

Internet Engineering Task Force (IETF)  
Request for Comments: 8343  
Obsoletes: 7223  
Category: Standards Track  
ISSN: 2070-1721

M. Bjorklund  
Tail-f Systems  
March 2018

## A YANG Data Model for Interface Management

### Модель данных YANG для управления интерфейсами

#### Тезисы

В этом документе определена модель данных YANG для управления сетевыми интерфейсами. Предполагается добавление (augment) расширений для конкретных типов интерфейсов к определенной здесь базовой модели. Модель включает данные конфигурации и состояния (информация о состоянии и счетчики статистики).

Определенная здесь модель данных YANG соответствует архитектуре хранилищ конфигурации NMDA<sup>1</sup>, определенной в RFC 8342.

Данный документ отменяет RFC 7223.

#### Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF<sup>2</sup> и представляет согласованный взгляд сообщества IETF. Документ прошел открытое обсуждение и был одобрен для публикации IESG<sup>3</sup>. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <https://www.rfc-editor.org/info/rfc8343>.

#### Авторские права

Авторские права (Copyright (c) 2018) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

Этот документ является субъектом прав и ограничений, перечисленных в BCP 78 и IETF Trust Legal Provisions и относящихся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно, поскольку в них описаны права и ограничения, относящиеся к данному документу. Фрагменты программного кода, включенные в этот документ, распространяются в соответствии с упрощенной лицензией BSD, как указано в параграфе 4.e документа Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

## Оглавление

1. Введение.....	2
1.1. Отличия от RFC 7223.....	2
1.2. Терминология.....	2
1.3. Диаграммы деревьев.....	2
2. Цели.....	2
3. Модель данных интерфейса.....	3
3.1. Списки интерфейсов.....	3
3.2. Указание интерфейсов.....	4
3.3. Уровни интерфейсов.....	4
4. Связь с IF-MIB.....	4
5. Модуль YANG.....	5
6. Взаимодействие с IANA.....	19
7. Вопросы безопасности.....	19
8. Литература.....	20
8.1. Нормативные документы.....	20
8.2. Дополнительная литература.....	20
Приложение А. Пример модуля для интерфейса Ethernet.....	20
Приложение В. Пример модуля для связки интерфейсов Ethernet.....	21
Приложение С. Пример модуля для интерфейса VLAN.....	21
Приложение D. Пример отклика NETCONF <get-config>.....	22
Приложение E. Пример отклика NETCONF <get-data>.....	23
Приложение F. Примеры схем именования интерфейсов.....	24
F.1. Маршрутизатор с ограничениями для имен интерфейсов.....	24
F.2. Маршрутизатор с произвольными именами интерфейсов.....	25
F.3. Коммутатор Ethernet с ограничениями для имен интерфейсов.....	25
F.4. Типовой хост с ограничениями для имен интерфейсов.....	25

<sup>1</sup>Network Management Datastore Architecture — архитектура хранилища данных сетевого управления.

<sup>2</sup>Internet Engineering Task Force.

<sup>3</sup>Internet Engineering Steering Group.

F.5. Типовой хост с произвольными именами интерфейсов.....	26
Благодарности.....	26
Адрес автора.....	26

## 1. Введение

Этот документ определяет модель данных YANG [RFC7950] для управления сетевыми интерфейсами. Предполагается, что базовая модель будет дополнена (augment) моделями данных для конкретных типов интерфейсов.

Сетевые интерфейсы являются важной частью управления для множества протоколов Internet. Поэтому важно разработать общую модель данных для идентификации, настройки и мониторинга интерфейсов.

Модель включает данные конфигурации и состояния (информация о состоянии и счетчики статистики).

Эта версия модели данных интерфейсов поддерживает архитектуру хранилищ конфигурации NMDA [RFC8342].

### 1.1. Отличия от RFC 7223

Субдерево «/interfaces-state» с узлами данных config false было исключено и все узлы config false сейчас представлены в субдереве «/interfaces».

Серверы, не реализующие NMDA или желающие поддерживать клиентов, не реализующих NMDA, **могут** реализовать отмененное дерево «/interfaces-state».

### 1.2. 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.

Ниже приведены определения используемых в документе терминов.

#### ***system-controlled interface – управляемый системой интерфейс***

Интерфейс называют управляемым системой (system-controlled), если система создает или удаляет интерфейс независимо от того, был ли он явно настроен. Примерами являются интерфейсы, представляющий физические компоненты, которые могут добавляться в систему или удаляться из нее (например, линейные платы или подключаемые в процессе работы беспроводные интерфейсы). Управляемые системой интерфейсы могут появляться при включении той или иной функциональности (например, loopback-интерфейс при включении стека IP).

#### ***user-controlled interface – управляемый пользователем интерфейс***

Интерфейс называют управляемым пользователем (user-controlled), если создание интерфейса определяется его явной настройкой в хранилище рабочей конфигурации, а удаление – явным удалением конфигурации интерфейса из этого хранилища. Примерами являются интерфейсы VLAN, настроенные на управляемых системой интерфейсах Ethernet.

Ниже приведен список используемых терминов из [RFC8342].

- client - клиент;
- server — сервер;
- configuration — конфигурация;
- system state — состояние системы;
- operational state — операционное (рабочее) состояние;
- intended configuration — предполагаемая конфигурация;
- running configuration datastore — хранилище рабочей конфигурации;
- operational state datastore — хранилище данных операционного состояния.

Ниже приведен список используемых терминов из [RFC7950].

- augment - дополнение;
- data model - модель данных;
- data node - узел данных.

### 1.3. Диаграммы деревьев

Диаграммы деревьев в этом документе используют нотацию, определенную в [RFC8340].

## 2. Цели

В этом разделе описаны некоторые из целей разработки модели, представленной в разделе 5.

- Ясно, что существующие реализации будут отображать описанную здесь модель на свои фирменные, естественные модели данных. Для облегчения этого модель данных должна быть простой.
- Модель данных должна подходить для новых реализаций без отображения на естественную модель.
- Ссылки на интерфейсы должны быть как можно проще, предпочтительно с использованием одного leafref.

- Отображение на ifIndex [RFC2863], используемое протоколом SNMP<sup>1</sup> для указания интерфейсов, должно быть четким.
- Модель должна поддерживать уровни интерфейсов - как (1) простые, где один интерфейс работает поверх единственного другого, так и (2) более сложные, где один интерфейс может быть результатом агрегирования N других интерфейсов или N интерфейсов могут мультиплексироваться в один.
- Модель данных должна поддерживать представление предварительно подготовленной конфигурации интерфейса, т. е. возможность настройки конфигурации для отсутствующего в системе физического интерфейса. В устройствах, поддерживающих динамическое добавление и удаление физических интерфейсов рекомендуется также поддерживать предварительную настройку конфигурации интерфейса.
- Модель данных должна поддерживать как физические, так и логические интерфейсы.
- В модель данных следует включать доступные только для чтения счетчики, обеспечивающие сбор статистики переданных и принятых октетов и байтов, числа пакетов, принятых с ошибками, а также пакетов, которые не были переданы по причине ошибок.

### 3. Модель данных интерфейса

А этом документе определен модуль YANG «ietf-interfaces», который имеет показанную ниже структуру без субдерева «/interfaces-state».

```

module: ietf-interfaces
  +--rw interfaces
    +--rw interface* [name]
      +--rw name                string
      +--rw description?       string
      +--rw type                identityref
      +--rw enabled?           boolean
      +--rw link-up-down-trap-enable? enumeration {if-mib}?
      +--ro admin-status        enumeration {if-mib}?
      +--ro oper-status         enumeration
      +--ro last-change?       yang:date-and-time
      +--ro if-index            int32 {if-mib}?
      +--ro phys-address?       yang:phys-address
      +--ro higher-layer-if*    interface-ref
      +--ro lower-layer-if*    interface-ref
      +--ro speed?              yang:gauge64
      +--ro statistics
        +--ro discontinuity-time yang:date-and-time
        +--ro in-octets?         yang:counter64
        +--ro in-unicast-pkts?   yang:counter64
        +--ro in-broadcast-pkts? yang:counter64
        +--ro in-multicast-pkts? yang:counter64
        +--ro in-discards?      yang:counter32
        +--ro in-errors?        yang:counter32
        +--ro in-unknown-protos? yang:counter32
        +--ro out-octets?       yang:counter64
        +--ro out-unicast-pkts? yang:counter64
        +--ro out-broadcast-pkts? yang:counter64
        +--ro out-multicast-pkts? yang:counter64
        +--ro out-discards?     yang:counter32
        +--ro out-errors?      yang:counter32

```

#### 3.1. Списки интерфейсов

Представленная в документе модель данных использует плоский список интерфейсов (/interfaces/interface), каждый из которых указывается именем. Кроме того, каждый интерфейс имеет обязательный лист type.

Модуль iana-if-type [RFC7224] определяет отождествления YANG для типов интерфейсов в поддерживаемом IANA реестре «ifType definitions».

Имеется один список настроенных интерфейсов (/interfaces/interface) и отдельный список рабочих состояний интерфейсов (/interfaces-state/interface).

Предполагается, что модели данных для конкретных типов дополняют списки интерфейсов и возможно будут использовать лист type, чтобы сделать дополнения условными.

В качестве примера возможного дополнения для конкретного типа интерфейса рассмотрим приведенный ниже фрагмент кода YANG. Более полный пример приведен в Приложении А.

```

import interfaces {
  prefix "if";
}
import iana-if-type {
  prefix ianaift;
}

augment "/if:interfaces/if:interface" {

```

<sup>1</sup>Simple Network Management Protocol - простой протокол сетевого управления.

```
when "if:type = 'ianaift:ethernetCsmacd'";

    container ethernet {
        leaf duplex {
            ...
        }
    }
}
```

Для управляемых системой интерфейсов name является зависимым от устройства именем интерфейса.

Если устройство поддерживает произвольное именование управляемых пользователем интерфейсов, сервер анонсирует свойство arbitrary-names. Если сервер не анонсирует это свойство, имена управляемых пользователем интерфейсов **должны** соответствовать схеме именования для устройства. Способ получения клиентом информации о схемах именования таких устройств выходит за рамки документа. Примеры приведены в приложениях E.1 и E.2.

Когда системно-управляемый интерфейс создается системой в операционном состоянии, она пытается применить предполагаемую с именем нового интерфейса. Если такой конфигурации не найдено или заданный в конфигурации тип не соответствует типу реального интерфейса, система создает интерфейс без применения явной конфигурации.

При создании управляемого пользователем интерфейса имя этого интерфейса определяет конфигурация.

В зависимости от операционной системы и физической точки, к которой интерфейс подключается или из которой удаляется, для реализации может оказаться невозможным обеспечение предсказуемых и согласованных имен для управляемых системой интерфейсов в циклах вставки-удаления, а также в ожидании первой вставки. Поэтому возможность предоставлять конфигурации для таких интерфейсов зависит от реализации и не может предполагаться во всех случаях.

## 3.2. Указание интерфейсов

Интерфейс указывается именем, которое уникально в рамках сервера. Это свойство фиксируется в определениях типа (typedef) interface-ref, которые другим модулям YANG **следует** применять, когда нужно указать интерфейс.

## 3.3. Уровни интерфейсов

Не существует общего механизма настройки интерфейса так, чтобы он размещался «поверх» некоего другого интерфейса. Предполагается, что модели для конкретных типов интерфейсов будут определять свои узлы данных для уровней интерфейсов с помощью типов interface-ref для указания нижележащих уровней.

Ниже приведен пример модели с такими узлами. Более полный пример представлен в Приложении B.

```
import interfaces {
    prefix "if";
}
import iana-if-type {
    prefix ianaift;
}

augment "/if:interfaces/if:interface" {
    when "if:type = 'ianaift:ieee8023adLag'";

    leaf-list slave-if {
        type if:interface-ref;
        must "/if:interfaces/if:interface[if:name = current()]"
            + "/if:type = 'ianaift:ethernetCsmacd'" {
            description
                "Ведомый (slave) интерфейс должен иметь тип
                'ethernetCsmacd'.";
        }
    }
}
// Другие параметры настройки связи, время восстановления и пр.
}
```

Хотя параметры уровней настраиваются в моделях для конкретных типов интерфейсов, два базовых leaf-list (higher-layer-if и lower-layer-if) представляют доступную только для чтения иерархию уровней интерфейсов.

## 4. Связь с IF-MIB

Если устройство реализует IF-MIB [RFC2863], каждая запись списка /interfaces/interface обычно отображается на один элемент ifEntry. Лист if-index **должен** содержать ifIndex соответствующего элемента ifEntry.

В большинстве случаев name из записи /interfaces/interface отображается на ifName. База IF-MIB позволяет двум разным ifEntry иметь общее имя ifName. Поддерживающие такую возможность устройства, которые соответствуют также определенной в этом документе модели данных, не могут иметь взаимно-однозначного отображения между листом name и ifName.

Указанное в конфигурации описание (description) интерфейса традиционно отображается некоторыми реализациями в ifAlias. Этот документ разрешает такие отображения, но рекомендует разработчикам учитывать различия в пространстве значений и постоянстве этих объектов. Подробности приведены в определении листа description представленного в разделе 5 модуля YANG.

IF-MIB определяет также открытый для записи объект ifPromiscuousMode. Поскольку агенты SNMP обычно не реализуют этот объект как конфигурационный, он не отображается в модуль ietf-interfaces.

Объект ifMtu из IF-MIB не отображается в модуль ietf-interfaces. Предполагается, что модули YANG для конкретных типов интерфейсов будут представлять MTU с помощью дополнения модели ietf-interfaces.

В IF-MIB имеется множество счетчиков, существующих в 32-битовом и 64-битовом варианте. 64-битовые счетчики были добавлены для поддержки интерфейсов со скоростью выше 20000000 бит/с. Современные реализации обычно поддерживают такие интерфейсы, поэтому модель данных включает лишь 64-битовые счетчики. Отметим, что NETCONF и SNMP могут различаться по частоте обращения к этим счетчикам. Например, многие реализации SNMP кэшируют значения счетчиков в течение некоторого времени.

Объекты ifDescr и ifConnectorPresent из IF-MIB не отображаются в модуль ietf-interfaces.

Ниже приведены таблицы сопоставления узлов данных YANG и объектов IF-MIB.

*Узлы YANG и соответствующие объекты IF-MIB.*

<i>Узел данных YANG в /interfaces/interface</i>	<i>Объект IF-MIB</i>
name	ifName
type	ifType
description	ifAlias
admin-status	ifAdminStatus
oper-status	ifOperStatus
last-change	ifLastChange
if-index	ifIndex
link-up-down-trap-enable	ifLinkUpDownTrapEnable
phys-address	ifPhysAddress
higher-layer-if and lower-layer-if	ifStackTable
speed	ifSpeed and ifHighSpeed
discontinuity-time	ifCounterDiscontinuityTime
in-octets	ifHCInOctets
in-unicast-pkts	ifHCInUcastPkts
in-broadcast-pkts	ifHCInBroadcastPkts
in-multicast-pkts	ifHCInMulticastPkts
in-discards	ifInDiscards
in-errors	ifInErrors
in-unknown-protos	ifInUnknownProtos
out-octets	ifHCOctets
out-unicast-pkts	ifHCOUcastPkts
out-broadcast-pkts	ifHCOBroadcastPkts
out-multicast-pkts	ifHCOMulticastPkts
out-discards	ifOutDiscards
out-errors	ifOutErrors

## 5. Модуль YANG

Этот модуль YANG импортирует определения типов (typedef) из [RFC6991].

```
<CODE BEGINS> file "ietf-interfaces@2018-02-20.yang"

module ietf-interfaces {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-interfaces";
  prefix if;

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF NETMOD (Network Modeling) Working Group";

  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    Editor: Martin Bjorklund
           <mailto:mbj@tail-f.com>";

  description
    "Этот модуль содержит набор определений YANG для управления
    сетевыми интерфейсами.

    Авторские права (Copyright (c) 2018) принадлежат IETF Trust
    и лицам, указанным как авторы кода. Все права защищены.

    Распространение и использование в исходной и двоичной форме
    с изменениями или без них разрешается в соответствии с условиями,
    указанными в упрощенной лицензии BSD, изложенной в разделе 4.с
    Правового положения IETF Trust применительно к документам IETF
    (http://trustee.ietf.org/license-info).
```

Эта версия модуля YANG является частью RFC 8343, где правовые аспекты выражены более полно.";

```
revision 2018-02-20 {
  description
    "Обновление для поддержки NMDA.";
  reference
    "RFC 8343: A YANG Data Model for Interface Management";
}

revision 2014-05-08 {
  description
    "Первый выпуск.";
  reference
    "RFC 7223: A YANG Data Model for Interface Management";
}

/*
 * Определения типов
 */

typedef interface-ref {
  type leafref {
    path "/if:interfaces/if:interface/if:name";
  }
  description
    "Этот тип применяется моделями данных, которым нужно
    указывать настроенные интерфейсы.";
}

/*
 * Отождествления
 */

identity interface-type {
  description
    "Базовое отождествление, из которого выводятся конкретные типы.";
}

/*
 * Возможности
 */

feature arbitrary-names {
  description
    "Это свойство показывает, что управляемые пользователем интерфейсы
    могут иметь произвольные имена.";
}

feature pre-provisioning {
  description
    "Это свойство показывает, что устройство поддерживает подготовленные
    заранее конфигурации интерфейсов, т. е. можно настроить интерфейс,
    которого еще нет в системе.";
}

feature if-mib {
  description
    "Это свойство указывает поддержку устройством IF-MIB.";
  reference
    "RFC 2863: The Interfaces Group MIB";
}

/*
 * Узлы данных
 */

container interfaces {
  description
    "Параметры интерфейса.";

  list interface {
    key "name";

    description
      "Список интерфейсов устройства.

      Статус интерфейса из этого списка доступен в рабочем состоянии.
```

Если конфигурация управляемого системой интерфейса не может использоваться системой (например, установленное оборудование не соответствует типу интерфейса), конфигурация не применяется к управляемому системой интерфейсу, показанному в операционном состоянии. Если конфигурация управляемого пользователем интерфейса не может применяться системой, настроенный интерфейс не включается в операционное состояние.

Системно-управляемые интерфейсы, созданные системой, всегда присутствуют в этом списке операционного состояния, даже если они не настроены.";

```
leaf name {
  type string;
  description
    "Имя интерфейса .

    Устройство МОЖЕТ ограничивать разрешенные для этого листа значения, возможно в зависимости от типа интерфейса. Для управляемых системой интерфейсов этот лист содержит зависимое от устройства имя интерфейса.

    Когда клиент пытается создать конфигурацию для управляемого системой интерфейса, которого нет в операционной состоянии, сервер МОЖЕТ отклонить запрос, если система не поддерживает предварительной настройки интерфейсов или имя указывает интерфейс, который не может присутствовать в системе. Сервер NETCONF ДОЛЖЕН вернуть отклик rpc-error с error-tag 'invalid-value' в таком случае.

    Если устройство поддерживает предварительную настройку интерфейсов, анонсируется свойство 'pre-provisioning'.

    Если устройство разрешает произвольные имена для управляемых пользователем интерфейсов, анонсируется свойство 'arbitrary-names'.

    Когда система создает управляемый пользователем интерфейс, он включается в рабочее состояние с тем же именем.

    Реализация сервера МОЖЕТ отображать этот лист на объект MIB ifName. Такая реализация должна использовать тот или иной механизм обработки различий в размере и разрешенных символах для этого листа и ifName. Определение таких механизмов выходит за рамки этого документа.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifName";
}

leaf description {
  type string;
  description
    "Текстовое описание интерфейса.

    Реализация сервера МОЖЕТ отображать этот лист на объект MIB ifAlias. Такие реализации должны использовать тот или иной механизм для обработки различий в размере и разрешенных символах для этого листа и ifAlias. Определение таких механизмов выходит за рамки документа.

    Поскольку ifAlias определяется для сохранения в энергонезависимой памяти, реализация MIB ДОЛЖНА отображать ifAlias на значение 'description' в постоянном хранилище.
  reference
    "RFC 2863: The Interfaces Group MIB - ifAlias";
}

leaf type {
  type identityref {
    base interface-type;
  }
  mandatory true;
  description
    "Тип интерфейса.
```

При создании записи для интерфейса сервер **МОЖЕТ** инициализировать лист `type` действующим значением, например, если можно вывести тип из имени интерфейса.

Если клиент пытается установить для типа интерфейса значение, которое никогда не используется системой (например, тип не поддерживается или не соответствует имени), сервер **ДОЛЖЕН** отвергнуть запрос. Сервер **NETCONF ДОЛЖЕН** вернуть отклик `rpc-error` с `error-tag 'invalid-value'` в таком случае.";

```
reference
  "RFC 2863: The Interfaces Group MIB - ifType";
}

leaf enabled {
  type boolean;
  default "true";
  description
    "Этот лист содержит настроенное, желаемое состояние интерфейса.

    Системы, реализующие IF-MIB, используют значение этого листа в
    хранилище running для установки в IF-MIB.ifAdminStatus
    значения up или down после инициализации ifEntry как описано в
    RFC 2863.

    Изменение этого листа в хранилище running отражается в
    ifAdminStatus, но при изменении ifAdminStatus с помощью
    SNMP этот лист не меняется.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifAdminStatus";
}

leaf link-up-down-trap-enable {
  if-feature if-mib;
  type enumeration {
    enum enabled {
      value 1;
      description
        "Указывает следует ли генерировать уведомления SNMP
        linkUp/linkDown для этого интерфейса.";
    }
    enum disabled {
      value 2;
      description
        "Указывает следует ли генерировать уведомления SNMP
        linkUp/linkDown для этого интерфейса.";
    }
  }
  description
    "Указывает следует ли генерировать уведомления SNMP
    linkUp/linkDown для этого интерфейса.

    Если этот узел не настроен, значение enabled применяется
    сервером для интерфейсов, которые не работают поверх другого
    интерфейса (т. е. нет записей lower-layer-if), и disabled
    в остальных случаях.";
  reference
    "RFC 2863: The Interfaces Group MIB -
    ifLinkUpDownTrapEnable";
}

leaf admin-status {
  if-feature if-mib;
  type enumeration {
    enum up {
      value 1;
      description
        "Готов пропускать пакеты.";
    }
    enum down {
      value 2;
      description
        "Не готов пропускать пакеты и не находится в режиме теста.";
    }
    enum testing {
      value 3;
      description

```



```

        "Находится в режиме тестирования.";
    }
}
config false;
mandatory true;
description
    "Желаемое состояние интерфейса.

    Этот лист имеет такую же семантику, как ifAdminStatus.";
reference
    "RFC 2863: The Interfaces Group MIB - ifAdminStatus";
}

leaf oper-status {
    type enumeration {
        enum up {
            value 1;
            description
                "Готов пропускать пакеты.";
        }
        enum down {
            value 2;
            description
                "Интерфейс не пропускает никаких пакетов.";
        }
        enum testing {
            value 3;
            description
                "В режиме тестирования. Не может пропускать рабочих
                пакетов.";
        }
        enum unknown {
            value 4;
            description
                "Состояние не может быть определено по каким-то причинам.";
        }
        enum dormant {
            value 5;
            description
                "Ожидание внешнего события.";
        }
        enum not-present {
            value 6;
            description
                "Отсутствует тот или иной (обычно аппаратный) компонент.";
        }
        enum lower-layer-down {
            value 7;
            description
                "Не работает вследствие состояния нижележащего интерфейса.";
        }
    }
}
config false;
mandatory true;
description
    "Текущее операционное состояние интерфейса.

    Этот лист имеет такую же семантику, как ifOperStatus.";
reference
    "RFC 2863: The Interfaces Group MIB - ifOperStatus";
}

leaf last-change {
    type yang:date-and-time;
    config false;
    description
        "Время перехода интерфейса в текущее операционное состояние.
        Если текущее состояние началось до предыдущей реинициализации
        локальной подсистемы сетевого управления, этого узла не будет.";
    reference
        "RFC 2863: The Interfaces Group MIB - ifLastChange";
}

leaf if-index {
    if-feature if-mib;
    type int32 {

```

```
    range "1..2147483647";
  }
  config false;
  mandatory true;
  description
    "Значение ifIndex для ifEntry, представленной этим интерфейсом.";
  interface.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifIndex";
}

leaf phys-address {
  type yang:phys-address;
  config false;
  description
    "Адрес интерфейса на его протокольном подуровне. Например,
    для интерфейса 802.x этот объект обычно содержит MAC1-адрес.
    Зависящие от среды модули должны определять порядок битов
    и байтов, а также формат значения для этого объекта. Для
    интерфейсов, не имеющих такого адреса (например,
    последовательная линия), этого узла не будет.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifPhysAddress";
}

leaf-list higher-layer-if {
  type interface-ref;
  config false;
  description
    "Список ссылок на интерфейсы, работающие поверх этого
    интерфейса.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifStackTable";
}

leaf-list lower-layer-if {
  type interface-ref;
  config false;
  description
    "Список ссылок на интерфейсы под этим интерфейсом.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifStackTable";
}

leaf speed {
  type yang:gauge64;
  units "bits/second";
  config false;
  description
    "Оценка текущей пропускной способности интерфейса в бит/с.
    Для интерфейсов, не меняющих пропускной способности или
    не позволяющих точно оценить ее, в этом узле следует
    указывать номинальную пропускную способность. Для
    интерфейсов, не использующих концепцию пропускной
    способности, этот узел не присутствует.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifSpeed, ifHighSpeed";
}

container statistics {
  config false;
  description
    "Набор связанных с интерфейсом объектов статистики.";

  leaf discontinuity-time {
    type yang:date-and-time;
    mandatory true;
    description
      "Время последнего события, когда один или несколько
      счетчиков интерфейса подверглись разрыву. Если таких
      событий не было с момента последней реинициализации
      локальной подсистемы управления, указывается время с
      момента этой реинициализации.";
  }
}
```

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

```
leaf in-octets {
  type yang:counter64;
  description
    "Общее число октетов, принятых на интерфейсе с учетом
    символов кадрирования.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifHCInOctets";
}

leaf in-unicast-pkts {
  type yang:counter64;
  description
    "Число пакетов, доставленных этим подуровнем на
    вышележащий (под)уровень, которые не были направлены по
    широковещательному или групповому адресу этого подуровня.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifHCInUcastPkts";
}

leaf in-broadcast-pkts {
  type yang:counter64;
  description
    "Число пакетов, доставленных этим подуровнем на
    вышележащий (под)уровень, которые были направлены по
    широковещательному адресу этого подуровня.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB -
    ifHCInBroadcastPkts";
}

leaf in-multicast-pkts {
  type yang:counter64;
  description
    "Число пакетов, доставленных этим подуровнем на
    вышележащий (под)уровень, которые были направлены по
    групповому адресу этого подуровня. Для протокола
    уровня MAC учитываются групповые (Group) и
    функциональные (Functional) адреса.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB -
    ifHCInMulticastPkts";
}

leaf in-discards {
  type yang:counter32;
  description
    "Число входящих пакетов, которые несмотря на отсутствие
    ошибок были выбраны для отбрасывания с целью предотвратить
    их доставку протоколу вышележащего уровня. Одной из
    возможных причин отбрасывания является освобождение буферов.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifInDiscards";
}

leaf in-errors {
  type yang:counter32;
```

```
description
  "Для ориентированных на пакеты интерфейсов - число входящих
  пакетов с ошибками, препятствующими передаче протоколу
  вышележащего уровня. Для ориентированных на символы
  интерфейсов и интерфейсов с фиксированным размером блоков -
  число входящих блоков передачи с ошибками, препятствующими
  доставке протоколу вышележащего уровня.

  Разрыв значения этого счетчика может происходить при
  реинициализации системы управления или иных событиях
  указанных временем 'discontinuity-time'. ";
reference
  "RFC 2863: The Interfaces Group MIB - ifInErrors";
}

leaf in-unknown-protos {
  type yang:counter32;
  description
    "Для ориентированных на пакеты интерфейсов - число принятых
    через интерфейс пакетов, которые были отброшены по причине
    неизвестного или не поддерживаемого протокола. Для символично-
    ориентированных интерфейсов и интерфейсов с фиксированным
    размером блока и поддержкой мультиплексирования протоколов -
    число принятых через интерфейс блоков передачи, которые были
    отброшены по причине неизвестного или не поддерживаемого
    протокола. Для интерфейсов, не поддерживающих
    мультиплексирование протоколов, этот счетчик не присутствует.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
  reference
    "RFC 2863: The Interfaces Group MIB - ifInUnknownProtos";
}

leaf out-octets {
  type yang:counter64;
  description
    "Общее число переданных интерфейсом октетов, включая символы
    кадрирования.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
  reference
    "RFC 2863: The Interfaces Group MIB - ifHCOutOctets";
}

leaf out-unicast-pkts {
  type yang:counter64;
  description
    "Общее число пакетов, для которых протокол вышележащего
    уровня запросил передачу по адресу, не являющемуся
    широковещательным или групповым адресам для этого
    подуровня, включая отброшенные и не переданные пакеты.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
  reference
    "RFC 2863: The Interfaces Group MIB - ifHCOutUcastPkts";
}

leaf out-broadcast-pkts {
  type yang:counter64;
  description
    "Общее число пакетов, для которых протокол вышележащего
    уровня запросил передачу, направленных по
    широковещательным адресам для этого подуровня,
    включая отброшенные и не переданные пакеты.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
```

```

reference
  "RFC 2863: The Interfaces Group MIB -
    ifHCOutBroadcastPkts";
}

leaf out-multicast-pkts {
  type yang:counter64;
  description
    "Общее число пакетов, для которых протокол вышележащего
    уровня запросил передачу, направленных по
    групповым адресам для этого подуровня, включая
    отброшенные и не переданные пакеты. Для протокола уровня
    MAC это включает групповые и функциональные адреса.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB -
      ifHCOutMulticastPkts";
}

leaf out-discards {
  type yang:counter32;
  description
    "Число исходящих пакетов, которые были выбраны для
    отбрасывания, несмотря на отсутствие препятствующих
    передаче ошибок. Возможной причиной такого отбрасывания
    является освобождение буферного пространства.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifOutDiscards";
}

leaf out-errors {
  "Для пакетно-ориентированных интерфейсов - число исходящих
  пакетов, которые не были переданы по причине ошибок. Для
  символьных интерфейсов и интерфейсов с постоянным размером
  блока - число блоков, не переданных из-за ошибок.

  Разрыв значения этого счетчика может происходить при
  реинициализации системы управления или иных событиях
  указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifOutErrors";
}
}
}

/*
 * Устаревшие определения типов
 */
typedef interface-state-ref {
  type leafref {
    path "/if:interfaces-state/if:interface/if:name";
  }
  status deprecated;
  description
    "Этот тип применяется моделями данных, которым нужно
    указывать работающие интерфейсы.";
}

/*
 * Устаревшие узлы данных состояния
 */
container interfaces-state {
  config false;
  status deprecated;
  description
    "Узлы данных для операционных состояний интерфейсов.";
}

```

```
list interface {
  key "name";
  status deprecated;

  description
    "Список интерфейсов в устройстве.

    Системно-управляемые интерфейсы, созданные системой, всегда
    присутствуют в этом списке, независимо от их конфигурации.";

  leaf name {
    type string;
    status deprecated;
    description
      "Имя интерфейса.

      Реализация сервера МОЖЕТ отображать этот лист на объект
      MIB ifName. Такие реализации должны использовать тот или
      иной механизм для обработки различий в размере и разрешенных
      символах для этого листа и ifName. Определение механизмов
      выходит за рамки этого документа.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifName";
  }

  leaf type {
    type identityref {
      base interface-type;
    }
    mandatory true;
    status deprecated;
    description
      "Тип интерфейса.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifType";
  }

  leaf admin-status {
    if-feature if-mib;
    type enumeration {
      enum up {
        value 1;
        description
          "Готов пропускать пакеты.";
      }
      enum down {
        value 2;
        description
          "Не готов пропускать пакеты и не находится в режиме теста.";
      }
      enum testing {
        value 3;
        description
          "Находится в режиме тестирования.";
      }
    }
    mandatory true;
    status deprecated;
    description
      "Желаемое состояние интерфейса.

      Этот лист имеет такую же семантику, как ifAdminStatus.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifAdminStatus";
  }

  leaf oper-status {
    type enumeration {
      enum up {
        value 1;
        description
          "Готов пропускать пакеты.";
      }
      enum down {
        value 2;
        description
```

```

    "Интерфейс не пропускает никаких пакетов.";
  }
  enum testing {
    value 3;
    description
      "В режиме тестирования. Не может пропускать рабочих
        пакетов.";
  }
  enum unknown {
    value 4;
    description
      "Состояние не может быть определено по каким-то причинам.";
  }
  enum dormant {
    value 5;
    description
      "Ожидание внешнего события.";
  }
  enum not-present {
    value 6;
    description
      "Отсутствует тот или иной (обычно аппаратный) компонент.";
  }
  enum lower-layer-down {
    value 7;
    description
      "Не работает вследствие состояния нижележащего интерфейса.";
  }
}
mandatory true;
status deprecated;
description
  "Текущее операционное состояние интерфейса.

  Этот лист имеет такую же семантику, как ifOperStatus.";
reference
  "RFC 2863: The Interfaces Group MIB - ifOperStatus";
}

leaf last-change {
  type yang:date-and-time;
  status deprecated;
  description
    "Время перехода интерфейса в текущее операционное состояние.
     Если текущее состояние началось до предыдущей реинициализации
     локальной подсистемы сетевого управления, этого узла не будет.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifLastChange";
}

leaf if-index {
  if-feature if-mib;
  type int32 {
    range "1..2147483647";
  }
  mandatory true;
  status deprecated;
  description
    "Значение ifIndex для ifEntry, представленной этим интерфейсом.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifIndex";
}

leaf phys-address {
  type yang:phys-address;
  status deprecated;
  description
    "Адрес интерфейса на его протокольном подуровне. Например,
     для интерфейса 802.x этот объект обычно содержит MAC-адрес.
     Зависящие от среды модули должны определять порядок битов
     и байтов, а также формат значения для этого объекта. Для
     интерфейсов, не имеющих такого адреса (например,
     последовательная линия), этого узла не будет.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifPhysAddress";
}

```

```
leaf-list higher-layer-if {
  type interface-state-ref;
  status deprecated;
  description
    "Список ссылок на интерфейсы, работающие поверх этого
    интерфейса.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifStackTable";
}

leaf-list lower-layer-if {
  type interface-state-ref;
  status deprecated;
  description
    "Список ссылок на интерфейсы под этим интерфейсом.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifStackTable";
}

leaf speed {
  type yang:gauge64;
  units "bits/second";
  status deprecated;
  description
    "Оценка текущей пропускной способности интерфейса в бит/с.
    Для интерфейсов, не меняющих пропускной способности или
    не позволяющих точно оценить ее, в этом узле следует
    указывать номинальную пропускную способность. Для
    интерфейсов, не использующих концепцию пропускной
    способности, этот узел не присутствует.";
  reference
    "RFC 2863: The Interfaces Group MIB -
    ifSpeed, ifHighSpeed";
}

container statistics {
  status deprecated;
  description
    "Набор связанных с интерфейсом объектов статистики.";

  leaf discontinuity-time {
    type yang:date-and-time;
    mandatory true;
    status deprecated;
    description
      "Время последнего события, когда один или несколько
      счетчиков интерфейса подверглись разрыву. Если таких
      событий не было с момента последней реинициализации
      локальной подсистемы управления, указывается время с
      момента этой реинициализации.";
  }

  leaf in-octets {
    type yang:counter64;
    status deprecated;
    description
      "Общее число октетов, принятых на интерфейсе с учетом
      символов кадрирования.

      Разрыв значения этого счетчика может происходить при
      реинициализации системы управления или иных событиях
      указанных временем 'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifHCInOctets";
  }

  leaf in-unicast-pkts {
    type yang:counter64;
    status deprecated;
    description
      "Число пакетов, доставленных этим подуровнем на
      вышележащий (под)уровень, которые не были направлены по
      широковещательному или групповому адресу этого подуровня.

      Разрыв значения этого счетчика может происходить при
```



```

    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
    reference
    "RFC 2863: The Interfaces Group MIB - ifHCInUcastPkts";
}

leaf in-broadcast-pkts {
    type yang:counter64;
    status deprecated;
    description
    "Число пакетов, доставленных этим подуровнем на
    вышележащий (под)уровень, которые были направлены по
    широковещательному адресу этого подуровня.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
    reference
    "RFC 2863: The Interfaces Group MIB -
    ifHCInBroadcastPkts";
}

leaf in-multicast-pkts {
    type yang:counter64;
    status deprecated;
    description
    "Число пакетов, доставленных этим подуровнем на
    вышележащий (под)уровень, которые были направлены по
    групповому адресу этого подуровня. Для протокола
    уровня MAC учитываются групповые (Group) и
    функциональные (Functional) адреса.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
    reference
    "RFC 2863: The Interfaces Group MIB -
    ifHCInMulticastPkts";
}

leaf in-discards {
    type yang:counter32;
    status deprecated;
    description
    "Число входящих пакетов, которые несмотря на отсутствие
    ошибок были выбраны для отбрасывания с целью предотвратить
    их доставку протоколу вышележащего уровня. Одной из
    возможных причин отбрасывания является освобождение буферов.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
    reference
    "RFC 2863: The Interfaces Group MIB - ifInDiscards";
}

leaf in-errors {
    type yang:counter32;
    status deprecated;
    description
    "Для ориентированных на пакеты интерфейсов - число входящих
    пакетов с ошибками, препятствующими передаче протоколу
    вышележащего уровня. Для ориентированных на символы
    интерфейсов и интерфейсов с фиксированным размером блоков -
    число входящих блоков передачи с ошибками, препятствующими
    доставке протоколу вышележащего уровня.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'. ";
    reference
    "RFC 2863: The Interfaces Group MIB - ifInErrors";
}

leaf in-unknown-protos {
    type yang:counter32;

```

```
status deprecated;
description
  "Для ориентированных на пакеты интерфейсов - число принятых
  через интерфейс пакетов, которые были отброшены по причине
  неизвестного или не поддерживаемого протокола. Для символично-
  ориентированных интерфейсов и интерфейсов с фиксированным
  размером блока и поддержкой мультиплексирования протоколов -
  число принятых через интерфейс блоков передачи, которые были
  отброшены по причине неизвестного или не поддерживаемого
  протокола. Для интерфейсов, не поддерживающих
  мультиплексирование протоколов, этот счетчик не присутствует.

  Разрыв значения этого счетчика может происходить при
  реинициализации системы управления или иных событиях
  указанных временем 'discontinuity-time'.";
reference
  "RFC 2863: The Interfaces Group MIB - ifInUnknownProtos";
}

leaf out-octets {
  type yang:counter64;
  status deprecated;
  description
    "Общее число переданных интерфейсом октетов, включая символы
    кадрирования.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifHCOutOctets";
}

leaf out-unicast-pkts {
  type yang:counter64;
  status deprecated;
  description
    "Общее число пакетов, для которых протокол вышележащего
    уровня запросил передачу по адресу, не являющемуся
    широковещательным или групповым адресам для этого
    подуровня, включая отброшенные и не переданные пакеты.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifHCOutUcastPkts";
}

leaf out-broadcast-pkts {
  type yang:counter64;
  status deprecated;
  description
    "Общее число пакетов, для которых протокол вышележащего
    уровня запросил передачу, направленных по
    широковещательным адресам для этого подуровня,
    включая отброшенные и не переданные пакеты.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB -
      ifHCOutBroadcastPkts";
}

leaf out-multicast-pkts {
  type yang:counter64;
  status deprecated;
  description
    "Общее число пакетов, для которых протокол вышележащего
    уровня запросил передачу, направленных по
    групповым адресам для этого подуровня, включая
    отброшенные и не переданные пакеты. Для протокола уровня
    MAC это включает групповые и функциональные адреса.
```

```

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
reference
  "RFC 2863: The Interfaces Group MIB -
    ifHCOutMulticastPkts";
}

leaf out-discards {
  type yang:counter32;
  status deprecated;
  description
    "Число исходящих пакетов, которые были выбраны для
    отбрасывания, несмотря на отсутствие препятствующих
    передаче ошибок. Возможной причиной такого отбрасывания
    является освобождение буферного пространства.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifOutDiscards";
}

leaf out-errors {
  type yang:counter32;
  status deprecated;
  description
    "Для пакетно-ориентированных интерфейсов - число исходящих
    пакетов, которые не были переданы по причине ошибок. Для
    символьных интерфейсов и интерфейсов с постоянным размером
    блока - число блоков, не переданных из-за ошибок.

    Разрыв значения этого счетчика может происходить при
    реинициализации системы управления или иных событиях
    указанных временем 'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifOutErrors";
}
}
}
}
}
}
}
}

<CODE ENDS>

```

## 6. Взаимодействие с IANA

Этот документ регистрирует URI в реестре «IETF XML Registry» [RFC3688]. В соответствии с форматом RFC 3688 выполнена приведенная ниже регистрация.

```

URI: urn:ietf:params:xml:ns:yang:ietf-interfaces
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-interfaces
namespace: urn:ietf:params:xml:ns:yang:ietf-interfaces
prefix: if
reference: RFC 8343

```

## 7. Вопросы безопасности

Описанный в этом документе модуль YANG определяет схему данных, которая предназначена для доступа по протоколам сетевого управления, таким как NETCONF [RFC6241] или RESTCONF [RFC8040]. Нижним уровнем NETCONF является защищенный транспорт с обязательной поддержкой SSH<sup>1</sup> [RFC6242]. Нижним уровнем RESTCONF является HTTPS с обязательной поддержкой TLS [RFC5246].

Модель управления доступом NETCONF [RFC8341] обеспечивает способы ограничения доступа отдельным пользователям NETCONF или RESTCONF заданным подмножеством всех доступных операций и содержимого NETCONF или RESTCONF.

В модуле YANG имеется множество узлов данных, доступных для чтения, создания и удаления (например, с принятым по умолчанию `config true`). Эти узлы могут быть конфиденциальными или уязвимыми в некоторых сетевых средах. Операции записи (например, `<edit-config>`) в эти узлы без подобающей защиты могут оказывать негативное влияние на работу сети. Ниже перечислены субдеревья и узлы данных с указанием уязвимости.

<sup>1</sup>Secure Shell — защищенная оболочка.

**/interfaces/interface**

Этот список указывает настроенные в устройстве интерфейсы. Несанкционированный доступ к списку может вынуждать устройство игнорировать пакеты, которые оно должно принимать и обрабатывать.

**/interfaces/interface/enabled**

Этот лист управляет включением и выключением интерфейса. Несанкционированный доступ к списку может вынуждать устройство игнорировать пакеты, которые оно должно принимать и обрабатывать.

## 8. Литература

### 8.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<https://www.rfc-editor.org/info/rfc2863>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), 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>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [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>>.

### 8.2. Дополнительная литература

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

## Приложение А. Пример модуля для интерфейса Ethernet

В этом приложении приведен простой пример определения модуля для интерфейса Ethernet, демонстрирующий условное добавление зависимых от среды конфигурационных параметров в список базовых интерфейсов. Показано также условное добавление параметров рабочего состояния. Пример не является законченным модулем для Ethernet.

```
module example-ethernet {
  namespace "http://example.com/ethernet";
  prefix "eth";

  import ietf-interfaces {
    prefix if;
  }
  import iana-if-type {
    prefix ianaift;
  }

  // Конфигурационные параметры для интерфейсов Ethernet
  augment "/if:interfaces/if:interface" {
    when "if:type = 'ianaift:ethernetCsmacd'";

    container ethernet {
      container transmission {
        choice transmission-params {
          case auto {
            leaf auto-negotiate {
```

```

        type empty;
    }
}
case manual {
    container manual {
        leaf duplex {
            type enumeration {
                enum "half";
                enum "full";
            }
        }
        leaf speed {
            type enumeration {
                enum "10Mb";
                enum "100Mb";
                enum "1Gb";
                enum "10Gb";
            }
        }
    }
}
}
leaf duplex {
    type enumeration {
        enum "half";
        enum "full";
    }
    config false;
}
}
// Другие параметры, относящиеся к Ethernet ...
}
}
}

```

### Приложение В. Пример модуля для связки интерфейсов Ethernet

В этом приложении представлен пример определения уровней интерфейсов. Определен интерфейс связки Ethernet, объединяющей несколько интерфейсов Ethernet в один логический интерфейс.

```

module example-ethernet-bonding {
    namespace "http://example.com/ethernet-bonding";
    prefix "bond";

    import ietf-interfaces {
        prefix if;
    }
    import iana-if-type {
        prefix ianaift;
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ieee8023adLag'";

        leaf-list slave-if {
            type if:interface-ref;
            must "/if:interfaces/if:interface[if:name = current()]"
                + "/if:type = 'ianaift:ethernetCsmacd'" {
                description
                    "Ведомый (slave) интерфейс должен иметь тип 'ethernetCsmacd'.";
            }
        }
        leaf bonding-mode {
            type enumeration {
                enum round-robin;
                enum active-backup;
                enum broadcast;
            }
        }
        // Другие параметры конфигурации связки, время восстановления и пр.
    }
}

```

### Приложение С. Пример модуля для интерфейса VLAN

В этом приложении представлен пример определения интерфейса VLAN.

```

module example-vlan {
    namespace "http://example.com/vlan";

```

```

prefix "vlan";

import ietf-interfaces {
  prefix if;
}
import iana-if-type {
  prefix ianaift;
}

augment "/if:interfaces/if:interface" {
  when "if:type = 'ianaift:ethernetCsmacd' or
        if:type = 'ianaift:ieee8023adLag'";
  leaf vlan-tagging {
    type boolean;
    default false;
  }
}

augment "/if:interfaces/if:interface" {
  when "if:type = 'ianaift:l2vlan'";

  leaf base-interface {
    type if:interface-ref;
    must "/if:interfaces/if:interface[if:name = current()]"
      + "/vlan:vlan-tagging = 'true'" {
      description
        "На базовом интерфейсе должны быть разрешены теги VLAN.";
    }
  }
  leaf vlan-id {
    type uint16 {
      range "1..4094";
    }
    must "../base-interface" {
      description
        "При определении vlan-id должен быть указан базовый интерфейс.";
    }
  }
}
}
}

```

## Приложение D. Пример отклика NETCONF <get-config>

В этом приложении дан пример отклика на запрос NETCONF <get-config> для хранилища рабочей конфигурации на устройстве, реализующем приведенный выше пример модели данных.

```

<rpc-reply
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  message-id="101">
  <data>
    <interfaces
      xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
      xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type"
      xmlns:vlan="http://example.com/vlan">

      <interface>
        <name>eth0</name>
        <type>ianaift:ethernetCsmacd</type>
        <enabled>>false</enabled>
      </interface>

      <interface>
        <name>eth1</name>
        <type>ianaift:ethernetCsmacd</type>
        <enabled>>true</enabled>
        <vlan:vlan-tagging>true</vlan:vlan-tagging>
      </interface>

      <interface>
        <name>eth1.10</name>
        <type>ianaift:l2vlan</type>
        <enabled>>true</enabled>
        <vlan:base-interface>eth1</vlan:base-interface>
        <vlan:vlan-id>10</vlan:vlan-id>
      </interface>

      <interface>
        <name>lo1</name>

```

```

    <type>ianaift:softwareLoopback</type>
    <enabled>true</enabled>
  </interface>

</interfaces>
</data>
</rpc-reply>

```

## Приложение E. Пример отклика NETCONF <get-data>

В этом приложении дан пример отклика на запрос NETCONF <get-data> для хранилища операционного состояния на устройстве, реализующем приведенный выше пример модели данных.

В этом примере используется аннотация origin, определенная в модуле «ietf-origin» [RFC8342].

```

<rpc-reply
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  message-id="101">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-datastores">
    <interfaces
      xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
      xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type"
      xmlns:vlan="http://example.com/vlan"
      xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin">

      <interface or:origin="or:intended">
        <name>eth0</name>
        <type>ianaift:ethernetCsmacd</type>
        <enabled>false</enabled>
        <admin-status>down</admin-status>
        <oper-status>down</oper-status>
        <if-index>2</if-index>
        <phys-address>00:01:02:03:04:05</phys-address>
        <statistics>
          <discontinuity-time>
            2013-04-01T03:00:00+00:00
          </discontinuity-time>
          <!-- Здесь показаны счетчики -->
        </statistics>
      </interface>

      <interface or:origin="or:intended">
        <name>eth1</name>
        <type>ianaift:ethernetCsmacd</type>
        <enabled>true</enabled>
        <admin-status>up</admin-status>
        <oper-status>up</oper-status>
        <if-index>7</if-index>
        <phys-address>00:01:02:03:04:06</phys-address>
        <higher-layer-if>eth1.10</higher-layer-if>
        <statistics>
          <discontinuity-time>
            2013-04-01T03:00:00+00:00
          </discontinuity-time>
          <!-- Здесь показаны счетчики -->
        </statistics>
        <vlan:vlan-tagging>true</vlan:vlan-tagging>
      </interface>

      <interface or:origin="or:intended">
        <name>eth1.10</name>
        <type>ianaift:l2vlan</type>
        <enabled>true</enabled>
        <admin-status>up</admin-status>
        <oper-status>up</oper-status>
        <if-index>9</if-index>
        <lower-layer-if>eth1</lower-layer-if>
        <statistics>
          <discontinuity-time>
            2013-04-01T03:00:00+00:00
          </discontinuity-time>
          <!-- Здесь показаны счетчики -->
        </statistics>
        <vlan:base-interface>eth1</vlan:base-interface>
        <vlan:vlan-id>10</vlan:vlan-id>
      </interface>
    </interfaces>
  </data>
</rpc-reply>

```

```

<!-- Этот интерфейс не настроен -->
<interface or:origin="or:system">
  <name>eth2</name>
  <type>ianaift:ethernetCsmacd</type>
  <admin-status>down</admin-status>
  <oper-status>down</oper-status>
  <if-index>8</if-index>
  <phys-address>00:01:02:03:04:07</phys-address>
  <statistics>
    <discontinuity-time>
      2013-04-01T03:00:00+00:00
    </discontinuity-time>
    <!-- Здесь показаны счетчики -->
  </statistics>
</interface>

<interface or:origin="or:intended">
  <name>lol</name>
  <type>ianaift:softwareLoopback</type>
  <enabled>true</enabled>
  <admin-status>up</admin-status>
  <oper-status>up</oper-status>
  <if-index>1</if-index>
  <statistics>
    <discontinuity-time>
      2013-04-01T03:00:00+00:00
    </discontinuity-time>
    <!-- Здесь показаны счетчики -->
  </statistics>
</interface>

</interfaces>
</data>
</rpc-reply>

```

## Приложение F. Примеры схем именованя интерфейсов

В этом приложении приведены примеры стратегии развертывания.

В примерах используется модель данных ex-vlan (Приложение C) для показа настройки управляемых пользователем интерфейсов.

### F.1. Маршрутизатор с ограничениями для имен интерфейсов

В этом примере маршрутизатор имеет 4 линейных платы с восемью портами на каждой. Гнезда для плат имеют физические номера от 0 до 3, а порты на каждой плате - от 0 до 7. Каждая плата имеет порты Fast Ethernet или Gigabit Ethernet.

Обусловленными устройством именами физических интерфейсов будут fastethernet-N/M или gigabitethernet-N/M.

Имена интерфейсов VLAN ограничены формой <physical-interface-name>.<subinterface-number>.

Предполагается, что схема именованя известна оператору. Реализация автоматически инициализирует значение type в соответствии с именем интерфейса.

Сервер NETCONF не анонсирует возможность arbitrary-names в сообщении <hello>.

Оператор может настроить физический интерфейс, передавая команду <edit-config>, содержащую

```

<interface nc:operation="create">
  <name>fastethernet-1/0</name>
</interface>

```

При получении сервером такого запроса он будет устанавливать для листа type значение ianaift:ethernetCsmacd. Если клиент передаст команду <get-config> после приведенной выше команды <edit-config>, он получит

```

<interface>
  <name>fastethernet-1/0</name>
  <type>ianaift:ethernetCsmacd</type>
</interface>

```

Клиент может настроить интерфейс VLAN с помощью команды <edit-config>, содержащей

```

<interface nc:operation="create">
  <name>fastethernet-1/0.10005</name>
  <type>ianaift:l2vlan</type>
  <vlan:base-interface>fastethernet-1/0</vlan:base-interface>
  <vlan:vlan-id>5</vlan:vlan-id>
</interface>

```

Если клиент попытается изменить тип физического интерфейса командой <edit-config>, содержащей

```

<interface nc:operation="merge">
  <name>fastethernet-1/0</name>
  <type>ianaift:tunnel</type>

```



```
</interface>
```

сервер вернет ошибку `invalid-value`, поскольку новый тип не соответствует имени.

## F.2. Маршрутизатор с произвольными именами интерфейсов

В этом примере маршрутизатор имеет 4 линейных платы с восемью портами на каждой. Гнезда для плат имеют физические номера от 0 до 3, а порты на каждой плате - от 0 до 7. Каждая плата имеет порты Fast Ethernet или Gigabit Ethernet.

Обусловленными устройством именами физических интерфейсов будут `fastethernet-N/M` или `gigabithernet-N/M`.

Реализация не ограничивает управляемые пользователем имена. Это позволяет оператору более гибко применять конфигурации к разным интерфейсам. Однако это несколько усложняет отображение имен интерфейсов, найденных в других протоколах, на записи в конфигурации.

Сервер NETCONF анонсирует возможность `arbitrary-names` в сообщении `<hello>`.

Физические интерфейсы настраиваются в соответствии с приложением F.1.

Оператор может настроить интерфейс VLAN с помощью команды `<edit-config>`, содержащей

```
<interface nc:operation="create">
  <name>acme-interface</name>
  <type>ianaift:l2vlan</type>
  <vlan:base-interface>fastethernet-1/0</vlan:base-interface>
  <vlan:vlan-id>5</vlan:vlan-id>
</interface>
```

При необходимости оператор может перенести конфигурацию с именем `acme-interface` на другой физический интерфейс с помощью команды `<edit-config>`, содержащей

```
<interface nc:operation="merge">
  <name>acme-interface</name>
  <vlan:base-interface>fastethernet-1/1</vlan:base-interface>
</interface>
```

## F.3. Коммутатор Ethernet с ограничениями для имен интерфейсов

В этом примере коммутатор Ethernet имеет множество портов, каждый из которых указывается простым номером.

Зависящие от устройства имена физических интерфейсов являются номерами, соответствующими номерам физических портов.

Оператор может настроить физический интерфейс с помощью команды `<edit-config>`, содержащей

```
<interface nc:operation="create">
  <name>6</name>
</interface>
```

Когда сервер получает такой запрос, он устанавливает для листа `type` значение `ianaift:ethernetCsmacd`. Если клиент выполнит команду `<get-config>` после показанной выше команды `<edit-config>`, он получит

```
<interface>
  <name>6</name>
  <type>ianaift:ethernetCsmacd</type>
</interface>
```

## F.4. Типовой хост с ограничениями для имен интерфейсов

В этом примере обычный хост имеет интерфейсы, именуемые ядром. Система идентифицирует физические интерфейсы по именам, назначенным операционной системой.

Имена интерфейсов VLAN ограничены формой `<physical-interface-name>:<vlan-number>`.

Сервер NETCONF не анонсирует возможность `arbitrary-names` в сообщении `<hello>`.

Оператор может настроить интерфейс с помощью команды `<edit-config>`, содержащей

```
<interface nc:operation="create">
  <name>eth8</name>
</interface>
```

Когда сервер получает такой запрос, он устанавливает для листа `type` значение `ianaift:ethernetCsmacd`. Если клиент выполнит команду `<get-config>` после показанной выше команды `<edit-config>`, он получит

```
<interface>
  <name>eth8</name>
  <type>ianaift:ethernetCsmacd</type>
</interface>
```

Клиент может настроить интерфейс VLAN с помощью команды `<edit-config>`, содержащей

```
<interface nc:operation="create">
  <name>eth8:5</name>
  <type>ianaift:l2vlan</type>
  <vlan:base-interface>eth8</vlan:base-interface>
  <vlan:vlan-id>5</vlan:vlan-id>
</interface>
```

## F.5. Типовой хост с произвольными именами интерфейсов

В этом примере обычный хост имеет интерфейсы, именуемые ядром. Система идентифицирует физические интерфейсы по именам, назначенным операционной системой.

Реализация не ограничивает управляемые пользователем имена. Это позволяет оператору более гибко применять конфигурации к разным интерфейсам. Однако это несколько усложняет отображение имен интерфейсов, найденных в других протоколах, на записи в конфигурации.

Сервер NETCONF анонсирует возможность `arbitrary-names` в сообщении `<hello>`.

Физические интерфейсы настраиваются как в приложении F.4.

Оператор может настроить интерфейс VLAN с помощью команды `<edit-config>`, содержащей

```
<interface nc:operation="create">
  <name>acme-interface</name>
  <type>ianaift:12vlan</type>
  <vlan:base-interface>eth8</vlan:base-interface>
  <vlan:vlan-id>5</vlan:vlan-id>
</interface>
```

При необходимости оператор может перенести конфигурацию с именем `acme-interface` на другой физический интерфейс с помощью команды `<edit-config>`, содержащей

```
<interface nc:operation="merge">
  <name>acme-interface</name>
  <vlan:base-interface>eth3</vlan:base-interface>
</interface>
```

## Благодарности

Автор благодарен Alexander Clemm, Per Hedeland, Ladislav Lhotka и Juergen Schoenwaelder за ценные замечания.

## Адрес автора

Martin Bjorklund

Tail-f Systems

Email: [mbj@tail-f.com](mailto:mbj@tail-f.com)

Перевод на русский язык

Николай Малых

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