760. *Прим. перев.*

- операции через явные соединения;
- аутентификацию (проверку подлинности);
- защиту целостности;
- защиту конфиденциальности.

В [RFC4742] задано сопоставление с протоколом Secure Shell (SSH) [RFC4251], которые является обязательным для поддержки транспортым протоколом. Другие протоколы включают SOAP [RFC4743], блочный расширяемый протокол обмена (Blocks Extensible Exchange Protocol или BEEP) [RFC4744], и защиту транспортого уровня (Transport Layer Security или TLS) [RFC5539].

2.2. YANG

YANG является языком моделирования данных для NETCONF. Он позволяет описывать иерархии узлов данных (node) и ограничения для них. YANG определяет модели данных и способы манипулирования ими через операции протокола NETCONF.

Каждый модуль YANG задаёт модель данных, однозначно указываемую URI пространства имён. Эти модели данных можно расширять, что позволяет объединять стандартные и фирменные модели данных. Модели строятся из организационных контейнеров, списков узлов данных и формирующих данные листьев дерева данных.

```
module example-ospf {
    namespace "http://example.org/netconf/ospf";
    prefix ospf;

    import network-types { // Доступ к определениям другого модуля
        prefix nett;
    }

    container ospf {   // Объявляет тег верхнего уровня
        list area {    // Объявляет список узлов area
            key name;  // Ключ name указывает членов списка
            leaf name {
                type nett:area-id;
            }
            list interface {
                key name;
                leaf name {
                    type nett:interface-name;
                }
                leaf priority {
                    description "Заданный приоритет маршрутизатора";
                    type uint8;  // Тип задаёт ограничение для
                                 // действительных значений priority.
                }
                leaf metric {
                    type uint16 {
                        range 1..65535;
                    }
                }
                leaf dead-interval {
                    units seconds;
                    type uint16 {
                        range 1..65535;
                    }
                }
            }
        }
    }
}
```

Модуль YANG задаёт модель данных в терминах данных, их иерархической организации и ограничений. YANG определяет, как эти данные представляются в XML и пименяются в операциях NETCONF. Основные операторы YANG указаны в таблице.

Оператор	***Описание***
augment	Расширяет имеющиеся иерархии данных
choice	Задаёт взаимоисключающие варианты
container	Определяет уровень в иерархии данных
extension	Добавляет новый оператор в YANG
feature	Указывает необязательную часть модели
grouping	Группирует определения данных в набор для многократного использования
key	Задаёт листья ключей для списков
leaf	Определяет лист в иерархии данных
leaf-list	Лист, который может присутствовать неоднократно
list	Иерархия, которая может присутствовать неоднократно
notification	Определяет уведомление
rpc	Задаёт входные и выходные параметры для операции RPC
typedef	Определяет новый тип
uses	Встраивает содержимое grouping

2.2.1. Ограничения

YANG позволяет разработчику ввести для модели данных ограничения, предотвращающие невозможные или нелогичные данные. Эти ограничения сообщают клиенту о данных, передаваемых устройств, а также позволяет

клиенту знать, какие данные устройство будет воспринимать, чтобы клиент мог отправлять корреткные данные. Ограничения применяются к данным конфигурации, но могут использоваться также для rpc и уведомлений.

Основным ограничением служит оператор type, который определяет содержимое листа данного именованного типа. Краткое описание основных ограничений YANG приведено в таблице.

Оператор	***Описание***
length	Ограничивает размер строки
mandatory	Требует наличия узла
max-elements	Ограничивает число элементов в списке сверху
min-elements	Ограничивает число элементов в списке снизу
must	Выражение XPath должно иметь значение true
pattern	Должно быть выполнено регулярное выражение
range	Значение должно входить в диапазон
type	Значение должно присуутствовать в данных
leafref[1]	
unique	Значение должно быть уникальным в данных
when	Узел присутствует только при XPath со значением true

В операторах must и when применяются выражения XPath [W3CXPATH] для задания условий, которые семантически оцениваются по отношению к иерархии данных, но ни клиент, ни сервер не обязаны реализовать спецификацию XPath и могут применять любые средства для выполнения заданных условий.

2.2.2. Гибкость

YANG использует тип union и операторы choice и feature для предоставления разработчикам гибкости при задании моделей данных. Тип union позволяет листу воспринимать разные типы данных, таких как integer или слово unbounded (не ограничено).

```
type union {
    type int32;
    type enumeration {
        enum "unbounded";
    }
}
```

Оператор choice содержит список взаимоисключающих узлов, из которых действительная конфигурация может содержать лишь 1 (case). Оператор feature позволяет разработчику указать необязательные части модели, а устройству - указать, какие из этих частей оно реализует.

Оператор deviation позволяет устройству указать части модуля YANG, которые не реализованы в устройстве полностью. Хотя устройствам рекомендуется полностью соблюдать соглашения, представленные в модуле YANG, в реальности это может соблюдаться не всегда. Отклонения позволяют выразить такие ограничения, а не оставлять их для обнаружения как ошибки в процессе работы.

2.2.3. Модель расширяемости

Язык XML включает концепцию пространства имён, позволяющую комбинировать элементы XML из разных источников в одной иерархии без риска возникновения конфликтов. Модули YANG определяют содержимое для конкретных пространств имён, но один модуль может дополнять определение из другого модуля, внося элементы из пространства имён этого модуля в иерархию своего пространства имён.

Поскольку модуль может дополнять определение из другого модуля, иерархия определений может расти за счёт добавления источников в базовую иерархию. Эти дополнения указываются с использованием пространства имён исходных модулей, что помогает избежать конфликтов имён при изменении модулей со временем. Например, если приведённая выше конфигурация OSPF была стандартной, модуль от производителя может дополнить её своими расширениями.

```
module vendorx-ospf {
    namespace "http://vendorx.example.com/ospf";
    prefix vendorx;
    import example-ospf {
        prefix ospf;
    }

    augment /ospf:ospf/ospf:area/ospf:interface[2] {
        leaf no-neighbor-down-notification {
            type empty;
            description "Не сообщать другим протоколам
                       + " об отключении соседа.";
        }
    }
}
```

Элемент <no-neighbor-down-notification> тогда помещается в пространство имён vendorx.

```
<ospf xmlns="http://example.org/netconf/ospf"
      xmlns:vendorx="http://vendorx.example.com/ospf">

  <area>
    <name>0.0.0.0</name>
```

[1] В оригинале ошибочно сказано reference, см. https://www.rfc-editor.org/errata/eid3012. *Прим. перев.*
[2] В оригинале ошибочно сказано interfaces, см. https://www.rfc-editor.org/errata/eid3356. *Прим. перев.*

```
          <interface>
            <name>ge-0/0/0.0</name>
            <priority>30</priority>
            <vendorx:no-neighbor-down-notification/>
          </interface>
        </area>
      </ospf>
```

Дополнения интегрируются с базовыми модулями, что позволяет извлекать, архивировать, загружать и удалять их в естественной иерархии. Если клиентское приложение запрашивает конфигурацию для конкретной области OSPF, оно получит часть иерахии для этой области вместе с данными дополнения.

2.3. Трансляции YANG

Язык моделирования данных YANG является центральной частью группы связанных технологий. Сам язык YANG, описанный в [RFC6020], задаёт синтаксис и операторы, их назначение и способы комбинирования для построения иерархии узлов, описывающей модель данных.

Этот документ задаёт концепцию содержимого XML «в линии» для операций NETCONF с моделями данных из модулей YANG. Это включает базовое сопоставление узлов дерева данных YANG с элементами XML, а также механизмы, применяемые в <edit-config> для манипуляций с данными, таких как упорядочивание узлов в списке.

YANG использует обычный и легко описываемый синтаксис, предназначенный в первую очередь для человека. Этот синтаксис удобен для электронной почты, утилит diff и patch, в также подходит для форматных ограничений RFC.

2.3.1. YIN

В некоторых средах встраивание синтаксического анализатора YANG может быть неприемлемым. Для таких случаев грамматика XML для YANG определена как YIN (YANG Independent Notation - независимая нотация), что позволяет применять синтаксические анализаторы XML, доступные как в открытом коде, так и в коммерческих версиях. Трансляция между YANG и YIN является прямой, обратимой и не вносит потерь. Операторы YANG преобразуются в элементы XML, сохраняя структуру и содержимое YANG, но позволяя использовать готовые анализаторы XML вместо анализатора YANG. YIN поддерживает полную семантическую эквивалентность YANG.

2.3.2. DSDL (RELAX NG)

Поскольку содержимое NETCONF кодируется в XML, естественно применять для проверки языки схем XML. Для упрощения этого YANG предоставляет стандартизованное сопоставление модулей YANG с языками DSDL [RFC6110], основным компонентом которых является RELAX NG. DSDL считается лучшим выбором в качестве стандартного языка схем, поскольку он учитывает не только грамматику и типы данных документа XML, но и семантические ограничения, а также правила изменения набора информации в документе. Кроме того, DSDL обеспечивает формальные средства для координации нескольких независимых схем и определения способа применения схем к разным частям документа. Это полезно, поскольку содержимое YANG обычно собрано из нескольких словарей.

2.4. Типы YANG

YANG поддерживает множество встроенных типов и позволяет выводить из них производные типы расширяемым способом. Новые типы могут задавать дополнительные ограничения для значений данных. Доступна стандартная библиотка типов для использования в YANG [RFC6021]. Эти модули YANG задают часто применяемые типы данных дял связанных с IETF стандартов.

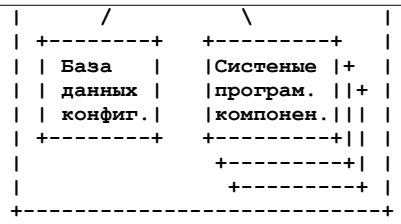
2.5. Рекомендации IETF

Задан дополнительный набор рекомендаций для авторов и рецензентов спецификаций Standards-Track с модулями YANG [RFC6087]. Эти рекомендации следует применять как базу для обзора других документов с моделями YANG.

3. Работа с YANG

3.1. Создание решений на основе NETCONF и YANG

В типовом решении на базе YANG клиент и сервер управляются содержимым модулей YANG. Сервер содержит определения модулей как метаданные, доступные машине NETCONF, а та обрабатывает запросы, применяет метаданные для анализа и проверки запросов, выполняет запрошеннную операцию и возвращает результат клиенту.



Для применения YANG должны быть определены модули, для решения конкретного набора задач. Эти модули загружаются, компилируются и кодируются в сервер.

Ниже показан пример типичного взаимодействия между клиентом и сервером.

- Клиентское приложение [C] организует сессию NETCONF с сервером (устройством) [S].
- [C] и [S] обмениваются сообщениями <hello> со списком возможностей, поддерживаемых каждой стороной, что позволяет [C] узнать о модулях, поддерживаемых [S].
- [C] создаёт и передаёт операцию, заданную в модуле YANG, в кодировке XML в элементе NETCONF <rpc>.
- [S] получает и анализирует элемент <rpc>.
- [S] проверяет содержимое запроса по модели данных их модуля YANG.
- [S] выполняет запрошенную операцию, возможно меняя содержимое хранилища данных конфигурации.
- [S] создаёт отклик, включающий сам отклик на операцию, запрошенные данные и возникшие ошибки.
- [S] передаёт отклик в кодировке XML внутри элемента NETCONF <rpc-reply>.
- [C] получает и анализирует элемент <rpc-reply>.
- [C] проверяет и должным образом обрабатывает отклик.

Отметим, что клиент и сервер не обязаны обрабатывать модули YANG именно так. Содержимое модели данных может быть жёстко встроено (hard code) в сервер вместо обработки его базовой машиной. Клиент может быть нацелен на конкретную модель YANG, а не на общую обработку. Таким клиентом может быть простой сценарий оболочки (shell script), который помещает аргументы в шаблон содержимого XML и передаёт серверу.

3.2. Выполнение требований операторов

NETCONF и YANG решают многие из задач, поставленных на семинаре IAB NM.

- Простота применения - язык YANG удобен для человека, прост и удобочитаем. Многие каверзные вопросы сохраняются из-за сложности проблемной области, но YANG стремится сделать их более ясными и простыми для решения.
- Данные конфигурации и состояния - YANG чтеко отделяет данные конфигурации от иных типов данных.
- Транзакции - NETCONF обеспечивает простой механизм транзакций.
- Генерация различий - модуль YANG даёт достаточно сведения для генерации различия, которое требуется для замены одного модуля другим.
- Дамп и восстановление - NETCONF позволяет сохранять и восстанавливать данные конфигурации. Это может быть выполнено для определённого модуля YANG.
- Настройка всей сети - NETCONF поддерживает отказоустойчивые транзакции для настройки в масштабе сети за счёт операций представления и подтверждённого представления. При попытке внесения изменений, затрагивающих не одно устройство, эти возможности упрощают обработку отказов и позволяют создавать заведомо неделимые транзакции (успех или отказ всех операций одной транзакции - всё или ничего).
- Удобство для текста - модули YANG и определяемые в них данные удобны для текста.
- Обработка конфигурации - NETCONF позволяет различать передачу и активацию данных конфигурации.
- Ориентированность на задачи - модуль YANG может указывать конкретные задачи как операции RPC. Клиент может вызвать операцию RPC или обратиться к любым базовым данным напрямую.
- Полнота охвата - можно задать модули YANG полностью охватывающие естественные возможности устройства. Это избавляет от обращений к командному интерфейсу (command line interface или CLI) с помощью таких инструментов, как Expect [SWEXPECT].
- Своевременность - модули YANG можно связать с операциями CLI для незамедлительной доступности естественных операций и данных.
- Сложности реализации - гибкость YANG позволяет упростить внедрение модулей. Добавление возможностей и замена третьей нормальной формы естественной иерархией данных позволяет снизить сложность.
- Простой язык моделирования данных - YANG обладает достаточными возможностями для применения в других ситуациях. В частности, можно интегрировать встроенный API и естественный интерфейс CLI для упрощения инфраструктуры.
- Поддержка разных языков - YANG использует символы Unicode UTF-8 [RFC3629].
- Сопоставление событий - YANG объединяет операции RPC, уведомления, данные конфигурации и состояния, разрешая внутренние ссылки. Например, поле уведомления можно пометить как указывающее партнёра BGP и клиентское приложение сможет легко найти этого партнёра в данных конфигурации.
- Стоимость внедрения - были приложены значительные усилия для снижения расходов на внедрение.

- Удобный для человека синтаксис - синтаксис YANG оптимизирован для чтения, особенно при рецензировании, поскольку оно чаще всего применяется человеком.
- Постобработка - применение XML расширяет возможности постобработки данных, возможно с применением основанных на XML технологий, таких как XPath [W3CXPATH], XQuery [W3CXQUERY], XSLT [W3CXSLT].
- Семантическое несоответствие - более богатые и описательные модели данных снижают вероятность семантического несоответствия. За счёт возможности определять новые примитивы модули YANG будут более содержательными, что позволит усилить соблюдение правил и ограничений.
- Безопасность - NETCONF работает по транспортным протоколам, защищённым SSH или TLS, обеспечивающим защищённую связь и аутентификацию с использованием проверенной технологии. Защищенный транспорт может использовать имеющуюся инфраструктуру управления ключами и свидетельствами, что снижает расходы на внедрение.
- Надёжность - NETCONF и YANG - цельные и надёжные технологии. NETCONF работает на основе соединений и включает механизмы автоматического восстановления при потере соединения.
- Внесение изменений - модели на основе YANG поддерживают операции внесения изменений - добавление, вставка, редактирование, удаление чётко определены.
- Ориентированность на методы - YANG позволяет задавать новые операции RPC, включая их имена, по сути являющиеся методами. Входные и выходные параметры операций RPC также задаются в модулях YANG.

3.3. Роли при создании решений

Создание решений на основе NETCONF и YANGтребует взаимодействия с множеством разных групп. Разработчики моделей должны понимать, как создать полезные модели, придающие структуру и смысл данных, обеспечивая им максимальную гибкость на будущее. Рецензенты должны быстро определить, является ли структура точной. Разработчикам устройств нужно закодировать модель данных в их устройства, а разработчикам приложений - закодировать свои приложения, чтобы воспользоваться преимуществами модели данных. Имеются разные стратегии для выполнения каждой из этих задач, кратко описанные в последующих параграфах.

3.3.1. Разработчики моделей

Разработчики определяют модель данных на основе своих глубоких знаний о моделируемой предметной области. Модели следует быть максимально простой и выразительной. Организацию модели следует ориентировать на текущие задачи, не забывая о возможности её расширения за счёт других модулей и приспособлении к будущим задачам.

Моделирование рассматривается также в разделе 4. Вопросы моделирования.

3.3.2. Рецензенты

Роль рецензента, пожалуй, самая важная и время, которое рецензент готов уделить, очень ценно. Чтобы помочь рецензенту, YANG подчёркивает удобочитаемость, естественную иерархию данных и простые, короткие операторы.

3.3.3. Разработчики устройств

Модель YANG информирует разработчика устройства о том, какие данные моделируются. Разработчик читает модели YANG и создаёт код, поддерживающий модель. Модель описывает иерархию данных и связанные ограничения, а описания и справочные материалы помогают разработчикам устройств понять, как преобразовать представление модели в свою реализацию устройства.

3.3.3.1. Базовая поддержка содержимого

Модель YANG может быть скомпилирована в основанную на YANG машину на стороне клиента и сервера. Входные и выходные данные могут проверяться. Хранилище данных конфигурации также может проверяться в соответствии с заданными в модели ограничениями.

Модули сериализации и десериализации для генерации и приёма содержимого NETCONF могут управляться метаданными модели. При получении данных метаданные проверяются для контроля пригодности входящих элементов XML.

3.3.3.2. Определения XML

Модуль YANG задаёт кодирование XML для данных, передаваемых через NETCONF. Правила кодирования неизменны, поэтому модуль YANG можно использовать для проверки соответствия содержимого NETCONF правилам.

3.3.4. Разработчики приложений

Модель YANG информирует разработчика приложения о том, какие данные моделируются. Разработчик может просмотреть модули и принять одно из трёх возможных представлений, рассмотренных ниже с учётом влияния YANG на их устройство. В реальности большинство приложений применяют сочетание этих подходов.

3.3.4.1. Жёсткое встраивание

Приложение может кодироваться для конкретного, хорошо известного содержимого модулей YANG, напрямую реализуя их организацию, правила и логику на основе явных сведений. Например, можно написать сценарий для смены доменного имени группы устройств с использованием стандартного модуля YANG, включающего такие узлы. Этот сценарий примет новое доменное имя как аргумент и поместит его в строку, содержащую остальной код XML, как этого требует модуль YANG. Это содержимое передаётся через NETCONF на каждое из устройств.

Этот тип приложений удобен для небольших, фиксированных задач, где издержки обеспечения гибкости превосходят затраты на жёсткое встраивание сведений непосредственно в приложение.

3.3.4.2. Снизу вверх

Приложение может использовать базовый восходящий подход к настройке, концентрируясь непосредственно на данных устрйюства и обрабатывая их без особого понимания. Модули YANG могут служить для управления работой YANG-эквивалента «браузера MIB». Такое приложение работает с данными конфигурации устройства на основе их организации в модуле YANG. Например, интерфейс GUI может представлять простую визуализацию, где элементы иерархии YANG отображаются иерархией папок или панелей GUI. Щелчок на строке раскрывает содержимое соответствующей иерархии XML. Этот тип GUI легко создать путём генерации таблиц стилей XSLT по моделям данных YANG. Затем можно воспользоваться машиной XSLT для преобразования данных конфигурации в набор web-страниц.

Модули YANG позволяют приложению применять ограничения без понимания семантики модуля YANG.

3.3.4.3. Сверху вниз

Нисходящий подход позволяет приложению видеть данные конфигурации, отличающиеся от стандартных и/или фирменных модулей YANG. Приложение может организовать свою модель организации данных и представить её пользователю. Когда приложению нужно передать данные на устройство, оно преобразует эти данные из ориентированного на задачу представления в формат, подходящий для конкретного устройства. Преобразованием управляет приложение, что позволяет менять и обновлять это преобразование без воздействия на устройство. Например, можно создать приложение, которое моделирует сети VPN в сетевом представлении. Приложению потребуется преобразовать эти высокоуровневые определения VPN в данные конфигурации, которые будут передаваться конкретным устройствам в составе VPN.

Даже при таком подходе модули YANG полезны, поскольку их можно использовать для моделирования VPN. Например, в представленном ниже коде моделируется список VPN с протоколами, топологией и инерфейсами, а также списком классификаторов.

```
list example-bgpvpn {
    key name;
    leaf name { ... }
    leaf protocol {
        type enumeration {
            enum bgpvpn;
            enum l2vpn;
        }
    }
    leaf topology {
        type enumeration {
            enum hub-n-spoke;
            enum mesh;
        }
    }
    list members {
        key "device interface";
        leaf device { ... }
        leaf interface { ... }
    }
    list classifiers {
        ...
    }
}
```

Приложение может использовать такой модуль YANG для управления своей работой, создания экземпляров VPN в базе данных и последующего выталкивания конфигурации этих VPN в отдельные устройства с использованием стандартной модели устройства (например, example-bgpvpn.yang) или преобразования стандартного содержимого устройств в тот или иной фирменный формат для нестандартных устройств.

4. Вопросы моделирования

Ниже рассматриваются соображения, которые разработчику следует учитывать при создании моделей YANG.

4.1. Принятые по умолчанию значения

Концепция принятых по умолчанию значений проста, но детали и представление этих значений, а также взаимодействие с данными конфигурации могут вызывать затруднения. NETCONF отдает принятые по умолчанию значения на откуп модели данных, а YANG обеспечивает гибкость реализации устройства а плане обработки принятых по умолчанию значений. Требование состоит в том, что устройство «**должно** вести себя так, будто лист имеется в дереве данных с принятым по умолчанию значением. Это позволяет реализации устройства выбрать способ обработки принятых по умолчанию значений.

Одним из вариантов является представление конфигурации как набора инструкций по настройке устройства. Если значение данных в этих инструкциях совпадает с принятым по умолчанию, его следует сохранять как часть конфигурации, но если значение не задано явно, оно не считается частью конфигурации.

Другим вариантом является «подрезка» значений, совпадающих с принятыми по умолчанию, путём неявного их удаления из хранилища данных конфигурации. Установка для листа принятого по умолчанию значения фактически удаляет этот лист.

Устройство может сообщать все принятые по умолчанию значения, независимо от их фактической установки. Это облегчает работу клиента, которому нужны принятые по умолчанию значения, но может существенно увеличить размер данных конфигурации.

Эти варианты отражают схемы обработки принятых по умолчанию значений для широко распространённых сетевых устройств и их поддержка позволяет YANG снизить расходы на реализацию и внедрение на основе моделей YANG.

Когда клиент извлекает данные из устройства, он должен быть готов к отсутствию узлов с принятыми по умолчанию значениями, поскольку сервер не обязан передавать такие элементы. Это позволяет устройству реализовать любую из первых двух схем обработки принятых по умолчанию значений.

Независимо от выбора реализации устройство может поддерживать свойство with-defaults [RFC6243] и давать клиенту возможность выбрать желаемую обработку принятых по умолчанию значений.

При оценке выражений XPath для таких конструкций, как must и when, контекст оценки будет включать подходящие значения, принятые по умолчанию, поэтому разработчик модели может рассчитывать на согласованное поведение всех устройств.

4.2. Соответствие

При разработке моделей данных нужно принимать во внимание ряд противоречивых аспектов:

- полезность;
- соответствие;
- гибкость;
- расширяемость;
- отклонения.

Чтобы модель была интересной, она должна быть полезно, решающей задачи более простым и мощным способом, нежели это возможно без модели. Модели следует обеспечивать максимальную полезность в предметной области задачи. Разработчикам следует создавать модели для максимального числа устройств, которые могут корректно реализовать модель. Если модель слишком специализирована или включает очень много допущений об устройстве, сложность и стоимость реализации такой модели приведут к низкому качеству и проблемам совместимости, снижая ценность модели.

Разработчики могут применять в моделях оператор feature, чтобы предоставить устройствам некоторую гибкость за счёт разделения модели, позволяющего устройству указать реализованные в нем части модели. Например, при наличии в модели свойства logging устройство, не имеющее хранилища для записи журнала, может сообщить клиенту, что оно не поддерживает эту функцию модули.

Модели можно расширять с помощью операторов augment и разработчику следует рассматривать возможность расширения модели. Такие дополнения могут задавать производители, приложения и органы стандартизации.

Отклонения позволяют устройствам указать неполную совместимость реализации с моделью. Например, после публикации модели разработчик может решить сделать конкретный узел настраиваемым, хотя стандартная модель считает его данными состояния. Реализация обычно сообщает значение и может заявить об отклонении своего поведения от стандартного. Приложения, способные узнать о таком отклонении, могут учесть это, но другие приложения могут продолжать рассматривать реализацию, как соответствующую модели.

Иногда реализации могут принимать решения, препятствующие соответствию стандартам. Такие случаи прискорбны, но могут возникать на практике, а разработчики моделей и приложений принимают такие решения на свой страх и риск. Реализацией, выдающей целочисленный лист как «корову» (cow), будет сложно управлять, но приложения должны быть готовы к такому поведению устройств в «полевых» условиях.

Несмотря на это, клиентам и серверам следует считать модуль YANG соглашением, которое обе стороны согласны выполнять. Разработчику модели следует чётко указать условия такого соглашения, а разработчикам клиентов и серверов - стремиться корректно и точно реализовать модель данных, описанную модулем YANG.

4.3. Разные данные

Различать данные конфигурации, рабочего состояния и статистики важно понимать разработчикам моделей и тем, кто планирует расширять протокол NETCONF. Далее приведена некоторая предыстория и даны определения и примеры.

4.3.1. Предпосылки

На семинаре IAB NM операторы сформулировали два требования, включённые в [RFC3535].

2. Необходимо чётко различать данные конфигурации, данные, описывающие рабочее состояние, и статистику. Для некоторых устройств очень сложно определить, какие параметры настроены административно, а какие получены от других механизмов, таких как протоколы маршрутизации.

3. Требуется возможность раздельно получать от устройств данные конфигурации, данные рабочего состояния и статистику, а также возможность сравнения однотипных данных между устройствами.

Протокол NETCONF, заданный в RFC 4741, различает два типа данных - данные конфигурации и данные состояния:

> Данные конфигурации представляют собой набор доступных для записи данных, требуемых для перехода системы из принятого по умолчанию начального состояния в её текущее состояние.
>
> Данные состояния - это дополнительные сведения о системе, не являющиеся данными состояния, такие как доступные лишь для чтения сведения о состоянии и собранной статистике.

NETCONF не следует заданному операторами различию между данными конфигурации, данными рабочего состояния и статистикой, относя два последних вида данных к данным состояния.

4.3.2. Определения

Ниже приведено определение данных конфигурации, состояния и статистики, заимствованное из предыдущей работы.

- Данные конфигурации представляют собой набор доступных для записи данных, требуемых для перехода системы из принятого по умолчанию начального состояния в её текущее состояние [RFC4741].

- Данные работчего состояния - это набор сведений полученных системой в процессе работы и влияющих на поведение системы подобно данным конфигурации. В отличие от данных конфигурации, данные рабочего

состояния имеют временный характер и изменяются в результате взаимодействий с внутренними компонентами и другими системами по специализтолванным протоколам.

- Статистические данные представляют собой доступные лишь для чтения данные, созданные самой системой. Они описывают производительность системы и её компонентов.

Приведённые ниже примеры помогут понять разницу между данными конфигурации, состояния и статистики.

4.3.2.1. Пример 1 - таблица маршрутизации IP

Таблицы маршрутизации IP могут содержать заданные статически записи (данные конфигурации), а также записи, полученные от протоколов маршрутизации, таких как OSPF (данные рабочего состояния). Кроме того, машина маршрутизации может собирать статистику, например, о частоте использования конкретной записи в таблице.

4.3.2.2. Пример 2 - интерфейсы

Сетевые интерфейсы обычно имеют большое число атрибутов, связанных с типом интерфейса, а в некоторых случаях - с подключённым к интерфейсу кабелем. Примерами являются значения MTU на интерфейсе или скорость интерфейса Ethernet.

Во многих случаях системы используют атрибуты, обнаруженные при инициализации интерфейса. Эти атрибуты, как таковые, относятся к рабочему состоянию, однако обычно имеется возможность переопределить обнаруженные атрибуты статическими данными состояния, например, установить определённое значение MTU[1] или задать скорость для интерфейса Ethernet.

Система будет записывать статистику (счётчики) для числа пакетов и байтов, а также ошибки при передаче и приёме на каждом интерфейсе.

4.3.2.3. Пример 3 - учётные записи

Системы обычно поддерживают статические данные конфигурации о настроенных в системе учётных записях. Кроме того, системы могут динамически получать сведения об учётных записях извне (например, по протоколу LDAP[2], от систем NIS[3]), являющиеся данными рабочего состояния. Сведения об использовании учётной записи являются применом данных статистики.

Отметим, что сведения, предоставляемые системе для создания новой учетной записи, могут дополняться данными конфигурации, определяемыми системой при создании учетной записи (например, уникальный идентификатор записи). Хотя система может создавать такие данные, они обычно становятся частью статических данных конфигурации системы, поскольку не являются временными.

4.3.3. Допущения

Основное внимание в YANG уделяется данным конфигурации. Не существует единого механизма разделить данные рабочего состояния и статистику, поскольку NETCONF считает те и другие данными состояния. В этом параграфе описано несколько вариантов решения этой задачи.

4.3.3.1. Модели данных

Первым решением является модель данных с явным разделением данных конфигурации и рабочего состояния. Это ведёт к дублированию структур данных и может плохо масштабироваться с точки зрения моделирования. Например, настроенное значение режима дуплекса и его рабочее значение могут быть разными листьями в модели данных.

4.3.3.2. Дополнительные операции по извлечению рабочего состояния

Протокол NETCONF может расширяться новыми операциями, которые, в частности, позволяют извлекать всё рабочее состояние , например, операция <get-ops> (и, возможно, также операция <get-stats>).

4.3.3.3. Хранилище данных рабочего состояния

Другим вариантом может быть введение нового хранилища данных, представляющего рабочее состояние. Операция <get-config> для такого хранилища <operational> будет возвращать рабочее состояние, определяющее поведение устройства, а не его статические и явно заданные состояния конфигурации.

4.4. Решения

В настоящее время единственным жизнеспособным вариантом является раздельное моделирование конфигурации и рабочего состояния. Листья конфигурации будут указывать желаемые значения, а листья рабочего состояния - текущие значения на устройстве. В примере с режимом дуплекса будет два листа - duplex (config true) и op-duplex (config false).

В некоторых случаях могут применяться раздельные листья, в других - раздельные списки, которые можно организовать разными способами с различными ограничениями. Операторы ключей, сортировки и ограничений, такие как must, unique и when, могут различаться в данных конфигурации и рабочего состояния. Например, настроенные статические маршруты могут быть списком, отличным от списка рабочей таблицы маршрутизации, поскольку применяемые ключи и сортировка могут различаться.

5. Вопросы безопасности

Этот документ описывает архитектуру управления сетями с применением NETCONF и YANG и не влияет на безопасность Internet.

[1] Maximum Transmission Unit - максимальный передаваемый блок.
[2] Lightweight Directory Access Protocol - облегченный протокол доступа к каталогам.
[3] Network Information Service - сетеавая информационная система.

6. Литература

6.1. Нормативные документы

[ISODSDL] International Organization for Standardization, "Document Schema Definition Languages (DSDL) - Part 1: Overview", ISO/IEC 19757-1, November 2004.

[RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, May 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4251] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[RFC4741] Enns, R., "NETCONF Configuration Protocol", RFC 4741, December 2006.

[RFC4742] Wasserman, M. and T. Goddard, "Using the NETCONF Configuration Protocol over Secure SHell (SSH)", RFC 4742, December 2006.

[RFC4743] Goddard, T., "Using NETCONF over the Simple Object Access Protocol (SOAP)", RFC 4743, December 2006.

[RFC4744] Lear, E. and K. Crozier, "Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)", RFC 4744, December 2006.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, July 2008.

[RFC5539] Badra, M., "NETCONF over Transport Layer Security (TLS)", RFC 5539, May 2009.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, January 2011.

[RFC6110] Lhotka, L., "Mapping YANG to Document Schema Definition Languages and Validating NETCONF Content", RFC 6110, February 2011.

[RFC6243] Bierman, A. and B. Lengyel, "With-defaults Capability for NETCONF", RFC 6243, June 2011.

[SWEXPECT] "The Expect Home Page", <http://expect.sourceforge.net/>.

[W3CXPATH] DeRose, S. and J. Clark, "XML Path Language (XPath) Version 1.0", World Wide Web Consortium Recommendation REC-xpath-19991116, November 1999, <http://www.w3.org/TR/1999/REC-xpath-19991116>.

[W3CXQUERY] Boag, S., "XQuery 1.0: An XML Query Language", W3C WD WD-xquery-20050915, September 2005.

[W3CXSD] Walmsley, P. and D. Fallside, "XML Schema Part 0: Primer Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-0-20041028, October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028>.

[W3CXSLT] Clark, J., "XSL Transformations (XSLT) Version 1.0", World Wide Web Consortium Recommendation REC-xslt-19991116, November 1999, <http://www.w3.org/TR/1999/REC-xslt-19991116>.

6.2. Дополнительная литература

[RCDML] Presuhn, R., Ed., "Requirements for a Configuration Data Modeling Language", Work in Progress, February 2008.

Адрес автора

Phil Shafer
Juniper Networks
EMail: phil@juniper.net

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru