

Internet Engineering Task Force (IETF)
Request for Comments: 8530
Category: Standards Track
ISSN: 2070-1721

L. Berger
C. Hopps
LabN Consulting, L.L.C.
A. Lindem
Cisco Systems
D. Bogdanovic
X. Liu
Volta Networks
March 2019

YANG Model for Logical Network Elements

Модель YANG для логических элементов сети

Аннотация

Этот документ определяет модуль для логических элементов сети YANG LNE, соответствующих архитектуре хранилищ данных управления сетью (Network Management Datastore Architecture или NMDA). Модуль может служить для управления разделением логических ресурсов, которые могут присутствовать на сетевом устройстве. Примерами таких логических устройств являются логические системы и логические маршрутизаторы. Модель YANG в этом документе соответствует архитектуре NMDA, описанной в RFC 8342.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF¹ и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG². Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <https://www.rfc-editor.org/info/rfc8530>.

Авторские права

Copyright (c) 2019. Авторские права принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с упрощённой лицензией BSD, как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

Оглавление

1. Введение.....	2
1.1. Терминология.....	2
2. Обзор.....	2
3. Логические элементы сети.....	3
3.1. Создание экземпляра LNE и выделение ресурсов.....	3
3.2. Управление LNE с точки зрения LNE.....	3
3.3. Управление LNE с точки зрения хоста.....	3
4. Вопросы безопасности.....	4
5. Взаимодействие с IANA.....	4
6. Модель логического элемента сети.....	4
7. Литература.....	6
7.1. Нормативные документы.....	6
7.2. Дополнительная литература.....	7
Приложение А. Примеры.....	7
А.1. LNE, управляемый хост-устройством.....	7
А.1.1. Данные конфигурации.....	9
А.1.2. Данные состояния.....	11
А.2. LNE с самоуправлением.....	16
А.2.1. Данные конфигурации.....	17
А.2.1.1. Логический элемент vnf1.....	17
А.2.1.2. Логический элемент vnf2.....	18
А.2.2. Данные состояния.....	19
А.2.2.1. Хост-устройство.....	19
А.2.2.2. Логический элемент vnf1.....	20
А.2.2.3. Логический элемент vnf2.....	22
Благодарности.....	23
Адреса авторов.....	23

¹Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

²Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

Этот документ определяет соответствующий архитектуре NMDA модуль YANG [RFC6020] для поддержки создания логических сетевых элементов (logical network element или LNE) на сетевом устройстве. LNE - это независимо поддерживаемое сетевое устройство на основе ресурсов, выделенных ему хостом или родительским сетевым устройством. Элемент LNE, работающий на сетевом хосте, концептуально похож на виртуальную машину, работающую на хост-системе. В терминах виртуализации можно назвать LNE гостем (Guest), а содержащее его сетевое устройство - хостом (Host). Хотя LNE можно реализовать с помощью технологий виртуализации на хосте, это совсем не обязательно. Модель YANG в этом документе соответствует архитектуре хранилищ сетевой информации NMDA, определённой в [RFC8342].

Документ также определяет дополнения, требуемые для выделения хостом ресурсов данному LNE. Поскольку модель управления интерфейсом [RFC8343] является единственным модулем, определяющим в настоящее время ресурсы хоста, этот документ определяет лишь одно дополнение, охватывающее назначение интерфейса элементу LNE. Предполагается, что будущие модули, определяющие поддержку управления ресурсами хост-устройства, будут при необходимости поддерживать выделение контролируемых ресурсов элементам LNE.

Поскольку элемент LNE является независимо управляемым устройством, у каждого LNE будет набор моделируемых в YANG данных, независимый от хост-устройства и других LNE. Например, несколько LNE могут иметь свой интерфейс Tunnel0, который не будет конфликтовать с интерфейсами других LNE и не будет присутствовать в модели интерфейсов хоста. LNE будет иметь свои интерфейсы управления, возможно включая независимые серверы NETCONF, RESTCONF и т. п. для поддержки настройки своих моделей YANG. Реализация может полностью переименовать выделенные интерфейсы и, например, на хосте выделенный LNE интерфейс может называться Ethernet0/1, а в LNE - eth1.

В дополнение к стандартным интерфейсам управления реализация хост-устройства может обеспечивать доступ к конфигурации LNE и операционные модели YANG напрямую с хост-системы. Такой доступ может происходить через точку монтирования yang-schema-mount [RFC8528], в которой обращаться к корневому уровню моделей YANG LNE.

Примеры терминов производителей для LNE включают logical system, logical router, virtual switch, chassis, fabric.

1.1. Терминология

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

Предполагается знакомство читателя с концепциями YANG [RFC7950] и YANG Schema Mount [RFC8528]. В документе применяется графическое представление моделей данных, описанное в [RFC8340].

2. Обзор

В этом документе рассматриваются сетевые устройства, которые поддерживают протоколы и функции, определённые в рамках IETF Routing Area, например, маршрутизаторы, межсетевые экраны, хосты. Такие устройства могут быть физическими или виртуальными, например, классический маршрутизатор с нестандартным (custom) оборудованием или маршрутизатор внутри виртуальной машины на сервере, реализующей функцию виртуальной сети (virtual network function или VNF). Каждое устройство может выделять свои ресурсы в элементы LNE, каждый из которых является управляемым логическим устройством. Примеры терминов производителей для LNE включают logical system, logical router, virtual switch, chassis, fabric. Каждый LNE может также поддерживать функции маршрутизации и пересылки VPN (VPN Routing and Forwarding или VRF) и экземпляра виртуальной коммутации (Virtual Switching Instance или VSI), которые далее называются экземплярами сети (Network Instance или NI). Это разделение представлено на рисунке 1.



Рисунок 1. Связи между элементами модуля.

Модель для LNE представлена в разделе 3. Логические элементы сети, а модель для NI - в [RFC8529].

Текущая модель управления интерфейсом [RFC8343] затрагивается определениями LNE и NI. Этот документ и [RFC8529] определяют дополнения к модели интерфейса для поддержки LNE и NI. Похожие элементы (возможно, только для LNE) могут потребоваться включить и в будущие модули для оборудования и QoS.

Интерфейсы являются важной частью конфигурации и рабочего состояния любого сетевого устройства. Обычно они включают комбинацию необработанных (raw) физических интерфейсов, интерфейсов канального уровня, настройку адресов и логические интерфейсы, которые могут быть не привязаны ни к какому физическому интерфейсу. Некоторые системные службы, а также протоколы L2 и L3 тоже могут связывать свои данные конфигурации и рабочего состояния в различными типами интерфейсов (эти связи для простоты не показаны). Модель управления интерфейсом задана в [RFC8343]. Модуль logical-network-element дополняет имеющиеся модели управления интерфейсами, добавляя идентификатор, который применяется на интерфейсах для указания связанного LNE. Относящиеся к интерфейсам дополнение показано ниже.

```

module: ietf-logical-network-element
  augment /if:interfaces/if:interface:
    +--rw bind-lne-name? ->
      /logical-network-elements/logical-network-element/name

```

Модель интерфейса в [RFC8343] структурирована так, чтобы включить все интерфейсы в плоский список без учёта логического выделения ресурсов, поддерживаемых на устройстве. Лист (leaf) и bind-lne-name обеспечивают связь между интерфейсом и соответствующим LNE. Отметим, что в настоящее время при назначении интерфейсу как LNE, так и NI нужно сначала выделять интерфейс для LNE, а потом в модели интерфейса этого LNE связывать представление LNE для этого интерфейса с экземпляром NI с помощью механизма, заданного в [RFC8529].

3. Логические элементы сети

Элементы LNE поддерживают способность некоторых устройств разделять ресурсы на независимые логические коммутаторы и маршрутизаторы. Поддержка устройствами нескольких LNE зависит от реализации. Системам без таких возможностей не требуется включать поддержку модуля logical-network-element. В физических устройствах некоторые аппаратные возможности совместно используются разделами (partition), но экземпляры протоколов плоскости управления (например, маршрутизации), таблицы и конфигурация поддерживаются отдельно. Например, для логических маршрутизаторов или VNF может создаваться несколько логических экземпляров, использующих один программный образ. Модель поддерживает настройку множества экземпляров на одном устройстве за счёт создания списка LNE, каждый из которых имеет свою конфигурацию и рабочее состояние в части протоколов маршрутизации и коммутации. Модель LNE можно представить в форме

```

module: ietf-logical-network-element
  +--rw logical-network-elements
    +--rw logical-network-element* [name]
      +--rw name string
      +--rw managed? boolean
      +--rw description? string
      +--mp root
  augment /if:interfaces/if:interface:
    +--rw bind-lne-name?
      -> /logical-network-elements/logical-network-element/name

  notifications:
    +---n bind-lne-name-failed
      +--ro name -> /if:interfaces/interface/name
      +--ro bind-lne-name
      | -> /if:interfaces/interface/lne:bind-lne-name
      +--ro error-info? string

```

Узел name указывает логический элемент сети, managed показывает, будет ли сервер, обеспечивающий хост-устройство, предоставлять клиенту сведения LNE через структуру root. Корнем данных конкретного LNE является точка монтирования схемы (root). Узел bind-lne-name служит для связывания интерфейса с LNE, а bind-lne-name-failed применяется при некоторых типах отказов.

Корень LNE **должен** содержать хотя бы библиотеку YANG [RFC7895] и модуль интерфейса [RFC8343].

3.1. Создание экземпляра LNE и выделение ресурсов

Элементами LNE может управлять клиент с использованием имеющегося списка операций. При создании записей списка порождается новый экземпляр LNE. Предполагается, что модели, монтируемые в корень LNE, будут зависеть от реализации сервера. При удалении записи списка уничтожается имеющийся элемент LNE. Дополнительные сведения можно найти в параграфе 7.8.6 [RFC7950].

После создания экземпляра LNE с ним можно связать ресурсы хост-устройства. Как отмечено ранее, этот документ дополняет ietf-interfaces листом bind-lne-name для поддержки таких ассоциаций с интерфейсами. При указании в bind-lne-name действительного имени LNE реализация **должна** предпринять все требуемые шаги по назначению интерфейса элементу LNE или вернуть сообщение об ошибке (см. ниже) с указанием причины отказа в назначении. Назначение может завершаться отказом во время обработки набора или по завершении асинхронной обработки. В последнем случае сведения об ошибке передаются через уведомление.

При успешном назначении интерфейса элементу LNE реализация **должна** также обеспечить LNE доступные ресурсы, предоставив LNE созданный системой интерфейс. Имя этого интерфейса определяется локально и может совпадать или отличаться от имени, применяемого в контексте хост-устройства. Созданный системой интерфейс **следует** раскрывать через экземпляр подэлемента интерфейса [RFC8343] конкретного LNE.

3.2. Управление LNE с точки зрения LNE

Предполагается, что каждый экземпляр LNE будет поддерживать функции управления из контекста корня LNE через сервер, предоставляющий информацию с корнем LNE в качестве корня устройства. Функции управления, работающие в контексте LNE, доступны через стандартные интерфейсы управления LNE, например, NETCONF и SNMP. Исходная конфигурация LNE, как и исходная конфигурация хост-устройства определяется локальной реализацией.

При доступе к LNE через его интерфейс управления будет представлен образ (representation) сетевого устройства, но область действия будет ограничена конкретным LNE. Обычные механизмы YANG и NETCONF вместе с требуемым экземпляром библиотеки YANG [RFC7895] могут применяться для идентификации доступных модулей. Каждый поддерживаемый модель будет представляться как модуль верхнего уровня. В связанных с ресурсами модулях будут отражаться лишь связанные с LNE ресурсы, такие как интерфейсы, оборудование и (возможно) QoS. С точки зрения системы управления не будет различий между доступным представлением LNE и физическим сетевым устройством.

3.3. Управление LNE с точки зрения хоста

Существует несколько подходов к реализации, позволяющих сетевому устройству поддерживать модуль logical-network-element и множество LNE. Некоторые варианты предоставляют функциям управления, работающим на

сетевом устройстве, доступ к конфигурации и рабочему состоянию LNE, но даже при наличии такой возможности управление LNE с сетевого устройства может быть запрещено правилами.

Независимо от выбранного метода реализации упомянутый выше логический узел `managed` служит для контроля возможности управления LNE из контекста сетевого устройства. При `managed false` элементом LNE невозможно управлять с хост-системы и управление разрешено лишь из контекста LNE, как описано в параграфе 3.2. Попытки доступа к информации ниже корневого узла, для которого установлено `managed false`, **должны** приводит к указанному ниже сообщению об ошибке. В некоторых реализациях смена значения может оказаться невозможной. Например, при реализации LNE с использованием виртуальной машины и традиционных технологий гипервизора вполне вероятно, что возможно будет лишь значение `managed false`.

Реализация самостоятельно решает вопрос возможности доступа и изменения информации из контекста LNE или даже из контекста хост-устройства. При установке `managed true`, к сведениям LNE **нужно** предоставлять доступ из контекста хост-устройства. Когда в связанном определении `schema-mount` установлено `config true`, для данных LNE **нужно** также предоставлять возможность изменения из контекста хост-устройства. Когда сведения LNE доступны из контекста хост-устройства и LNE, те же сведения **должны** быть доступны через элемент `root` с изменёнными в соответствии с [RFC8528] путями.

Реализация **может** представлять схему LNE с использованием подхода `inline` или `shared-schema`, как описано в [RFC8528], выбор которого полностью определяется реализацией. Вариант `inline` обычно предполагается в случаях, когда для листа `managed` всегда установлено значение `false`. Вариант `shared-schema` считается более полезным в случаях, когда все LNE используют общую схему. При использовании `shared-schema` с точкой монтирования LNE библиотека YANG с корнем в точке монтирования **должна** соответствовать схеме, определённой в соответствии с модулем `ietf-yang-schema-mount`.

Помимо двух модулей, всегда присутствующих для LNE, поскольку сам элемент LNE является сетевым устройством, все модули, которые могут присутствовать в сетевом устройстве верхнего уровня, **могут** быть представлены и для LNE. Предполагается, что список доступных модулей будет зависеть от реализации, как и метод поддержки элементов LNE. Примеры использования LNE представлены в Приложении A.

4. Вопросы безопасности

Заданный в этом документе модуль YANG определяет схему для данных, которая разработана для доступа через протоколы управления сетью, такие как NETCONF [RFC6241] и RESTCONF [RFC8040]. Нижним уровнем NETCONF является защищённый транспорт с обязательной реализацией Secure Shell (SSH) [RFC6242]. Нижним уровнем RESTCONF служит HTTPS с обязательной реализацией защищённого транспорта TLS [RFC8446].

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

Данные LNE содержат сведения о конфигурации устройства и сети, поэтому безопасность этих сведений важна, но принципиально не отличается от защиты других сведений о конфигурации устройства или интерфейса, рассмотренной в таких документах, как [RFC8343], [RFC7317], [RFC8349].

Уязвимы могут быть параметры и ветви с `config true`, указанные ниже.

/logical-network-elements/logical-network-element

Этот список задаёт логические элементы сети и связанные с ними конфигурации.

/logical-network-elements/logical-network-element/managed

Хотя этот лист содержится в предыдущем списке, он заслуживает особого внимания, поскольку определяет доступность сведений из точки монтирования LNE из контекста хост-устройства и LNE. Содержимое этого листа может быть чувствительным в средах, где LNE управляется стороной, отличной от хост-устройства и не желающей делиться сведениями LNE с оператором хост-устройства.

/if:interfaces/if:interface/bind-lne-name

Этот лист указывает экземпляр LNE, которому назначен интерфейс. Реализациям следует обращать пристальное внимание к разрешению изменений в этом листе, поскольку удаление интерфейса из LNE может оказывать серьёзное влияние на работу LNE, подобно физическому удалению интерфейса из устройства. Реализации могут отвергать переназначение, используя описанные выше сообщения об ошибках.

Несанкционированный доступ к любому из этих списков может негативно влиять на безопасность локального устройства и сети. Это может приводить к сбоям в работе сети, доставке пакетов в непредусмотренные места и иным проблемам.

5. Взаимодействие с IANA

Этот документ регистрирует URI в реестре IETF XML Registry [RFC3688].

URI: `urn:ietf:params:xml:ns:yang:ietf-logical-network-element`

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

Документ регистрирует модуль YANG в реестре YANG Module Names [RFC6020].

```
name:          ietf-logical-network-element
namespace:    urn:ietf:params:xml:ns:yang:ietf-logical-network-element
prefix:       lne
reference:    RFC 8530
```

6. Модель логического элемента сети

Структура определённой в документе модели описана в представленном ниже модуле YANG.

```
<CODE BEGINS> file "ietf-logical-network-element@2019-01-25.yang"
module ietf-logical-network-element {
  yang-version 1.1;
```

```

// Пространство имен

namespace "urn:ietf:params:xml:ns:yang:ietf-logical-network-element";
prefix lne;

// Импорт некоторых базовых типов

import ietf-interfaces {
  prefix if;
  reference
    "RFC 8343: A YANG Data Model for Interface Management";
}
import ietf-yang-schema-mount {
  prefix yangmnt;
  reference
    "RFC 8528: YANG Schema Mount";
}

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

  Author: Lou Berger
         <mailto:lberger@labn.net>

  Author: Christian Hopps
         <mailto:chopps@chopps.org>

  Author: Acee Lindem
         <mailto:acee@cisco.com>

  Author: Dean Bogdanovic
         <mailto:ivandean@gmail.com>";
description
  "Этот модуль служит для поддержки множества логических
  сетевых элементов в одной физической или виртуальной системе.

  Авторские права (Copyright (c) 2019) принадлежат IETF Trust
  и лицам, указанным в качестве авторов кода. Все права защищены.

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

  Эта версия данного модуля YANG является частью RFC 8530, где
  правовые вопросы рассмотрены более полно.";

revision 2019-01-25 {
  description
    "Исходный выпуск.";
  reference
    "RFC 8530: YANG Model for Logical Network Elements";
}

// Операторы определения устройства верхнего уровня

container logical-network-elements {
  description
    "Позволяет сетевому устройству поддерживать множество экземпляров
    логических сетевых элементов (устройств).";
  list logical-network-element {
    key "name";
    description
      "Список логических сетевых элементов.";
    leaf name {
      type string;
      description
        "Уникальный на устройстве идентификатор LNE.";
    }
    leaf managed {
      type boolean;
      default "true";
      description
        "Значение true разрешает хосту доступ к сведениям LNE через
        корень точки монтирования. Это значение неизменно во всех
        реализациях.";
    }
    leaf description {
      type string;
      description
        "Описание логического элемента сети.";
    }
  }
}

```

```
}
container root {
  description
    "Контейнер для точки монтирования.";
  yangmnt:mount-point "root" {
    description
      "Корень для моделей, поддерживаемых на LNE. Эта точка
      монтирования может быть встроенной (inline) в
      зависимости от реализации сервера. В неё всегда
      НУЖНО включать библиотеку YANG и экземпляр интерфейса.

      Кода связанный лист managed имеет значение false, при
      любой операции доступа к информации ниже корня НУЖНО
      возвращать отказ с error-tag access-denied и
      error-app-tag lne-not-managed.";
  }
}
}
}

// Оператор дополнения (augment)
augment "/if:interfaces/if:interface" {
  description
    "Добавляет узел для идентификации логического элемента,
    связанного с интерфейсом. Применяется к интерфейсам, которые
    могут быть назначены логическим элементам.

    Отметим, что будет возвращаться стандартная ошибка, если
    указанный leafref отсутствует. Если интерфейс по какой-либо
    причине не может быть назначен, операцию НУЖНО прервать с
    возвратом error-tag operation-failed и error-app-tag
    lne-assignment-failed. СЛЕДУЕТ также предоставлять значимый
    элемент error-info с указанием причины отказа в назначении.";
  leaf bind-lne-name {
    type leafref {
      path "/logical-network-elements/logical-network-element/name";
    }
    description
      "Идентификатор LNE, с которым связан интерфейс.";
  }
}

// Оператор уведомления (notification)
notification bind-lne-name-failed {
  description
    "Указывает ошибку при связывании интерфейса с LNE. Создается
    лишь после первоначального успеха при установке bind-lne-name.";
  leaf name {
    type leafref {
      path "/if:interfaces/if:interface/if:name";
    }
    mandatory true;
    description
      "Имя интерфейса, связанного с отказом.";
  }
  leaf bind-lne-name {
    type leafref {
      path "/if:interfaces/if:interface/lne:bind-lne-name";
    }
    mandatory true;
    description
      "Значение bind-lne-name, связанное с отказом.";
  }
  leaf error-info {
    type string;
    description
      "Указывает причину отказа (необязательный элемент).";
  }
}
}
}
<CODE ENDS>
```

7. Литература

7.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](https://www.rfc-editor.org/info/rfc2119), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](https://www.rfc-editor.org/info/rfc3688), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](https://www.rfc-editor.org/info/rfc6020), DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, [RFC 8341](#), DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 8343](#), DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [RFC 8446](#), DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount", [RFC 8528](#), DOI 10.17487/RFC8528, March 2019, <<https://www.rfc-editor.org/info/rfc8528>>.

7.2. Дополнительная литература

- [DEVICE-MODEL] Lindem, A., Berger, L., Bogdanovic, D., and C. Hopps, "Network Device YANG Logical Organization", Work in Progress, draft-ietf-rtgwg-device-model-02, March 2017.
- [OSPF-YANG] Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "YANG Data Model for OSPF Protocol", Work in Progress¹, draft-ietf-ospf-yang-21, January 2019.
- [RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", [RFC 7317](#), DOI 10.17487/RFC7317, August 2014, <<https://www.rfc-editor.org/info/rfc7317>>.
- [RFC7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", [RFC 7895](#), DOI 10.17487/RFC7895, June 2016, <<https://www.rfc-editor.org/info/rfc7895>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, [RFC 8340](#), DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8348] Bierman, A., Bjorklund, M., Dong, J., and D. Romascanu, "A YANG Data Model for Hardware Management", [RFC 8348](#), DOI 10.17487/RFC8348, March 2018, <<https://www.rfc-editor.org/info/rfc8348>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", [RFC 8349](#), DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", [RFC 8529](#), DOI 10.17487/RFC8529, March 2019, <<https://www.rfc-editor.org/info/rfc8529>>.

Приложение А. Примеры

В последующих параграфах представлены примеры использования элементов LNE.

А.1. LNE, управляемый хост-устройством

Ниже описан пример модели LNE, применяющей монтирование схемы для родительского управления. Пример устройства поддерживает несколько экземпляров LNE (логических маршрутизаторов), каждый из которых поддерживает интерфейсы L2 и L3 [RFC8343], базу маршрутных данных [RFC8349] и протокол OSPF [OSPF-YANG]. Каждое из этих свойств задано моделью YANG и свойство объединяются с использованием монтирования схемы YANG, как показано ниже. В примере указаны не все монтируемые модули, например, может монтироваться также модель управления оборудованием, заданная в [RFC8348].

```

module: ietf-logical-network-element
  +--rw logical-network-elements
    +--rw logical-network-element* [name]
      +--rw name string
      +--mp root
        +--ro yanglib:modules-state/
          | +--ro module-set-id string
          | +--ro module* [name revision]
          |   +--ro name yang:yang-identifier
        +--rw sys:system/
          | +--rw contact? string
          | +--rw hostname? inet:domain-name
          | +--rw authentication {authentication}?
          |   +--rw user-authentication-order* identityref
          |   +--rw user* [name] {local-users}?
          |   +--rw name string

```

¹Опубликовано в [RFC 9129](#). Прим. перев.

```

|         +--rw password?          ianach:crypt-hash
|         +--rw authorized-key* [name]
|         +--rw name                string
|         +--rw algorithm           string
|         +--rw key-data            binary
+--ro sys:system-state/
|   ...
+--rw rt:routing/
|   +--rw router-id?                yang:dotted-quad
|   +--rw control-plane-protocols
|       +--rw control-plane-protocol* [type name]
|           +--rw ospf:ospf/
|               +--rw areas
|                   +--rw area* [area-id]
|                       +--rw interfaces
|                           +--rw interface* [name]
|                               +--rw name if:interface-ref
|                                   +--rw cost?  uint16
+--rw if:interfaces/
    +--rw interface* [name]
        +--rw name                string
        +--rw ip:ipv4!/
            | +--rw address* [ip]
            | ...
module: ietf-interfaces
+--rw interfaces
+--rw interface* [name]
+--rw name                string
+--rw lne:bind-lne-name?  string
+--ro oper-status         enumeration

module: ietf-yang-library
+--ro modules-state
+--ro module-set-id      string
+--ro module* [name revision]
+--ro name                yang:yang-identifier

module: ietf-system
+--rw system
| +--rw contact?          string
| +--rw hostname?         inet:domain-name
| +--rw authentication {authentication}?
|     +--rw user-authentication-order* identityref
|     +--rw user* [name] {local-users}?
|         +--rw name                string
|         +--rw password?           ianach:crypt-hash
|         +--rw authorized-key* [name]
|             +--rw name                string
|             +--rw algorithm           string
|             +--rw key-data            binary
+--ro system-state
+--ro platform
| +--ro os-name?          string
| +--ro os-release?      string

```

Для реализации приведённой выше схемы устройство применяет показанный ниже экземпляр монтирования схемы.

```

"ietf-yang-schema-mount:schema-mounts": {
  "mount-point": [
    {
      "module": "ietf-logical-network-element",
      "label": "root",
      "shared-schema": {}
    }
  ]
}

```

Применяя реализацию монтирования схемы YANG, оператор может создавать экземпляры логических маршрутизаторов. Маршрутизатору может быть назначен интерфейс, к которому это маршрутизатор будет иметь доступ. Затем на этом интерфейсе можно включить протокол OSPF.

Для этой реализации родительская сессия управления имеет доступ к схемам родительской хост-системы и дочерним логическим маршрутизаторам. Кроме того, каждый логический маршрутизатор может предоставлять свою сессию управления, имеющую показанную ниже схему.

```

module: ietf-yang-library
+--ro modules-state
+--ro module-set-id      string
+--ro module* [name revision]
+--ro name                yang:yang-identifier

module: ietf-system
+--rw system
| +--rw contact?          string
| +--rw hostname?         inet:domain-name
| +--rw authentication {authentication}?
|     +--rw user-authentication-order* identityref

```

```

|     +--rw user* [name] {local-users}?
|     +--rw name          string
|     +--rw password?     ianach:crypt-hash
|     +--rw authorized-key* [name]
|     +--rw name          string
|     +--rw algorithm     string
|     +--rw key-data      binary
+--ro system-state
  +--ro platform
    +--ro os-name?       string
    +--ro os-release?   string

module: ietf-routing
  rw-- routing
    +--rw router-id?          yang:dotted-quad
    +--rw control-plane-protocols
      +--rw control-plane-protocol* [type name]
        +--rw ospf:ospf/
          +--rw areas
            +--rw area* [area-id]
              +--rw interfaces
                +--rw interface* [name]
                  +--rw name       if:interface-ref
                  +--rw cost?     uint16

module: ietf-interfaces
  +--rw interfaces
    +--rw interface* [name]
      +--rw name          string
      +--ro oper-status   enumeration

```

A.1.1. Данные конфигурации

Далее приведён пример с настройкой двух коинетских элементов LNE.

```

{
  "ietf-logical-network-element:logical-network-elements": {
    "logical-network-element": [
      {
        "name": "cust1",
        "root": {
          "ietf-system:system": {
            "authentication": {
              "user": [
                {
                  "name": "john",
                  "password": "$0$password"
                }
              ]
            }
          }
        },
        "ietf-routing:routing": {
          "router-id": "192.0.2.1",
          "control-plane-protocols": {
            "control-plane-protocol": [
              {
                "type": "ietf-routing:ospf",
                "name": "1",
                "ietf-ospf:ospf": {
                  "af": "ipv4",
                  "areas": {
                    "area": [
                      {
                        "area-id": "203.0.113.1",
                        "interfaces": {
                          "interface": [
                            {
                              "name": "eth1",
                              "cost": 10
                            }
                          ]
                        }
                      }
                    ]
                  }
                }
              ]
            }
          }
        },
        "ietf-interfaces:interfaces": {
          "interface": [
            {
              "name": "eth1",
              "type": "iana-if-type:ethernetCsmacd",
              "ietf-ip:ipv4": {
                "address": [

```

```

        {
          "ip": "192.0.2.11",
          "prefix-length": 24
        }
      ]
    },
    "ietf-ip:ipv6": {
      "address": [
        {
          "ip": "2001:db8:0:2::11",
          "prefix-length": 64
        }
      ]
    }
  ]
}
},
{
  "name": "cust2",
  "root": {
    "ietf-system:system": {
      "authentication": {
        "user": [
          {
            "name": "john",
            "password": "$0$password"
          }
        ]
      }
    }
  },
  "ietf-routing:routing": {
    "router-id": "192.0.2.2",
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "ietf-routing:ospf",
          "name": "1",
          "ietf-ospf:ospf": {
            "af": "ipv4",
            "areas": {
              "area": [
                {
                  "area-id": "203.0.113.1",
                  "interfaces": {
                    "interface": [
                      {
                        "name": "eth1",
                        "cost": 10
                      }
                    ]
                  }
                }
              ]
            }
          }
        }
      ]
    }
  },
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1",
        "type": "iana-if-type:ethernetCsmacd",
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.11",
              "prefix-length": 24
            }
          ]
        }
      }
    ]
  }
}
},
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",

```

```

    "type": "iana-if-type:ethernetCsmacd",
    "ietf-ip:ipv4": {
      "address": [
        {
          "ip": "192.0.2.10",
          "prefix-length": 24
        }
      ]
    },
    "ietf-ip:ipv6": {
      "address": [
        {
          "ip": "2001:db8:0:2::10",
          "prefix-length": 64
        }
      ]
    }
  },
  {
    "name": "cust1:eth1",
    "type": "iana-if-type:ethernetCsmacd",
    "ietf-logical-network-element:bind-lne-name": "cust1"
  },
  {
    "name": "cust2:eth1",
    "type": "iana-if-type:ethernetCsmacd",
    "ietf-logical-network-element:bind-lne-name": "cust2"
  }
]
},
"ietf-system:system": {
  "authentication": {
    "user": [
      {
        "name": "root",
        "password": "$0$password"
      }
    ]
  }
}
}
}

```

A.1.2. Данные состояния

Следующий пример показывает возможные данные состояний, связанные с предыдущей конфигурацией.

```

{
  "ietf-logical-network-element:logical-network-elements": {
    "logical-network-element": [
      {
        "name": "cust1",
        "root": {
          "ietf-yang-library:modules-state": {
            "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
            "module": [
              {
                "name": "iana-if-type",
                "revision": "2014-05-08",
                "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
                "conformance-type": "import"
              },
              {
                "name": "ietf-inet-types",
                "revision": "2013-07-15",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
                "conformance-type": "import"
              },
              {
                "name": "ietf-interfaces",
                "revision": "2014-05-08",
                "feature": [
                  "arbitrary-names",
                  "pre-provisioning"
                ],
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-ip",
                "revision": "2014-06-16",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
                "conformance-type": "implement"
              }
            ]
          }
        }
      }
    ]
  }
}

```

```

    {
      "name": "ietf-ospf",
      "revision": "2018-03-03",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-routing",
      "revision": "2018-03-13",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-system",
      "revision": "2014-08-06",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-system",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-yang-library",
      "revision": "2016-06-21",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-yang-library",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-yang-types",
      "revision": "2013-07-15",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-yang-types",
      "conformance-type": "import"
    }
  ]
},
"ietf-system:system-state": {
  "platform": {
    "os-name": "NetworkOS"
  }
},
"ietf-routing:routing": {
  "router-id": "192.0.2.1",
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-routing:ospf",
        "name": "1",
        "ietf-ospf:ospf": {
          "af": "ipv4",
          "areas": {
            "area": [
              {
                "area-id": "203.0.113.1",
                "interfaces": {
                  "interface": [
                    {
                      "name": "eth1",
                      "cost": 10
                    }
                  ]
                }
              }
            ]
          }
        }
      }
    ]
  }
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "phys-address": "00:01:02:A1:B1:C1",
      "statistics": {
        "discontinuity-time": "2017-06-26T12:34:56-05:00"
      },
      "ietf-ip:ipv4": {
        "address": [
          {
            "ip": "192.0.2.11",
            "prefix-length": 24
          }
        ]
      }
    }
  ],
},

```

```

    "ietf-ip:ipv6": {
      "address": [
        {
          "ip": "2001:db8:0:2::11",
          "prefix-length": 64
        }
      ]
    }
  ]
}
},
{
  "name": "cust2",
  "root": {
    "ietf-yang-library:modules-state": {
      "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
      "module": [
        {
          "name": "iana-if-type",
          "revision": "2014-05-08",
          "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
          "conformance-type": "import"
        },
        {
          "name": "ietf-inet-types",
          "revision": "2013-07-15",
          "namespace":
            "urn:ietf:params:xml:ns:yang:ietf-inet-types",
          "conformance-type": "import"
        },
        {
          "name": "ietf-interfaces",
          "revision": "2014-05-08",
          "feature": [
            "arbitrary-names",
            "pre-provisioning"
          ],
          "namespace":
            "urn:ietf:params:xml:ns:yang:ietf-interfaces",
          "conformance-type": "implement"
        },
        {
          "name": "ietf-ip",
          "revision": "2014-06-16",
          "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
          "conformance-type": "implement"
        },
        {
          "name": "ietf-ospf",
          "revision": "2018-03-03",
          "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
          "conformance-type": "implement"
        },
        {
          "name": "ietf-routing",
          "revision": "2018-03-13",
          "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
          "conformance-type": "implement"
        },
        {
          "name": "ietf-system",
          "revision": "2014-08-06",
          "namespace": "urn:ietf:params:xml:ns:yang:ietf-system",
          "conformance-type": "implement"
        },
        {
          "name": "ietf-yang-library",
          "revision": "2016-06-21",
          "namespace":
            "urn:ietf:params:xml:ns:yang:ietf-yang-library",
          "conformance-type": "implement"
        },
        {
          "name": "ietf-yang-types",
          "revision": "2013-07-15",
          "namespace":
            "urn:ietf:params:xml:ns:yang:ietf-yang-types",
          "conformance-type": "import"
        }
      ]
    }
  },
  "ietf-system:system-state": {
    "platform": {
      "os-name": "NetworkOS"
    }
  }
}

```

```

    }
  },
  "ietf-routing:routing": {
    "router-id": "192.0.2.2",
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "ietf-routing:ospf",
          "name": "1",
          "ietf-ospf:ospf": {
            "af": "ipv4",
            "areas": {
              "area": [
                {
                  "area-id": "203.0.113.1",
                  "interfaces": {
                    "interface": [
                      {
                        "name": "eth1",
                        "cost": 10
                      }
                    ]
                  }
                }
              ]
            }
          }
        }
      ]
    }
  },
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "phys-address": "00:01:02:A1:B1:C2",
        "statistics": {
          "discontinuity-time": "2017-06-26T12:34:56-05:00"
        },
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.11",
              "prefix-length": 24
            }
          ]
        }
      }
    ]
  },
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "phys-address": "00:01:02:A1:B1:C0",
        "statistics": {
          "discontinuity-time": "2017-06-26T12:34:56-05:00"
        },
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.10",
              "prefix-length": 24
            }
          ]
        },
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:2::10",
              "prefix-length": 64
            }
          ]
        }
      }
    ]
  },
  {
    "name": "cust1:eth1",

```

```

    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "phys-address": "00:01:02:A1:B1:C1",
    "statistics": {
      "discontinuity-time": "2017-06-26T12:34:56-05:00"
    },
    "ietf-logical-network-element:bind-lne-name": "cust1"
  },
  {
    "name": "cust2:eth1",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "phys-address": "00:01:02:A1:B1:C2",
    "statistics": {
      "discontinuity-time": "2017-06-26T12:34:56-05:00"
    },
    "ietf-logical-network-element:bind-lne-name": "cust2"
  }
]
},
"ietf-system:system-state": {
  "platform": {
    "os-name": "NetworkOS"
  }
},
"ietf-yang-library:modules-state": {
  "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
  "module": [
    {
      "name": "iana-if-type",
      "revision": "2014-05-08",
      "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
      "conformance-type": "import"
    },
    {
      "name": "ietf-inet-types",
      "revision": "2013-07-15",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
      "conformance-type": "import"
    },
    {
      "name": "ietf-interfaces",
      "revision": "2014-05-08",
      "feature": [
        "arbitrary-names",
        "pre-provisioning"
      ],
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-ip",
      "revision": "2014-06-16",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-logical-network-element",
      "revision": "2017-03-13",
      "feature": [
        "bind-lne-name"
      ],
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-logical-network-element",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-ospf",
      "revision": "2018-03-03",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-routing",
      "revision": "2018-03-13",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-system",
      "revision": "2014-08-06",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-system",
      "conformance-type": "implement"
    }
  ],

```

```

{
  "name": "ietf-yang-library",
  "revision": "2016-06-21",
  "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-library",
  "conformance-type": "implement"
},
{
  "name": "ietf-yang-schema-mount",
  "revision": "2017-05-16",
  "namespace":
    "urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount",
  "conformance-type": "implement"
},
{
  "name": "ietf-yang-types",
  "revision": "2013-07-15",
  "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
  "conformance-type": "import"
}
]
},
"ietf-yang-schema-mount:schema-mounts": {
  "mount-point": [
    {
      "module": "ietf-logical-network-element",
      "label": "root",
      "shared-schema": {}
    }
  ]
}
}
}

```

A.2. LNE с самоуправлением

Здесь приведён пример модели LNE с монтированием схемы для независимого от потомка управления. Устройство поддерживает несколько экземпляров LNE (логических маршрутизаторов), каждый из которых поддерживает интерфейсы L2 и L3 [RFC8343], базу маршрутных данных [RFC8349] и протокол OSPF [OSPF-YANG]. Каждое из этих свойств задано моделью YANG и свойства объединяются с использованием монтирования схемы YANG, как показано ниже.

```

module: ietf-logical-network-element
  +--rw logical-network-elements
    +--rw logical-network-element* [name]
      +--rw name string
      +--mp root
        // Внутренние модули LNE не видны родительскому
        // управлению. Потомок управляет своими модулями,
        // включая ietf-routing и ietf-interfaces.

module: ietf-interfaces
  +--rw interfaces
    +--rw interface* [name]
      +--rw name string
      +--rw lne:bind-lne-name? string
      +--ro oper-status enumeration

module: ietf-yang-library
  +--ro modules-state
    +--ro module-set-id string
    +--ro module* [name revision]
      +--ro name yang:yang-identifier

module: ietf-system
  +--rw system
  | +--rw contact? string
  | +--rw hostname? inet:domain-name
  | +--rw authentication {authentication}?
  |   +--rw user-authentication-order* identityref
  |   +--rw user* [name] {local-users}?
  |     +--rw name string
  |     +--rw password? ianach:crypt-hash
  |     +--rw authorized-key* [name]
  |       +--rw name string
  |       +--rw algorithm string
  |       +--rw key-data binary
  +--ro system-state
    +--ro platform
    | +--ro os-name? string
    | +--ro os-release? string

```

Для реализации этой схемы устройство применяет показанный ниже экземпляр монтирования схемы.

```

"ietf-yang-schema-mount:schema-mounts": {
  "mount-point": [
    {
      "module": "ietf-logical-network-element",

```

```

    "label": "root",
    "inline": {}
  }
}

```

С использованием монтирования схемы YANG оператор может создавать экземпляры логических маршрутизаторов, каждый из которых имеет свои встроенные (inline) модули. Логическому маршрутизатору может быть назначен интерфейс, к которому маршрутизатор имеет доступ. Каждый логический маршрутизатор независимо управляет своим набором модулей, который может различаться у разных маршрутизаторов. Ниже приведён пример набора схем, реализованного одним из логических маршрутизаторов.

```

module: ietf-yang-library
  +--ro modules-state
    +--ro module-set-id string
    +--ro module* [name revision]
      +--ro name yang:yang-identifier

module: ietf-system
  +--rw system
    | +--rw contact? string
    | +--rw hostname? inet:domain-name
    | +--rw authentication {authentication}?
    |   +--rw user-authentication-order* identityref
    |   +--rw user* [name] {local-users}?
    |     +--rw name string
    |     +--rw password? ianach:crypt-hash
    |     +--rw authorized-key* [name]
    |       +--rw name string
    |       +--rw algorithm string
    |       +--rw key-data binary
  +--ro system-state
    +--ro platform
      | +--ro os-name? string
      | +--ro os-release? string

module: ietf-routing
  +--rw routing
    +--rw router-id? yang:dotted-quad
    +--rw control-plane-protocols
      +--rw control-plane-protocol* [type name]
        +--rw ospf:ospf/
          +--rw areas
            +--rw area* [area-id]
              +--rw interfaces
                +--rw interface* [name]
                  +--rw name if:interface-ref
                  +--rw cost? uint16

module: ietf-interfaces
  +--rw interfaces
    +--rw interface* [name]
      +--rw name string
      +--ro oper-status enumeration

```

A.2.1. Данные конфигурации

Каждый из дочерних виртуальных маршрутизаторов управляется через свои сессии и данные конфигурации.

A.2.1.1. Логический элемент vnf1

Ниже показан пример данных конфигурации для LNE с именем vnf1.

```

{
  "ietf-system:system": {
    "authentication": {
      "user": [
        {
          "name": "john",
          "password": "$0$password"
        }
      ]
    }
  },
  "ietf-routing:routing": {
    "router-id": "192.0.2.1",
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "ietf-routing:ospf",
          "name": "1",
          "ietf-ospf:ospf": {
            "af": "ipv4",
            "areas": {
              "area": [
                {
                  "area-id": "203.0.113.1",
                  "interfaces": {

```



```

    "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
    "conformance-type": "import"
  },
  {
    "name": "ietf-inet-types",
    "revision": "2013-07-15",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
    "conformance-type": "import"
  },
  {
    "name": "ietf-interfaces",
    "revision": "2014-05-08",
    "feature": [
      "arbitrary-names",
      "pre-provisioning"
    ],
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-ip",
    "revision": "2014-06-16",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-logical-network-element",
    "revision": "2017-03-13",
    "feature": [
      "bind-lne-name"
    ],
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-logical-network-element",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-system",
    "revision": "2014-08-06",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-system",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-yang-library",
    "revision": "2016-06-21",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-library",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-yang-schema-mount",
    "revision": "2017-05-16",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-yang-types",
    "revision": "2013-07-15",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
    "conformance-type": "import"
  }
]
},
"ietf-yang-schema-mount:schema-mounts": {
  "mount-point": [
    {
      "module": "ietf-logical-network-element",
      "label": "root",
      "inline": {}
    }
  ]
}
}
}

```

A.2.2.2. Логический элемент vnf1

Ниже показаны данные состояния для элемента LNE с именем vnf1.

```

{
  "ietf-yang-library:modules-state": {
    "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
    "module": [
      {
        "name": "iana-if-type",
        "revision": "2014-05-08",
        "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
        "conformance-type": "import"
      }
    ]
  }
}

```

```

    },
    {
      "name": "ietf-inet-types",
      "revision": "2013-07-15",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
      "conformance-type": "import"
    },
    {
      "name": "ietf-interfaces",
      "revision": "2014-05-08",
      "feature": [
        "arbitrary-names",
        "pre-provisioning"
      ],
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-ip",
      "revision": "2014-06-16",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-ospf",
      "revision": "2018-03-03",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-routing",
      "revision": "2018-03-13",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-system",
      "revision": "2014-08-06",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-system",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-yang-library",
      "revision": "2016-06-21",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-library",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-yang-types",
      "revision": "2013-07-15",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
      "conformance-type": "import"
    }
  ]
},

"ietf-system:system-state": {
  "platform": {
    "os-name": "NetworkOS"
  }
},

"ietf-routing:routing": {
  "router-id": "192.0.2.1",
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-routing:ospf",
        "name": "1",
        "ietf-ospf:ospf": {
          "af": "ipv4",
          "areas": {
            "area": [
              {
                "area-id": "203.0.113.1",
                "interfaces": {
                  "interface": [
                    {
                      "name": "eth1",
                      "cost": 10
                    }
                  ]
                }
              }
            ]
          }
        }
      }
    ]
  }
}
}

```

```

    }
  ]
}
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "phys-address": "00:01:02:A1:B1:C1",
      "statistics": {
        "discontinuity-time": "2017-06-26T12:34:56-05:00"
      },
      "ietf-ip:ipv4": {
        "address": [
          {
            "ip": "192.0.2.11",
            "prefix-length": 24
          }
        ]
      }
    }
  ]
}
}
}
}

```

A.2.2.3. Логический элемент vnf2

Ниже показаны данные состояния для элемента LNE с именем vnf2.

```

{
  "ietf-yang-library:modules-state": {
    "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
    "module": [
      {
        "name": "iana-if-type",
        "revision": "2014-05-08",
        "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
        "conformance-type": "import"
      },
      {
        "name": "ietf-inet-types",
        "revision": "2013-07-15",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
        "conformance-type": "import"
      },
      {
        "name": "ietf-interfaces",
        "revision": "2014-05-08",
        "feature": [
          "arbitrary-names",
          "pre-provisioning"
        ],
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-ip",
        "revision": "2014-06-16",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-ospf",
        "revision": "2018-03-03",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-routing",
        "revision": "2018-03-13",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-system",
        "revision": "2014-08-06",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-system",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-yang-library",
        "revision": "2016-06-21",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-library",

```

```

    "conformance-type": "implement"
  },
  {
    "name": "ietf-yang-types",
    "revision": "2013-07-15",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
    "conformance-type": "import"
  }
]
},
"ietf-system:system-state": {
  "platform": {
    "os-name": "NetworkOS"
  }
},
"ietf-routing:routing": {
  "router-id": "192.0.2.2",
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-routing:ospf",
        "name": "1",
        "ietf-ospf:ospf": {
          "af": "ipv4",
          "areas": {
            "area": [
              {
                "area-id": "203.0.113.1",
                "interfaces": {
                  "interface": [
                    {
                      "name": "eth1",
                      "cost": 10
                    }
                  ]
                }
              }
            ]
          }
        }
      }
    ]
  }
},
"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "eth1",
      "type": "iana-if-type:ethernetCsmacd",
      "oper-status": "up",
      "phys-address": "00:01:02:A1:B1:C2",
      "statistics": {
        "discontinuity-time": "2017-06-26T12:34:56-05:00"
      },
      "ietf-ip:ipv4": {
        "address": [
          {
            "ip": "192.0.2.11",
            "prefix-length": 24
          }
        ]
      }
    }
  ]
}
}

```

Благодарности

Команда разработчиков Routing Area Yang Architecture включала Acee Lindem, Anees Shaikh, Christian Hopps, Dean Bogdanovic, Lou Berger, Qin Wu, Rob Shakir, Stephane Litkowski, Yan Gang. Полезные обзорные комментарии также представили Martin Bjorklund, John Scudder, Dan Romascanu, Taylor Yu.

Этот документ был выведен из Network Device YANG Logical Organization [DEVICE-MODEL].

Спасибо Alvaro Retana за рецензию IESG.

Адреса авторов

Lou Berger
 LabN Consulting, L.L.C.
 Email: lberger@labn.net

Christian Hopps

LabN Consulting, L.L.C.

Email: chopps@chopps.org

Acee Lindem

Cisco Systems

301 Midenhall Way

Cary, NC 27513

United States of America

Email: acee@cisco.com

Dean Bogdanovic

Volta Networks

Email: ivandean@gmail.com

Xufeng Liu

Volta Networks

Email: xufeng.liu.ietf@gmail.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru