Internet Engineering Task Force (IETF)
Request for Comments: 8641
Category: Standards Track
ISSN: 2070-1721

A. Clemm
Futurewei
E. Voit
Cisco Systems

September 2019

Subscription to YANG Notifications for Datastore Updates

Подписка на уведомления YANG об обновлении хранилища данных

Аннотация

Этот документ описывает механизм, позволяющий приложениям-подписчикам запросить настраиваемый продолжающийся поток обновления от хранилища данных YANG. Такая видимость обновлений открывает новые возможности на основе отражения и мониторинга удалённых состояний конфигурации и работы.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. 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/rfc8641.

Оглавление

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

1. Введение

Традиционные подходы к обеспечению видимости управляемых объектов из удалённой системы основаны на опросе. При опросах данные периодически запрашиваются и извлекаются клиентом с сервера для поддержки их актуальности. Однако с управлением на основе опросов связаны некоторые проблемы.

- Опросы связаны с существенной задержкой, которая препятствует использованию многих приложений.
- Циклы опроса могут пропускаться, а запросы могут задерживаться или теряться, особенно в нагруженной сети, когда потребность в данных велика.
- Запросы могут подвергаться флуктуациям, приводящим к различию интервалов опроса. Полученные данные сложно калибровать и сравнивать.
- Для приложений, отслеживающих изменения, многие циклы удалённого опроса создают нежелательную и, в конечном итоге, расточительную нагрузку на сеть, устройства и приложения, особенно при редких измерениях.

Эффективной альтернативой запросам является получение автоматических продолжающихся обновлений от нужного набора хранилищ данных. Поэтому есть потребность в службе, которая (1) позволяет приложениям подписываться на обновления хранилищ данных и (2) позволяет серверам (их ещё называют издателями - publisher) выталкивать (push) данные передавая обновления, по сути, в режиме потока. Требования к таким службам приведены в [RFC7923].

Этот документ предлагает соответствующее решение на основе [RFC8639]. Дополнением к этой работе являются дополнения (augment) моделей данных YANG, расширенные RPC и определяемые хранилищами уведомления. Варианты транспорта, представленные в [RFC8639], хорошо подходят для этого решения.

2. Определения

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.

Документ использует термины из [RFC7950], [RFC8341], [RFC8342], [RFC8639] и добавляет термины, указанные ниже.

Datastore node - узел хранилища данных
Узел созданного экземпляра дерева данных YANG, связанный с хранилищем данных. В этом документе такие узлы часто называются объектами.

Datastore node update - обновление узла хранилища данных
Элемент данных, содержащий текущее значение узла datastore на момент создания обновления, а также путь к узлу хранилища данных.

Datastore subscription - описание хранилища данных
Подписка на поток обновления для узла хранилища данных.

Datastore subtree - субдерево (ветвь) хранилища данных
Узел хранилища данных и све его потомки datastore.

On-change subscription - подписка на изменения
Подписка на хранилище данных с уведомлениями при обнаружении изменения в хранилище.

Periodic subscription - периодическая подписка
Подписка на хранилище данных с уведомлениями через заданный интервал времени.

Selection filter - фильтр выбора
Оценка и/или выбор критерия, применяемого к целевому набору объектов.

Update record - запись обновления
Представление обновления одного или нескольких узлов хранилищ данных. Кроме того, запись может содержать сведения о том, какой тип обновления привёл у обновлению узла datastore (например, были ли узел datastore добавлен, изменён или удалён). В запись могут включаться другие метаданные, такие как идентификатор подписки, для которой создана запись. В документе записи обновления часто называются просто обновлением.

Update trigger - триггер обновления
Механизм, определяющий необходимость создания записи обновления.

YANG-Push
Механизм подписки и выталкивания обновлений хранилища данных, описанный в этом документе.

3. Обзор решения

Этот документ задаёт решение, обеспечивающее услуги подписки на обновления от хранилища данных. Поддерживаются динамические и настраиваемые подписки на обновления. Подписка задаёт, когда уведомления (их называют также push-обновлениями) следует передавать и какие данные включать в записи обновления. Обновления хранилищ данных последовательно выталкиваются (push) от издателя в соответствии с подпиской.

3.1. Модель подписки

Подписки YANG-Push определены с использованием модели данных YANG. Эта модель расширяет модель подписки, определённую в [RFC8639], возможностью подписки на обновления хранилищ данных, в частности, задания триггеров обновления, указывающих условия генерации записей обновления, а также их состав. Главные отличия указаны ниже.

- Указание фильтров выбора, определяющих узлы и/или ветви хранилища данных, для которых выталкиваются обновления.
- Указание правил обновления, включающих условия генерации и выталкивания новых записей. Имеется два типа подписок, различающиеся режимом отправки уведомлений - периодически или по факту изменения.
  - Для периодической подписки триггер обновлений указывается двумя параметрами, определяющими выталкивание обновлений. Эти параметру задают (1) интервал выталкивания обновления и (2) время привязки - anchor-time, т. е. точку отсчёта для вычисления моментов сборки и отправки периодических обновлений.
  - Для подписки on-change триггером обновлений является обнаружение изменений в сведениях, на которые организована подписка. Параметры управления описаны ниже.
    - dampening-period. При подписке на изменения уведомления следует передавать как можно быстрее. Однако частая отправка последовательности изменений может оказаться нежелательной, поскольку может требовать значительных ресурсов издателя или получателя. Для защиты от истощения ресурсов **может** применяться интервал демпфирования, задающий время ожидания перед генерацией последовательных записей обновления для одной подписки. Интервал демпфирования применяется ко всем узлам хранилища данных, выбранным в одной подписке. Это означает, что при обновлении одного или нескольких объектов запись для них создаётся незамедлительно (при отсутствии интервала демпфирования) или в конце заданного интервала демпфирования. Если в интервале демпфирования какой-либо объект меняется неоднократно, включается лишь значение на момент создания записи обновления. Интервал демпфирования начинается по завершении сборки записи обновления.
    - Параметр change-type может применяться для сокращения числа типов изменений, для которых передаются уведомления (например, можно передавать уведомления только при создании или удалении объекта, но не при его изменении).
    - Параметр sync-on-start определяет, будет ли применяться выталкивание всех обновлений (push-update, параграф 3.7) в начале подписки. Такая ранняя синхронизация задаёт «опорную точку» для последующих обновлений.
- Кодирование (с использованием anydata) обновлений, выталкиваемых периодически или по изменению.

3.2. Согласование правил подписки

Запрос на динамическую подписку **следует** отклонена, если издатель понимает, что не может предоставить записи обновления в соответствии с запросом RPC establish-subscription или modify-subscription. В этом случае подписчик сможет быстро передать новый запрос RPC с другими параметрами.

Подписчику не следует пытаться угадать подходящие параметры. Поэтому для сокращения числа попыток подписки в динамической подписке поддерживается простое согласование параметров подписки между издателем и подписчиком. Согласование происходит путём включения дополнительной информации в отклик об ошибке для запроса RPC. Учёт возвращённых в сообщении об ошибке сведений позволит повысить вероятность успеха при последующем вызове RPC. Подсказки включают предлагаемые интервалы периодических обновлений, приемлемы интервалы демпфирования, оценку числа объектов, возвращаемых при использовании предложенного фильтра отбора. Однако не даётся гарантии, что последующий запрос в соответствии с этими подсказками будет воспринят.

3.3. Обновления при изменении

Подписка on-change позволяет получать обновления при изменении целевых объектов. Такая подписка особенно полезна при нечастом изменении данных, требующем быстрого оповещения с минимальной задержкой после события.

Подписки на изменения обычно сложнее в реализации, чем периодические подписки, поэтому могут поддерживаться не всеми реализациями или не для каждого объекта.

Восприятие или отклонение запроса подписки на изменения, когда подписка включает объекты, для которых не поддерживаются уведомления при изменении, определяется реализацией издателя. Издатель **может** воспринять подписку на изменения даже при наличии в запросе объектов, для которых on-change не поддерживается. В этом случае обновления передаются лишь для поддерживаемых объектов, а прочие объекты исключаются из записей обновления даже при смене их значений. Чтобы подписчик мог определить объекты, поддерживающие уведомления при изменении, издатель помечает объекты должным образом. Поэтому подписчик принимает на себя ответственность за выбор нужных объектов. Маркировка объектов описана в параграфе 3.10.

Издатель **может** просто отклонять подписки on-change, содержащие объекты, для которых не поддерживаются уведомления при изменении. В случае настраиваемой подписки издатель **может** приостановить подписку.

Чтобы избавить получателей от лавины повторяющихся обновлений для подписки на быстро меняющиеся объекты или объекты с колеблющимися значениями, в подписке на изменения можно задать интервал демпфирования. После создания записи для объекта с таким интервалом новые записи о его изменениях не будут создаваться, пока не

закончится интервал демпфирования. В конце периода демпфирования передаются текущие значения для всех изменённых объектов. Изменёнными считаются также объекты, созданные или удалённые в интервале демпфирования. Если объект вернулся к первоначальному значению или был создан и удалён в интервале демпфирования, текущее значение (а не промежуточные обновления) все равно будет передано. Это указывает, что на объекте происходили изменения.

Подписки на изменения можно уточнять, указывая лишь некоторые типы изменений. Например, подписчик может получать уведомления лишь о создании и удалении объектов, но не о смене их значений.

С учётом сказанного выше процесс создания записи об обновлении в подписке на изменения показан ниже.

