

NETCONF Event Notifications

Уведомления о событиях NETCONF

Статус документа

В этом документе описан проект стандартного протокола Internet, предложенного сообществу Internet, документ служит приглашением к дискуссии в целях развития предложенного протокола. Текущее состояние стандартизации протокола можно узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться свободно.

Аннотация

Этот документ задаёт механизмы асинхронной доставки уведомлений для протокола настройки сети (Network Configuration или NETCONF), являющиеся необязательным свойством протокола NETCONF. Документ определяет возможности и операции, требуемые для поддержки этой службы.

Оглавление

1. Введение.....	1
1.1. Определения терминов.....	2
1.2. Мотивация.....	2
1.3. Уведомления о событиях в NETCONF.....	2
2. Связанные с уведомлениями операции.....	3
2.1. Подписка на уведомления о событиях.....	3
2.1.1. <create-subscription>.....	3
2.1.1.1. Пример использования.....	3
2.2. Передача уведомлений о событиях.....	4
2.2.1. <notification>.....	4
2.3. Прерывание подписки.....	4
3. Концепции поддержки.....	4
3.1. Обмен возможностями.....	4
3.1.1. Идентификатор возможности.....	4
3.1.2. Пример возможности.....	4
3.2. Потоки событий.....	4
3.2.1. Определение потока событий.....	5
3.2.2. Формат содержимого потока событий.....	5
3.2.3. Принятый по умолчанию поток событий.....	5
3.2.4. Источники потоков событий.....	5
3.2.5. Обнаружение потоков событий.....	5
3.2.5.1. Извлечение имени с помощью операции <get>.....	5
3.2.5.2. Подписка на поток событий.....	6
3.2.5.2.1. Фильтрация содержимого потока событий.....	6
3.3. Воспроизведение уведомлений.....	6
3.3.1. Обзор.....	6
3.3.2. Организация подписки с воспроизведением.....	6
3.4. Схема управления уведомлениями.....	7
3.5. Данные подписки.....	8
3.6. Детали фильтрации.....	8
3.6.1. Фильтрация.....	8
3.7. Поток сообщений.....	9
4. Схема XML для уведомлений о событиях.....	9
5. Примеры фильтрации.....	11
5.1. Фильтр ветвей.....	12
5.2. Фильтры XPath.....	13
6. Способность чередования.....	13
6.1. Описание.....	13
6.2. Зависимости.....	13
6.3. Идентификатор возможности.....	13
6.4. Новые операции.....	13
6.5. Изменения имеющихся операций.....	13
7. Вопросы безопасности.....	13
8. Взаимодействие с IANA.....	14
9. Благодарности.....	14
10. Нормативные документы.....	14

1. Введение

Протокол [NETCONF] концептуально делится на 4 уровня, показанные на рисунке 1.

Уровень	Пример
Содержимое	Данные конфигурации
Операции	<get-config>, <edit-config>, <notification>
RPC	<rpc>, <rpc-reply>
Транспортный	ВЕЕР, SSH, SSL, консоль
протокол	

Рисунок 1.

Этот документ определяет механизмы, обеспечивающие асинхронную доставку уведомлений для протокола [NETCONF] и являющиеся необязательным свойством протокола NETCONF. Документ определяет возможности и операции, требуемые для поддержки этой службы.

1.1. Определения терминов

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

Element

Элемент [XML].

Subscription - подписка

Соглашение и метод для получения уведомлений о событиях через сеанс NETCONF, концепция, связанная с доставкой уведомлений (если они имеются для отправки), включающая адресата и выбор уведомлений и привязанная к сроку действия сессии.

Operation - операция

Операция протокола NETCONF [NETCONF]. В этом документе - операция NETCONF для поддержки уведомлений.

Event - событие

Что-то произошедшее, что может представлять интерес - изменение конфигурации, отказ, смена состояния, пересечение порога, внешний ввод в систему. Зачастую событие ведёт к асинхронному сообщению, иногда называемому уведомлением, которое передаётся заинтересованным сторонам для информирования о событии.

Replay - воспроизведение

Способность передать или повторить записанные уведомления по запросу. Эти уведомления передаются асинхронно. Функция реализуется сервером NETCONF и вызывается клиентом NETCONF.

Stream - поток

Набор уведомлений о событиях, соответствующих неким критериям пересылки, и доступный клиентам NETCONF для подписки.

Filter - фильтр

Параметр, указывающий подмножество событий, которые представляют интерес. Фильтр определяется одним или несколькими элементами фильтрации [NETCONF], каждый из которых задаёт честь единого фильтра.

1.2. Мотивация

Целью этой работы является обеспечение отправки асинхронных сообщений, соответствующих модели данных (содержимому) и модели безопасности, применяемым в реализации NETCONF. Работа нацелена на выполнение указанных ниже требований.

- В исходном выпуске следует поддерживать уведомления для операций настройки.
- Следует обеспечить возможность использования одной модели для уведомлений и операций настройки.
- Следует поддерживать разумные ограничения размера сообщений (не слишком короткие).
- Уведомления следует доставлять с помощью ориентированного на соединения механизма.
- Следует обеспечить механизм подписки на уведомления с учётом того, что сервер NETCONF не передаёт уведомлений до их запроса, а поток уведомлений инициирует клиент NETCONF.
- На сервере NETCONF следует поддерживать механизм фильтрации передаваемых уведомлений.
- Содержащихся в уведомлениях сведений должно быть достаточно для независимого от транспорта анализа. Иными словами, содержимое должно полностью описывать уведомление без привлечения данных протокола.
- Серверу следует поддерживать возможность воспроизведения записанных уведомлений.

1.3. Уведомления о событиях в NETCONF

Этот документ задаёт механизм, с помощью которого клиенты NETCONF указывают заинтересованность в получении уведомлений о событиях от сервера NETCONF путём организации подписки. Сервер NETCONF отвергает или воспринимает запрос и в случае принятия начинает передачу уведомлений о событиях клиенту NETCONF по мере возникновения событий в системе. Эти уведомления продолжаются до завершения сеанса NETCONF или прерывания подписки по иным причинам. Подписка на уведомления о событиях предоставляет клиенту NETCONF опции для указания интересующих событий, задаваемые в момент организации подписки. Организованную подписку нельзя изменить.

Сервер NETCONF **должен** воспринимать и обрабатывать операцию <close-session> даже при наличии активных подписок на уведомления. Сервер NETCONF **может** воспринимать и обрабатывать другие команды, иначе они будут отвергаться и сервер **должен** будет возвращать ошибку resource-denied. Поддержку обработки других команд сервер NETCONF анонсирует через возможность (свойство) :interleave.

2. Связанные с уведомлениями операции

2.1. Подписка на уведомления о событиях

Подписку на уведомления о событиях инициирует клиент NETCONF, а сервер NETCONF отвечает на запрос. Подписка привязана к одному потоку на весь срок своего действия. При организации подписки на уведомления о событиях указываются интересующие события.

2.1.1. <create-subscription>

Эта операция инициирует подписку на уведомления о событиях, которые будут асинхронно передаваться инициатору до прерывания подписки.

Параметры

Stream

Необязательный параметр <stream> указывает интересующий поток событий. При отсутствии параметра передаётся заданный по умолчанию поток событий NETCONF.

Filter

Необязательный параметр <filter> указывает интересующие события. Формат параметра совпадает с форматом параметров фильтров в протокольных операциях NETCONF. При отсутствии параметра передаются все уведомления, не заблокированные другими параметрами (см. параграф 3.6. Детали фильтрации).

Start Time

Параметр <startTime> служит для запуска функции воспроизведения и указывает момент, с которого нужно начинать воспроизведение. При отсутствии <startTime> воспроизведения не происходит. Указанное время не может быть больше (позже) текущего. Если <startTime> указывает время раньше первой сохраненной записи, воспроизведение начинается с первой доступной записи. Параметр использует тип dateTime и соответствует [RFC3339]. Реализации должны поддерживать указание часового пояса.

Stop Time

Необязательный параметр <stopTime> применяется с необязательной функцией воспроизведения для указания самых новых интересующих обновлений. При отсутствии <stopTime> уведомления передаются до прерывания подписки. Этот параметр должен использоваться с <startTime> и меть большее значение. Параметр <stopTime> может указывать будущее время. Параметр использует тип dateTime и соответствует [RFC3339]. Реализации должны поддерживать указание часового пояса.

Позитивный отклик

Если сервер NETCONF способен выполнить запрос, он передаёт элемент <ok>.

Негативный отклик

Включается элемент <grpc-errord> в <grpc-reply>, если запрос не может быть выполнен по какой-либо причине. Запрос на подписку будет приводит к отказу при наличии ошибок в синтаксисе фильтра или указании имени отсутствующего потока.

Если в запросе указан параметр <stopTime> без <startTime>, следует возвращать

```
Tag: missing-element
Error-type: protocol
Severity: error
Error-info: <bad-element>: startTime
Description: Запрошенный элемент отсутствует.
```

Если запрошена функция воспроизведения, а сервер NETCONF не поддерживает её, возвращается отклик

```
Tag: operation-failed
Error-type: protocol
Severity: error
Error-info: none
Description: Запрос не может быть выполнен, поскольку возникла ошибка, для которой нет другой причины.
```

Если указано значение <stopTime> раньше времени <startTime>, возвращается ошибка

```
Tag: bad-element
Error-type: protocol
Severity: error
Error-info: <bad-element>: stopTime
Description: Значение элемента некорректно, например, ошибочный тип, выход за пределы диапазона, несоответствие шаблону.
```

Если указано значение <startTime> больше текущего времени, возвращается ошибка

```
Tag: bad-element
Error-type: protocol
Severity: error
Error-info: <bad-element>: startTime
Description: Значение элемента некорректно, например, ошибочный тип, выход за пределы диапазона, несоответствие шаблону.
```

2.1.1.1. Пример использования

Ниже приведён пример организации подписки. Более сложные случаи представлены в разделе 5.

```
<netconf:rpc message-id="101"
```

```

xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0"
<create-subscription
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
</create-subscription>
</netconf:rpc>

```

2.2. Передача уведомлений о событиях

После организации подписки сервер NETCONF передаёт через соединение асинхронные уведомления об событиях.

2.2.1. <notification>

Уведомление о событии передаётся клиенту, инициировавшему команду <create-subscription>, асинхронно по интересующим событиям (т. е. соответствующим критериям фильтра). Уведомление о событии - это полный документ XML с корректным форматом. Отметим, что <notification> является не вызовом удалённой процедуры (Remote Procedure Call или RPC), а элементом верхнего уровня, указывающим одностороннее сообщение как уведомление.

Параметры

eventTime

Время события, указанное его источником. Параметр использует тип dateTime и соответствует [RFC3339]. Реализации должны поддерживать указание часового пояса.

Включается также зависимое от уведомления содержимое при его наличии, которое выходит за рамки этого документа.

Отклика на уведомление не передаётся.

2.3. Прерывание подписки

Закрывание подписки на уведомления о событиях может быть выполнено с помощью операции <close-session> из сеанса, где подписка была организована, путём прерывания сессии NETCONF (<kill-session>) или базовой транспортной сессии из другого сеанса. Если указано время остановки подписки, она будет завершаться по достижении этого момента. В этом случае сессия NETCONF может оставаться активной.

3. Концепции поддержки

3.1. Обмен возможностями

Возможность обрабатывать и передавать уведомления о событиях анонсируется при обмене возможностями между клиентом и сервером NETCONF.

3.1.1. Идентификатор возможности

```
urn:ietf:params:netconf:capability:notification:1.0
```

3.1.2. Пример возможности

```

<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:xml:ns:netconf:base:1.0
    </capability>
    <capability>
      urn:ietf:params:netconf:capability:startup:1.0
    </capability>
    <capability>
      urn:ietf:params:netconf:capability:notification:1.0
    </capability>
  </capabilities>
  <session-id>4</session-id>
</hello>

```

3.2. Поток событий

Поток событий определяется как набор уведомлений о событиях, соответствующих неким критериям пересылки.



Рисунок 2.

На рисунке 2 показан поток уведомлений и концепции, рассмотренные в этом документе. Рисунок не предписывает и не препятствует реализации. Компоненты системы (с1..сn) генерируют уведомления, которые передаются центральному обработчику для классификации и распространения. Этот обработчик проверяет каждое уведомление и сравнивает уведомления о событиях с набором определений потоков. При совпадении уведомление о событии включается в соответствующий поток (1..n). Уведомление может включаться в несколько потоков событий.

После получения сервером NETCONF внутреннего события из потока оно преобразуется сервером в подходящий код XML и элемент <notification> готов к передаче во все сессии NETCONF с подпиской на этот поток. После создания элемента <notification> сервер применяет правила контроля доступа. Если у сессии нет права получать <notification>, элемент исключается для этой сессии и обработка внутреннего события для этой сессии завершается.

Когда клиент NETCONF подписывается на данный поток событий, заданные пользователем фильтры (при наличии) применяются к потоку и соответствующие уведомления о событиях пересылаются серверу NETCONF для распространения этому клиенту NETCONF. Фильтр переносится от клиента на сервер при операции <create-subscription> и применяется к каждому элементу <notification> в потоке. Фильтрация описана в параграфе 3.6.

Может также поддерживаться служба записи (log) уведомлений и в этом случае центральный элемент записывает уведомления о событиях в журнал. Сервер NETCONF позднее может извлекать записанные уведомления для необязательной функции воспроизведения, описанной в параграфе 3.3. Воспроизведение уведомлений.

3.2.1. Определение потока событий

Потоки событий предопределены на управляемом устройстве. Настройка потоков событий выходит за рамки этого документа, однако предполагается, что такие потоки заранее организуются производителем (предварительная настройка), настраиваются пользователем (например, в конфигурации устройства) или задаются комбинацией этих вариантов. Производители могут разрешать настройку потоков событий по протоколу NETCONF (например, с помощью операции <edit-config> operation).

3.2.2. Формат содержимого потока событий

Содержимое всех потоков событий, доступных клиентам NETCONF (например, в уведомлениях от сервера NETCONF) должно кодироваться в XML.

3.2.3. Принятый по умолчанию поток событий

Реализация сервера NETCONF с поддержкой уведомлений должна поддерживать поток уведомлений о событиях NETCONF. Этот поток содержит уведомления NETCONF XML о событиях, поддерживаемые сервером NETCONF. Строка NETCONF используется при анонсировании поддержки потока событий в операции <get> для <streams> и операции <create-subscription>. Определение уведомлений о событиях и их содержимого, за исключением <eventTime>, выходит за рамки этого документа.

3.2.4. Источники потоков событий

За исключением принятого по умолчанию потока событий (NETCONF), спецификация дополнительных источников событий (например, SNMP¹, syslog) выходит за рамки этого документа. Реализация сервера NETCONF может использовать любой желаемый источник событий при организации поддерживаемых потоков событий.

3.2.5. Обнаружение потоков событий

Клиент NETCONF извлекает список поддерживаемых сервером NETCONF событий с помощью операции <get>.

3.2.5.1. Извлечение имени с помощью операции <get>

Список доступных потоков событий извлекается путём запроса ветви <streams> с помощью операции <get>. Доступные для запрашивающей сессии потоки событий возвращаются в отклике с элементами <name> и <description>, где <name> является обязательным элементом с уникальным в рамках сервера NETCONF именем. При отсутствии доступных потоков событий (с учётом заданных в операции <get> фильтров) возвращается пустой отклик.

Дополнительные сведения о потоке включают информацию о поддержке воспроизведения и, при наличии такой поддержки, - метку времени наиболее раннего уведомления, которое можно воспроизвести.

Ниже приведён пример, показывающий извлечение списка доступных событий с помощью операции <get>.

```
<rpc message-id="101"
  xmlns="urn:iETF:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <netconf xmlns="urn:iETF:params:xml:ns:netmod:notification">
        <streams/>
      </netconf>
    </filter>
  </get>
</rpc>
```

Сервер NETCONF возвращает список доступных для подписки потоков - NETCONF, SNMP, syslog-critical.

```
<rpc-reply message-id="101"
  xmlns="urn:iETF:params:xml:ns:netconf:base:1.0">
  <data>
    <netconf xmlns="urn:iETF:params:xml:ns:netmod:notification">
      <streams>
        <stream>
          <name>NETCONF</name>
          <description>Принятый по умолчанию поток событий NETCONF</description>
        </stream>
      </streams>
    </netconf>
  </data>
```

¹Simple Network Management Protocol - простой протокол управления сетью.

```

    <replaySupport>true</replaySupport>
    <replayLogCreationTime>
      2007-07-08T00:00:00Z
    </replayLogCreationTime>
  </stream>
</stream>
  <name>SNMP</name>
  <description>Уведомления SNMP</description>
  <replaySupport>>false</replaySupport>
</stream>
<stream>
  <name>syslog-critical</name>
  <description>Критическая и высокая важность</description>
  <replaySupport>true</replaySupport>
  <replayLogCreationTime>
    2007-07-01T00:00:00Z
  </replayLogCreationTime>
</stream>
</streams>
</netconf>
</data>
</rpc-reply>

```

3.2.5.2. Подписка на поток событий

Клиент NETCONF может запросить у сервера NETCONF список доступных потоков событий для этой сессии и затем подать запрос `<create-subscription>` с именем нужного потока. Запрос `<create-subscription>` без указания имени организует подписку на принятый по умолчанию поток событий NETCONF.

3.2.5.2.1. Фильтрация содержимого потока событий

Набор уведомлений, доставляемых в потоке событий, можно уточнить с помощью заданного пользователем фильтра при организации подписки (`<create-subscription>`). Это временный фильтр, связанный с подпиской и не меняющий конфигурацию потока событий. Элемент фильтрации применяется к содержимому `<notification>`, а не к самому элементу. Примеры представлены в разделе 5. Возможна фильтрация по ветвям и выражениям XPATH.

Поддержка XPATH для свойства `Notification` анонсируется как часть обычного анонса возможностей XPATH. Если поддержка XPATH анонсируется с возможностями XPATH, будет поддерживаться и фильтрация XPATH. Если эта возможность не анонсируется, фильтрация уведомлений по XPATH не будет поддерживаться.

3.3. Воспроизведение уведомлений

3.3.1. Обзор

Воспроизведение даёт возможность подписки на события из недавнего прошлого и в некоторых случаях передавать все записанные уведомления при первом обращении конкретного клиента NETCONF. Эти уведомления передаются обычным способом.

Воспроизведение уведомлений задаётся включением в запрос подписки параметра `<startTime>`, указывающего время первого воспроизводимого уведомления. Можно задать также время окончания параметром `<stopTime>`. Если этот параметр не указан, уведомления передаются до прерывания подписки.

Для потока уведомлений с поддержкой воспроизведения не предполагается неограниченное хранение уведомлений для выполнения любого запроса на воспроизведение. Клиенты могут запросить `<replayLogCreationTime>` и `<replayLogAgedTime>`, чтобы узнать доступные для воспроизведения временные рамки.

Фактическое число доступных для воспроизведения уведомлений в данный момент времени зависит от реализации сервера NETCONF. Параметры управления для этого аспекта функции воспроизведения выходят за рамки документа.

Воспроизведение зависит от потока уведомлений, способного поддерживать ту или иную форму записи (log), хотя и не накладывает ограничений на размер или форму журнала, а также его местоположение на устройстве. Узнать о поддержке воспроизведения для потока можно с помощью операции `<get>` для элемента `<streams>` в схеме `the Notification Management` и просмотра полученного объекта `<replaySupport>`. Эта схема также включает элемент `<replayLogCreationTime>`, указывающий самую раннюю из доступных в журнале записей.

3.3.2. Организация подписки с воспроизведением

Эта функция использует необязательные параметры команды `<create-subscription>`, называемые `<startTime>` и `<stopTime>`. Параметр `<startTime>` указывает начальную дату и время интересующих событий для воспроизведения и указывает, что подписка будет обеспечивать воспроизведение уведомлений. Событие до указанного момента не будут воспроизводиться. Параметр `<stopTime>` задаёт дату и время завершения интересующих событий для воспроизведения уведомлений. При отсутствии параметра воспроизведение будет продолжаться до прерывания подписки.

Отметим, что параметры `<startTime>` и `<stopTime>` связаны со временем генерации событий их источником.

Уведомление `<replayComplete>` передаётся для указания передачи всех воспроизводимых уведомлений и применять его в иных целях недопустимо. Если для подписки не задано время остановки, сессия становится обычной сессией NETCONF. После этого сервер NETCONF будет воспринимать операции `<rpc>`, даже если ранее он не принимал их по причине отсутствия чередования. В случае подписки без указания времени остановки после отправки уведомления `<replayComplete>` можно ожидать, что будут переданы любые уведомления, созданные после начала подписки, а затем уведомления будут передаваться по мере их появления в системе.

Уведомления `<replayComplete>` и `<notificationComplete>` не фильтруются и будут передаваться всегда для подписки с воспроизведением при указании параметров `<startTime>` и `<stopTime>`, соответственно.

3.4. Схема управления уведомлениями

Эта схема служит для получения сведений о поддерживаемых в системе потоках уведомлений. Она содержит определения уведомлений <replayComplete> и <notificationComplete>, передаваемых для индикации того, что все уведомления при воспроизведении переданы и подписка прервана, соответственно.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:ncEvent="urn:ietf:params:xml:ns:netconf:notification:1.0"
  xmlns:manageEvent="urn:ietf:params:xml:ns:netmod:notification"
  targetNamespace="urn:ietf:params:xml:ns:netmod:notification"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xml:lang="ru" version="1.0">
  <xs:annotation>
    <xs:documentation xml:lang="ru">
      Схема, которая может служить для получения сведений о
      текущих потоках событий. Она также включает уведомления
      replayComplete и notificationComplete.
    </xs:documentation>
  </xs:annotation>

  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xs:import namespace="urn:ietf:params:xml:ns:netconf:base:1.0"
    schemaLocation="netconf.xsd"/>
  <xs:import namespace=
    "urn:ietf:params:xml:ns:netconf:notification:1.0"
    schemaLocation="notification.xsd"/>

  <xs:element name="netconf" type="manageEvent:Netconf"/>

  <xs:complexType name="Netconf">
    <xs:sequence>
      <xs:element name="streams" >
        <xs:annotation>
          <xs:documentation>
            Список потоков событий, поддерживаемых системой.
            По запросу возвращается набор потоков в соответствии
            с правами пользователя.
          </xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence minOccurs="1" maxOccurs="unbounded">
            <xs:element name="stream">
              <xs:annotation>
                <xs:documentation>
                  Имя и описание потока, а также иные сведения.
                </xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="name"
                    type="ncEvent:streamNameType">
                    <xs:annotation>
                      <xs:documentation>
                        Имя потока событий. Для заданного по
                        умолчанию потока событий NETCONF должно
                        использоваться значение NETCONF.
                      </xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="description"
                    type="xs:string">
                    <xs:annotation>
                      <xs:documentation>
                        Описание потока событий с указанием типа
                        событий, передаваемых в этом потоке.
                      </xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="replaySupport"
                    type="xs:boolean">
                    <xs:annotation>
                      <xs:documentation>
                        Указывает, поддерживается ли воспроизведение
                        для этого потока.
                      </xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="replayLogCreationTime"
                    type="xs:dateTime" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>
                        Временная метка создания журнала для функции
```

воспроизведения в этом потоке. Отметим, что это может быть раньше наиболее давнего уведомления в журнале. Объект обновляется при сбросе журнала по какой-либо причине. Объект ДОЛЖЕН присутствовать, если поддерживается воспроизведение.

```

    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="replayLogAgedTime"
  type="xs:dateTime" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      Временная метка последнего устаревшего
      уведомления в журнале. Объект ДОЛЖЕН
      присутствовать, если поддерживается
      воспроизведение и объекты удалялись из
      журнала в результате старения.
    </xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="ReplayCompleteNotificationType">
  <xs:complexContent>
    <xs:extension base="ncEvent:NotificationContentType"/>
  </xs:complexContent>
</xs:complexType>

<xs:element name="replayComplete"
  type="manageEvent:ReplayCompleteNotificationType"
  substitutionGroup="ncEvent:notificationContent">
  <xs:annotation>
    <xs:documentation>
      Это уведомление передаётся для индикации завершения
      воспроизводимой части подписки.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:complexType name="NotificationCompleteNotificationType">
  <xs:complexContent>
    <xs:extension base="ncEvent:NotificationContentType"/>
  </xs:complexContent>
</xs:complexType>

<xs:element name="notificationComplete"
  type="manageEvent:NotificationCompleteNotificationType"
  substitutionGroup="ncEvent:notificationContent">
  <xs:annotation>
    <xs:documentation>
      Это уведомление передаётся для индикации завершения
      подписки на уведомления. Оно передаётся, если при
      организации подписки был задан параметр stopTime.
    </xs:documentation>
  </xs:annotation>
</xs:element>

</xs:schema>

```

3.5. Данные подписки

Подписки представляют меняющиеся сведения о состоянии и срок их действия определяет сессия или параметр <stopTime>.

3.6. Детали фильтрации

Если задан элемент фильтра для поиска определённых данных и таких данных нет в конкретном уведомлении о событии, уведомление будет отфильтровано. Например, если задана проверка severity=critical в уведомлении о событии конфигурации, где это поле не поддерживается, уведомление не пройдёт через фильтр.

Для фильтрации ветвей (subtree) совпадение означает непустой набор узлов, для фильтров XPath следует применять механизмы [XPATN] для преобразования возвращённого значения в логическое (boolean).

3.6.1. Фильтрация

Фильтрация задаётся явно при организации подписки с помощью параметра filter. Фильтр существует лишь как параметр подписки.

3.7. Поток сообщений

На рисунке 3 показан поток сообщений между клиентом (C) и сервером (S) NETCONF при организации подписки и в начале потока уведомлений. Подписка задаёт <startTime>, поэтому сервер начинает воспроизведение записанных уведомлений. Возможен обмен несколькими запросами-откликами в процессе организации подписки, но на рисунке это не показано.

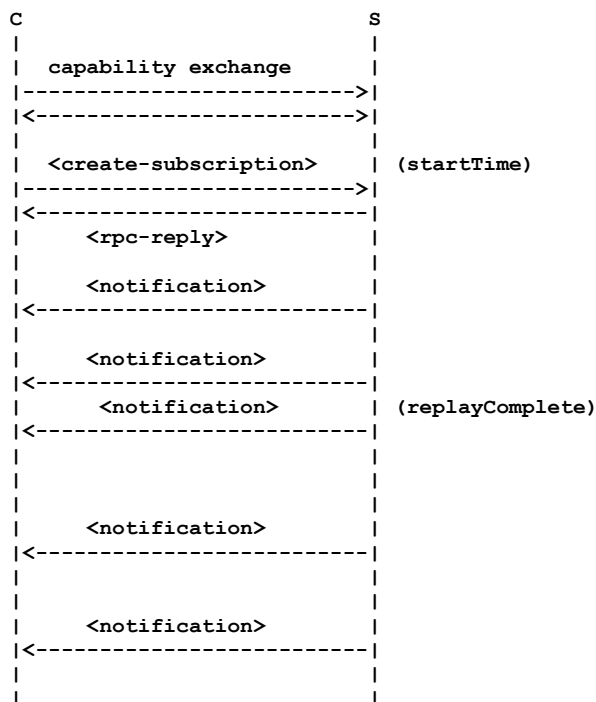


Рисунок 3.

На рисунке 4 показан обмен сообщениями между клиентом (C) и сервером (S) NETCONF при организации подписки и в начале потока уведомлений. Подписка задаёт параметры <startTime> и <stopTime>, поэтому начинается воспроизведение записанных уведомлений с последующим переходом сессии в обычный режим NETCONF запрос-отклик после передачи уведомлений <replayComplete> и <notificationComplete>, когда становится возможной обычная обработка запросов <rpc>. Возможен обмен множеством пар rpc/rpc-reply до организации подписки, но на рисунке это не показано.

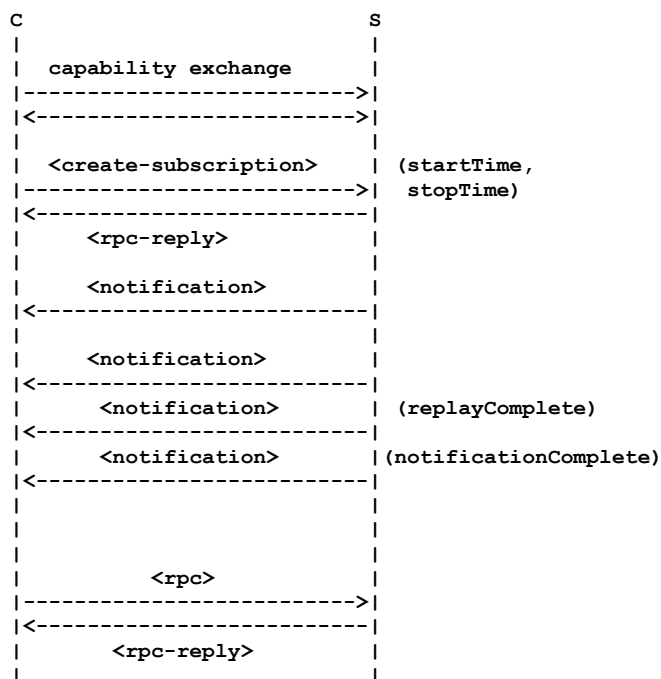


Рисунок 4.

4. Схема XML для уведомлений о событиях

Ниже показана схема [XMLSchema], определяющая уведомления о событиях NETCONF.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0"
  targetNamespace=
    "urn:ietf:params:xml:ns:netconf:notification:1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xml:lang="ru">
  
```

```

<!-- Импорт стандартных определений XML -->

<xs:import namespace="http://www.w3.org/XML/1998/namespace"
  schemaLocation="http://www.w3.org/2001/xml.xsd">
  <xs:annotation>
    <xs:documentation>
      Импорт обращается к группам атрибутов xml: для получения
      xml:lang, объявленного в элементе error-message.
    </xs:documentation>
  </xs:annotation>
</xs:import>

<!-- Импорт базовых определений netconf -->
<xs:import namespace="urn:ietf:params:xml:ns:netconf:base:1.0"
  schemaLocation="netconf.xsd"/>

<!-- ***** Симметричные операции *****-->

<!-- Операция <create-subscription> -->

<xs:complexType name="createSubscriptionType">
  <xs:complexContent>
    <xs:extension base="netconf:rpcOperationType">
      <xs:sequence>
        <xs:element name="stream"
          type="streamNameType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              Необязательный параметр для указания
              потока интересующий событий. По умолчанию
              передаётся поток событий NETCONF.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="filter"
          type="netconf:filterInlineType"
          minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              Необязательный параметр для указания
              набора интересующих событий. Формат
              параметра совпадает с форматом
              фильтров операций протокола NETCONF.
              По умолчанию передаются все события,
              не заблокированные другими
              параметрами.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="startTime" type="xs:dateTime"
          minOccurs="0" >
          <xs:annotation>
            <xs:documentation>
              Параметр для иницирования функции
              воспроизведения, указывающий время
              начала. Отсутствие параметра указывает,
              что воспроизведение не применяется.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="stopTime" type="xs:dateTime"
          minOccurs="0" >
          <xs:annotation>
            <xs:documentation>
              Необязательный параметр для применения с
              функцией воспроизведения, для указания
              интереса к уведомлениям до указанного
              момента. При отсутствии параметра
              уведомления передаются до прерывания
              подписки. Параметр должен использоваться
              вместе с startTime.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="streamNameType">
  <xs:annotation>
    <xs:documentation>
      Имя потока событий.
    </xs:documentation>
  </xs:annotation>
</xs:simpleType>
<xs:restriction base="xs:string"/>

```

```

</xs:simpleType>

<xs:element name="create-subscription"
  type="createSubscriptionType"
  substitutionGroup="netconf:rpcOperation">
  <xs:annotation>
    <xs:documentation>
      Команда для организации подписки на уведомления. Она
      принимает имя потока уведомления и фильтр для указания
      содержимого подписки. Имеются также временные параметры
      startTime и stopTime, служащие для выбора интервала
      интересующих событий при воспроизведении уведомлений.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<!-- ***** Односторонние операции *****-->

<!-- Операция <Notification> -->
<xs:complexType name="NotificationContentType"/>

<xs:element name="notificationContent"
  type="NotificationContentType" abstract="true"/>

<xs:complexType name="NotificationType">
  <xs:sequence>
    <xs:element name="eventTime" type="xs:dateTime">
      <xs:annotation>
        <xs:documentation>
          Время события, указанное его источником.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="notificationContent"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="notification" type="NotificationType"/>
</xs:schema>

```

5. Примеры фильтрации

В этом разделе даны примеры различных методов фильтрации содержимого подписки на уведомления. Для примеров нужны некоторые предположения о содержимом уведомлений. Здесь предполагается, что определение схемы включает на верхнем уровне элемент <event>, включающий класс события (например, отказ, статус, конфигурация), оповещающих объект и важность или рабочее состояние. Для генерации примеров использовалась приведенная ниже фиктивная схема.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://example.com/event/1.0"
  xmlns="http://example.com/event/1.0"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ncEvent="urn:ietf:params:xml:ns:netconf:notification:1.0">

  <xs:import namespace=
    "urn:ietf:params:xml:ns:netconf:notification:1.0"
    schemaLocation="notification.xsd"/>

  <xs:complexType name="eventType">
    <xs:complexContent>
      <xs:extension base="ncEvent:NotificationContentType">
        <xs:sequence>
          <xs:element name="eventClass" />
          <xs:element name="reportingEntity">
            <xs:complexType>
              <xs:sequence>
                <xs:any namespace="##any"
                  processContents="lax"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:choice>
            <xs:element name="severity"/>
            <xs:element name="operState"/>
          </xs:choice>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="event"
    type="eventType"
    substitutionGroup="ncEvent:notificationContent"/>

</xs:schema>

```

Приведённое выше фиктивное определение может приводить к показанному ниже списку уведомлений, используемых в примерах этого раздела section.

```
<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:01:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>fault</eventClass>
    <reportingEntity>
      <card>Ethernet0</card>
    </reportingEntity>
    <severity>major</severity>
  </event>
</notification>

<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:02:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>fault</eventClass>
    <reportingEntity>
      <card>Ethernet2</card>
    </reportingEntity>
    <severity>critical</severity>
  </event>
</notification>

<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:04:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>fault</eventClass>
    <reportingEntity>
      <card>ATM1</card>
    </reportingEntity>
    <severity>minor</severity>
  </event>
</notification>

<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:10:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>state</eventClass>
    <reportingEntity>
      <card>Ethernet0</card>
    </reportingEntity>
    <operState>enabled</operState>
  </event>
</notification>
```

5.1. Фильтр ветвей

Фильтрация по ветвям XML мало подходит для создания сложных фильтров, поскольку она поддерживает лишь сравнение на равенство и применение логических операторов ИЛИ (OR), например, в ветви event выбрать все уведомления с severity=critical, severity=major или severity=minor. Тем не менее, такие фильтры можно применять для определения простых фильтров пересылки уведомлений, как показано ниже.

Далее показано, как выбрать отказы с важностью critical, major или minor. Выражение для фильтра имеет форму ((fault & severity=critical) | (fault & severity=major) | (fault & severity=minor)).

```
<netconf:rpc netconf:message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <filter netconf:type="subtree">
      <event xmlns="http://example.com/event/1.0">
        <eventClass>fault</eventClass>
        <severity>critical</severity>
      </event>
      <event xmlns="http://example.com/event/1.0">
        <eventClass>fault</eventClass>
        <severity>major</severity>
      </event>
      <event xmlns="http://example.com/event/1.0">
        <eventClass>fault</eventClass>
        <severity>minor</severity>
      </event>
    </filter>
  </create-subscription>
</netconf:rpc>
```

Ниже показано, как выбрать состояние, настройку или отказ сетевого адаптера Ethernet0. Выражение для фильтра имеет вид (state | config | (fault & (card=Ethernet0))).

```
<netconf:rpc netconf:message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
```

```

xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<filter netconf:type="subtree">
  <event xmlns="http://example.com/event/1.0">
    <eventClass>state</eventClass>
  </event>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>config</eventClass>
  </event>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>fault</eventClass>
    <reportingEntity>
      <card>Ethernet0</card>
    </reportingEntity>
  </event>
</filter>
</create-subscription>
</netconf:rpc>

```

5.2. Фильтры XPath

Пример фильтра [XPath] показывает, как выбрать уведомления EventClass с важностью critical, major или minor. Выражение для фильтра имеет вид ((fault) & ((severity=critical) | (severity=major) | (severity = minor))).

```

<netconf:rpc netconf:message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <filter netconf:type="xpath"
      xmlns:ex="http://example.com/event/1.0"
      select="/ex:event[ex:eventClass='fault' and
        (ex:severity='minor' or ex:severity='major'
          or ex:severity='critical')]"/>
    </create-subscription>
  </netconf:rpc>

```

Далее показано, как выбрать состояние, настройку или отказ сетевого адаптера Ethernet0. Выражение для фильтра имеет вид (state | config | (fault & card=Ethernet0))

```

<netconf:rpc message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <filter netconf:type="xpath"
      xmlns:ex="http://example.com/event/1.0"
      select="/ex:event[
        (ex:eventClass='state' or ex:eventClass='config') or
        ((ex:eventClass='fault' and ex:card='Ethernet0'))]"/>
    </create-subscription>
  </netconf:rpc>

```

6. Способность чередования

6.1. Описание

Свойство :interleave указывает, что партнёр NETCONF способен чередовать подписку на уведомления с другими операциями NETCONF. Это означает, что сервер NETCONF **должен** принимать, обрабатывать и отвечать на запросы NETCONF в сессии, где активна подписка на уведомления. Такая возможность способствует расширяемости за счёт снижения общего числа сессий NETCONF, требуемых оператору или управляющему приложению.

6.2. Зависимости

Эта возможность зависит от поддержки уведомлений.

6.3. Идентификатор возможности

Возможность :interleave указывается строкой urn:ietf:params:netconf:capability:interleave:1.0

6.4. Новые операции

Нет.

6.5. Изменения имеющихся операций

Когда передаётся запрос <create-subscription> при наличии другой активной подписки в той же сессии, возвращается ошибка, показанная ниже.

```

Tag: operation-failed
Error-type: protocol
Severity: error
Error-info: none
Description: Запрос невозможно выполнить, поскольку операция
столкнулась в отказом по причинам, не охватываемым другими
условиями ошибок.

```

7. Вопросы безопасности

Соображения безопасности из [NETCONF] применимы к свойству уведомлений (Notification capability).

Модель контроля доступа и выбор транспорта будут оказывать существенное влияние на безопасность решения.

Элементы <notification> никогда не передаются до завершения организации транспортного уровня и уровня NETCONF, включая обмен возможностями, а также до определения и проверки подлинности менеджера.

Рекомендуется позаботиться о безопасности:

- вызовов <create-subscription>;
- операций <get> для доступных лишь для чтения узлов;
- содержимого <notification>.

Безопасное выполнение означает применение защищённого транспорта и гарантии наличия у пользователя прав на выполнение функций, запрашивающих определённую часть содержимого NETCONF. При получении вызова <get> для заданного этим документом содержимого клиентам следует предоставлять возможность просмотра лишь того содержимого, к которому им предоставлен доступ. Вызов <create-subscription> можно считать отложенной операцией <get> для содержимого, к которому могут получать доступ разные пользователи различными способами.

Разные права доступа отражаются в элементах <notification>, к которым разные пользователи имеют доступ.

Одна из возможных проблем безопасности связана с доставкой данных из потоков, не относящихся к NETCONF, таких как syslog и SNMP. Эти данные могут быть более (или менее) уязвимыми при транспортировке через NETCONF по сравнению с доставкой по обычно используемым для них протоколам, в зависимости от свидетельств (credential) защиты двух подсистем. Сервер NETCONF отвечает за контроль доступа к содержимому потока.

Содержимое уведомлений и имена потоков событий могут включать чувствительные сведения и следует позаботиться о том, чтобы они были доступны лишь уполномоченным пользователям. Серверу NETCONF **недопустимо** включать в уведомления какие-либо данные, к которым пользователю не предоставлен доступ.

Если подписка создана с параметром <stopTime>, сессия NETCONF будет возвращаться к обычному режиму NETCONF (запрос-отклик) по окончании воспроизведения. Клиенты отвечают за закрытие сессии, когда она становится ненужной.

Если враждебный или содержащий ошибки клиент NETCONF передаёт множество запросов <create-subscription>, они аккумулируются и могут истощать ресурсы системы. В таких ситуациях подписка может быть прервана путём завершения базовых сессий NETCONF с помощью операции <kill-session>.

8. Взаимодействие с IANA

Этот документ регистрирует три URI для пространства имён NETCONF XML в реестре IETF XML в соответствии с [RFC3688]. Отметим, что URN возможностей соответствуют параграфу 10.3 в [NETCONF].

Индекс	Идентификатор возможности
:notification	urn:iETF:params:netconf:capability:notification:1.0
:interleave	urn:iETF:params:netconf:capability:interleave:1.0

URI: urn:iETF:params:xml:ns:netmod:notification

URI: urn:iETF:params:xml:ns:netconf:notification:1.0

Registrant Contact: The IESG.

XML: N/A, запрошенный URI является пространством имён XML.

Кроме того, агентство IANA зарегистрировало схему XML, заданную в разделе 4.

9. Благодарности

Спасибо Gilbert Gagnon, Greg Wilbur, Kim Curran за их вклад на раннем этапе работы над документом. Редакторы признательны за вклад в работу на сессии редактирования в Ванкувере Orly Nicklass, James Balestriere, Yoshifumi Atarashi, Glenn Waters, Alexander Clemm, Dave Harrington, Dave Partain, Ray Atarashi, David Perkins, а также на сессии редактирования в Монреале - Balazs Lengyel, Phil Shafer, Rob Enns, Andy Bierman, Dan Romascanu, Bert Wijnen, Simon Leinen, Juergen Schoenwaelder, Hideki Okita, Vincent Cridlig, Martin Bjorklund, Olivier Festor, Radu State, Brian Trammell, William Chow. Спасибо Li Yan за многочисленные отзывы и Suresh Krishnan - за общий обзор документа.

10. Нормативные документы

[NETCONF] Enns, R., Ed., "NETCONF Configuration Protocol", [RFC 4741](#), December 2006.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](#), January 2004.

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", W3C XML, February 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>.

[XMLSchema] Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", W3C <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/structures.html>, October 2004.

[XPath] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", W3C <http://www.w3.org/TR/1999/REC-xpath-19991116>, November 1999.

Адреса авторов

Sharon Chisholm
Nortel
3500 Carling Ave
Nepean, Ontario K2H 8E9
Canada
E-Mail: schishol@nortel.com

Hector Trevino
Cisco
Suite 400
9155 E. Nichols Ave
Englewood, CO 80112
USA
EMail: htrevino@cisco.com

Полное заявление авторских прав

К этому документу применимы права, лицензии и ограничения, указанные в BCP 78, и, за исключением указанного там, авторы сохраняют свои права.

Этот документ и содержащаяся в нем информация представлены "как есть" и автор, организация, которую он/она представляет или которая выступает спонсором (если таковой имеется), Internet Society и IETF отказываются от каких-либо гарантий (явных или подразумеваемых), включая (но не ограничиваясь) любые гарантии того, что использование представленной здесь информации не будет нарушать чьих-либо прав, и любые предполагаемые гарантии коммерческого использования или применимости для тех или иных задач.

Интеллектуальная собственность

IETF не принимает какой-либо позиции в отношении действительности или объема каких-либо прав интеллектуальной собственности (Intellectual Property Rights или IPR) или иных прав, которые, как может быть заявлено, относятся к реализации или использованию описанной в этом документе технологии, или степени, в которой любая лицензия, по которой права могут или не могут быть доступны, не заявляется также применение каких-либо усилий для определения таких прав. Сведения о процедурах IETF в отношении прав в документах RFC можно найти в BCP 78 и BCP 79.

Копии раскрытия IPR, предоставленные секретариату IETF, и любые гарантии доступности лицензий, а также результаты попыток получить общую лицензию или право на использование таких прав собственности разработчиками или пользователями этой спецификации, можно получить из сетевого репозитория IETF IPR по ссылке <http://www.ietf.org/ipr>.

IETF предлагает любой заинтересованной стороне обратить внимание на авторские права, патенты или использование патентов, а также иные права собственности, которые могут потребоваться для реализации этого стандарта. Информацию следует направлять в IETF по адресу ietf-ipr@ietf.org.

Перевод на русский язык

Николай Малых
nmalykh@protokols.ru