

Internet Engineering Task Force (IETF)  
Request for Comments: 8639  
Category: Standards Track  
ISSN: 2070-1721

E. Voit  
Cisco Systems  
A. Clemm  
Futurewei  
A. Gonzalez Prieto  
Microsoft  
E. Nilsen-Nygaard  
A. Tripathy  
Cisco Systems  
September 2019

## Subscription to YANG Notifications

Подписка на уведомления YANG

### Аннотация

Этот документ определяет модель данных YANG и связанные механизмы для подписки получателей на поток событий от издателя. Применение этих методов позволяет подписчику запрашивать и получать продолжающийся настраиваемый поток создаваемой издателем информации.

### Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF<sup>1</sup> и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG<sup>2</sup>. Дополнительную информацию о стандартах Internet можно найти в разделе 2 RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <https://www.rfc-editor.org/info/rfc8639>.

### Авторские права

Авторские права (Copyright (c) 2019) 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).

Документ может содержать материалы из IETF Document или IETF Contribution, опубликованных или публично доступных до 10 ноября 2008 года. Лица, контролирующие авторские права на некоторые из таких документов, могли не предоставить IETF Trust права разрешать внесение изменений в такие документы за рамками процессов IETF Standards. Без получения соответствующего разрешения от лиц, контролирующих авторские права этот документ не может быть изменён вне рамок процесса IETF Standards, не могут также создаваться производные документы за рамками процесса IETF Standards за исключением форматирования документа для публикации или перевода с английского языка на другие языки.

## Оглавление

1. Введение.....	2
1.1. Мотивация.....	2
1.2. Терминология.....	2
1.3. Обзор решения.....	3
1.4. Отличия от RFC 5277.....	3
2. Решение.....	4
2.1. Потоки событий.....	4
2.2. Фильтры потоков событий.....	4
2.3. QoS.....	4
2.4. Динамическая подписка.....	5
2.4.1. Конечный автомат динамической подписки.....	5
2.4.2. Организация динамической подписки.....	5
2.4.2.1. Запрос воспроизведения записей о событиях.....	6
2.4.3. Изменение динамической подписки.....	6
2.4.4. Удаление динамической подписки.....	7
2.4.5. Уничтожение динамической подписки.....	7
2.4.6. Отказы RPC.....	7
2.5. Настраиваемая подписка.....	8
2.5.1. Конечный автомат настраиваемой подписки.....	8

<sup>1</sup>Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

<sup>2</sup>Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

2.5.2. Организация настраиваемой подписки.....	9
2.5.3. Изменение настраиваемой подписки.....	10
2.5.4. Удаление настраиваемой подписки.....	10
2.5.5. Сброс получателя настраиваемой подписки.....	10
2.5.6. Воспроизведение для настраиваемой подписки.....	10
2.5.7. Транспортная связность для настраиваемой подписки.....	10
2.6. Доставка записей о событиях.....	11
2.7. Уведомления об изменении состояния подписки.....	11
2.7.1. subscription-started.....	11
2.7.2. subscription-modified.....	12
2.7.3. subscription-terminated.....	12
2.7.4. subscription-suspended.....	13
2.7.5. subscription-resumed.....	13
2.7.6. subscription-completed.....	13
2.7.7. replay-completed.....	13
2.8. Мониторинг подписки.....	14
2.9. Поддержка модуля YANG ietf-subscribed-notifications.....	14
3. Диаграммы деревьев модели YANG.....	14
3.1. Контейнер streams.....	14
3.2. Контейнер filters.....	14
3.3. Контейнер subscriptions.....	15
4. Модуль YANG для подписки на уведомления о событиях.....	15
5. Взаимодействие с IANA.....	30
6. Вопросы реализации.....	30
7. Транспортные требования.....	30
8. Вопросы безопасности.....	30
9. Литература.....	32
9.1. Нормативные документы.....	32
9.2. Дополнительная литература.....	33
Приложение А. Пример дополнения настроенного транспорта.....	33
Благодарности.....	34
Адреса авторов.....	34

## 1. Введение

Этот документ задаёт модель данных YANG и связанные с ней механизмы подписки на уведомления о потоках событий у издателя. Это фактически обеспечивает возможность публикации по подписке, когда настраиваемые потребности в сведениях и права доступа для каждого целевого получателя воспринимаются издателем до того, как соответствующие записи будут размещены и вытолкнуты (push). Подписчик получает продолжающийся настроенный поток публикуемых издателем сведений.

Описанная здесь функциональность не зависит от транспорта, а для настройки или динамического оповещения о подписках может служить такой транспорт, как протокол настройки сети (Network Configuration Protocol или NETCONF) [RFC6241] или RESTCONF [RFC8040]. Привязки для доставки записей о событиях по протоколу NETCONF и RESTCONF описаны в [RFC8640] и [RESTCONF-Notif], соответственно.

Модель данных YANG, заданная здесь, соответствует архитектуре хранилищ данных управления сетью (Network Management Datastore Architecture или NMDA), определённой в [RFC8342].

### 1.1. Мотивация

Ограничения для подписки, указанные в [RFC5277], были частично смягчены в [RFC7923]. Мотивом этой работы является разрешение оставшихся вопросов. Основные возможности, поддерживаемые этим документом, включают:

- несколько подписок в одной транспортной сессии;
- поддержка динамической и настраиваемой подписки;
- изменение имеющейся подписки;
- рабочие счётчики на уровне подписки;
- согласование параметров подписки (на основе советов, возвращаемых как часть отклонённого запроса);
- уведомления о смене состояния подписки (например, приостановка издателем, изменение параметров);
- независимость от транспорта.

### 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.

#### *Client* - клиент

Определён в [RFC8342].

#### *Configuration* - конфигурация

Определён в [RFC8342].

#### *Configuration datastore* - хранилище конфигурации

Определён в [RFC8342].

**Configured subscription - настраиваемая подписка**

Подписка, организованная путём настройки в хранилище конфигурации.

**Dynamic subscription - динамическая подписка**

Подписка, динамически задаваемая подписчиком вызовом удалённой процедуры (Remote Procedure Call или RPC).

**Event - событие**

Что-то, представляющее интерес, например, смена конфигурации, отказ, смена статуса, достижение порога, внешнее событие в системе.

**Event occurrence time - время события**

Временная метка, соответствующая моменту, когда порождающий процесс решил, что событие произошло.

**Event record - запись о событии**

Набор сведений о событии.

**Event stream - поток событий**

Продолжающийся, хронологически упорядоченный набор событий, собранных в том или ином контексте.

**Event stream filter - фильтр потока событий**

Критерий оценки, применяемый к записям потока событий. Записи проходят фильтр при соответствии критериям.

**Notification message - уведомляющее сообщение (уведомление)**

Сведения для получателя, указывающие, что произошло одно или несколько событий.

**Publisher - издатель**

Сущность, отвечающая за поток уведомляющих сообщений в соответствии с подпиской.

**Receiver - получатель**

Целью (элемент), которой издатель выталкивает записи о событиях в соответствии с подпиской. Для динамической подписки получатель и подписчик совпадают.

**Subscriber - подписчик**

Клиент, способный запросить и согласовать условия генерации и выталкивания издателем записей о событиях.

Для динамической подписки получатель и подписчик совпадают.

**Subscription - подписка**

Соглашение с издателем, предусматривающее сведения, для которых один или несколько получателей желают, чтобы издатель выталкивал (push) их без дополнительного запроса.

Используемые в документе диаграммы деревьев YANG соответствуют нотации [RFC8340].

### 1.3. Обзор решения

Этот документ описывает независимый от транспорта механизм для подписки и получения содержимого потока событий у издателя. Механизм работает на основе подписки. Поддерживается два типа подписки, указанных ниже.

1. Динамическая подписка, когда подписчик инициирует организацию подписки от издателя через RPC. Если издатель способен обслужить запрос, он воспринимает его и начинает выталкивать уведомления для подписчика. При неспособности издателя обслужить запрос он возвращает сообщение об ошибке, которое **может** включать советы по параметрам подписки, которые позволят воспринять новый запрос.
2. Настраиваемые подписки, которыми можно управлять с помощью конфигурации, чтобы издатель мог передавать получателю уведомления. Поддержка настраиваемой подписки необязательна, а её доступность анонсируется через YANG feature.

Ниже указаны дополнительные характеристики, по которым различается динамическая и настраиваемая подписка.

- Срок действия динамической подписки привязан к создавшей её транспортной сессии. Для ориентированного на соединения транспорта с поддержкой состояния, такого как NETCONF, потеря транспортной сессии ведёт к немедленному прекращению всех связанных с ней динамических подписок. Для транспорта без организации соединений или поддержки состояния, такого как HTTP, отсутствие подтверждений приёма последовательного набора уведомлений и/или сообщений keep-alive может вызывать прерывание динамической подписки. Это отличается от настраиваемой подписки, срок действия которой определяет соответствующая конфигурация, присутствующая у издателя. Привязка к конфигурации означает, что (1) настроенные подписчики могут сохраняться при перезагрузке и (2) даже после полного отключения издателя от сети.
- Настраиваемые подписки может менять любой клиент настройки конфигурации с правами записи для настроек подписки. Динамические подписки можно изменять лишь вызовами RPC от исходного подписчика или путём изменения данных конфигурации, указываемых подпиской.

Отметим, что в одной подписке не смешиваются динамические и настраиваемые операции. В частности, настраиваемую подписку нельзя изменить или удалить с использованием RPC, заданных в этом документе. Динамическую подписку нельзя напрямую изменить или удалить операциями настройки конфигурации. Однако можно выполнить операцию настройки, опосредованно влияющую на динамическую подписку. При смене значения настроенного фильтра, указанного имеющейся динамической подпиской, передаваемые записи событий могут смениться.

Отметим также, что спецификации транспорта, основанные на этом документе, **должны** детализировать жизненный цикл динамических и настраиваемых (если они поддерживаются) подписок.

Издатель **может** прервать динамическую подписку в любой момент, а также **может** приостановить отправку уведомлений для любой динамической подписки и одного или нескольких получателей настраиваемой подписки. Такое прекращение или приостановка определяется самим издателем.

### 1.4. Отличия от RFC 5277

Этот документ предназначен для обеспечения расширенных возможностей подписки, исходно заданной в [RFC5277]. Важно понимать, что используется повторно или заменено, особенно при расширении имеющихся реализаций на основе [RFC5277]. Основные различия между этими двумя документами указаны ниже.

- Этот документ задаёт независимую от транспорта возможность, а [RFC5277] привязан к NETCONF.

- Для новых операций применяется описанная здесь модель данных вместо модели из параграфа 3.4 в [RFC5277].
- Операции RPC из этого документа заменяют <create-subscription> из раздела 4 в [RFC5277].
- Применяется сообщение <notification> из раздела 4 в [RFC5277].
- Содержимое потока событий NETCONF идентично в обоих случаях.
- Издатель **может** одновременно реализовать Notification Management Schema и RPC из [RFC5277] и этого документа.
- В отличие от [RFC5277], этот документ позволяет в одной транспортной сессии смешивать уведомления и RPC из разных подписок.
- Время остановки (stop-time) для подписки можно задать как часть воспроизведения (replay) уведомления. Это поддерживает возможность, аналогичную параметру <stopTime> из [RFC5277]. Однако в данной спецификации параметр stop-time может применяться и без воспроизведения.

## 2. Решение

В соответствии с обзором в параграфе 1.3 этот раздел подробно описывает общий контекст, конечные автоматы и подсистемы, которые могут быть собраны для обеспечения возможности подписки на события от издателя.

### 2.1. Поток событий

Поток событий - это именованный элемент у издателя, представляющий постоянно обновляемый набор записей о событиях, заданных в YANG. Запись о событии является воплощением оператора YANG notification. Если оператор notification задан как потомок узла данных, экземпляр включает иерархию узлов, указывающую узел данных в хранилище (параграф 7.16.2 в [RFC7950]). Каждый поток событий доступен для подписки. Идентификация способов а) определения потока событий (кроме потока NETCONF), б) определения и генерации записей о событиях и с) привязки записей к потоку событий выходят за рамки этого документа.

В этом документе определено единственное имя для потока событий - NETCONF. Поток событий NETCONF содержит все сведения о записи события NETCONF, поддерживаемые издателем, за исключением случаев явного исключения записи из потока. В дополнение к NETCONF реализация **может** задать дополнительные потоки событий.

Поскольку заданные в YANG записи о событиях создаются системой, они могут быть привязаны к одному или нескольким потокам. Запись о событии распространяется подписчикам, когда (1) подписка включает указанный поток и (2) фильтры подписки не исключают эту запись для получателя.

Могут применяться правила управления доступом для исключения записей о событиях из потока событий при отсутствии доступа на чтение. В параграфе 3.4.6 [RFC8341] приведён пример такого контроля. Отметим, что в соответствии с параграфом 2.7 этого документа, подписка на уведомления о смене состояния никогда не фильтруется.

Если для записей о событии в потоке не задано правил контроля доступа, получателю **должен** разрешаться доступ ко всем записям. Если права доступа для подписчика меняются во время действия подписки и доступ к потоку событий становится невозможным, подписка **должна** прерываться.

Записи о событиях **недопустимо** доставлять получателям с изменением порядка из включения в поток событий.

### 2.2. Фильтры потоков событий

Этот документ задаёт расширяемый механизм фильтрации. Фильтр является логическим тестом, помещаемым в запись о событии. Результат false ведёт к исключению записи из доставки получателю. Фильтр никогда не вырезает сведений из записи о событии до того, как запись будет инкапсулирована в уведомление. Поддерживается два необязательных варианта синтаксиса фильтрации потоков на основе [XPath] и ветвей (субдеревьев) [RFC6241].

Если в подписке не задан фильтр потока событий, получателю передаются все записи из потока.

### 2.3. QoS

Этот документ предоставляет несколько параметров качества обслуживания (Quality of Service или QoS), задающих обработку подписки по отношению к иному трафику между издателем и получателем.

- Маркировка dscp для дифференцированной приоритизации уведомлений при передаче через сеть.
- «Вес» для выделения подписке пропускной способности пропорциональной ему среди других подписок.
- «Зависимость» (dependency) от других подписок.

Если издатель поддерживает свойство (feature) dscp, подписка с листом dscp **должна** приводить к соответствующей маркировке DSCP<sup>1</sup> [RFC2474] в заголовке IP каждого уведомления, включая уведомления о смене состояния. Издатель **должен** соблюдать маркировку DSCP для исходящего от него трафика подписки.

Для разных кодов DSCP нужны разные транспортные соединения, поэтому при использовании TCP издатель с поддержкой свойства dscp должен гарантировать, что уведомления из подписки передаются в одной транспортной сессии TCP, где весь трафик подписки использует значение листа dscp. Если это невозможно гарантировать, любой вызов RPC establish-subscription **следует** отклонять с ошибкой dscp-unavailable.

Для параметра weighting при одновременном извлечении из очереди уведомлений из нескольких подписок для получателя издатель **должен** выделить для каждой подписки пропускную способность, пропорциональную «весу» этой подписки. Поддержка weighting не обязательна для издателя и зависит от свойства (feature) qos.

<sup>1</sup>Differentiated Services Code Point - код дифференцированного обслуживания.

Если для подписки установлен параметр `dependency`, любые буферизованные уведомления, содержащие события, выбранные родительской подпиской, **должны** выталкиваться из очереди (`dequeue`) раньше уведомлений зависимой подписки. Если уведомления зависят друг от друга, первыми **должны** выходить из очереди те, которые были в ней дольше. Если `dependency` в RPC ссылается не на несуществующую или больше не доступную подписку, зависимость (`dependency`) **должна** удаляться без уведомления. Возможность `dependency` необязательна для издателя и её поддержка указывается свойством (`feature`) `qos`.

Параметры `dependency` и `weighting` будут соблюдаться и применяться только для подписок с одинаковым значением листа `dscp`.

Имеются другие типы перегрузки возможностей (`capacity`) издателя, которые данная спецификация не рассматривает. Например, не обсуждается приоритизация подписок при перегрузке CPU у издателя. При выборе реализации такие соображения необходимо учитывать.

## 2.4. Динамическая подписка

Для управления динамическими подписками служат операции протокола (в форме RPC в соответствии с параграфом 7.14 [RFC7950]) выполняемые по отношению к элементам издателя (целям). Эти RPC разрабатывались с возможностью расширения, чтобы их можно было выполнять для целей подписки вне потока событий. Примеры таких дополнений RPC приведены в модели данных YANG, представленной в [RFC8641].

### 2.4.1. Конечный автомат динамической подписки

На рисунке 1 показан конечный автомат издателя для динамической подписки, каждое состояние которого указано отдельным блоком. Важно отметить, что подписка не существует у издателя, пока не будет воспринят вызов RPC `establish-subscription`. Простого запроса подписчика недостаточно для того, чтобы подписка была видна извне. Состояния `start` и `end` показаны для отражения событий начала и завершения подписки.

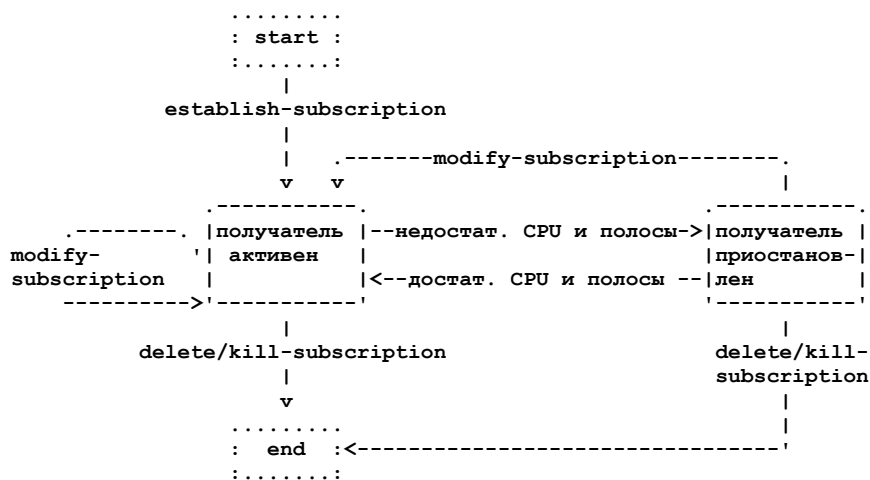


Рисунок 1. Конечный автомат издателя для динамической подписки.

Ниже указаны интересные аспекты конечного автомата.

- Успешный вызов RPC `establish-subscription` или `modify-subscription` переводит подписку в состояние `active`.
- Отказ RPC `modify-subscription` не меняет состояния подписки и потоковые обновления не меняются.
- RPC `delete-subscription` или `kill-subscription` завершают подписку, как и достижение `stop-time`.
- Издатель может приостановить подписку при нехватке ресурсов CPU или пропускной способности для её обслуживания. Это анонсируется подписчику в уведомлении о смене состояния `subscription-suspended`.
- Подписчик может изменить приостановленную подписку (например, в попытке использовать меньше ресурсов) и успешное изменение возвращает подписку в состояние `active`.
- Даже без запроса `modify-subscription` издатель может вернуть подписку в состояние `active` при наличии достаточных ресурсов. Это анонсируется подписчику в уведомлении о смене состояния `subscription-resumed`.

### 2.4.2. Организация динамической подписки

RPC `establish-subscription` позволяет подписчику запросить создание подписки. Операция принимает на входе ряд параметров, описанных ниже.

- Имя потока - `stream`, указывающее целевой поток событий, к которому применяется подписка.
- Фильтр событий потока, который может сокращать число выталкиваемых подписчику записей.
- Если применяемый RPC использует несколько форм кодирования, параметр `encoding` задаёт тип кодирования. При отсутствии этого параметра **должно** применяться кодирование RPC.
- Необязательное время остановки подписки - `stop-time`. При отсутствии `stop-time` уведомления будут передаваться до момента прерывания подписки.
- Необязательный параметр `replay-start-time`, значение которого **должно** быть в прошлом, указывающий что в подписке нужно воспроизвести события с этого момента. Воспроизведение событий описано в параграфе 2.4.2.1. При отсутствии `replay-start-time` подписка начинается с момента её организации.

Если издатель может выполнить запрос `establish-subscription`, он возвращает идентификатор подписки и незамедлительно запускает поток уведомлений о событиях.

Ниже приведена диаграмма дерева establish-subscription. Объекты дерева представлены в модуле YANG (раздел 4).

```

+---x establish-subscription
+---w input
| +---w (target)
| | +--:(stream)
| | | +---w (stream-filter)?
| | | | +--:(by-reference)
| | | | | +---w stream-filter-name
| | | | | | stream-filter-ref
| | | | | +--:(within-subscription)
| | | | | | +---w (filter-spec)?
| | | | | | | +--:(stream-subtree-filter)
| | | | | | | | +---w stream-subtree-filter? <anydata>
| | | | | | | | | {subtree}?
| | | | | | | +--:(stream-xpath-filter)
| | | | | | | | +---w stream-xpath-filter?
| | | | | | | | | yang:xpath1.0 {xpath}?
| | | | | +---w stream
| | | | | | stream-ref
| | | | +---w replay-start-time?
| | | | | yang:date-and-time {replay}?
| +---w stop-time?
| | yang:date-and-time
| +---w dscp?
| | | {dscp}?
| | | | inet:dscp
| +---w weighting?
| | | {qos}?
| | | | uint8
| +---w dependency?
| | | subscription-id {qos}?
| +---w encoding?
| | | encoding
+--ro output
+--ro id
| | subscription-id
+--ro replay-start-time-revision?
| | yang:date-and-time
| | | {replay}?

```

Рисунок 2. Диаграмма дерева RPC establish-subscription.

Издатель **может** отвергнуть RPC establish-subscription по многим причинам, как описано в параграфе 2.4.6. Передаваемое в результате сообщение об ошибке RPC **может** включать детали входных параметров, позволяющие организовать подписку следующим вызовом RPC establish-subscription. Такие советы **должны** передаваться в контейнере yang-data establish-subscription-stream-error-info, включённом в сообщение об ошибке RPC.

Ниже представлено дерево yang-data establish-subscription-stream-error-info. Объекты этого дерева описаны в модуле YANG (раздел 4).

```

yang-data establish-subscription-stream-error-info
+--ro establish-subscription-stream-error-info
+--ro reason?
| | identityref
+--ro filter-failure-hint?
| | string

```

Рисунок 3. Диаграмма дерева yang-data establish-subscription-stream-error-info.

### 2.4.2.1. Запрос воспроизведения записей о событиях

Воспроизведение позволяет организовать подписку, способную передать записи о событиях недавнего прошлого. Иными словами, при инициализации подписки передаются все записи целевого потока событий, которые соответствуют фильтру, произошли после replay-start-time и до stop-time, если имеется stop-time. Завершение записей о прошлых событиях указывается сменой состояния подписки на replay-completed. Затем могут следовать любые записи о событиях после организации подписки. Для конкретной подписки все записи событий доставляются в порядке размещения в потоке событий.

Функция воспроизведения необязательна и её поддержка зависит от поддержки потока событий в том или ином журнале (log). Этот документ не ограничивает размер и форму журнала, место его размещения у издателя, условия очистки.

Включение replay-start-time в RPC establish-subscription указывает запрос воспроизведения. Если replay-start-time содержит значение раньше, чем поддерживается в хранящейся истории событий, подписка воспринимается, а в replay-start-time-revision возвращаемого отклика **должно** указываться фактическое время начала воспроизведения. В подписку на воспроизведение может включаться параметр stop-time, которые **может** указывать время до текущего момента, но **должен** указывать время после replay-start-time. Если replay-start-time указывает время после указанного в записях буфера воспроизведения, издатель **должен** передать уведомления replay-completed сразу после отклика об успехе RPC establish-subscription.

Если поток событий поддерживает воспроизведение, лист replay-support присутствует в записи списка /streams/stream для потока событий. Не предполагается, что поток событий с поддержкой воспроизведения имеет неограниченную возможность сохранять уведомления для любого последующего запроса на воспроизведение. Чтобы оценить временные рамки доступных для воспроизведения событий, подписчик может считать значения листьев replay-log-creation-time и replay-log-aged-time, показанные на рисунке 18 для дерева YANG и в разделе 4 с определением элементов модуля YANG. Фактический размер журнала воспроизведения в каждый момент времени зависит от издателя. Параметры управления журналом воспроизведения выходят за рамки этого документа.

### 2.4.3. Изменение динамической подписки

Операция modify-subscription позволяет изменить условия имеющейся динамической подписки. Подписку можно менять неограниченное число раз. Изменение динамической подписки возможно лишь через RPC с использованием транспортной сессии, соединяющей с подписчиком. Если издатель принимает запрос на изменение подписки, он подтверждает это подписчику и сразу же начинает передавать сообщения в соответствии с новыми условиями.

Подписку, созданную через конфигурацию, нельзя изменить с помощью этого вызова RPC. Однако можно использовать настройку для изменения объектов, указанных в подписке (таких, как указанных фильтр).

Ниже приведено дерево для modify-subscription. Объекты дерева описаны в модуле YANG (раздел 4).

```

+---x modify-subscription
  +---w input
    +---w id
      |
      | subscription-id
    +---w (target)
      | +---: (stream)
      |   +---w (stream-filter)?
      |     +---: (by-reference)
      |       | +---w stream-filter-name
      |       |   |
      |       |   | stream-filter-ref
      |       |   |
      |       |   +---: (within-subscription)
      |       |     +---w (filter-spec)?
      |       |       +---: (stream-subtree-filter)
      |       |         | +---w stream-subtree-filter? <anydata>
      |       |         |   {subtree}?
      |       |         +---: (stream-xpath-filter)
      |       |           +---w stream-xpath-filter?
      |       |             yang:xpath1.0 {xpath}?
    +---w stop-time?
      yang:date-and-time
  
```

Рисунок 4. Дерево modify-subscription RPC.

Если издатель принимает запрошенные изменения приостановленной в данный момент подписки, подписка незамедлительно возобновляется (т. е. возвращается в состояние active). Издатель может сразу же приостановить измененную подписку, передав уведомление subscription-suspended до отправки какой-либо записи обновления.

Если издатель отвергает запрос RPC, состояние подписки не изменяется, т. е. запрос не имеет влияния. Отклонение RPC по любой причине указывается ошибкой RPC, как указано в параграфе 2.4.6. Содержимое отклика с ошибкой RPC **может** включать советы, которые (при их принятии) могут обеспечить успех следующей подписки. Эти советы должны передаваться в контейнере yang-data modify-subscription-stream-error-info, помещаемом в отклик об ошибке RPC.

Ниже приведено дерево для modify-subscription-stream-error-info RPC yang-data. Объекты дерева описаны в модуле YANG (раздел 4).

```

yang-data modify-subscription-stream-error-info
  +---ro modify-subscription-stream-error-info
    +---ro reason?          identityref
    +---ro filter-failure-hint? string
  
```

Рисунок 5. Дерево modify-subscription-stream-error-info RPC yang-data.

#### 2.4.4. Удаление динамической подписки

Операция delete-subscription позволяет отменить имеющуюся подписку. Если издатель принимает и подтверждает запрос, ему **недопустимо** предавать какие-либо уведомления для подписки.

Ниже приведено дерево для delete-subscription. Объекты дерева описаны в модуле YANG (раздел 4).

```

+---x delete-subscription
  +---w input
    +---w id      subscription-id
  
```

Рисунок 6. Дерево delete-subscription RPC.

Удаление динамической подписки возможно лишь через RPC с использованием транспортной сессии, соединяющей с подписчиком. Настраиваемую подписку нельзя удалить с помощью RPC.

#### 2.4.5. Уничтожение динамической подписки

Операция kill-subscription позволяет оператору прервать динамическую подписку, не связанную с транспортной сессией, используемой RPC. Издатель **должен** прервать любую динамическую подписку, указанную параметром id в запросе RPC, если такая подписка существует.

Настраиваемую подписку нельзя удалить с помощью этого вызова RPC, но можно удалить её обычной операцией настройки. Издатели **должны** отвергать любые RPC, пытающиеся уничтожить настраиваемую подписку.

Ниже приведено дерево для kill-subscription. Объекты дерева описаны в модуле YANG (раздел 4).

```

+---x kill-subscription
  +---w input
    +---w id      subscription-id
  
```

Рисунок 7. Дерево kill-subscription RPC.

#### 2.4.6. Отказы RPC

При неудачном выполнении RPC издатель возвращает связанные с этим сведения как часть отклика RPC об ошибке. Обработка ошибок транспортного уровня **должна** выполняться до описанной в этом параграфе обработки ошибок RPC. Во всех случаях сведения об ошибке RPC, возвращённые издателем, будут использовать имеющиеся структуры RPC транспортного уровня, такие как можно видеть в NETCONF (Приложение А к [RFC6241]) или RESTCONF (параграф 7.1 в [RFC8040]). Эти структуры **должны** обеспечивать кодирование связанных с подписками ошибок, указанных ниже и определённых в представленной здесь модели данных YANG.

В результате такого разнообразия способ кодирования ошибок подписки в отклике RPC об ошибке зависит от транспорта. Действительные ошибки, которые могут возникать для каждого типа RPC, указаны ниже.

**establish-subscription modify-subscription delete-subscription kill-subscription**  
 dscp-unavailable filter-unsupported no-such-subscription no-such-subscription  
 encoding-unsupported insufficient-resources  
 filter-unsupported no-such-subscription  
 insufficient-resources  
 replay-unsupported

Пример основанного на NETCONF отклика с сообщением об ошибке из приведённого списка можно найти в отклике на ошибку no-such-subscription, представленном на рисунке 10 в [RFC8640].

Имеется один финальный набор независимых от транспорта элементов для ошибок RPC, включённых в определённую здесь модель данных YANG - 3 структуры yang-data, позволяющие издателю предоставить получателям любые сведения об ошибке, которые не помещаются в имеющиеся структуры RPC транспортных уровней.

1. Структура establish-subscription-stream-error-info **должна** возвращаться с заполненным листом reason, если причина ошибки не включена где-то в транспортной части отклика от отказа RPC establish-subscription. Структура **должна** передаваться, если включены советы по преодолению ошибки RPC.
2. Структура modify-subscription-stream-error-info **должна** возвращаться с заполненным листом reason, если причина ошибки не включена где-то в транспортной части отклика от отказа RPC modify-subscription. Структура **должна** передаваться, если включены советы по преодолению ошибки RPC.
3. Структура delete-subscription-error-info **должна** возвращаться с заполненным листом reason, если причина ошибки не включена где-то в транспортной части отклика от отказа RPC delete-subscription или kill-subscription.

## 2.5. Настраиваемая подписка

Настраиваемые подписки организуются через конфигурацию и могут быть изменены любым клиентом настройки, имеющим соответствующие полномочия. Изменение или прерывание подписки через конфигурацию возможно в любой момент. В рамках одной транспортной сессии **должно** поддерживаться множество подписок.

Настраиваемые подписки имеют ряд отличий от динамических подписок:

- сохранение после перезагрузки издателя;
- сохранение при недоступности транспорта;
- возможность передавать уведомления не одному получателю (получатели не знают друг о друге).

Поддержка настраиваемой подписки не обязательна для издателей и указывается анонсированием свойства configured. Для получателя настраиваемой подписки необязательна поддержка динамической подписки. Однако, если для настраиваемой подписки требуется воспроизведение пропущенных записей о событиях, получателю настоятельно рекомендуется поддерживать динамическую подписку, которая позволяет организовать повтор пропущенных записей.

В дополнение к описанным в параграфе 2.4.2 параметрам динамической подписки для настраиваемых подписок доступны указанные ниже параметры.

- Параметр transport указывает транспортный протокол, используемый для всех получателей подписки.
- Один или несколько получателей записей о событиях, указываемых по именам (name).
- Необязательные параметры, указывающие точку выхода трафика от издателя:
  - source-interface указывает выходной интерфейс издателя (поддержка этого параметра не обязательна и анонсируется свойством interface-designation);
  - source-address указывает адрес IP для указания в уведомлениях, отправляемых получателям;
  - source-vrf указывает экземпляр виртуальной маршрутизации и пересылки (Virtual Routing and Forwarding или VRF) для отправки уведомлений получателям, определённый в [RFC8529] (поддержка VRF не обязательна для издателя и анонсируется свойством supports-vrf).

Если ни один из этих параметров не задан, уведомления **должны** передаваться через принятый по умолчанию интерфейс издателя.

Дерево, включающее эти параметры, приведено на рисунке 20 в параграфе 3.3, а параметры описаны в модуле YANG (раздел 4).

### 2.5.1. Конечный автомат настраиваемой подписки

На рисунке 8 показан конечный автомат издателя для настраиваемой подписки, описывающий 3 состояния (valid, invalid, concluded) и переходы между ними. Указаны также начальное (start) и конечное (end) состояния для создания и удаления подписки. Создание или изменение настраиваемой подписки вызывает оценку издателем возможности подписки (valid или invalid) на основании своих критериев. Если подписка возможна (valid), она начинает работать - линия (1) на рисунке.

Подписка может перейти из состояния valid в состояние invalid одним из двух способов. Во-первых, это может быть изменение, делающее подписку непригодной - (2) на рисунке. Во-вторых, издатель может решить, что подписка больше не поддерживается. Это может быть связано с неожиданным, но существенным ростом числа записей в потоке событий, снижением возможностей CPU, указанием более сложного фильтра или другими подписками, потребляющими много ресурсов - (3) на рисунке. В любом случае передаётся уведомление subscription-terminated всем получателям с состоянием active или suspended. Подписчик с состоянием valid может быть переведен в состояние concluded по достижении времени остановки - (5). В этом случае передаётся уведомление subscription-concluded всем получателям с состоянием active или suspended. Подписка может быть удалена через настройку - (4).





По причине отсутствия явной привязки к имеющейся транспортной сессии операции настройки должны включать дополнительные параметры по сравнению с динамической подпиской. Эти параметры идентифицируют каждого получателя, способ соединения с ним и, возможно, привязку к определённому выходному интерфейсу издателя. Параметры зависящей от получателя транспортной связности **должны** настраиваться через определяемые транспортом дополнения (augment) к этой спецификации (2.5.7. Транспортная связность для настраиваемой подписки).

После организации подписки издатель сразу же передаёт каждому получателю уведомление о смене состояния подписки `subscription-started`. Вполне возможно, что после настройки, перезагрузки или даже работы в установленном состоянии транспортная сессия может стать недоступной получателю. В таком случае при наличии данных для доставки в активной подписке применяются зависящие от транспорта операции «звонка домой» (`call home`) [RFC8071] для организации соединения. Когда транспорт будет доступен, уведомления можно вытолкнуть получателю.

Для активных настраиваемых подписок разрешается буферизовать записи о событиях даже после отправки уведомления `subscription-started`. Однако при потере событий (а не просто задержке) из-за исчерпания буфера воспроизведения должно передаваться новое уведомление `subscription-started`, указывающее разрыв в записях.

Пример создания подписки с помощью операций настройки по протоколу NETCONF приведён в Приложении А.

### 2.5.3. Изменение настраиваемой подписки

Настраиваемые подписки можно изменять конфигурационными операциями применительно к ветви `subscriptions`.

Если изменение добавляет получателей, для них устанавливается состояние `connecting`. При удалении получателя ему передаётся уведомление о смене статуса подписки `subscription-terminated`, если он был активен или приостановлен.

Если изменение влияет на правила для подписки, издатель передаёт активным получателям уведомление `subscription-modified`. Для приостановленных получателей такое уведомление задерживается до момента восстановления подписки. Отметим, что в этом случае уведомление `subscription-modified` сообщает получателю о возобновлении подписки, поэтому передавать дополнительно уведомление `subscription-resumed` не требуется. Если во время приостановки произошло несколько изменений, получателю требуется передавать лишь уведомление `subscription-modified`, описывающее последнее изменение.

### 2.5.4. Удаление настраиваемой подписки

Подписчиков можно удалить с помощью операций настройки применительно к ветви верхнего уровня `subscriptions`.

Сразу после удаления подписки издатель передаёт всем её получателям уведомление об этом (`subscription-terminated`).

### 2.5.5. Сброс получателя настраиваемой подписки

Возможны случаи, когда настраиваемую подписку для получателя требуется сбросить (`reset`). Это выполняется с помощью действия `reset` в модуле YANG (`/subscriptions/subscription/receivers/receiver/reset`). Действие может быть полезно при тайм-ауте издателя для попытки связаться с получателем. При таком сбросе иницируется транспортная сессия, если это требуется, и передаётся новое уведомление `subscription-started`. Действие не влияет на транспортную связность, если требуемое соединение уже имеется.

### 2.5.6. Воспроизведение для настраиваемой подписки

Возможно воспроизведение (повтор) для настраиваемой подписки, выполняемое через конфигурацию её объекта `configured-replay`. Установка этого объекта включает потоковую передачу буферизованных записей потока событий подписки с момента последнего перезапуска издателя каждому настроенному получателю.

Воспроизведение записей о событиях, созданных с момента перезапуска, полезно. Это позволяет передать получателю записи о событиях, созданные до организации транспортного соединения с ним. Установка времени перезапуска как стартовой точки воспроизведения позволяет исключить повтор записей о событиях, созданных до перезапуска. Это также гарантирует, что каждому получателю будут переданы одни и те же записи, независимо от скорости транспортного соединения с ним. Установка времени перезапуска в качестве стартовой точки воспроизведения позволяет чётко задать временные рамки.

В результате при активизации любого получателя настраиваемой подписки буферизованные записи будут переданы ему сразу после уведомления `subscription-started`. Если издатель знает последнюю переданную получателю запись и не перезапустился, первой воспроизводимой записью будет следующая запись, прошедшая фильтр отбора. В иных случаях первой будет запись, соответствующая фильтру отбора, следующая за самым поздним из моментов `replay-log-creation-time`, `replay-log-aged-time` или времени последней загрузки издателя. Листья `replay-log-creation-time` и `replay-log-aged-time` описаны в параграфе 2.4.2.1. Время последней загрузки издателя гарантирует отсутствие дубликатов записей из предшествующего перезагрузке периода.

Вполне возможно желание получателя воспроизвести записи из потока событий до последнего перезапуска издателя. Если такие записи имеются и настроено воспроизведение, издатель **должен** передать время записи о событии, непосредственно предшествующее `replay-start-time` в листе `replay-previous-event-time`. Благодаря наличию `replay-previous-event-time`, получатель будет знать, что имеются события, записанные до перезапуска. Кроме того, если получатель уже принимал записи о событиях с тем же идентификатором подписки, он может определить интервал времени для записей о событиях, которые не были получены. Зная это, получатель может использовать динамическую подписку для извлечения любых событий, помещённых в поток перед последним перезапуском.

Остальные функции воспроизведения совпадают с описанными в параграфе 2.4.2.1 для динамической подписки.

### 2.5.7. Транспортная связность для настраиваемой подписки

Эта спецификация не зависит от транспорта, однако для поддержки настраиваемой подписки зачастую нужна транспортная связность и параметры её организации зависят от транспорта. Поэтому модуль YANG, описанный в разделе 4, не может напрямую задавать и раскрывать эти транспортные параметры.

Реализации должны поддерживать процесс организации соединений, для чего в заданную этим документом модель данных YANG включён узел, куда можно добавить (augment) зависящие от транспорта параметры для отдельного получателя (/subscriptions/subscription/receivers/receiver). Добавление транспортных параметров в этот узел позволяет разработчикам включать объекты YANG, требуемые для поддержки процесса организации соединений.

В результате издатель, поддерживающий свойство configured, **должен** также поддерживать хотя бы одну модель данных YANG, дополняющую транспортные параметры в /subscriptions/subscription/receivers/receiver. Пример такого дополнения приведён в Приложении А.

## 2.6. Доставка записей о событиях

После организации подписки (динамической или настраиваемой) издатель передаёт поток записей о событиях в уведомлениях в соответствии с условиями подписки. Для динамической подписки уведомления передаются в сессии, использованной для организации подписки. Для настраиваемой подписки уведомления передаются через соединения, заданные транспортом и каждым получателем подписки.

Уведомление передаётся получателю, если запись о событии не заблокирована фильтром или контролем доступа для получателя. Уведомление **должно** включать объект <eventTime>, как указано в разделе 4 [RFC5277]. Этот объект <eventTime> **должен** размещаться на верхнем уровне структурированной записи YANG о событии.

Ниже приведён пример XML [W3C.REC-xml-20081126], заимствованный из параграфа 4.2.10 в [RFC7950], который иллюстрирует корректное сообщение.

```
<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-09-01T10:00:00Z</eventTime>
  <link-failure xmlns="https://acme.example.com/system">
    <if-name>so-1/2/3.0</if-name>
    <if-admin-status>up</if-admin-status>
    <if-oper-status>down</if-oper-status>
  </link-failure>
</notification>
```

Рисунок 10. Уведомление настраиваемой подписки.

В параграфе 2.2.1 [RFC5277] сказано, что уведомление передаётся подписчику, вызвавшему <create-subscription>. Этот документ интерпретирует заявление [RFC5277] более широко, позволяя передавать уведомления подписчику, вызвавшему establish-subscription или заданному в конфигурации подписчику, которому было отправлено уведомление subscription-started.

При старте или изменении динамической подписки вызовом establish-subscription или modify-subscription, соответственно, записи о событиях, соответствующие вновь применённому фильтру, **недопустимо** передавать, пока не будет отправлен отклик RPC.

При старте или изменении настраиваемой подписки записи о событиях, соответствующие вновь применённому фильтру, **недопустимо** передавать, пока не будет отправлен отклик subscription-started или subscription-modified, соответственно.

## 2.7. Уведомления об изменении состояния подписки

В дополнение к отправке получателем записей о событиях издатель **должен** передавать уведомления о смене статуса подписки при возникновении событий, связанных с управлением подпиской.

Уведомления о смене состояния подписки отличаются от других уведомлений тем, что они не включаются в поток событий. Вместо этого уведомления включаются (в соответствии с этим разделом) в последовательность уведомлений, передаваемых конкретному получателю. Уведомления о смене статуса подписки не могут фильтроваться или отбрасываться и доставляются лишь получателю, для которых статус подписки изменился. Уведомления о смене состояния подписки легко отличить от других уведомлений по расширению YANG subscription-state-notif, служащему для пометки уведомлений об изменении состояния подписки.

В последующих параграфах описаны все уведомления о смене состояния подписки.

### 2.7.1. subscription-started

This notification indicates that a configured subscription has started, and event records may be sent. Included in this subscription state change notification are all the parameters of the subscription, except for (1) transport connection information for one or more receivers and (2) origin information indicating where notification messages will egress the publisher. Note that if a referenced filter from the "filters" container has been used in the subscription, the notification still provides the contents of that referenced filter under the "within-subscription" subtree.

Note that for dynamic subscriptions, no "subscription-started" notifications are ever sent.

Below is a tree diagram for "subscription-started". All objects contained in this tree are described in the YANG module in Section 4.

```
+---n subscription-started {configured}?
  +--ro id
  |   subscription-id
  +--ro (target)
  |   +--:(stream)
  |   |   +--ro (stream-filter)?
  |   |   |   +--:(by-reference)
  |   |   |   |   +--ro stream-filter-name
  |   |   |   |   |   stream-filter-ref
  |   |   |   +--:(within-subscription)
  |   |   +--ro (filter-spec)?
```

```

| | | +--:(stream-subtree-filter)
| | | | +--ro stream-subtree-filter? <anydata>
| | | | | {subtree}?
| | | +--:(stream-xpath-filter)
| | | | +--ro stream-xpath-filter? yang:xpath1.0
| | | | | {xpath}?
| | +--ro stream stream-ref
| | +--ro replay-start-time?
| | | yang:date-and-time {replay}?
| | +--ro replay-previous-event-time?
| | | yang:date-and-time {replay}?
+--ro stop-time?
| | yang:date-and-time
+--ro dscp? inet:dscp
| | {dscp}?
+--ro weighting? uint8 {qos}?
+--ro dependency?
| | subscription-id {qos}?
+--ro transport? transport
| | {configured}?
+--ro encoding? encoding
+--ro purpose? string
| | {configured}?

```

Рисунок 11. Дерево уведомления *subscription-started*

### 2.7.2. *subscription-modified*

This notification indicates that a subscription has been modified by configuration operations. It is delivered directly after the last event records processed using the previous subscription parameters, and before any event records processed after the modification.

Below is a tree diagram for "subscription-modified". All objects contained in this tree are described in the YANG module in Section 4.

```

+---n subscription-modified
+--ro id
| | subscription-id
+--ro (target)
| | +--:(stream)
| | | +--ro (stream-filter)?
| | | | +--:(by-reference)
| | | | | +--ro stream-filter-name
| | | | | | stream-filter-ref
| | | | +--:(within-subscription)
| | | | | +--ro (filter-spec)?
| | | | | | +--:(stream-subtree-filter)
| | | | | | | +--ro stream-subtree-filter? <anydata>
| | | | | | | | {subtree}?
| | | | | | +--:(stream-xpath-filter)
| | | | | | | +--ro stream-xpath-filter? yang:xpath1.0
| | | | | | | | {xpath}?
| | | +--ro stream stream-ref
| | | +--ro replay-start-time?
| | | | yang:date-and-time {replay}?
+--ro stop-time?
| | yang:date-and-time
+--ro dscp? inet:dscp
| | {dscp}?
+--ro weighting? uint8 {qos}?
+--ro dependency?
| | subscription-id {qos}?
+--ro transport? transport
| | {configured}?
+--ro encoding? encoding
+--ro purpose? string
| | {configured}?

```

Рисунок 12. Дерево уведомления *subscription-modified*.

A publisher most often sends this notification directly after the modification of any configuration parameters impacting a configured subscription. But it may also be sent at two other times:

1. If a configured subscription has been modified during the suspension of a receiver, the notification will be delayed until the receiver's suspension is lifted. In this situation, the notification indicates that the subscription has been both modified and resumed.
2. A "subscription-modified" subscription state change notification **MUST** be sent if the contents of the filter identified by the subscription's "stream-filter-ref" leaf have changed. This state change notification is to be sent for a filter change impacting any active receivers of a configured or dynamic subscription.

### 2.7.3. *subscription-terminated*

This notification indicates that no further event records for this subscription should be expected from the publisher. A publisher may terminate the sending of event records to a receiver for the following reasons:

1. Configuration that removes a configured subscription, or a "kill-subscription" RPC that ends a dynamic subscription. These are identified via the reason "no-such-subscription".
2. A referenced filter is no longer accessible. This reason is identified by the "filter-unavailable" identity.
3. The event stream referenced by a subscription is no longer accessible by the receiver. This reason is identified by the "stream-unavailable" identity.
4. A suspended subscription has exceeded some timeout. This reason is identified by the "suspension-timeout" identity.

Each reason listed above derives from the "subscription-terminated-reason" base identity specified in the YANG data model in this document.

Below is a tree diagram for "subscription-terminated". All objects contained in this tree are described in the YANG module in Section 4.

```
+---n subscription-terminated
  +--ro id          subscription-id
  +--ro reason      identityref
```

Рисунок 13. Диаграмма дерева уведомления subscription-terminated.

Note: This subscription state change notification MUST be sent to a dynamic subscription's receiver when the subscription ends unexpectedly. This might happen when a "kill-subscription" RPC is successful or when some other event, not including reaching the subscription's "stop-time", results in a publisher choosing to end the subscription.

### 2.7.4. subscription-suspended

This notification indicates that a publisher has suspended the

sending of event records to a receiver and also indicates the possible loss of events. Suspension happens when capacity constraints stop a publisher from serving a valid subscription. The two conditions where this is possible are:

1. "insufficient-resources", when a publisher is unable to produce the requested event stream of notification messages, and
2. "unsupportable-volume", when the bandwidth needed to get

generated notification messages to a receiver exceeds a threshold.

These conditions are encoded in the "reason" object. No further notifications will be sent until the subscription resumes or is terminated.

Below is a tree diagram for "subscription-suspended". All objects contained in this tree are described in the YANG module in Section 4.

```
+---n subscription-suspended
  +--ro id          subscription-id
  +--ro reason      identityref
```

Рисунок 14. Дерево уведомления subscription-suspended.

### 2.7.5. subscription-resumed

This notification indicates that a previously suspended subscription has been resumed under the unmodified terms previously in place. Subscribed event records generated after the issuance of this subscription state change notification may now be sent.

Below is a tree diagram for "subscription-resumed". All objects contained in this tree are described in the YANG module in Section 4.

```
+---n subscription-resumed
  +--ro id          subscription-id
```

Рисунок 15. Дерево уведомления subscription-resumed.

### 2.7.6. subscription-completed

This notification indicates that a subscription that includes a "stop-time" has successfully finished passing event records upon reaching that time.

Below is a tree diagram for "subscription-completed". All objects contained in this tree are described in the YANG module in Section 4.

```
+---n subscription-completed {configured}?
  +--ro id          subscription-id
```

Рисунок 16. Дерево уведомления subscription-completed.

### 2.7.7. replay-completed

This notification indicates that all of the event records prior to the current time have been passed to a receiver. It is sent before any notification messages containing an event record with a timestamp later than (1) the "stop-time" or (2) the subscription's start time.

If a subscription does not contain a "stop-time" or has a "stop-time" that has not been reached, then after the "replay-completed" notification has been sent, additional event records will be sent in sequence as they arise naturally on the publisher.

Below is a tree diagram for "replay-completed". All objects contained in this tree are described in the YANG module in Section 4.

```
+---n replay-completed {replay}?
  +--ro id      subscription-id
```

Рисунок 17. Дерево уведомления `replay-completed`.

## 2.8. Мониторинг подписки

In the operational state datastore, the "subscriptions" container maintains the state of all dynamic subscriptions as well as all configured subscriptions. Using datastore retrieval operations [RFC8641] or subscribing to the "subscriptions" container (Section 3.3) allows the state of subscriptions and their connectivity to receivers to be monitored.

Each subscription in the operational state datastore is represented as a list element. Included in this list are event counters for each receiver, the state of each receiver, and the subscription parameters currently in effect. The appearance of the leaf "configured-subscription-state" indicates that a particular subscription came into being via configuration. This leaf also indicates whether the current state of that subscription is "valid", "invalid", or "concluded".

To understand the flow of event records in a subscription, there are two counters available for each receiver. The first counter is "sent-event-records", which shows the number of events identified for sending to a receiver. The second counter is "excluded-event-records", which shows the number of event records not sent to a receiver. "excluded-event-records" shows the combined results of both access control and per-subscription filtering. For configured subscriptions, counters are reset whenever the subscription's state is evaluated as "valid" (see (1) in Figure 8).

Dynamic subscriptions are removed from the operational state datastore once they expire (reaching "stop-time") or when they are terminated. While many subscription objects are shown as configurable, dynamic subscriptions are only included in the operational state datastore and as a result are not configurable.

## 2.9. Поддержка модуля YANG `ietf-subscribed-notifications`

Publishers supporting this document MUST indicate support of the YANG module "ietf-subscribed-notifications" in the YANG library of the publisher. In addition, if supported, the optional features "encode-xml", "encode-json", "configured", "supports-vr", "qos", "xpath", "subtree", "interface-designation", "dscp", and "replay" MUST be indicated.

## 3. Диаграммы деревьев модели YANG

This section contains tree diagrams for nodes defined in Section 4. For tree diagrams of subscription state change notifications, see Section 2.7. For the tree diagrams for the RPCs, see Section 2.4.

### 3.1. Контейнер `streams`

A publisher maintains a list of available event streams as operational data. This list contains both standardized and vendor-specific event streams. This enables subscribers to discover what streams a publisher supports.

Below is a tree diagram for the "streams" container. All objects contained in this tree are described in the YANG module in Section 4.

```
+--ro streams
  +--ro stream* [name]
    +--ro name           string
    +--ro description   string
    +--ro replay-support? empty {replay}?
    +--ro replay-log-creation-time yang:date-and-time
    | {replay}?
    +--ro replay-log-aged-time?  yang:date-and-time
    | {replay}?
```

Рисунок 18. Дерево контейнера `streams`.

### 3.2. Контейнер `filters`

The "filters" container maintains a list of all subscription filters that persist outside the lifecycle of a single subscription. This enables predefined filters that may be referenced by more than one subscription.

Below is a tree diagram for the "filters" container. All objects contained in this tree are described in the YANG module in Section 4.

```
+--rw filters
  +--rw stream-filter* [name]
    +--rw name           string
    +--rw (filter-spec)?
    | +--:(stream-subtree-filter)
    | | +--rw stream-subtree-filter? <anydata> {subtree}?
    | +--:(stream-xpath-filter)
    | +--rw stream-xpath-filter?  yang:xpath1.0 {xpath}?
```

Рисунок 19. Дерево контейнера `filters`.

### 3.3. Контейнер subscriptions

The "subscriptions" container maintains a list of all subscriptions on a publisher, both configured and dynamic. It can be used to retrieve information about the subscriptions that a publisher is serving.

Below is a tree diagram for the "subscriptions" container. All objects contained in this tree are described in the YANG module in Section 4.

```

+--rw subscriptions
  +--rw subscription* [id]
    +--rw id
      | subscription-id
    +--rw (target)
      | +--:(stream)
      |   +--rw (stream-filter)?
      |     | +--:(by-reference)
      |     | | +--rw stream-filter-name
      |     | | stream-filter-ref
      |     | +--:(within-subscription)
      |     |   +--rw (filter-spec)?
      |     |     +--:(stream-subtree-filter)
      |     |     | +--rw stream-subtree-filter? <anydata>
      |     |     | {subtree}?
      |     |     +--:(stream-xpath-filter)
      |     |     | +--rw stream-xpath-filter?
      |     |     | yang:xpath1.0 {xpath}?
      |     +--rw stream stream-ref
      |     +--ro replay-start-time?
      |     | yang:date-and-time {replay}?
      |     +--rw configured-replay? empty
      |     | {configured,replay}?
    +--rw stop-time?
      | yang:date-and-time
    +--rw dscp? inet:dscp
      | {dscp}?
    +--rw weighting? uint8 {qos}?
    +--rw dependency?
      | subscription-id {qos}?
    +--rw transport? transport
      | {configured}?
    +--rw encoding? encoding
    +--rw purpose? string
      | {configured}?
    +--rw (notification-message-origin)? {configured}?
      | +--:(interface-originated)
      | | +--rw source-interface?
      | | | if:interface-ref {interface-designation}?
      | +--:(address-originated)
      |   +--rw source-vrf?
      |     | -> /ni:network-instances/network-instance/name
      |     | {supports-vrf}?
      |   +--rw source-address?
      |     | inet:ip-address-no-zone
    +--ro configured-subscription-state? enumeration
      | {configured}?
    +--rw receivers
      +--rw receiver* [name]
        +--rw name string
        +--ro sent-event-records?
          | yang:zero-based-counter64
        +--ro excluded-event-records?
          | yang:zero-based-counter64
        +--ro state enumeration
        +---x reset {configured}?
          +--ro output
            +--ro time yang:date-and-time
  
```

Рисунок 20. Дерево контейнера subscriptions.

## 4. Модуль YANG для подписки на уведомления о событиях

This module imports typedefs from [RFC6991], [RFC8343], [RFC8341], [RFC8529], and [RFC8040]. It references [RFC6241], [XPath] ("XML Path Language (XPath) Version 1.0"), [RFC7049], [RFC8259], [RFC7950], [RFC7951], and [RFC7540].

```

<CODE BEGINS> file "ietf-subscribed-notifications@2019-09-09.yang"
module ietf-subscribed-notifications {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications";
  prefix sn;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-interfaces {
    prefix if;
  }
}
  
```

```
reference
  "RFC 8343: A YANG Data Model for Interface Management";
}
import ietf-netconf-acm {
  prefix nacm;
  reference
    "RFC 8341: Network Configuration Access Control Model";
}
import ietf-network-instance {
  prefix ni;
  reference
    "RFC 8529: YANG Data Model for Network Instances";
}
import ietf-restconf {
  prefix rc;
  reference
    "RFC 8040: RESTCONF Protocol";
}
import ietf-yang-types {
  prefix yang;
  reference
    "RFC 6991: Common YANG Data Types";
}

organization
  "IETF NETCONF (Network Configuration) Working Group";
contact
  "WG Web: <https://datatracker.ietf.org/wg/netconf/>
  WG List: <mailto:netconf@ietf.org>

  Author: Alexander Clemm
         <mailto:ludwig@clemm.org>

  Author: Eric Voit
         <mailto:evoit@cisco.com>

  Author: Alberto Gonzalez Prieto
         <mailto:alberto.gonzalez@microsoft.com>

  Author: Einar Nilsen-Nygaard
         <mailto:einarnn@cisco.com>

  Author: Ambika Prasad Tripathy
         <mailto:ambtripa@cisco.com>";
description
  "Этот модуль задаёт модель данных YANG для описания подписки на
  записи о событиях и получения соответствующих уведомлений.

  Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
  СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
  НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
  ВСП 14 (RFC 2119) (RFC 8174) тогда и только тогда, когда они
  указаны заглавными буквами, как показано здесь.

  Авторские права (Copyright (c) 2019) принадлежат IETF Trust и
  лицам, указанным как авторы. Все права защищены.

  Распространение и применение модуля в исходной или двоичной
  форме с изменениями или без таковых разрешено в соответствии с
  лицензией Simplified BSD License, изложенной в параграфе 4.с
  IETF Trust's Legal Provisions Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  Эта версия модуля YANG является частью RFC 8639, где правовые
  аспекты приведены более полно.";

revision 2019-09-09 {
  description
    "Исходный выпуск.";
  reference
    "RFC 8639: A YANG Data Model for Subscriptions to
    Event Notifications";
}

/*
 * Свойства (функции)
 */
feature configured {
  description
    "Указывает поддержку настройки подписок.";
}

feature dscp {
  description
    "Указывает поддержку издателем возможности указывать код
    дифференцированного обслуживания (DSCP) в исходящих пакетах.";
```



```
}

feature encode-json {
  description
    "Указывает поддержку кодирования JSON для уведомлений.";
}

feature encode-xml {
  description
    "Указывает поддержку кодирования XML для уведомлений.";
}

feature interface-designation {
  description
    "Указывает, что издатель выполняется все взаимодействия с
    получателем в рамках настраиваемой подписки через один
    назначенный выходной интерфейс.";
}

feature qos {
  description
    "Указывает поддержку издателем абсолютной зависимости трафика
    одной подписки от другой, а также взвешенное распределение
    пропускной способности между подписками. Это относится к функциям
    качества обслуживания (QoS), позволяющим дифференцированно
    обрабатывать уведомления между издателем и получателями.";
}

feature replay {
  description
    "Указывает поддержку воспроизведения записи событий, которое
    позволяет передать поток записей о событиях в хронологическом
    порядке.";
}

feature subtree {
  description
    "Указывает поддержку фильтрации по ветвям дерева YANG.";
  reference
    "RFC 6241: Network Configuration Protocol (NETCONF),
    Section 6";
}

feature supports-vrf {
  description
    "Указывает поддержку издателем конфигурации VRF для
    настраиваемых подписок. Поддержка VRF для динамических
    подписок не требует этой функции.";
  reference
    "RFC 8529: YANG Data Model for Network Instances,
    Section 6";
}

feature xpath {
  description
    "Указывает поддержку фильтрации XPath.";
  reference
    "XML Path Language (XPath) Version 1.0
    (https://www.w3.org/TR/1999/REC-xpath-19991116)";
}

/*
 * Расширения
 */

extension subscription-state-notification {
  description
    "Этот оператор применим только к уведомлениям и указывает, что
    это уведомление о смене состояния подписки и поэтому оно не
    передается в обычном потоке событий и для его получения не нужна
    специальная подписка. Оператор может применяться лишь как
    субоператор оператора YANG notification и не предназначен для
    применения за пределами этого модуля YANG.";
}

/*
 * Идентификаторы
 */

/* Ошибки RPC и уведомлений */

identity delete-subscription-error {
  description
    "Базовый идентификатор для проблем, обнаруженных при попытке
    выполнить запрос RPC delete-subscription или kill-subscription.";
}

identity establish-subscription-error {
```

```
description
  "Базовый идентификатор для проблем, обнаруженных при попытке
  выполнить запрос RPC establish-subscription.";
}

identity modify-subscription-error {
  description
    "Базовый идентификатор для проблем, обнаруженных при попытке
    выполнить запрос RPC modify-subscription.";
}

identity subscription-suspended-reason {
  description
    "Базовый идентификатор для проблем, сообщённых получателю как
    часть уведомления subscription-suspended.";
}

identity subscription-terminated-reason {
  description
    "Базовый идентификатор для проблем, сообщённых получателю как
    часть уведомления subscription-terminated.";
}

identity dscp-unavailable {
  base establish-subscription-error;
  if-feature "dscp";
  description
    "Издатель не способен промаркировать приоритет уведомлений так,
    чтобы это не менялось при передаче через сеть.";
}

identity encoding-unsupported {
  base establish-subscription-error;
  description
    "Невозможно представить уведомления в желаемом формате.";
}

identity filter-unavailable {
  base subscription-terminated-reason;
  description
    "Указанного фильтра нет, т. е. получатель указал фильтр,
    которого не существует или он не имеет к фильтру прав доступа.";
}

identity filter-unsupported {
  base establish-subscription-error;
  base modify-subscription-error;
  description
    "Невозможно разобрать фильтр. Причиной может быть синтаксическая
    ошибка или излишняя сложность фильтра для издателя.";
}

identity insufficient-resources {
  base establish-subscription-error;
  base modify-subscription-error;
  base subscription-suspended-reason;
  description
    "У издателя недостаточно ресурсов для поддержки запрошенной
    подписки. Например, выделенный CPU слишком ограничен для
    генерации нужного набора уведомлений.";
}

identity no-such-subscription {
  base modify-subscription-error;
  base delete-subscription-error;
  base subscription-terminated-reason;
  description
    "Указанной подписки нет. Это может быть связано с ошибочным
    идентификатором подписки, её принадлежностью другому подписчику
    или с указанием настраиваемой подписки.";
}

identity replay-unsupported {
  base establish-subscription-error;
  if-feature "replay";
  description
    "Невозможно выполнить воспроизведение для этой подписки, т. е.
    издатель не может предоставить запрошенные исторические сведения
    для потока событий этому получателю.";
}

identity stream-unavailable {
  base subscription-terminated-reason;
  description
    "Недоступный для подписки данным получателем поток событий.";
}
```

```
identity suspension-timeout {
  base subscription-terminated-reason;
  description
    "Прерывание приостановленной подписки. Издатель отменил подписку,
     время приостановки которой достигло предела.";
}

identity unsupported-volume {
  base subscription-suspended-reason;
  description
    "У издателя нет достаточной пропускной способности сети для
     передачи сгенерированных сведения, предназначенных получателю.";
}

/* Идентификаторы кодирования */

identity configurable-encoding {
  description
    "Если транспортный идентификатор выведен из этого отождествления,
     это означает поддержку настройки кодирования. Примером
     настраиваемого кодирования может служить новый идентификатор,
     такой как encode-cbor. Такое отождествление может применять
     configurable-encoding в качестве своей базы. Это позволяет
     динамической подписке, кодированной в JSON (RFC 8259), запросить
     кодирование уведомлений в (CBOR) (RFC 7049). Дополнительные
     детали в части настраиваемого кодирования можно найти в
     транспортном документе, основанном на этой спецификации.";
  reference
    "RFC 8259: The JavaScript Object Notation (JSON) Data
     Interchange Format
     RFC 7049: Concise Binary Object Representation (CBOR)";
}

identity encoding {
  description
    "Базовый идентификатор для представления кодировки данных.";
}

identity encode-xml {
  base encoding;
  if-feature "encode-xml";
  description
    "Кодирование данных в XML, как описано в RFC 7950.";
  reference
    "RFC 7950: The YANG 1.1 Data Modeling Language";
}

identity encode-json {
  base encoding;
  if-feature "encode-json";
  description
    "Кодирование данных в JSON, как описано в RFC 7951.";
  reference
    "RFC 7951: JSON Encoding of Data Modeled with YANG";
}

/* Идентификатор для транспорта */

identity transport {
  description
    "Идентификатор для базового механизма передачи уведомлений.";
}

/*
 * Определения типов
 */

typedef encoding {
  type identityref {
    base encoding;
  }
  description
    "Задаёт кодированием данных, например, для подписки.";
}

typedef stream-filter-ref {
  type leafref {
    path "/sn:filters/sn:stream-filter/sn:name";
  }
  description
    "Этот тип служит для ссылки на фильтр потока событий.";
}

typedef stream-ref {
  type leafref {
```

```
path "/sn:streams/sn:stream/sn:name";
}
description
  "Указывает предоставляемый системой поток событий.";
}

typedef subscription-id {
  type uint32;
  description
    "Тип для идентификаторов подписки.";
}

typedef transport {
  type identityref {
    base transport;
  }
  description
    "Задаёт транспорт для передачи уведомлений получателю.";
}

/*
 * Группировки
 */
grouping stream-filter-elements {
  description
    "Задаёт базу для фильтров, применяемых к потокам событий.";
  choice filter-spec {
    description
      "Спецификация фильтра для этого запроса.";
    anydata stream-subtree-filter {
      if-feature "subtree";
      description
        "Критерии оценки потока событий в форме фильтра по ветвям,
        определённого в разделе 6 RFC 6241.

        Фильтр ветвей применяется к представлению отдельных
        выделенных записей в потоке событий.

        Если фильтр возвращает непустой набор узлов, записи событий
        включаются в уведомление, отправляемое получателем.";
      reference
        "RFC 6241: Network Configuration Protocol (NETCONF),
        Section 6";
    }
  }
  leaf stream-xpath-filter {
    if-feature "xpath";
    type yang:xpath1.0;
    description
      "Критерии оценки потока событий в форме выражения XPath 1.0.

      Выражение XPath оценивается для представления отдельных
      выделенных записей в потоке событий.

      Результат выражения XPath преобразуется в логическое значение
      с использованием стандартных правил XPath 1.0. Значение true
      указывает включение записи в уведомление, передаваемое
      получателем.

      Выражения оцениваются в описанном ниже контексте XPath.
      - Набор объявлений пространств имён - это набор пар
      «префикс, пространство имён» для всех модулей YANG,
      реализованных сервером, где префиксом служит имя модуля
      YANG, а пространство имён задаёт оператор namespace в
      модуле YANG.
      Если лист кодируется в XML, все объявления пространств
      имён в области действия листа stream-xpath-filter
      добавляются к набору объявлений пространств имён. Если
      префикс, найденный в XML, уже есть в наборе объявлений,
      применяется пространство имён в XML.
      - Набор привязок переменных пуст.
      - Библиотека функций состоит из ядра библиотеки функций и
      функций XPath из раздела 10 в RFC 7950.
      - Узлом контекста служит корневой узел хранилища.";
    reference
      "XML Path Language (XPath) Version 1.0
      (https://www.w3.org/TR/1999/REC-xpath-19991116)
      RFC 7950: The YANG 1.1 Data Modeling Language,
      Section 10";
  }
}
}

grouping update-qos {
  description
    "Описывает сведения QoS для подписки, передаваемые нижележащему
    транспорту для приоритизации и обработки.";
```

```

leaf dscp {
  if-feature "dscp";
  type inet:dscp;
  default "0";
  description
    "Желаемый уровень транспортного приоритета для уведомлений,
    инкапсулирующих содержимое подписки. Этот приоритет
    применяется для всех получателей данной подписки.";
}
leaf weighting {
  if-feature "qos";
  type uint8 {
    range "0 .. 255";
  }
  description
    "Относительный вес для подписки. Большее значение соответствует
    большим ресурсам. Это позволяет базовому транспорту обоснованно
    распределять ресурсы между подписками.";
  reference
    "RFC 7540: Hypertext Transfer Protocol Version 2 (HTTP/2),
    Section 5.3.2";
}
leaf dependency {
  if-feature "qos";
  type subscription-id;
  description
    "Указывает subscription-id родительской подписки, которая имеет
    абсолютный приоритет при наличии у неё push-обновлений, готовых
    к передаче от издателя. Иными словами, при наличии у родителя
    данных для передачи никакие потоковые объекты текущей подписки
    не следует передавать.

    Если зависимость задана в конфигурации или вызовом RPC, но
    указанного subscription-id нет, зависимость игнорируется. Если
    указанная подписка удалена, зависимость тоже удаляется.";
  reference
    "RFC 7540: Hypertext Transfer Protocol Version 2 (HTTP/2),
    Section 5.3.1";
}
}

grouping subscription-policy-modifiable {
  description
    "Описывает все объекты подписки, которые могут меняться.";
  choice target {
    mandatory true;
    description
      "Указывает источник сведений для подписки, а также желаемый
      набор информации из этого источника.";
    case stream {
      choice stream-filter {
        description
          "Фильтр событий для подписки, указываемый ссылкой на
          глобальный список фильтров или явно заданный в подписке.";
        case by-reference {
          description
            "Применять фильтр, настраиваемый отдельно.";
          leaf stream-filter-name {
            type stream-filter-ref;
            mandatory true;
            description
              "Ссылка на имеющийся фильтр потока событий, для
              применения к подписке.";
          }
        }
        case within-subscription {
          description
            "Локальное определение позволяет фильтру событий иметь
            такой же жизненный цикл, как у подписки.";
          uses stream-filter-elements;
        }
      }
    }
  }
}
leaf stop-time {
  type yang:date-and-time;
  description
    "Указывает время, когда передачу уведомлений для подписки
    следует прекратить. Если время stop-time не задано, уведомления
    будут передаваться до прерывания подписки. При наличии
    replay-start-time значение stop-time должно быть больше. Если
    replay-start-time не задано, stop-time при организации подписки
    должно указывать будущее время.";
}
}
}

```

```

grouping subscription-policy-dynamic {
  description
    "Описывает информацию о подписке, которая может быть передана
    только через вызовы RPC, заданные в этой модели данных.";
  uses subscription-policy-modifiable {
    augment "target/stream" {
      description
        "Добавляет объекты, которые можно изменить этим вызовом RPC.";
      leaf stream {
        type stream-ref {
          require-instance false;
        }
        mandatory true;
        description
          "Указывает поток событий, рассматриваемый для подписки.";
      }
      leaf replay-start-time {
        if-feature "replay";
        type yang:date-and-time;
        config false;
        description
          "Служит для включения функции replay в динамической
          подписке для записей о событиях, произошедших не ранее
          указанного времени. Если replay-start-time не указано,
          это не будет подписка на воспроизведение и выталкивание
          событий следует начинать сразу. Недопустимо указывать
          текущее или будущее время.";
      }
    }
  }
  uses update-qos;
}

grouping subscription-policy {
  description
    "Описывает полный набор правил для динамических и настраиваемых
    подписок, за исключением получателей и сетевых сведений,
    специфических для издателя, таких как интерфейс, который следует
    использовать для передачи уведомлений.";
  uses subscription-policy-dynamic;
  leaf transport {
    if-feature "configured";
    type transport;
    description
      "Для настраиваемой подписки задаёт транспорт, применяемый при
      доставке сообщений получателям этой подписки.";
  }
  leaf encoding {
    when 'not(../transport) or derived-from(../transport,
    "sn:configurable-encoding")';
    type encoding;
    description
      "Тип кодирования уведомлений. Для динамической подписки без
      указания кодирования в RPC establish-subscription будет
      применяться кодирование, использованное в RPC. Если для
      настраиваемой подписки значение не задано явно, применяется
      заданное по умолчанию кодирование базового транспорта.";
  }
  leaf purpose {
    if-feature "configured";
    type string;
    description
      "Текст, позволяющий настраиваемому объекту задать источник и
      другую специфику данной подписки.";
  }
}

/*
 * RPC
 */

rpc establish-subscription {
  description
    "Этот вызов RPC позволяет подписчику организовать (возможно, и
    согласовать) подписку от своего имени. При успешном выполнении
    подписка будет действовать на протяжении всей ассоциации
    подписчика с издателем или до прерывания подписки. Если возникает
    ошибка или издатель не может выполнить требования подписки,
    возвращается ошибка RPC и подписка не организуется. В этом случае
    error-info в отклике RPC МОЖЕТ включать параметры, применение
    которых повысит вероятность успеха следующего запроса
    establish-subscription.";
  input {
    uses subscription-policy-dynamic;
    leaf encoding {
      type encoding;
    }
  }
}

```

```

description
  "Тип кодирования для данных подписки. Если типе не указан при
  вызове RPC, издатель ДОЛЖЕН применять кодирование,
  использованное для RPC.";
}
}
output {
  leaf id {
    type subscription-id;
    mandatory true;
    description
      "Идентификатор, используемый для этой подписки.";
  }
  leaf replay-start-time-revision {
    if-feature "replay";
    type yang:date-and-time;
    description
      "При запросе воспроизведения представляет начальный момент
      воспроизведения из буфера запрошенного потока событий.
      Значение этого листа совпадает с replay-log-aged-time при
      его наличии, иначе совпадает с replay-log-creation-time.
      Получателю воспроизводятся все записи после этого времени.
      Этот лист передаётся лишь в том случае, когда указанное
      время начала позже запрошенного подписчиком времени.";
  }
}
}
}
rc:yang-data establish-subscription-stream-error-info {
  container establish-subscription-stream-error-info {
    description
      "Если любой из параметров RPC establish-subscription не подходит
      для этого потока событий, подписка не организуется и отклик об
      ошибке RPC ДОЛЖЕН указывать причину отказа. Этот элемент
      yang-data МОЖЕТ включаться как структурированные данные отклика
      об ошибке RPC для указания причины отказа. Элемент yang-data
      ДОЛЖЕН включаться, если подписчику возвращаются подсказки.";
    leaf reason {
      type identityref {
        base establish-subscription-error;
      }
      description
        "Указывает причину отказа в подписке на этот поток событий.";
    }
    leaf filter-failure-hint {
      type string;
      description
        "Сведения о причине отказа в подписке. Синтаксис и семантика
        подсказки зависят от реализации.";
    }
  }
}
}
}
rc modify-subscription {
  description
    "Этот вызов RPC позволяет подписчику изменить параметры
    динамической подписки. При успешном изменении новые параметры
    сохраняются до завершения подписки или новой смены параметров.
    В случае ошибки или невозможности установить запрошенные
    параметры подписка не изменяется и сохраняются исходные
    параметры. В этом случае отклик об ошибке RPC МОЖЕТ включать в
    error-info параметры, которые повысят вероятность успеха при
    последующем вызове modify-subscription. Успешное выполнение
    modify-subscription переводит приостановленную подписку в
    состояние active.";
  input {
    leaf id {
      type subscription-id;
      mandatory true;
      description
        "Идентификатор для использования с этой подпиской.";
    }
    uses subscription-policy-modifiable;
  }
}
}
rc:yang-data modify-subscription-stream-error-info {
  container modify-subscription-stream-error-info {
    description
      "Этот элемент yang-data МОЖЕТ возвращаться как часть отклика об
      ошибке RPC modify-subscription для потока событий. Этот элемент
      ДОЛЖЕН использоваться при возврате советов подписчику.";
    leaf reason {
      type identityref {
        base modify-subscription-error;
      }
    }
  }
}
}
}

```

```

    description
      "Сведения в отклике об ошибке RPC modify-subscription,
      указывающие причину отказа от изменения подписки.";
  }
  leaf filter-failure-hint {
    type string;
    description
      "Сведения о неприемлемости фильтра для подписки. Синтаксис
      и семантика подсказки зависят от реализации.";
  }
}

rpc delete-subscription {
  description
    "Этот вызов RPC позволяет подписчику удалить созданную им подписку
    с помощью RPC establish-subscription. При возникновении ошибки
    сервер возвращает отклик rpc-error, где поле error-info МОЖЕТ
    содержать структуру delete-subscription-error-info.";
  input {
    leaf id {
      type subscription-id;
      mandatory true;
      description
        "Идентификатор удаляемой подписки. С помощью этого вызова RPC
        можно удалить лишь подписку, созданную вызовом RPC
        establish-subscription из того же источника.";
    }
  }
}

rpc kill-subscription {
  nasm:default-deny-all;
  description
    "Этот вызов RPC позволяет оператору удалить динамическую подписку,
    независимо от исходного подписчика и базовой транспортной сессии.
    При возникновении ошибки сервер возвращает отклик rpc-error, где
    поле error-info МОЖЕТ содержать delete-subscription-error-info.";
  input {
    leaf id {
      type subscription-id;
      mandatory true;
      description
        "Идентификатор удаляемой подписки. С помощью этого вызова RPC
        можно удалить лишь подписки, организованные
        с помощью establish-subscription.";
    }
  }
}

rc:yang-data delete-subscription-error-info {
  container delete-subscription-error-info {
    description
      "При отказе RPC delete-subscription или kill-subscription
      подписка не удаляется и отклик об ошибке RPC ДОЛЖЕН указывать
      причину этого отказа. Эта структура yang-data МОЖЕТ включаться
      отклик об ошибке RPC для указания причины отказа.";
    leaf reason {
      type identityref {
        base delete-subscription-error;
      }
      mandatory true;
      description
        "Указывает причину отказа при удалении подписки.";
    }
  }
}

/*
 * Уведомления
 */

notification replay-completed {
  sn:subscription-state-notification;
  if-feature "replay";
  description
    "Уведомление, указывающее, что все воспроизводимые уведомления
    были переданы.";
  leaf id {
    type subscription-id;
    mandatory true;
    description
      "Указывает подписку для воздействия.";
  }
}

```



```

notification subscription-completed {
  sn:subscription-state-notification;
  if-feature "configured";
  description
    "Указывает, что для подписки завершена передача записей о событиях
    по значению stop-time.";

  leaf id {
    type subscription-id;
    mandatory true;
    description
      "Указывает аккуратно завершённую подписку.";
  }
}

notification subscription-modified {
  sn:subscription-state-notification;
  description
    "Указывает, что подписка была изменена. Передаваемые после этого
    уведомления будут соответствовать новым условиям подписки. Для
    полноты это уведомление о смене статуса подписки включает
    исходные и изменённые параметры подписки.";
  leaf id {
    type subscription-id;
    mandatory true;
    description
      "Указывает подписку для воздействия.";
  }
  uses subscription-policy {
    refine "target/stream/stream-filter/within-subscription" {
      description
        "Фильтр для подписки. Если задан лист stream-filter-name,
        фильтр для подписки берётся из контейнера filters, иначе
        он задаётся явно как часть подписки.";
    }
  }
}

notification subscription-resumed {
  sn:subscription-state-notification;
  description
    "Указывает, что ранее приостановленная подписка возобновлена и
    уведомления снова будут передаваться. Кроме того,
    subscription-resumed указывает, что параметры подписки не
    изменились с момента передачи последней записи о событии.";
  leaf id {
    type subscription-id;
    mandatory true;
    description
      "Указывает подписку для воздействия.";
  }
}

notification subscription-started {
  sn:subscription-state-notification;
  if-feature "configured";
  description
    "Указывает, что подписка активирована и уведомления будут
    передаваться.";
  leaf id {
    type subscription-id;
    mandatory true;
    description
      "Указывает подписку для воздействия.";
  }
  uses subscription-policy {
    refine "target/stream/replay-start-time" {
      description
        "Указывает время, используемое для воспроизведения потока
        записанных событий. Значение совпадает с большим (поздним) из
        времени события в предыдущей записи, переданной получателю,
        replay-log-creation-time, replay-log-aged-time или
        последней загрузки издателя.";
    }
    refine "target/stream/stream-filter/within-subscription" {
      description
        "Фильтр, применяемый к подписке. Если задано значение
        stream-filter-name, фильтр подписки берётся из контейнера
        filters, иначе указывается явно как часть подписки.";
    }
    augment "target/stream" {
      description
        "Добавляет параметры, связанные с уведомлением
        subscription-started.";
      leaf replay-previous-event-time {
        when '../replay-start-time';
      }
    }
  }
}

```

```

    if-feature "replay";
    type yang:date-and-time;
    description
      "При наличии в буфере воспроизведения хотя бы одного события
      до replay-start-time этот лист указывает время события,
      сгенерированного непосредственно перед replay-start-time.

      Если получатель ранее принимал записи событий для этой
      настраиваемой подписки, он может сравнить это время с
      ранее полученной записью о событии. При несовпадении
      (возможно из-за перезагрузки), может быть инициализировано
      динамическое воспроизведение отсутствующих записей.";
  }
}
}

notification subscription-suspended {
  sn:subscription-state-notification;
  description
    "Указывает приостановку подписки издателем. Уведомления не будут
    передаваться до возобновления подписки. Это уведомление нужно
    передавать только получателям подписки, оно не является
    уведомлением общего назначения.";
  leaf id {
    type subscription-id;
    mandatory true;
    description
      "Указывает подписку для воздействия.";
  }
  leaf reason {
    type identityref {
      base subscription-suspended-reason;
    }
    mandatory true;
    description
      "Указывает условия, вызвавшие приостановку подписки.";
  }
}

notification subscription-terminated {
  sn:subscription-state-notification;
  description
    "Указывает прерывание подписки.";
  leaf id {
    type subscription-id;
    mandatory true;
    description
      "Указывает подписку для воздействия.";
  }
  leaf reason {
    type identityref {
      base subscription-terminated-reason;
    }
    mandatory true;
    description
      "Указывает условия, вызвавшие прерывание подписки.";
  }
}

/*
 * Узлы данных
 */
container streams {
  config false;
  description
    "Сведения из встроенного потока событий от издателя.";
  list stream {
    key "name";
    description
      "Указывает встроенные потоки событий, поддерживаемые издателем";
    leaf name {
      type string;
      description
        "Дескриптор предоставляемого системой потока событий,
        состоящего из набора последовательных записей, каждая из
        которых характеризуется своим доменом и семантикой.";
    }
    leaf description {
      type string;
      description
        "Описание потока событий, включающее такие сведения, как тип
        записей о событиях, доступных в этом потоке событий.";
    }
  }
  leaf replay-support {
    if-feature "replay";

```

```

    type empty;
    description
      "Указывает, что в потоке событий есть запись для
      воспроизведения.";
  }
  leaf replay-log-creation-time {
    when '../replay-support';
    if-feature "replay";
    type yang:date-and-time;
    mandatory true;
    description
      "Временная метка создания журнальной записи, применяемая для
      поддержки функции воспроизведения в данном потоке событий.
      Это время может быть раньше наиболее ранней из доступных в
      журнале записей. Объект обновляется при сбросе журнала.";
  }
  leaf replay-log-aged-time {
    when '../replay-support';
    if-feature "replay";
    type yang:date-and-time;
    description
      "Временная метка последней записи о событии, удалённой из
      журнала. Метка указывает, как далеко распространяется журнал
      в прошлое, если журнал не достигает прошлого времени
      replay-log-creation-time. Этот объект ДОЛЖЕН присутствовать,
      если поддерживается воспроизведение и какая-либо запись
      удалена из журнала.";
  }
}
}
}
container filters {
  description
    "Список настраиваемых фильтров, которые применимы к подписке. Это
    позволяет многократно применять заданные сложные фильтры.";
  list stream-filter {
    key "name";
    description
      "Список заданных фильтров, применимых к подпискам.";
    leaf name {
      type string;
      description
        "Имя для указания фильтра.";
    }
    uses stream-filter-elements;
  }
}
}
container subscriptions {
  description
    "Список активных в данный момент подписок, служащий для управления
    и мониторинга подписок. Это включает подписки, организованные
    через примитивы RPC и организованные путём настройки.";

  list subscription {
    key "id";
    description
      "Идентификатор и конкретные параметры подписки. Подписки из
      этого списка могли быть созданы с использованием канала
      управления, RPC или настройки конфигурации.

      Если для удаления подписки применяется RPC kill-subscription
      или операции настройки, сообщение subscription-terminated
      передаётся всем активным и приостановленным получателям.";
    leaf id {
      type subscription-id;
      description
        "Идентификатор подписки, уникальный у издателя.";
    }
    uses subscription-policy {
      refine "target/stream/stream" {
        description
          "Указывает поток событий, рассматриваемый для этой подписки.
          Если поток событий удалён или не может быть указан активной
          подпиской, передаётся уведомление subscription-terminated с
          указанием причины stream-unavailable. Если настраиваемая
          подписка указывает отсутствующий поток событий, подписка
          переводится в состояние invalid.";
      }
      refine "transport" {
        description
          "Для настраиваемой подписки этот лист задаёт транспорт,
          применяемый для доставки сообщений, предназначенных всем
          получателям этой подписки. Этот объект обязателен для
          подписок в хранилище данных конфигурации. Объект
          (1) необязателен для динамических подписок в хранилище
          данных рабочего состояния и его (2) не следует применять
          для других типов динамической подписки.";
      }
    }
  }
}

```

```
}
augment "target/stream" {
  description
    "Позволяет добавлять объекты в настраиваемую подписку.";
  leaf configured-replay {
    if-feature "configured";
    if-feature "replay";
    type empty;
    description
      "Присутствие этого листа указывает, что воспроизведение
      для настраиваемой подписки следует начинать с указанного
      момента в журнале событий или с момента загрузки издателя
      (что позднее).";
  }
}
}
choice notification-message-origin {
  if-feature "configured";
  description
    "Указывает выходной интерфейс издателя для отправки
    уведомлений.";
  case interface-originated {
    description
      "Уведомления предназначены для передачи через указанный
      интерфейс издателя.";
    leaf source-interface {
      if-feature "interface-designation";
      type if:interface-ref;
      description
        "Указывает интерфейс для передачи уведомлений.";
    }
  }
  case address-originated {
    description
      "Уведомления от издателя должны передаваться с конкретного
      адреса источника и/или контекста маршрутизации.";
    leaf source-vrf {
      if-feature "supports-vrf";
      type leafref {
        path "/ni:network-instances/ni:network-instance/ni:name";
      }
      description
        "VRF для отправки уведомлений от издателя.";
    }
    leaf source-address {
      type inet:ip-address-no-zone;
      description
        "Адрес источника в уведомлениях. Если VRF существует, но
        этого объекта нет, ДОЛЖЕН применяться заданный по умолчанию
        адрес издателя для этого экземпляра VRF.";
    }
  }
}
}
leaf configured-subscription-state {
  if-feature "configured";
  type enumeration {
    enum valid {
      value 1;
      description
        "Эта подписка поддерживается с её текущими параметрами.";
    }
    enum invalid {
      value 2;
      description
        "Эта подписка не поддерживается с текущими параметрами.";
    }
    enum concluded {
      value 3;
      description
        "Подписка неактивна, поскольку наступило время останова.
        Больше нет подписчиков со статусом active или suspended,
        но подписка ещё не удалена из конфигурации.";
    }
  }
}
config false;
description
  "Наличие этого листа указывает, что подписка организована
  настройкой, а не через канал управления или RPC. Значение
  указывает статус подписки, установленный издателем.";
}
container receivers {
  description
    "Набор получателей для подписки.";
  list receiver {
    key "name";
  }
}
```

```

min-elements 1;
description
  "Хост, считающийся получателем уведомлений в подписке. Для
  настраиваемой подписки транспортные параметры (или
  leafref для этих параметров) могут быть дополнены
  конкретным получателем в этом списке.";
leaf name {
  type string;
  description
    "Указывает уникального получателя для подписки.";
}

leaf sent-event-records {
  type yang:zero-based-counter64;
  config false;
  description
    "Число записей о событиях, переданных получателю. Счётчик
    инициализируется при организации динамической подписки
    или переходе настроенного получателя в состояние valid.";
}

leaf excluded-event-records {
  type yang:zero-based-counter64;
  config false;
  description
    "Число записей о событиях, явно удалённых фильтром событий
    потока или контролем доступа, чтобы они не были переданы
    получателю. Счётчик сбрасывается в 0 при каждой
    инициализации sent-event-records.";
}

leaf state {
  type enumeration {
    enum active {
      value 1;
      description
        "Указывает получателя всех подходящих сообщений для
        подписки.";
    }
    enum suspended {
      value 2;
      description
        "Получателю установлен статус suspended, поскольку
        издатель в данное время не может предоставить
        сообщения для подписки.";
    }
    enum connecting {
      value 3;
      if-feature "configured";
      description
        "Подписка настроена, но перед отправкой уведомлений
        нужно получить уведомление subscription-started о
        смене состояния подписки.

        Если было вызвано действие reset для получателя
        активной настраиваемой подписки, состояние должно
        быть изменено на connecting.";
    }
    enum disconnected {
      value 4;
      if-feature "configured";
      description
        "Отказ при отправке получателю уведомления
        subscription-started. Дополнительные попытки
        соединения в настоящее время не предпринимаются.";
    }
  }
}

config false;
mandatory true;
description
  "Состояние подписки с точки зрения конкретного получателя.
  С помощью этих сведений можно определить, генерирует ли
  издатель уведомления для этого получателя.";
}

action reset {
  if-feature "configured";
  description
    "Позволяет сбросить состояние для получателя настраиваемой
    подписки в connecting. Это даёт возможность заново
    инициализировать соединение.";
  output {
    leaf time {
      type yang:date-and-time;
      mandatory true;
      description
        "Время, когда издатель вернул для получателя состояние
        connecting.";
    }
  }
}

```

```
}  
  }  
    }  
  }  
}  
<CODE ENDS>
```

## 5. Взаимодействие с IANA

Агентство IANA зарегистрировало URI в субреестре ns реестра IETF XML Registry [RFC3688], доступном по ссылке <<https://www.iana.org/assignments/xml-registry>>, в соответствии с [RFC3688]

```
URI: urn:iETF:params:xml:ns:yang:ietf-subscribed-notifications
```

```
Registrant Contact: The NETCONF WG of the IETF.
```

```
XML: N/A; запрошенный URI является пространством имён XML.
```

Агентство IANA зарегистрировало модуль YANG в реестре YANG Module Names [RFC6020], доступном по ссылке <<https://www.iana.org/assignments/yang-parameters>>, в соответствии с [RFC6020]

```
Name: iETF-subscribed-notifications
```

```
Namespace: urn:iETF:params:xml:ns:yang:ietf-subscribed-notifications
```

```
Prefix: sn
```

```
Reference: RFC 8639
```

## 6. Вопросы реализации

Для поддержки реализаций со статическими и настраиваемыми подписками рекомендуется разделить домен идентификаторов подписки на статическую и динамическую часть. Это предотвратит конфликты, связанные с указанием для настраиваемой подписки значения `subscription-id`, уже выделенного динамически. Разумным решением является использование нижней половины пространства целочисленных идентификаторов для выделения внешними объектами (настраиваемая подписка). Верхняя половина пространства идентификаторов динамически распределяется издателем.

Если подписка не способна поместить серию отфильтрованных записей о событиях в пригодные для передачи сообщения, отправку для получателя следует приостановить с указанием причины `unsupported-volume`.

Для настраиваемых подписок операции выполняются над набором получателей с использованием идентификаторов подписки в качестве дескрипторов набора. Но для потоковых уведомлений сообщения о смене состояния подписки являются индивидуальными для получателя. В соответствии с этой спецификацией издатель не доставляет получателю каких-либо сведений о наличии других получателей. Но если оператор сети хочет, чтобы получатели сопоставляли результаты, полезно использовать идентификаторы подписки для такого сопоставления. Отметим, что различия в правах доступа разных получателей могут приводить к получению каждым из них своего набора записей о событиях.

Для настраиваемых подписок с воспроизведением получатели защищаются от дубликатов событий, вытолкнутых после перезагрузки издателя. Однако получатель может захотеть видеть записи о событиях, которые не удалось доставить непосредственно перед перезагрузкой. Доставка таких записей возможна путём указания `<eventTime>` [RFC5277] из последней записи о событии, полученной до уведомления `subscription-started` о смене состояния подписки. С помощью этого `<eventTime>` и `replay-start-time` из уведомления `subscription-started` можно организовать независимую динамическую подписку, где будут переданы записи, которые были созданы, но не отправлены получателю.

## 7. Транспортные требования

В этом разделе приведены требования к транспортировке уведомлений по подписке в поддержку описанного решения.

Транспорт, выбранный подписчиком для доступа к издателю, **должен** поддерживать множество запросов `establish-subscription` в одной транспортной сессии.

Для настраиваемой и динамической подписки издатель **должен** проверять подлинность получателя с помощью того или иного механизма транспортного уровня до начала отправки каких-либо записей о событиях. Кроме того, получатель **должен** аутентифицировать издателя на транспортном уровне. Результатом является взаимная проверка подлинности.

Настоятельно рекомендуется защищённый транспорт. Помимо этого издатель **должен** проверить наличие у получателя достаточных прав для выполнения запрашиваемой функции применительно к вовлечённому содержимому.

Спецификация транспорта на основе этого документа может (но не обязана) требовать использования одного логического канала для RPC и записей о событиях. Однако записи событий и уведомления о смене статуса подписки **должны** передаваться в одной транспортной сессии для соблюдения порядка доставки.

Спецификация транспорта **должна** указывать поддерживаемое кодирование. Если транспорт настраиваемой подписки поддерживает разное кодирование, спецификация **должна** задавать принятый по умолчанию вариант.

Если подписчик включает лист `dscp` в запрос `establish-subscription`, он должен понимать и учитывать, как соответствующий код DSCP будет представлен в домене издателя.

Дополнительные требования будут зависеть от выбора транспорта для подписки. Пример требований приведён в [RFC8640].

## 8. 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 защищённый транспорт с обязательной поддержкой SSH<sup>1</sup> [RFC6242]. Нижним уровнем RESTCONF является HTTPS с обязательной поддержкой TLS [RFC5246].

Модель управления доступом NETCONF (Network Configuration Access Control Model или NACM) [RFC8341] обеспечивает способы ограничения доступа отдельным пользователям NETCONF или RESTCONF заданным подмножеством всех доступных операций и содержимого NETCONF или RESTCONF.

При настраиваемой подписке можно использовать один или несколько издателей для перегрузки получателя. Чтобы предотвратить это, **не следует** передавать уведомления получателям, не поддерживающим эту спецификацию. Получателю, не желающему видеть уведомления, нужно лишь прервать или отвергнуть транспортные сессии с издателем.

Когда получатель видит новое сообщение subscription-started для подписки, где он уже принимает уведомления, это может говорить о том, что злоумышленник выполнил какие-то действия, которые нарушили связность получателя. Для извлечения событий, потерянных в течение интервала отсутствия связности получателю **следует** извлечь записи, созданные с момента последнего полученного события. Это можно сделать с помощью динамической подписки на воспроизведение с теми же фильтрами (в предположении поддержки издателем функции replay).

Для динамических подписок требуется защита реализаций от злонамеренных и ошибочных подписчиков, которые могут передавать большое число запросов establish-subscription, потребляя ресурсы системы. Операторам **следует** выполнять мониторинг таких случаев и при обнаружении предпринимать действия по ограничению потребляемых ресурсов, например, приостанавливая или прерывая часть подписок или прерывая транспортные сессии, если применяется сеансовый транспорт.

Механизмы воспроизведения, описанные в параграфах 2.4.2.1 и 2.5.6, предоставляют доступ к истории записей о событиях. Модель управления доступом, защищающая эти записи, по своему устройству может разрешить доступ пользователям, которые не имели полномочий на момент сбора сведений.

Использование DNS для поиска имён настраиваемых пользователей может приводить к ситуациям, когда имя распознаётся не так, как ожидает издатель и получатель будет отличаться от предполагаемого.

Злоумышленник, способный заставить издателя использовать некорректное время, может вызвать воспроизведение сообщений путём установки времени в прошлом и создать риск потери сообщений, передвинув время в будущее.

Многие узлы данных в описанном здесь модуле YANG позволяют доступ для записи, создания, удаления (writable/creatable/deletable, т. е. config true, как установлено по умолчанию). Такие узлы могут содержать конфиденциальные сведения или быть уязвимыми в некоторых сетевых средах. Операции записи (например, edit-config) для таких узлов без подобающей защиты могут отрицательно влиять на работу сети. Ниже перечислены ветви и узлы данных с указанием возможных уязвимостей.

#### **Контейнер /filters**

##### **stream-subtree-filter, stream-xpath-filter**

Обновление фильтра может повысить сложность расчётов для всех применяющих фильтр подписчиков.

#### **Контейнер /subscriptions**

Ниже приведены соображения, относящиеся только к операциям настройки конфигурации настраиваемых подписок.

##### **configured-replay**

Может использоваться для передачи подписчику большого числа записей о событиях.

##### **dependency**

Может использоваться для принудительного размещения важных событий в очереди после малозначимых.

##### **dscp**

Некорректная маркировка может привести к передаче трафика с приоритетом выше гарантированного.

##### **id**

Может переопределить имеющуюся подписку, возможно настроенную другим объектом.

##### **name**

Добавление ключевой записи может использоваться для попытки передать трафик нежелательному получателю.

##### **replay-start-time**

Может использоваться для выталкивания большого числа записей журнала, что ведёт к истощению ресурсов.

##### **source-address**

Заданный адрес может не позволить передачу трафика нужному получателю.

##### **source-interface**

Заданный интерфейс может не позволить передачу трафика нужному получателю.

##### **source-vrf**

Может разместить подписку в виртуальной сети, где получатели не имеют права просматривать её содержимое.

##### **stop-time**

Может использоваться для прерывания содержимого в неподходящий момент.

##### **stream**

Можно организовать подписку на поток событий, содержимого которого не разрешено для целевых получателей.

##### **stream-filter-name**

Можно установить фильтр, не соответствующий потоку событий.

##### **stream-subtree-filter**

Сложный фильтр может потребовать больших вычислительных ресурсов для данной подписки.

##### **stream-xpath-filter**

Сложный фильтр может потребовать больших вычислительных ресурсов для данной подписки.

##### **weighting**

Указание большого веса может перегрузить очереди других подписчиков.

<sup>1</sup>Secure Shell - защищённая оболочка.

Некоторые из доступных для чтения узлов модуля YANG могут считаться конфиденциальными или уязвимыми в отдельных сетевых средах. Для таких узлов важно контролировать операции чтения (например, get, get-config, notification). Ниже перечислены поддеревья и узлы данных с указанием возможных уязвимостей.

### Контейнер /streams

#### name

Если контроль доступа не настроен должным образом, могут раскрываться внутренние сведения системы.

#### replay-support

Если контроль доступа не настроен должным образом, могут раскрываться журнальные записи.

### Контейнер /subscriptions

#### excluded-event-records

Этот лист может давать сведения о фильтруемых записях событий. Оператору сети следует иметь полномочия для доступа к таким сведениям. Раскрытие получателю сведений о числе отфильтрованных событий может приводить к утечке информации о наличии фильтров, которые могут применяться для этого получателя.

#### subscription

Для разных групп операторов могут организовываться различные наборы подписок. Следует организовать контроль доступа, позволяющий видеть лишь разрешённые подписки.

Некоторые из операций RPC в этом модуле YANG могут быть чувствительны или уязвимы в той или иной сетевой среде. Важно контролировать доступ к таким операциям. Ниже указаны такие операции.

### Все RPC

Если ошибочный или враждебный подписчик передаёт неожиданно большое число RPC, это может приводить к избыточному расходу системных ресурсов издателя просто для определения приемлемости подписки. В таких случаях взаимодействие с подписчиками **может** прерываться путём разрыва транспортных сессий.

### RPC delete-subscription

Нет особых соображений.

### RPC establish-subscription, modify-subscription

Подписчики могут перегрузить ресурсы издателя, поэтому издатель **должен** гарантировать наличие ресурсов для выполнения запроса, а при недостатке ресурсов **должен** отвергать запрос.

### RPC kill-subscription

Вызов RPC kill-subscription **должен** быть защищён, чтобы только соединения с административными правами могли применять этот вызов RPC.

## 9. Литература

### 9.1. 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>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", [RFC 2474](https://www.rfc-editor.org/info/rfc2474), DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [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>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](https://www.rfc-editor.org/info/rfc5246), DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", [RFC 5277](https://www.rfc-editor.org/info/rfc5277), DOI 10.17487/RFC5277, July 2008, <<https://www.rfc-editor.org/info/rfc5277>>.
- [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](https://www.rfc-editor.org/info/rfc6241), 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](https://www.rfc-editor.org/info/rfc6242), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](https://www.rfc-editor.org/info/rfc6991), 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](https://www.rfc-editor.org/info/rfc7950), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", [RFC 7951](https://www.rfc-editor.org/info/rfc7951), DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](https://www.rfc-editor.org/info/rfc8040), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, [RFC 8174](https://www.rfc-editor.org/info/rfc8174), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, [RFC 8341](https://www.rfc-editor.org/info/rfc8341), DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.



- [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>>.
- [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>>.
- [W3C.REC-xml-20081126] Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <<https://www.w3.org/TR/2008/REC-xml-20081126>>.
- [XPath] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", November 1999, <<https://www.w3.org/TR/1999/REC-xpath-19991116>>.

## 9.2. Дополнительная литература

- [RESTCONF-Notif] Voit, E., Rahman, R., Nilsen-Nygaard, E., Clemm, A., and A. Bierman, "Dynamic subscription to YANG Events and Datastores over RESTCONF", Work in Progress<sup>1</sup>, draft-ietf-netconf-restconf-notif-15, June 2019.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <<https://www.rfc-editor.org/info/rfc7049>>.
- [RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", [RFC 7540](#), DOI 10.17487/RFC7540, May 2015, <<https://www.rfc-editor.org/info/rfc7540>>.
- [RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", [RFC 7923](#), DOI 10.17487/RFC7923, June 2016, <<https://www.rfc-editor.org/info/rfc7923>>.
- [RFC8071] Watsen, K., "NETCONF Call Home and RESTCONF Call Home", RFC 8071, DOI 10.17487/RFC8071, February 2017, <<https://www.rfc-editor.org/info/rfc8071>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, [RFC 8259](#), DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [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>>.
- [RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", [RFC 8640](#), DOI 10.17487/RFC8640, September 2019, <<https://www.rfc-editor.org/info/rfc8640>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", [RFC 8641](#), DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.

## Приложение А. Пример дополнения настроенного транспорта

В этом приложении представлен ненормативный пример расширения модуля YANG из раздела 4 для встраивания параметров конфигурации, требуемых процессом организации транспортного соединения. Пример не содержит полной модели транспорта и показывает соединение через воображаемый тип транспорта foo. Дополнительные сведения о настройке транспортной связности для настраиваемой подписки приведены в параграфе 2.5.7.

Представленный пример модуля YANG содержит два основных элемента. Первым является идентификатор транспорта foo, который позволяет агенту настройки указать foo в качестве типа транспорта для подписки. Вторым является дополнение YANG case foo через узел /subscriptions/subscription/receivers/receiver, определённый в разделе 4. Это дополнение задаёт параметры настройки транспорта address и port, которые нужны для соединения с получателем.

```

module example-foo-subscribed-notifications {
  yang-version 1.1;
  namespace
    "urn:example:foo-subscribed-notifications";

  prefix fsn;

  import ietf-subscribed-notifications {
    prefix sn;
  }
  import ietf-inet-types {
    prefix inet;
  }

  description
    "Указывает foo как поддерживаемый тип транспорта для
    уведомлений по подписке.";

  identity foo {
    base sn:transport;
    description
      "Тип транспорта foo доступен для использования в качестве

```

<sup>1</sup>Опубликовано в [RFC 8650](#). Прим. перев.

```
    транспортного протокола подписки на уведомления.";
}

augment
  "/sn:subscriptions/sn:subscription/sn:receivers/sn:receiver" {
  when 'derived-from(..../transport, "fsn:foo")';
  description
    "Это дополнение делает доступными для получателя параметры
    транспорта foo.";
  leaf address {
    type inet:host;
    mandatory true;
    description
      "Задаёт адрес для сообщений, направляемых получателю.";
  }
  leaf port {
    type inet:port-number;
    mandatory true;
    description
      "Задаёт номер порта для сообщений, направляемых получателю.";
  }
}
}
```

Рисунок 21. Пример транспортного дополнения для фиктивного протокола foo.

Этот пример модуля YANG для транспорта foo не предназначен для реального развёртывания. Для реальных транспортных технологий нужно определить соответствующий модуль YANG.

## Благодарности

Спасибо Andy Bierman, Tim Jenkins, Martin Bjorklund, Kent Watsen, Balazs Lengyel, Robert Wilton, Sharon Chisholm, Hector Trevino, Susan Hares, Michael Scharf, Guangying Zheng за ценные комментарии, дискуссии и отзывы.

## Адреса авторов

**Eric Voit**

Cisco Systems

Email: [evoit@cisco.com](mailto:evoit@cisco.com)

**Alexander Clemm**

Futurewei

Email: [ludwig@clemm.org](mailto:ludwig@clemm.org)

**Alberto Gonzalez Prieto**

Microsoft

Email: [alberto.gonzalez@microsoft.com](mailto:alberto.gonzalez@microsoft.com)

**Einar Nilsen-Nygaard**

Cisco Systems

Email: [einar@n@cisco.com](mailto:einar@n@cisco.com)

**Ambika Prasad Tripathy**

Cisco Systems

Email: [ambtripa@cisco.com](mailto:ambtripa@cisco.com)

## Перевод на русский язык

Николай Малых

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