1. Перед изменением или в начале интервала демпфирования применяется фильтрация и правила контроля доступа для проверки полномочий подписчика на просмотр соответствующих узлов хранилища (отфильтровываются ненужные узлы). Результатом является набор ветвей и узлов хранилища A.
2. Перед изменением или в конце интервала демпфирования применяется фильтрация и правила контроля доступа (возможно, новые). Результатом является набор ветвей и узлов хранилища B.
3. Создаётся запись для обновления путём применения записи YANG Patch [RFC8072] для перехода от A к B.
4. Если A и B имеются различия, аннулирующие друг друга, в запись YANG Patch помещается последнее изменение даже при совпадении нового значения с исходным (в результате отмены внесённых изменений последующими). Если изменения включают создание нового узла в хранилище и его последующее удаление, запись YANG Patch будет указывать удаление. Если же узел был удалён и создан заново, запись YANG Patch будет показывать создание узла.
5. Если полученная запись YANG Patch не пуста, она передаётся получателю.

Примечание. Если подписчик хочет задать разные интервалы демпфирования для разных объектов, он может организовать несколько подписок с разными фильтрами.

3.4. Вопросы надёжности

Подписка на обновления хранилища предназначена для избавления от опросов. Однако при этом важно, чтобы подписчики могли полагаться на подписку и быть уверенными в получении интересующих обновлений, не заботясь от том, что обновления могут быть незаметно отброшены. Иными словами, подписка является обещанием издателя предоставлять обновления получателям в соответствии с условиями подписки.

Однако имеется много причин, по которым издатель не может выполнять условия подписки, даже если та была создана с его согласия. Например, число узлов данных в хранилище может быть больше ожидаемого, а интервал - слишком коротким для быстрой отправки полной серии обновлений или внутренние проблемы могут мешать сбору объектов. По этим причинам предложенное в документе решение (1) требует от издателя уведомлять получателей при невозможности выполнять условия подписки и (2) предоставляет издателю возможность приостановить подписку в таких случаях. Это включает указание неполноты обновления в уведомлении push-update или push-change-update, а также отправку уведомлений subscription-suspended, когда это применимо. Описание этого дано в параграфе 3.11.1.

Издателю **следует** отвергать подписку, если выполнение запрошенных для неё условий маловероятно. В таких случаях предпочтительно, чтобы подписчик запросил менее ресурсоёмкие условия, нежели часто сталкивался с невыполнением условий.

Решение основано на [RFC8639], где указано, что любая потеря связности в базовом транспорте будет обнаруживаться и приведёт к прерыванию (для динамической подписки) или приостановке (для настраиваемой подписки), гарантируя, что потеря уведомлений об изменениях не пройдёт незамеченной.

3.5. Кодирование данных

3.5.1. Периодические подписки

При периодической подписке данные, включённые как часть записи обновления, соответствуют данным, которые можно было прочитать с помощью операции извлечения.

3.5.2. Подписки на изменения

При подписке на изменения записи обновления должны указывать не только значения изменённых узлов хранилища данных, но и типы изменений с момента предыдущего обновления. Поэтому правила кодирования данных в обновлениях on-change обычно следуют операциям YANG Patch, заданным в [RFC8072]. Эти операции указывают, что нужно применить к состоянию из предыдущего обновления, чтобы получить новое состояние. Отметим, что объекты, в обновлении могут включать не только данные конфигурации, но и иные объекты (включая рабочие данные), тогда как исправления (patch) [RFC8072] применяются лишь к данных конфигурации в конфигурационных хранилищах.

Издатель указывает тип изменения узла в хранилище данных с помощью операций YANG Patch - create служит для недавно созданных объектов (кроме записей упорядоченного пользователем списка), delete - для удалённых объектов (включая упорядоченные пользователем списки), replace - только при изменения значения объекта, insert - при вставке в список нового объекта, move - при перемещении записи в упорядоченном пользователем списке.

Однако патч должен делать больше, чем просто описывать отличие предыдущего состояния объекта от текущего. В соответствии с параграфом 3.3 требуется также определить, происходили ли изменения в интервале демпфирования. Для этого допустимо кодировать операцию YANG Patch так, чтобы её применение не приводило к изменению текущего состояния по сравнению с предыдущим. Это указывает, что на объекте происходили какие-то действия. Примером может служить патч, который указывает, операцию create для хранилища данных, когда получатель полагает, что объект уже существует, или операцию replace которая «меняет» прежнее состояние на такое же. Отметим, что это означает, что указанные в параграфе 2.5 [RFC8072] ошибки операций create и delete не являются ошибками в случае YANG-Push (т. е. считаются корректными операциями для YANG-Push).

3.6. Указание отбора в хранилище

В подписке должны быть указаны фильтры отбора и хранилище данных, к которому эти фильтры применяются. Эти сведения служат для выбора и последующей передачи данных из хранилища издателя получателям.

К подписке в каждый момент может применяться лишь 1 фильтр и запрос RPC, задающий новый фильтр, будет переписывать имеющийся. Ниже указаны типы фильтров, включённые в модель YANG-Push, которые можно применить к хранилищу данных.

subtree

Фильтр выбора ветвей указывает одну или несколько ветвей дерева. При указании этого фильтра записи обновления будут исходить лишь из узлов данных указанных ветвей хранилища. Синтаксис и семантика соответствуют заданным в разделе 6 [RFC6241].

xpath

Фильтр xpath - это выражение XPath[1], возвращающее набор узлов, для которых будут передаваться обновления.

Эти фильтры служат селекторами, определяющими объекты, попадающий в сферу действия подписки. Издатель должен поддерживать хотя бы один тип фильтров отбора.

XPath обеспечивает мощные средства фильтрации и при создании фильтров нужно соблюдать осторожность. Примером может быть фильтр XPath, который пропускает лишь узел хранилища с активным (up) интерфейсом. Получатель должен понимать влияние отсутствия или наличия объектов в каждом обновлении.

Когда набор критериев фильтра применяется к периодической подписке, критерии используются при каждом создании записи обновления и получателю передаются лишь узлы хранилища, соответствующие фильтру, к которым получатель имеет доступ. Если тот же фильтр применяется к подписке on-change, передаются сведения лишь для части узлов, поддерживающих уведомления при изменении. Узел хранилища, не поддерживающий on-change, не будет передаваться в push-update или push-change-update при подписке on-change (параграф 3.7).

3.7. Потоковые обновления

В отличие от традиционных запросов на извлечение данных, подписка на хранилище позволяет передать неограниченную по времени серию обновлений в форме потока. Для этого определены два базовых уведомления YANG - push-update и push-change-update.

Уведомление push-update задаёт полные, фильтруемые уведомления для обновлений хранилища данных в соответствии с условиями подписки. Этот тип уведомлений YANG применяется для постоянных уведомлений в периодической подписке и может также служить для подписки on-change в двух случаях. Во-первых, начальное уведомление push-update **должно** применяться при старте новой подписки для синхронизации получателя. Во-вторых, оно **может** передаваться, если позднее получатель решит снова синхронизировать подписку on-change. Запись обновления push-update содержит экземпляр ветви со всем содержимым, заданным подпиской. Это содержимое эквивалентно данным, которые были бы получены из хранилища явной операцией извлечения с использованием того же транспорта и фильтров.

Обновление push-change-update более распространено для подписок on-change. Запись обновления в этом случае включает набор изменений в узлах хранилища с момента отправки предыдущего сообщения. Иными словами, оно указывает узлы, которые были созданы, удалены или изменили свои значения. Когда в интервале демпфирования происходит несколько изменений и объект не удаляется, указывается последнее значение, т. е. для каждого объекта передаётся лишь одно изменение, а не вся история (иначе не было бы смысла задавать интервал демпфирования).

Сведения push-update и push-change-update кодируются и помещаются в уведомления, которые ставятся в выходную очередь для заданного транспорта.

На рисунке 1 приведён пример уведомления для подписки, отслеживающей рабочее состояние одного интерфейса Ethernet (в соответствии с [RFC8343]). Данные кодируются в XML [W3C.REC-xml-20081126] и передаются по протоколу управления сетью (Network Configuration Protocol или NETCONF) [RFC8640].

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
 <eventTime>2017-10-25T08:00:11.22Z</eventTime>
 <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
   <id>1011</id>
   <datastore-contents>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
       <interface>
         <name>eth0</name>
         <oper-status>up</oper-status>
       </interface>
     </interfaces>
   </datastore-contents>
 </push-update>
</notification>
```

Рисунок 1. Пример выталкивания.

На рисунке 2 приведён пример сообщения on-change для такой же подписки.

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
 <eventTime>2017-10-25T08:22:33.44Z</eventTime>
 <push-change-update
      xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
   <id>89</id>
   <datastore-changes>
     <yang-patch>
       <patch-id>0</patch-id>
       <edit>
```

[1]XPath - это язык запросов для выбора узлов в документе XML [XPATH].

```
          <edit-id>edit1</edit-id>
          <operation>replace</operation>
          <target>/ietf-interfaces:interfaces</target>
          <value>
            <interfaces
                 xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
              <interface>
                <name>eth0</name>
                <oper-status>down</oper-status>
              </interface>
            </interfaces>
          </value>
        </edit>
      </yang-patch>
    </datastore-changes>
  </push-change-update>
</notification>
```

Рисунок 2. Пример выталкивания для уведомления On-Change.

В примере patch-id имеет значение 0. В соответствии с [RFC8072] patch-id является произвольной строкой. При использовании YANG-Push издателю **следует** помещать в patch-id значение счётчика, начинающегося с 0 и инкрементируемое при каждой следующей генерации push-change-update для подписки. При использовании в качестве счётчика он **должен** сбрасываться в 0 при каждой ресинхронизации (т. е. при передаче push-update), а также по достижении максимального значения 4294967295 (предельное значение для типа uint32). Это позволяет легко обнаруживать потери или нарушение порядка записей обновления.

3.8. Управление подписками

Определённые в [RFC8639] RPC были усовершенствованы для поддержки подписки на хранилища данных. Были также добавлены коды ошибок, указывающие причины отказа в подписке и новые yang-data, которые **можно** применять для включения деталей входных параметров для успеха последующего вызова RPC.

Организация или изменение подписки на хранилище данных могут быть отвергнуты по разным причинам, включая запрос слишком большой ветви или неспособность издателя достаточно часто выталкивать записи обновлений. В таких случаях подписка не организуется и возвращается отклик RPC, указывающий причину отказа. В этот отклик **может** включаться набор параметров подписки, которые вероятно будут приняты при новом запросе. Подписчик может воспользоваться этими параметрами в будущих запросах.

Если при отказе в подписке на хранилище данных в отклик включаются советы по параметрам, их **следует** передавать в контейнере yang-data establish-subscription-datastore-error-info, помещённом в отклик об ошибке RPC вместо establish-subscription-stream-error-info, используемого в случае потоковой подписки. На рисунке 3 показано дерево establish-subscription-datastore-error-info с использованием нотации [RFC8340].

```
yang-data establish-subscription-datastore-error-info
  +--ro establish-subscription-datastore-error-info
     +--ro reason?                identityref
     +--ro period-hint?           centiseconds
     +--ro filter-failure-hint?   string
     +--ro object-count-estimate? uint32
     +--ro object-count-limit?    uint32
     +--ro kilobytes-estimate?    uint32
     +--ro kilobytes-limit?       uint32
```

Рисунок 3. Дерево establish-subscription-datastore-error-info.

В случае отказа на изменение подписки с включениям советов их **следует** помещать в контейнер yang-data modify-subscription-datastore-error-info в отклике об ошибке RPC вместо modify-subscription-stream-error-info, используемого для потоковой подписки. На рисунке 4 показано дерево modify-subscription-datastore-error-info.

```
yang-data modify-subscription-datastore-error-info
  +--ro modify-subscription-datastore-error-info
     +--ro reason?                identityref
     +--ro period-hint?           centiseconds
     +--ro filter-failure-hint?   string
     +--ro object-count-estimate? uint32
     +--ro object-count-limit?    uint32
     +--ro kilobytes-estimate?    uint32
     +--ro kilobytes-limit?       uint32
```

Рисунок 4. Дерево modify-subscription-datastore-error-info.

3.9. Проверка прав получателя

Получателю данных подписки **должны** передавать лишь уведомления, на которые он имеет соответствующие права доступа. Издатель **должен** предотвращать включение в выталкиваемые данные сведений, для которых у получателя нет прав доступа. Для этого нужно применять все соответствующие проверки при выталкивании обновления с удаление при необходимости данных из ветвей хранилища, к которым получателю не предоставлен доступ. Это позволяет проверять права доступа к выталкиваемым по подписке данным YANG как при их обычном извлечении (get).

К каждому push-update и push-change-update **должны** применяться правила контроля доступа, как показано на рисунке 5. Это включает проверку наличия прав чтения для любого нового объекта, выбранного после отправки предыдущего уведомления конкретному получателю. Издатель **должен** без уведомления исключать узлы данных, которые получателю не разрешено видеть. Для реализации этого **следует** применять концептуальную модель проверки полномочий из [RFC8341] (в частности, параграф 3.2.4), расширенную для узлов данных в уведомлениях, а не только в сообщениях <rpc-reply> в ответ на запросы <get> и <get-config>.

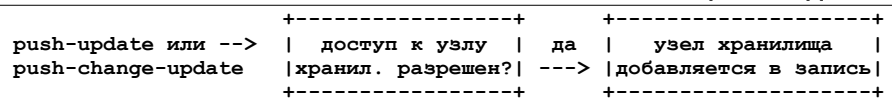

Рисунок 5. Контроль доступа для Push-обновлений.

Издатель **должен** разрешать указание в фильтре подписки отсутствующих данных и данных, к которым получателю не разрешён доступ. Это позволяет получателю отслеживать весь жизненный цикл определённого дерева хранилища данных без необходимости явно указывать каждый узел. Если после применения правил контроля доступа не остаётся элементов для записи обновления, результат зависит от типа подписки. Для периодической подписки **должно** передаваться пустое обновление push-update, чтобы клиент не думал о потере обновления. Для подписки on-change передавать уведомление push-update **недопустимо**, чтобы клиент не узнал об изменениях в узлах, к которым у него нет доступа для чтения. Изменениям в фильтрах **недопустимо** влиять на интервалы демпфирования.

Издатель **может** отклонить запрос establish-subscription для несуществующих данных или данных, к которым у получателя нет доступа. В этом случае **следует** возвращать unchanging-selection как идентификатор причины отказа. Кроме того, издатель **может** прервать динамическую подписку или приостановить настраиваемую при изменении прав доступа получателя или правил доступа к объектам подписки. В таких случаях издетелю **следует** указывать unchanging-selection как причину отказа при отправке subscription-terminated или subscription-suspended. Такая возможность позволяет издателю избежать необходимости поддерживать постоянную и полную фильтрацию содержимого подписки для каждой записи обновления, а также снижает выроятность утечки контролируемых объектов.

Если доступ для чтения ранее доступных узлов был утрачен в результате смены прав получателя, это **следует** указать как операцию delete для подписки on-change. Если не удаётся обработать такие изменения прав получателей с помощью delete, реализация издателя **должна** инициировать повторную организацию динамической подписки или повторную инициализацию настраиваемой подписки, чтобы установить соответствующие фильтры.

3.10. Узлы хранилища с уведомлением при изменении

В некоторых случаях издатель, поддерживающий уведомления при изменении, не способен выталкивать обновления on-change для некоторых типов объектов. Причиной этого могут быть частые изменения узлов хранилища данных (например, счётчиков октетов [RFC8343]), частые и малозначимые изменения мелких объектов (например, изменение температуры на 0,1 градуса) или неспособность реализации передавать такие уведомления для конкретного объекта.

В таких случаях для клиентских приложений важна возможность определить, для каких объектов поддерживаются и не поддерживаются уведомления при изменении. Иначе приложения не будут знать, могут ли они полагаться на подписку on-change для получения интересующих уведомлений об изменениях. Иными словами, если реализация не предоставляет решения или поддерживает поддерживает полные уведомления об изменениях, у клиентов такой реализации не будет возможности узнать область действия их подписки on-change.

Поэтому реализациям настоятельно рекомендуется предоставлять решение этой проблемы. Одним из вариантов может быть предоставление клиентам возможности узнать, для каких объектов поддерживаются уведомления при изменении, с помощью другой модели данных YANG. Пример такого решения представлен в [Yang-Push-Notif-Cap]. До стандартизации этого решения реализациям **следует** предоставлять свои решения.

3.11. Другие вопросы

3.11.1. Отказоустойчивость и надёжность

Важно, чтобы описанные в этом документе обновления (в частности, on-change) не терялись. Если же потеря уведомлений неизбежна, получатель должен знать об этом.

Записи обновлений в одной подписке **недопустимо** переупорядочивать до транспортировки.

Вполне возможно, что при некоторых обстоятельствах издатель поймёт, что он не способен включить в запись обновления полный набор объектов, заданных условиями подписки. В таких случаях он **должен** действовать, как указано ниже.

- Издатель **должен** установить флаг incomplete-update для любой записи, не содержащей всю информацию.
- Издатель **может** приостановить подписку в соответствии с [RFC8639]. Если издатель совсем не создаёт запись обновления, он **должен** приостановить подписку.
- При возобновлении подписки on-change издателю **следует** генерировать отличия (patch) от предыдущей. Если это невозможно и для подписки задано sync-on-start true, **можно** передать содержимое хранилища полностью через push-update (фактическая замена прежнего содержимого). Если и этот вариант невозможен, **должен** быть установлен флаг incomplete-update в следующем обновлении push-change-update.

Примечание. Вполне возможна постановка серии уведомлений push-change-update (и даже push-update) в очередь передачи на транспортном уровне. Издатель не обязан объединять записи обновления, переданные одновременно.

Действие получателя при наличии флага incomplete-update в записи обновления зависят от приложения. Можно просто ждать, не делая ничего, выполнить ресинхронизацию, активно извлекая все сведения для подписки, или прервать подписку и организовать новую, возможно с сокращением в ней числа объектов.

3.11.2. Возможности издателя

Предпочтительней отклонить запрос на подписку, нежели принять запрос, который невозможно выполнить.

Возможность поддержки подписки зависит от нескольких факторов, таких как триггер обновления в подпсике (при изменении или периодически), период передачи обновлений (короткий период требует больше ресурсов), объём данных в ветви хранилища, на которую организована подписка, число и сочетание других обслуживаемых подписок.

4. Модель YANG для управления Push-подписками на хранилища

4.1. Обзор

Модель данных YANG для подписок с выталкиванием на зранилища данных показана на рисунках 6 - 9 с использованием деревьев с нотацией [RFC8340]. Заданные здесь новые объекты схемы (не присутствующие в [RFC8639]) указаны префиксом yp. Для удобства читателей размер деревьев сокращен путём исключения некоторых узлов данных модуля YANG ietf-subscribed-notifications [RFC8639], не существенных для понимания, с заменой на «...».

Поскольку дерево достаточно велико, оно разделено на 4 части. На рисунке 6 показаны дополнения модуля YANG ietf-yang-push для конфигурации подписки, указанной в модуле YANG ietf-subscribed-notifications.

Модуль ietf-subscribed-notifications

```
...
+--rw filters
|  ...
|  +--rw yp:selection-filter* [filter-id]
|     +--rw yp:filter-id                   string
|     +--rw (yp:filter-spec)?
|        +--:(yp:datastore-subtree-filter)
|        |  +--rw yp:datastore-subtree-filter?   <anydata>
|        |          {sn:subtree}?
|        +--:(yp:datastore-xpath-filter)
|           +--rw yp:datastore-xpath-filter?     yang:xpath1.0
|                   {sn:xpath}?
+--rw subscriptions
   +--rw subscription* [id]
      |  ...
      +--rw (target)
      |  +--:(stream)
      |  |   ...
      |  +--:(yp:datastore)
      |     +--rw yp:datastore                      identityref
      |     +--rw (yp:selection-filter)?
      |        +--:(yp:by-reference)
      |        |  +--rw yp:selection-filter-ref
      |        |          selection-filter-ref
      |        +--:(yp:within-subscription)
      |           +--rw (yp:filter-spec)?
      |              +--:(yp:datastore-subtree-filter)
      |              |  +--rw yp:datastore-subtree-filter?
      |              |          <anydata> {sn:subtree}?
      |              +--:(yp:datastore-xpath-filter)
      |                 +--rw yp:datastore-xpath-filter?
      |                         yang:xpath1.0 {sn:xpath}?
      | ...
      +--rw (yp:update-trigger)
         +--:(yp:periodic)
         |  +--rw yp:periodic!
         |     +--rw yp:period         centiseconds
         |     +--rw yp:anchor-time?   yang:date-and-time
         +--:(yp:on-change) {on-change}?
            +--rw yp:on-change!
               +--rw yp:dampening-period?   centiseconds
               +--rw yp:sync-on-start?      boolean
               +--rw yp:excluded-change*    change-type
```

Рисунок 6. Структура модели данных - конфигурация подписки.

На рисунке 7 показаны дополнения из модуля YANG ietf-yang-push для RPC, заданных в модуле YANG ietf-subscribed-notifications [RFC8639]. В частности, эти дополнения касаются RPC establish-subscription и modify-subscription, в которые дабавлены параметры, требуемые для задания push-подписки на хранилища данных.

```
rpcs:
  +---x establish-subscription
  |  +---w input
  |  |  ...
  |  |  +---w (target)
  |  |  |  +--:(stream)
  |  |  |  |  ...
  |  |  |  +--:(yp:datastore)
  |  |  |     +---w yp:datastore                      identityref
  |  |  |     +---w (yp:selection-filter)?
  |  |  |        +--:(yp:by-reference)
  |  |  |        |  +---w yp:selection-filter-ref
  |  |  |        |          selection-filter-ref
  |  |  |        +--:(yp:within-subscription)
  |  |  |           +---w (yp:filter-spec)?
  |  |  |              +--:(yp:datastore-subtree-filter)
  |  |  |              |  +---w yp:datastore-subtree-filter?
  |  |  |              |          <anydata> {sn:subtree}?
  |  |  |              +--:(yp:datastore-xpath-filter)
  |  |  |                 +---w yp:datastore-xpath-filter?
  |  |  |                         yang:xpath1.0 {sn:xpath}?
  |  |  | ...
  |  |  +---w (yp:update-trigger)
```

```
|  |     +--:(yp:periodic)
|  |     |  +---w yp:periodic!
|  |     |     +---w yp:period         centiseconds
|  |     |     +---w yp:anchor-time?   yang:date-and-time
|  |     +--:(yp:on-change) {on-change}?
|  |        +---w yp:on-change!
|  |           +---w yp:dampening-period?   centiseconds
|  |           +---w yp:sync-on-start?      boolean
|  |           +---w yp:excluded-change*    change-type
|  +--ro output
|     +--ro id                             subscription-id
|     +--ro replay-start-time-revision?    yang:date-and-time
|              {replay}?
+---x modify-subscription
|  +---w input
|     ...
|     +---w (target)
|     |  ...
|     |  +--:(yp:datastore)
|     |     +---w yp:datastore                    identityref
|     |     +---w (yp:selection-filter)?
|     |        +--:(yp:by-reference)
|     |        |  +---w yp:selection-filter-ref
|     |        |          selection-filter-ref
|     |        +--:(yp:within-subscription)
|     |           +---w (yp:filter-spec)?
|     |              +--:(yp:datastore-subtree-filter)
|     |              |  +---w yp:datastore-subtree-filter?
|     |              |          <anydata> {sn:subtree}?
|     |              +--:(yp:datastore-xpath-filter)
|     |                 +---w yp:datastore-xpath-filter?
|     |                         yang:xpath1.0 {sn:xpath}?
|     | ...
|     +---w (yp:update-trigger)
|        +--:(yp:periodic)
|        |  +---w yp:periodic!
|        |     +---w yp:period         centiseconds
|        |     +---w yp:anchor-time?   yang:date-and-time
|        +--:(yp:on-change) {on-change}?
|           +---w yp:on-change!
|              +---w yp:dampening-period?   centiseconds
+---x delete-subscription
|  ...
+---x kill-subscription
   ...

yang-data (для сообщений об ошибках RPC)
  ...
```

Рисунок 7. Структура модели данных - RPC.

На рисунке 8 показаны дополнения из модуля YANG ietf-yang-push для уведомлений, заданных в модуле YANG ietf-subscribed-notifications. Эти дополнения позволяют включать параметры конфигурации подписки на хранилище данных в уведомления subscription-started и subscription-modified.

```
notifications:
  +---n replay-completed {replay}?
  |  ...
  +---n subscription-completed
  |  ...
  +---n subscription-started {configured}?
  |  |  ...
  |  +--ro (target)
  |  |  ...
  |  |  +--:(yp:datastore)
  |  |     +--ro yp:datastore                    identityref
  |  |     +--ro (yp:selection-filter)?
  |  |        +--:(yp:by-reference)
  |  |        |  +--ro yp:selection-filter-ref
  |  |        |          selection-filter-ref
  |  |        +--:(yp:within-subscription)
  |  |           +--ro (yp:filter-spec)?
  |  |              +--:(yp:datastore-subtree-filter)
  |  |              |  +--ro yp:datastore-subtree-filter?
  |  |              |          <anydata> {sn:subtree}?
  |  |              +--:(yp:datastore-xpath-filter)
  |  |                 +--ro yp:datastore-xpath-filter?
  |  |                         yang:xpath1.0 {sn:xpath}?
  |  ...
  |  +--ro (yp:update-trigger)
  |     +--:(yp:periodic)
  |     |  +--ro yp:periodic!
  |     |     +--ro yp:period         centiseconds
  |     |     +--ro yp:anchor-time?   yang:date-and-time
  |     +--:(yp:on-change) {on-change}?
  |        +--ro yp:on-change!
```

```
|           +--ro yp:dampening-period?   centiseconds
|           +--ro yp:sync-on-start?      boolean
|           +--ro yp:excluded-change*    change-type
+---n subscription-resumed
|  ...
+---n subscription-modified
|  ...
|  +--ro (target)
|  |  |  ...
|  |  +--:(yp:datastore)
|  |     +--ro yp:datastore                    identityref
|  |     +--ro (yp:selection-filter)?
|  |        +--:(yp:by-reference)
|  |        |  +--ro yp:selection-filter-ref
|  |        |          selection-filter-ref
|  |        +--:(yp:within-subscription)
|  |           +--ro (yp:filter-spec)?
|  |              +--:(yp:datastore-subtree-filter)
|  |              |  +--ro yp:datastore-subtree-filter?
|  |              |          <anydata> {sn:subtree}?
|  |              +--:(yp:datastore-xpath-filter)
|  |                 +--ro yp:datastore-xpath-filter?
|  |                         yang:xpath1.0 {sn:xpath}?
|  ...
|  +--ro (yp:update-trigger)?
|     +--:(yp:periodic)
|     |  +--ro yp:periodic!
|     |     +--ro yp:period         centiseconds
|     |     +--ro yp:anchor-time?   yang:date-and-time
|     +--:(yp:on-change) {on-change}?
|        +--ro yp:on-change!
|           +--ro yp:dampening-period?    centiseconds
|           +--ro yp:sync-on-start?       boolean
|           +--ro yp:excluded-change*     change-type
+---n subscription-terminated
|  ...
+---n subscription-suspended
   ...
```

Рисунок 8. Структура модели данных - уведомления.

На рисунке 9 приведены добавленные этим документов части модуля YANG ietf-yang-push, не являющиеся дополнениями других модулей YANG.

Модуль ietf-yang-push

```
rpcs:
  +---x resync-subscription {on-change}?
     +---w input
        +---w id    sn:subscription-id

yang-data (for placement into RPC error responses):
  +-- resync-subscription-error
  |  +--ro reason?                   identityref
  |  +--ro period-hint?              centiseconds
  |  +--ro filter-failure-hint?      string
  |  +--ro object-count-estimate?    uint32
  |  +--ro object-count-limit?       uint32
  |  +--ro kilobytes-estimate?       uint32
  |  +--ro kilobytes-limit?          uint32
  +-- establish-subscription-error-datastore
  |  +--ro reason?                   identityref
  |  +--ro period-hint?              centiseconds
  |  +--ro filter-failure-hint?      string
  |  +--ro object-count-estimate?    uint32
  |  +--ro object-count-limit?       uint32
  |  +--ro kilobytes-estimate?       uint32
  |  +--ro kilobytes-limit?          uint32
  +-- modify-subscription-error-datastore
     +--ro reason?                   identityref
     +--ro period-hint?              centiseconds
     +--ro filter-failure-hint?      string
     +--ro object-count-estimate?    uint32
     +--ro object-count-limit?       uint32
     +--ro kilobytes-estimate?       uint32
     +--ro kilobytes-limit?          uint32

   notifications:
     +---n push-update
     |  +--ro id?                   sn:subscription-id
     |  +--ro datastore-contents?   <anydata>
     |  +--ro incomplete-update?    empty
     +---n push-change-update {on-change}?
        +--ro id?                   sn:subscription-id
        +--ro datastore-changes
        |  +--ro yang-patch
        |     +--ro patch-id    string
```

```
|     +--ro comment?    string
|     +--ro edit* [edit-id]
|        +--ro edit-id      string
|        +--ro operation    enumeration
|        +--ro target       target-resource-offset
|        +--ro point?       target-resource-offset
|        +--ro where?       enumeration
|        +--ro value?       <anydata>
+--ro incomplete-update?    empty
```

Рисунок 9. Структура модели данных - новые части.

Некоторые компоненты модели данных описаны ниже.

4.2. Настройка подписки

Настраиваемые и динамические подписки представлены листом subscription. Ниже указаны новые параметры, расширяющие модель данных подписки из [RFC8639].

- Целевое хранилище, из которого извлекаются данные. Возможные хранилища включают указанные в [RFC8342]. Платформа может также поддерживать своё хранилище данных.
- Фильтр выбора интересующих узлов YANG в хранилище данных. Содержимое фильтра задаётся ссылкой на имеющийся фильтр или встроенным определением лишь для этой подписки. Указание фильтров ссылкой позволяет реализации избежать проверки приемлемости фильтра при запросе динамической подписки. Оператор case применяется для задания вариантов.
- Для периодической подписки триггерные обновления будут происходить на границах заданного интервала времени, которые могут быть рассчитаны по параметрам подписки:
  - period задаёт интервал между выталкиваниями обновлений;
  - anchor-time задаёт начальное время для отсчёта интервалов обновления; если этот параметр не задан, в качестве начального времени **должен** устанавливаться момент создания первой записи с обновлением.
- Для подписки on-change при условии завершения интервала демпфирования выталкивание происходит при обновлении информации для подписки. Подписки on-change имеют более сложную семантику, определяемую своим набором параметров:
  - dampening-period задаёт интервал, который должен пройти до отправки следующего обновления в подписке, т. е. каждое новое уведомление передаётся не раньше dampening-period после предыдущего;
  - excluded-change позволяет ограничить типы изменений, для которых следует передавать обновления (например, добавлять запись обновления только при создании объекта);
  - sync-on-start указывает, нужно ли передавать полное обновление всех данных в начале подписки.

4.3. Уведомления YANG

4.3.1. Уведомления о смене состояния

Применяются механизмы и уведомления для состояния подписки из [RFC8639]. Уведомления subscription-started и subscription-modified дополнены (augment) для включения объектов, связанных с хранилищем данных.

4.3.2. Уведомления для содержимого подписки

Кроме содержимого подписки уведомления push-update и push-change-update могут включать другие объекты.

- Идентификатор подписки (id) **должен** передаваться вместе с содержимым. Это позволяет получателю связать запись обновления с конкретной подпиской.
- Лист incomplete-update указывает, что не все изменения, произошедшие с момента предыдущего обновления, были включены в данное обновление. Иными словами, издатель не смог полностью выполнить свои обязательства по подписке, например, хранилище не смогло предоставить процессу издателя полный набор своих узлов. Для ресинхронизации подписки on-change издатель **может** передать следом push-update с моментальным снимком (snapshot) данных подписки.

4.4. YANG RPC

Подписки YANG-Push организуются, изменяются и удаляются с применением дополненных RPC из [RFC8639].

4.4.1. establish-subscription

Подписчик передаёт RPC establish-subscription с параметрами, указанными в параграфе 3.1. Например,

```
<netconf:rpc message-id="101"
    xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <establish-subscription
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"
      xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <yp:datastore
        xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
      ds:operational
    </yp:datastore>
    <yp:datastore-xpath-filter
        xmlns:ex="https://example.com/sample-data/1.0">
      /ex:foo
    </yp:datastore-xpath-filter>
```

```
    <yp:periodic>
      <yp:period>500</yp:period>
    </yp:periodic>
  </establish-subscription>
</netconf:rpc>
```

Рисунок 10. RPC establish-subscription.

Позитивный отклик включает идентификатор воспринятой подписки (id) и может иметь вид

```
<rpc-reply message-id="101"
   xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <id
     xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
      52
   </id>
</rpc-reply>
```

Рисунок 11. RPC establish-subscription с позитивным откликом.

Подписка может быть отвергнута по разным причинам, включая отсутствие прав на её организацию, нехватка у издателя возможностей для обслуживания, неспособность издателя извлекать содержимое хранилища данных с запрошенной периодичностью.

Если запрос отклонён из-за несособности издатедя обслужить его, издателю следует включить в отклик об ошибке советы, которые помогут подписчику задать приемлемые параметры для следующего запроса подписки. Эти подсказки включаются в структуру yang-data establish-subscription-error-datastore. Однако даже следование подсказкам не гарантирует успешной подписки.

Конкретные параметры, возвращаемые в отклике об ошибке RPC, зависят от применяемого для поддержки подписки транспорта. Для NETCONF такие параметры определены в [RFC8640]. Например, запрос NETCONF может иметь вид

```
    <rpc message-id="101"
         xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <establish-subscription
          xmlns=
            "urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"
          xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <yp:datastore
            xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
          ds:operational
        </yp:datastore>
        <yp:datastore-xpath-filter
            xmlns:ex="https://example.com/sample-data/1.0">
          /ex:foo
        </yp:datastore-xpath-filter>
        <yp:on-change>
          <yp:dampening-period>100</yp:dampening-period>
        </yp:on-change>
      </establish-subscription>
    </rpc>
```

Рисунок 12. Запрос establish-subscription, пример 2.

Издатель, который не может обслужить обновления при изменении, но способен доставить периодические обновления, может передать отклик NETCONF вида

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
  <rpc-error>
    <error-type>application</error-type>
    <error-tag>operation-failed</error-tag>
    <error-severity>error</error-severity>
    <error-path>/yp:periodic/yp:period</error-path>
    <error-info>
      <yp:establish-subscription-error-datastore>
        <yp:reason>yp:on-change-unsupported</yp:reason>
      </yp:establish-subscription-error-datastore>
    </error-info>
  </rpc-error>
</rpc-reply>
```

Рисунок 13. establish-subscription с откликом об ошибке, пример 2.

4.4.2. modify-subscription

Подписчик **может** вызвать RPC modify-subscription для организованной ранее подписки, включая в этот вызов желаемые значения параметров подписки. Не включённые в запрос параметры **должны** оставаться неизменными. На рисунке 14 показан пример попытки подписчика изменить период и фильтр XPath для хранилища данных в подписке с использованием NETCONF.

```
    <rpc message-id="102"
         xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <modify-subscription
          xmlns=
            "urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"
          xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <id>1011</id>
```

```
      <yp:datastore
          xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
        ds:operational
      </yp:datastore>
      <yp:datastore-xpath-filter
          xmlns:ex="https://example.com/sample-data/1.0">
        /ex:bar
      </yp:datastore-xpath-filter>
      <yp:periodic>
        <yp:period>250</yp:period>
      </yp:periodic>
    </modify-subscription>
  </rpc>
```

Рисунок 14. Запрос modify-subscription.

Издатель **должен** отвечать на запрос изменения подписки. Если запрос отвергнут, имеющаяся подписка не меняется и издатель **должен** возвратить отклик об ошибке RPC, который может включать советы, инкапсулированные в структуру yang-data modify-subscription-error-datastore. Подписку **можно** изменять неоднократно.

Конкретные параметры, возвращаемые в отклике об ошибке RPC, зависят от транспорта, применяемого для поддержки подписки. Для NETCONF такие параметры заданы в [RFC8640].

Настраиваемые подписки нельзя изменить с помощью RPC modify-subscription. Вместо этого нужно вносить изменения в конфигурацию.

4.4.3. delete-subscription

Для прекращения приёма обновлений и фактического удаления подписки, организованной ранее с помощью RPC establish-subscription подписчик может передать RPC delete-subscription, принимающего на входе лишь идентификатор подписки (id). Вызов RPC заимствован из [RFC8639].

4.4.4. resync-subscription

Этот вызов RPC поддерживается лишь для подписок on-change, ранее созданных вызовом RPC establish-subscription, например

```
  <rpc message-id="103"
       xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <resync-subscription
        xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
      <id>1011</id>
    </resync-subscription>
  </rpc>
```

Рисунок 15. resync-subscription.

При получении вызова издатель должен (1) воспринять запрос и быстро передать push-update или (2) передать соответствующий отклик об ошибке RPC. В этот отклик издатель **может** включить структуру данных yang-data resync-subscription-error с дополнительными сведениями о причине ошибки.

4.4.5. Синхронизация модуля YANG

Для запроса подписки подписчику нужно знать используемые издателем схемы хранилищ данных YANG. Эти схемы доступны в модуле библиотеки YANG ietf-yang-library.yang, заданном в [RFC8525]. Предполагается, что получатель знает о библиотеке YANG до начала подписки.

Набор модулей, выпусков, свойств и отклонений может меняться в процессе работы (если это поддерживает реализация издателя). Для этого библиотека YANG предоставляет простое уведомление yang-library-change, информирующее подписчика об изменении библиотеки. В этом случае может потребоваться информирование получателя об изменении модуля для корректной обработки обновлений, связанных с узлами хранилища данных.

5. Модуль YANG для YANG-Push

Этот модуль YANG импортирует определения типов из [RFC6991], идентификаторы из [RFC8342], расширение yang-data из [RFC8040], группировку yang-patch из [RFC8072]. Кроме того, модуль импортирует дополнения многих определений из [RFC8639], а также ссылается на [RFC6241], [XPATH] (XML Path Language (XPath) Version 1.0) и [RFC7950].

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

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-subscribed-notifications {
    prefix sn;
    reference
      "RFC 8639: Subscription to YANG Notifications";
  }
  import ietf-datastores {
    prefix ds;
```

```
    reference
      "RFC 8342: Network Management Datastore Architecture (NMDA)";
  }
  import ietf-restconf {
    prefix rc;
    reference
      "RFC 8040: RESTCONF Protocol";
  }
  import ietf-yang-patch {
    prefix ypatch;
    reference
      "RFC 8072: YANG Patch Media Type";
  }

  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>";

  description
    "Этот модуль содержит спецификации YANG для YANG-Push.

     Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
     СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
     НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
     BCP 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.c
     IETF Trust's Legal Provisions Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     Эта версия модуля YANG является частью RFC 8641, где правовые
     аспекты приведены более полно.";

  revision 2019-09-09 {
    description
      "Исходный выпуск.";
    reference
      "RFC 8641: Subscriptions to YANG Datastores";
  }

  /*
   * Свойства (функции)
   */

  feature on-change {
    description
      "Указывает поддержку подписки, вызываемой изменениями.";
  }

  /*
   * Отождествления (идентификаторы)
   */

  /* Идентификаторы типов ошибок для подписки на хранилище данных. */

  identity resync-subscription-error {
    description
      "Проблема при попытке выполнить запрос resync-subscription.";
  }

  identity cant-exclude {
    base sn:establish-subscription-error;
    description
      "Не удалось удалить набор параметров excluded-change. Это
       означает, что издатель не может ограничить уведомления
       push-change-update лишь типами изменений, запрошенные для
       этой подписки.";
  }

  identity datastore-not-subscribable {
    base sn:establish-subscription-error;
    base sn:subscription-terminated-reason;
```

```
    description
      "На это хранилище подписка невозможна.";
  }

  identity no-such-subscription-resync {
    base resync-subscription-error;
    description
      "Указанной подписки нет. Это может быть обусловлено ошибочным
       идентификатором подписки, идентификатором от другого
       подписчика или идентификатором настраиваемой подписки.";
  }

  identity on-change-unsupported {
    base sn:establish-subscription-error;
    description
      "Подписка on-change не поддерживается для каких-либо объектов,
       выбираемых этим фильтром.";
  }

  identity on-change-sync-unsupported {
    base sn:establish-subscription-error;
    description
      "Ни sync-on-start, ни ресинхронизация не поддерживаются для
       этой подписки. Эта ошибка может использоваться в двух случаях:
       (1) RPC establish-subscription включает sync-on-start, но
       издатель не поддерживает отправку push-update для этой
       подписки по причинам, отличным от on-change-unsupported и
       sync-too-big
       (2) вызван RPC resync-subscription для имеющейся периодической
       подписки или on-change без поддержки ресинхронизации.";
  }

  identity period-unsupported {
    base sn:establish-subscription-error;
    base sn:modify-subscription-error;
    base sn:subscription-suspended-reason;
    description
      "Запрошенный период или dampening-period слишком мал. Это
       возможно для периодических и on-change подписок (с
       демпфированием и без него). Могут дополнительно возвращаться
       советы по части приемлемого периода.";
  }

  identity update-too-big {
    base sn:establish-subscription-error;
    base sn:modify-subscription-error;
    base sn:subscription-suspended-reason;
    description
      "Размер деревьев периодического или on-change уведомления
       превышает допустимый. Советы с оценкой допустимого размера
       могут возвращаться как дополнение.";
  }

  identity sync-too-big {
    base sn:establish-subscription-error;
    base sn:modify-subscription-error;
    base resync-subscription-error;
    base sn:subscription-suspended-reason;
    description
      "Размер деревьев sync-on-start или ресинхронизации превышает
       допустимый. Могут дополнительно включаться советы с оценкой
       приемлемого размера.";
  }

  identity unchanging-selection {
    base sn:establish-subscription-error;
    base sn:modify-subscription-error;
    base sn:subscription-terminated-reason;
    description
      "Фильтр вряд ли выберет какие-либо узлы дерева данных. Это
       означает, что на основе текущих прав доступа подписчика
       издатель понимает, что выбор фильтром изменённых узлов дерева
       маловероятен. Примерами могут служит отсутствие ветви или
       прав у получателя, статические объекты, меняющиеся лишь
       при перезагрузке.";
  }

  /*
   * Определения типов
   */

  typedef change-type {
    type enumeration {
      enum create {
        description
          "Изменение говорит о создании нового узла в хранилище.";
```

```
      }
      enum delete {
        description
          "Изменение говорит об удалении узла из хранилища.";
      }
      enum insert {
        description
          "Изменение говорит о вставке нового узла хранилища с
           заданным пользователем порядком.";
      }
      enum move {
        description
          "Изменение говорит о смене порядка узлов в хранилище.";
      }
      enum replace {
        description
          "Изменение говорит о смене значения узла хранилища.";
      }
    }
    description
      "Указывает разные типы изменений в хранилище данных.

       Этот тип основан на операциях, заданных для YANG Patch, но
       отличается тем, что получатель может обрабатывать записи
       обновлений с операцией create на узле хранилища, который
       существует по мнению получателя, или удалять узел, которого
       по мнению получателя нет.";
    reference
      "RFC 8072: YANG Patch Media Type, Section 2.5";
  }

  typedef selection-filter-ref {
    type leafref {
      path "/sn:filters/yp:selection-filter/yp:filter-id";
    }
    description
      "Этот тип служит для указания фильтра отбора.";
  }

  typedef centiseconds {
    type uint32;
    description
      "Интервал времени в сотых долях секунды (0,01).";
  }

  /*
   * Определения групп
   */

  grouping datastore-criteria {
    description
      "Группировка для критериев, по которым объекты целевого
       хранилища данных следует включать в push-обновления.";
    leaf datastore {
      type identityref {
        base ds:datastore;
      }
      mandatory true;
      description
        "Хранилище, из которого извлекаются данные.";
    }
    uses selection-filter-objects;
  }

  grouping selection-filter-types {
    description
      "Группировка для типов селекторов объектов из хранилища.";
    choice filter-spec {
      description
        "Спецификация фильтра содержимого для этого запроса.";
      anydata datastore-subtree-filter {
        if-feature "sn:subtree";
        description
          "Указывает извлекаемые части целевого хранилища.";
        reference
          "RFC 6241: Network Configuration Protocol (NETCONF),
                     Section 6";
      }
      leaf datastore-xpath-filter {
        if-feature "sn:xpath";
        type yang:xpath1.0;
        description
          "Выражение XPath, указывающее извлекаемые части хранилища.

           Если выражение даёт набор узлов, все узлы этого набора
           выбираются фильтром. Иначе фильтр не возвращает ничего.
```

```
          Выражение оценивается в указанном ниже контексте.
          - Набор объявлений пространств имён - это набор пар
            «префикс-пространство имен» 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 selection-filter-objects {
  description
    "Определяет селектор для объектов из хранилища.";
  choice selection-filter {
    description
      "Источник фильтра отбора, применяемого для подписки. Это
       может быть (1) ссылка на глобальный список или (2) указание
       в самой подписке.";
    case by-reference {
      description
        "Фильтр, настроенный отдельно.";
      leaf selection-filter-ref {
        type selection-filter-ref;
        mandatory true;
        description
          "Ссылка на имеющийся фильтр отбора для применения
           к подписке.";
      }
    }
    case within-subscription {
      description
        "Локальное задание позволяет фильтру иметь жизненный цикл
         как у подписки.";
      uses selection-filter-types;
    }
  }
}

grouping update-policy-modifiable {
  description
    "Описывает связанные с хранилищем условия подписки, которые
     можно изменить в течение действия подписки.";
  choice update-trigger {
    description
      "Условия, требуемые для передачи подписчику записи
       обновления.";
    case periodic {
      container periodic {
        presence "Указывает периодическую подписку";
        description
          "У издателя запрашивается периодическая отправка
           получателю сведений о текущих значения узлов хранилища,
           определённых фильтром отбора.";
        leaf period {
          type centiseconds;
          mandatory true;
          description
            "Интервал между периодическими push обновлениями в
             сотых долях секунды (0,01).";
        }
        leaf anchor-time {
          type yang:date-and-time;
          description
            "Временная метка, до или после которой задаётся серия
             периодических обновлений. Следующая отправка будет
             по истечении целого числа интервалов передачи от
             anchor-time'. Например, при anchor-time, указывающем
             начало определённой минуты, и интервале в 1 минуту
             обновления будут передаваться в начале каждой минуты,
             когда подписка активна.";
        }
```

```
          }
        }
        case on-change {
          if-feature "on-change";
          container on-change {
            presence "Указывает подписку on-change.";
            description
              "У издателя запрошено уведомление получателя при смене
               в хранилище данных значений, заданных фильтром отбора.";
            leaf dampening-period {
              type centiseconds;
              default "0";
              description
                "Минимальный интервал между сборкой последовательных
                 записей обновления для одного получателя подписки.
                 Когда объект подписки изменился и истёк интервал
                 демпфирования (возможно, 0) после отправки предыдущей
                 записи обновления получателю, все изменившиеся объекты
                 и свойства подписки упорядочиваются и помещаются в
                 новую запись.";
            }
          }
        }
      }
    }

    grouping update-policy {
      description
        "Зависящие от хранилища условия подписки.";
      uses update-policy-modifiable {
        augment "update-trigger/on-change/on-change" {
          description
            "Объекты, которые нельзя изменить после организации
             подписки.";
          leaf sync-on-start {
            type boolean;
            default "true";
            description
              "Значение false (1) запрещает передавать уведомления
               push-update в подписке on-change и (2) ЗАПРЕЩАЕТ
               выталкивание полного набора объектов в соответствии
               с фильтром для этой подписки. Передаются лишь
               уведомления об изменениях (push-change-update). Значение
               true (задано по умолчанию) для упрощения синхронизации
               получателя разрешает передачу полного обновления в
               уведомлении push-update при старте подписки. После этого
               передаются лишь уведомления push-change-update, пока
               издатель не решит ресинхронизировать подписку с помощью
               нового уведомления push-update.";
          }
          leaf-list excluded-change {
            type change-type;
            description
              "Ограничивает набор изменений, вызывающих обновление.
               Например, при исключении операции replace будут
               передаваться лишь уведомления о создании и удалении.";
          }
        }
      }
    }

    grouping hints {
      description
        "Параметры, связанные с ошибкой подписки на хранилище данных.";
      leaf period-hint {
        type centiseconds;
        description
          "Возвращается при запросе слишком короткого периода. Этот
           совет может включать приемлемый интервал для периодической
           подписки или время демпфирования для подписки on-change.";
      }
      leaf filter-failure-hint {
        type string;
        description
          "Сведения о причине неприемлемости указанного фильтра
           для подписки.";
      }
      leaf object-count-estimate {
        type uint32;
        description
          "Если фильтр отбора может возвращать слишком много объектов,
           это значение указывает оценку числа таких объектов.";
      }
      leaf object-count-limit {
        type uint32;
        description
```

```
        "Если фильтр отбора возвращает слишком много объектов, это
         значение указывает верхний предел числа таких объектов.";
    }
    leaf kilobytes-estimate {
      type uint32;
      description
        "Если возвращаемая информация может выходить за пределы
         возможностей издателя, это значение будет указывать оценку
         максимального объёма данных после фильтра отбора.";
    }
    leaf kilobytes-limit {
      type uint32;
      description
        "Если возвращаемая информация может выходить за пределы
         возможностей издателя, это значение будет указывать верхний
         предел объёма данных после фильтра отбора.";
    }
  }

  /*
   * RPC
   */
  rpc resync-subscription {
    if-feature "on-change";
    description
      "Позволяет подписчику запросить выталкивание всех объектов
       в активной подписке on-change. Успешный вызов RPC ведёт к
       уведомлению push-update для всех узлов хранилища, к которым
       подписчику разрешён доступ. Этот вызов RPC возможен только
       из той же сессии, в которой работает подписка. В случае
       ошибки передаётся отклик с resync-subscription-error.";
    input {
      leaf id {
        type sn:subscription-id;
        mandatory true;
        description
          "Идентификатор подписки для ресинхронизации.";
      }
    }
  }

  rc:yang-data resync-subscription-error {
    container resync-subscription-error {
      description
        "При отказе RPC resync-subscription подписка не
         ресинхронизируется и сообщение об ошибке ДОЛЖНО указывать
         причину отказа. Эта структура yang-data МОЖЕТ включаться в
         отклик об ошибке для указания причины отказа.";
      leaf reason {
        type identityref {
          base resync-subscription-error;
        }
        mandatory true;
        description
          "Указывает причину отклонения запроса ресинхронизации.";
      }
      uses hints;
    }
  }

  augment "/sn:establish-subscription/sn:input" {
    description
      "Добавляет параметры подписки, применяемые к обновлениям
       хранилища данных, как входные параметры RPC.";
    uses update-policy;
  }

  augment "/sn:establish-subscription/sn:input/sn:target" {
    description
      "Добавляет хранилище данных как действительную цель для
       подписки в качестве входных данных RPC.";
    case datastore {
      description
        "Сведения о параметрах запроса для подписки на хранилище.";
      uses datastore-criteria;
    }
  }

  rc:yang-data establish-subscription-datastore-error-info {
    container establish-subscription-datastore-error-info {
      description
        "Если какой-либо параметр RPC establish-subscription не
         поддерживается для хранилища, подписка не организуется
         и отклик об ошибке RPC ДОЛЖЕН указывать причину отказа
         в подписке. Эта структура yang-data МОЖЕТ помещаться в
         отклик об ошибке RPC для указания причины отказа. Эта
```

```
          структура ДОЛЖНА включаться, если подписчику возвращаются
          подсказки.";
      leaf reason {
        type identityref {
          base sn:establish-subscription-error;
        }
        description
          "Причина отказа в подписке на целевое хранилище данных.";
      }
      uses hints;
    }
  }

  augment "/sn:modify-subscription/sn:input" {
    description
      "Дополнительные параметры подписки на обновления хранилища.";
    uses update-policy-modifiable;
  }

  augment "/sn:modify-subscription/sn:input/sn:target" {
    description
      "Добавляет хранилище данных как действительную цель для
       подписки в качестве входных данных RPC.";
    case datastore {
      description
        "Сведения о параметрах запроса для подписки на хранилище.";
      uses datastore-criteria;
    }
  }

  rc:yang-data modify-subscription-datastore-error-info {
    container modify-subscription-datastore-error-info {
      description
        "Эта структура yang-data МОЖЕТ включаться в отклик об ошибке
         RPC при отказе modify-subscription для хранилища данных.
         Структура ДОЛЖНА использоваться, если подписчику
         возвращаются подсказки.";
      leaf reason {
        type identityref {
          base sn:modify-subscription-error;
        }
        description
          "Причина отказа в изменении подписки.";
      }
      uses hints;
    }
  }

  /*
   * Уведомления
   */
  notification push-update {
    description
      "Уведомление с push-обновлением, с данными для подписки. При
       периодической подписке это уведомление передаётся для
       периодических обновлений. Оно также применяется для
       синхронизации обновлений в подписке on-change. Это уведомление
       нужно передавать лишь получателям подписки, оно не является
       уведомлением общего назначения, на которое можно подписаться
       любому получателю как на часть потока событий NETCONF.";
    leaf id {
      type sn:subscription-id;
      description
        "Указывает подписку, вызвавшую отправку уведомления.";
    }
    anydata datastore-contents {
      description
        "Обновлённые данные, содержащие снимок на момент обновления
         для всего набора данных подписки. Эти же данные были бы
         возвращены операцией get с таким же фильтром отбора.";
    }
    leaf incomplete-update {
      type empty;
      description
        "Флаг, указывающий неполноту обновления. Иными словами,
         издатель не смог выполнить все свои обязательства по
         подписке и передал неполный набор объектов.";
    }
  }

  notification push-change-update {
    if-feature "on-change";
    description
      "Pus-обновление on-change, которое нужно передавать лишь
       получателям подписки, оно не является уведомлением общего
       назначения, на которое можно подписаться любому получателю
```

```
        как на часть потока событий NETCONF.";
    leaf id {
      type sn:subscription-id;
      description
        "Указывает подписку, вызвавшую отправку уведомления.";
    }
    container datastore-changes {
      description
        "Набор изменений целевого хранилища, начиная с момента
         предыдущего обновления по условиям подписки.";
      uses ypatch:yang-patch;
    }
    leaf incomplete-update {
      type empty;
      description
        "Указывает, что включены не все изменения после предыдущего
         обновления. Иными словами, издатель не смог полностью
         выполнить свои обязательства по подписке, например, по
         причине большого числа произошедших изменений.";
    }
  }

  augment "/sn:subscription-started" {
    description
      "Добавляет связанные с хранилищем объекты в уведомление,
       передаваемое при старте подписки.";
    uses update-policy;

  augment "/sn:subscription-started/sn:target" {
    description
      "Позволяет включать хранилище в уведомление при старте
       подписки.";
    case datastore {
      uses datastore-criteria {
        refine "selection-filter/within-subscription" {
          description
            "Задаёт фильтр отбора и его местоположение. При указании
             selection-filter-ref фильтр берётся из контейнера
             filters, иначе задаётся явно как часть подписки.";
        }
      }
    }
  }

  augment "/sn:subscription-modified" {
    description
      "Добавляет связанные с хранилищем объекты в уведомление при
       изменении подписки.";
    uses update-policy;
  }

  augment "/sn:subscription-modified/sn:target" {
    description
      "Позволяет включать хранилище в уведомление при обновлении
       подписки.";
    case datastore {
      uses datastore-criteria {
        refine "selection-filter/within-subscription" {
          description
            "Задаёт фильтр отбора и его местоположение. При указании
             selection-filter-ref фильтр берётся из контейнера
             filters, иначе задаётся явно как часть подписки.";
        }
      }
    }
  }

  /*
   * Узлы данных
   */
  augment "/sn:filters" {
    description
      "Позволяет включать хранилище как часть критериев отбора
       для подписки.";
    list selection-filter {
      key "filter-id";
      description
        "Список заданных заранее фильтров, применяемых к подписке.";
      leaf filter-id {
        type string;
        description
          "Идентификатор фильтра отбора.";
      }
      uses selection-filter-types;
    }
  }
```

```
    augment "/sn:subscriptions/sn:subscription" {
      when 'yp:datastore';
      description
        "Добавляет в подписку на хранилище объекты, связанные с
         хранилищем, например, поток обновлений узлов хранилища.";
      uses update-policy;
    }

    augment "/sn:subscriptions/sn:subscription/sn:target" {
      description
        "Позволяет включать хранилище как часть критериев отбора
         для подписки.";
      case datastore {
        uses datastore-criteria;
      }
    }
  }
  <CODE ENDS>
```

6. Взаимодействие с IANA

Этот документ добавляет URI в реестр IETF XML Registry [RFC3688].

```
URI: urn:ietf:params:xml:ns:yang:ietf-yang-push
Registrant Contact: The IESG.
XML: N/A; запрошенный URI является пространством имён XML.
```

Документ добавляет модуль YANG в реестр YANG Module Names [RFC6020]

```
Name: ietf-yang-push
Namespace: urn:ietf:params:xml:ns:yang:ietf-yang-push
Prefix: yp
Reference: RFC 8641
```

7. Вопросы безопасности

Описанный в этом документе модуль YANG определяет схему данных, которая предназначения для доступа по протоколам сетевого управления, таким как NETCONF [RFC6241] или RESTCONF [RFC8040]. Нижним уровнем NETCONF является защищённый транспорт с обязательной поддержкой SSH [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.

В модуле YANG имеется множество узлов данных, доступных для чтения, создания и удаления (например, с принятым по умолчанию config true). Эти узлы могут быть конфиденциальными или уязвимыми в некоторых сетевых средах. Операции записи (например, <edit-config>) в эти узлы без подобающей защиты могут оказывать негативное влияние на работу сети. Следует отметить, что заданный здесь модель YANG дополняет модуль YANG из [RFC8639] и все соображения безопасности, отмеченные в [RFC8639], применимы и к подпискам на хранилища данных. Ниже перечислены субдеревья и узлы данных, добавленные в этом документе, с указанием уязвимости.

Ветвь selection-filter в контейнере filters

Эта ветвь позволяет подписчику указать узлы и ветви для включения в подписку на хранилище. Атакующий может попытаться изменить этот фильтр, например, для увеличения числа возвращаемых объектов с целью перегрузки получателя. Можно также исключить часть объектов из подписки, делая их изменения незаметными.

Ветвь datastore в выборе target списка subscription

Как и для ветви selection-filter, злоумышленник может пытаться изменить фильтруемые объекты для перегрузки получателя большим числом обновлений или сокрытия некоторых изменений.

Выбор update-trigger в списке subscription

Меняя триггер обновлений, злоумышленник может изменить передаваемые обновления для запутывания получателя или его перегрузки. Например, атакующий может поменять период отправки уведомлений в периодической подписке или интервал демпфирования в подписке on-change, увеличивая задержку передачи обновления, что может влиять на скорость отклика зависящих от обновлений приложений, или увеличивая число обновлений для истощения ресурсов получателя.

NACM обеспечивает средства для смягчения этих угроз на стороне издателя. Для смягчения угроз у получателей тот может отслеживать конфигурацию подписки на предмет неожиданных изменений и подписываться на обновления узлов хранилища данных YANG, представляющих подписку на хранилища. Поскольку объем таких данных невелик, сверхосторожный подписчик может даже сохранить периодический опрос для защиты подписки от скомпрометированных обновлений конфигурации самой подписки.

Некоторые из доступных для чтения узлов в этом модуле YANG могут быть конфиденциальны или уязвимы в той или иной сетевой среде. Важно контролировать доступ к таким объектам (например, get, get-config, notification). Ниже перечислены ветви и узлы, которые могут быть конфиденциальны или уязвимы.

Ветвь selection-filter в контейнере filters

Отсутствие должного контроля доступа может раскрывать компоненты системы для тех, кто не должен видеть их.

Ветвь datastore в выборе target списка subscription

Отсутствие должного контроля доступа может раскрывать компоненты системы для тех, кто не должен видеть их.

Выбор update-trigger в списке subscription

Отсутствие должного контроля доступа может раскрывать компоненты системы для тех, кто не должен видеть их.

Некоторые из операций RPC в этих модулях YANG могут быть чувствительны или уязвимы в той или иной сетевой среде. Важно контролировать доступ к таким операциям. Ниже указаны операции модуля ieft-dhcpv6-relay.yang, которые могут быть чувствительны или уязвимы.

RPC resync-subscription

Этот вызов RPC позволяет подписчику on-change щапросить выталкивание всех объектов подписки, что может приводить к передаче большого объёма данных. Злоумышленник может попытаться использовать RPC для истощения ресурсов сервера генерацией данных и последующей перегрузки получателя этими данными.

8. Литература

8.1. Нормативные документы

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8072] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Patch Media Type", RFC 8072, DOI 10.17487/RFC8072, February 2017, <https://www.rfc-editor.org/info/rfc8072>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/info/rfc8525>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[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>.

8.2. Дополнительная литература

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[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>.

[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>.

[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>.

[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>.

[Yang-Push-Notif-Cap] Lengyel, B., Clemm, A., and B. Claise, "Yang-Push Notification Capabilities", Work in Progress[1], draft-ietf-netconf-notification-capabilities-04, September 2019.

[1]Опубликовано в RFC 9196. *Прим. перев.*

Приложение A. Ошибки при подписке

A.1. Отказы RPC

Отклонение RPC по любой причине указывается откликом об ошибке RPC от издателя. Возвращаемые ошибки RPC включают (1) имеющиеся коды ошибок RPC на транспортном уровне, такие как приведены для NETCONF в [RFC6241], и (2) связанные с подпиской ошибки, заданные в модели данных YANG. В результате кодирование ошибок в откликах RPC зависит от транспорта.

Конкретные идентификаторы из модулей YANG ietf-subscribed-notifications [RFC8639] и ietf-yang-push могут возвращаться как часть сообщений об ошибках при неудачных попытках подписки на хранилища данных. Ошибки из модуля ietf-subscribed-notifications указаны в [RFC8639], а ошибки, определённые этим документом, приведены ниже.

establish-subscription	*modify-subscription*	*resync-subscription*
cant-exclude	period-unsupported	no-such-subscription-resync
datastore-not-subscribable	update-too-big	sync-too-big
on-change-unsupported	sync-too-big	
on-change-sync-unsupported	unchanging-selection	
period-unsupported		
update-too-big		
sync-too-big		
unchanging-selection		

В модель данных YANG включён финальный набор элементов для независимых от транспорта ошибок RPC в 4 структурах yang-data для отказов при подписке на хранилища данных.

1. Структура yang-data establish-subscription-error-datastore **должна** возвращаться, если сведения о причине ошибки RPC не включены в транспортную часть отклика об ошибке RPC establish-subscription. Она **должна** передаваться, если включены подсказки.
2. Структура yang-data modify-subscription-error-datastore **должна** возвращаться, если сведения о причине ошибки RPC не включены в транспортную часть отклика об ошибке RPC modify-subscription. Она **должна** передаваться, если включены подсказки.
3. Структура yang-data sn:delete-subscription-error **должна** возвращаться, если сведения о причине ошибки RPC не включены в транспортную часть отклика об ошибке RPC delete-subscription или kill-subscription.
4. Структура yang-data resync-subscription-error **должна** возвращаться, если сведения о причине ошибки RPC не включены в транспортную часть отклика об ошибке RPC resync-subscription.

A.2. Уведомления об отказах

Подписка может быть неожиданно прервана или приостановлена без вызова RPC или операции настройки. В таких случаях **должна** указываться причина отказа. Может возвращаться ряд ошибок в соответствующем уведомлении о смене состояния подписки. Для этого документ вводит указанные ниже идентификаторы ошибок, дополняющие ошибки, заданные в [RFC8639].

subscription-terminated	*subscription-suspended*
datastore-not-subscribable	period-unsupported
unchanging-selection	update-too-big
	synchronization-size

Благодарности

Спасибо Tim Jenkins, Martin Bjorklund, Kent Watsen, Susan Hares, Yang Geng, Peipei Guo, Michael Scharf, Guangying Zheng, Tom Petch, Henk Birkholz, Reshad Rahman, Qin Wu, Rohit Ranade, Rob Wilton за ценные комментарии, дискуссии и отзывы.

Участники работы

Ниже указаны люди, внёсшие существенные вклад в этот документ, которых следует считать соавторами. Их вклад состоит в создании модуля YANG, представленного в разделе 5.

Alberto Gonzalez Prieto
Microsoft
Email: alberto.gonzalez@microsoft.com

Ambika Prasad Tripathy
Cisco Systems
Email: ambtripa@cisco.com

Einar Nilsen-Nygaard
Cisco Systems
Email: einarnn@cisco.com

Andy Bierman
YumaWorks
Email: andy@yumaworks.com

Balazs Lengyel
Ericsson
Email: balazs.lengyel@ericsson.com

Адреса авторов

Alexander Clemm
Futurewei
Email: ludwig@clemm.org

Eric Voit
Cisco Systems
Email: evoit@cisco.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru