

Internet Engineering Task Force (IETF)
Request for Comments: 8913
Category: Standards Track
ISSN: 2070-1721

R. Civil
Ciena Corporation
A. Morton
AT&T Labs
R. Rahman

M. Jethanandani
Xoriant Corporation
K. Pentikousis, Ed.
Detecon
November 2021

Two-Way Active Measurement Protocol (TWAMP) YANG Data Model

Модель данных YANG для протокола TWAMP

Аннотация

Этот документ задаёт модель данных для реализации клиентов и серверов протокола двухсторонних активных измерений (Two-Way Active Measurement Protocol или TWAMP). Документ определяет модель данных TWAMP с помощью диаграмм классов унифицированного языка моделирования (Unified Modeling Language или UML) и формально определяет её с использованием языка моделирования данных YANG (RFC 7950). Модель данных совместима с архитектурой хранилищ данных управления сетью (Network Management Datastore Architecture или NMDA).

Статус документа

Документ содержит проект стандарта Internet (Standards Track).

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

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <https://www.rfc-editor.org/info/rfc8913>.

Авторские права

Copyright (c) 2021. Авторские права принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

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

Оглавление

1. Введение.....	2
1.1. Мотивация.....	2
1.2. Уровни требований.....	2
1.3. Организация документа.....	2
2. Область действия, модель и применимость.....	2
3. Обзор модели данных.....	3
3.1. Control-Client.....	3
3.2. Сервер.....	3
3.3. Session-Sender.....	4
3.4. Session-Reflector.....	4
4. Параметры модели данных.....	4
4.1. Control-Client.....	4
4.2. Сервер.....	5
4.3. Session-Sender.....	6
4.4. Session-Reflector.....	6
5. Модель данных.....	7
5.1. Диаграмма дерева YANG.....	7
5.2. Модуль YANG.....	8
6. Примеры моделей данных.....	23
6.1. Control-Client.....	23
6.2. Сервер.....	24
6.3. Session-Sender.....	24

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

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

6.4. Session-Reflector.....	25
7. Вопросы безопасности.....	26
8. Взаимодействие с IANA.....	27
9. Литература.....	27
9.1. Нормативные документы.....	27
9.2. Дополнительная литература.....	28
Приложение А. Подробные примеры моделей данных.....	28
А.1. Control-Client.....	28
А.2. Сервер.....	30
А.3. Session-Sender.....	30
А.4. Session-Reflector.....	31
Приложение В. Рабочие команды.....	32
Благодарности.....	32
Участник работы.....	32
Адреса авторов.....	32

1. Введение

Протокол TWAMP [RFC5357] служит для измерения параметров производительности сети, таких как задержка, пропускная способность и потеря пакетов, путём передачи пробных пакетов и измерения параметров их передачи через сеть. На сегодняшний день реализации TWAMP не поставляются со стандартной системой управления, поэтому разработчикам остаётся лишь предоставлять свой механизм. Документ решает этот вопрос, определяя модель с использованием диаграмм классов UML [UML] и формально задаёт модель данных TWAMP, совместимую с архитектурой NMDA) [RFC8342], используя язык YANG 1.1 [RFC7950].

1.1. Мотивация

В современных развёртываниях TWAMP отсутствие стандартизированной модели данных ограничивает гибкость динамической организации измерений на основе TWAMP в среде с оборудованием разных производителей. В крупной динамической инфраструктуре с виртуализацией, где сетевые функции размещаются в соответствии с алгоритмами оркестровки, фирменные механизмы управления измерениями TWAMP вносят существенные ограничения по части программируемости.

Есть две основные тенденции стандартизации управления TWAMP. Во-первых, предполагается, что в ближайшие годы крупномасштабное развёртывание TWAMP от разных производителей станет нормой. С точки зрения эксплуатации использование нескольких механизмов настройки TWAMP, зависящих от производителя, является дорогим и неэффективным по сравнению с применением одного стандартного механизма. Во-вторых, рост числа программно-управляемых и виртуализованных сетевых систем, основанных на динамических цепочках служб [NSC] с программируемыми плоскостями управления [RFC7426], требует чётко определённой модели данных для реализации TWAMP. Этот документ определяет такую модель данных TWAMP и задаёт её формально с использованием языка моделирования данных YANG 1.1 [RFC7950].

1.2. Уровни требований

Ключевые слова **должно** (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.

1.3. Организация документа

В разделе 2 описана область действия и применимость документа, раздел 3 содержит высокоуровневый обзор модели данных TWAMP. В разделе 4 описаны параметры конфигурации модели данных, а раздел 5 задаёт модель данных YANG для TWAMP. В разделе 6 приведены примеры, соответствующие описанной модели данных YANG, приложение А содержит подробное описание этих примеров.

2. Область действия, модель и применимость

Целью этого документа является спецификация независимой от производителя модели данных для реализаций TWAMP.

На рисунке 1 приведена логическая модель TWAMP, заимствованная из параграфа 1.2 спецификации TWAMP [RFC5357], с указателями на диаграммы UML [UML], приведённые в этом документе и связанные с моделями данных 4 логических элементов системы TWAMP - Control-Client, Server, Session-Sender, Session-Reflector. Руководство по обозначениям UML (Notation Guide) приведено в разделе 5 [UML].

В соответствии с TWAMP [RFC5357] непомеченные соединения на рисунке 1 не задаются и могут быть фирменными протоколами.

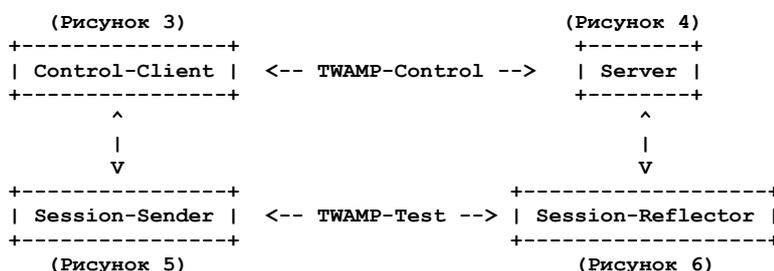


Рисунок 1. Логическая модель TWAMP.

Согласно TWAMP [RFC5357], реализация TWAMP может следовать упрощённой логической модели, в которой один узел может выступать в качестве Control-Client и Session-Sender, а другой быть сервером TWAMP и Session-Reflector. На рисунке 2 показана упрощённая логическая модель и взаимодействия между клиентом конфигурации и сервером TWAMP, например в NETCONF [RFC6241] или RESTCONF [RFC8040].

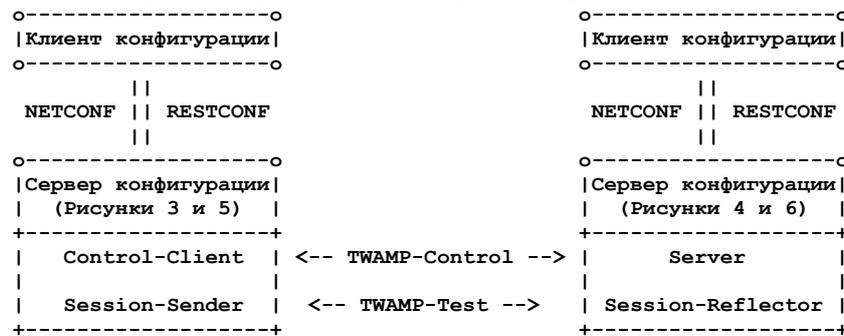


Рисунок 2. Упрощённая модель TWAMP и протоколы.

Определённая в этом документе модель ортогональна протоколу, используемому между клиентом и сервером конфигурации для передачи параметров настройки TWAMP.

Операционные действия, такие как запуск и остановка сессий TWAMP-Test, извлечение результатов теста и т. п., не рассматриваются в описанной здесь модели конфигурации. Как отмечено выше, такие действия не являются частью спецификации TWAMP [RFC5357] и поэтому выходят за рамки документа (см. Приложение В). Кроме того, для рабочего состояния может применяться информация из реестра метрик производительности (Performance Metrics Registry) [RFC8911] [PERF-METRICS], позволяющая создать независимую модель для параметров производительности, которые нужно собрать и извлечь.

3. Обзор модели данных

Модель данных TWAMP включает 4 категории элементов конфигурации.

1. Элементы глобальной конфигурации задаются на уровне устройства. Например, это может быть административный статус сеансов TWAMP, а в случае их разрешения - возможности сессии (например, Control-Client, сервер или оба сразу).
2. Атрибуты, задаваемые на уровне соединения TWAMP-Control, такие как IP-адрес сервера.
3. Атрибуты сессии TWAMP-Test, например, различные значения поля DSCP (Differentiated Services Code Point).
4. Атрибуты рабочего состояния реализации TWAMP.

Поскольку модель данных TWAMP описана в последующих разделах документа, читателю следует помнить о группировке функциональных элементов, показанной на рисунке 1.

3.1. Control-Client

TWAMP Control-Client имеет поле административного статуса, установленное на уровне устройства, которое показывает, включена ли функция на устройстве. Каждый TWAMP Control-Client может быть связан с одним или несколькими соединениями TWAMP-Control. Основные параметры управляющих соединений указаны ниже.

- Имя, которое может служить для однозначного указания конкретного управляющего соединения на Control-Client. Имя требуется для программируемости, поскольку в момент организации соединения TWAMP-Control доступна не вся информация об адресе IP и номере порта TCP, требуемая для однозначного указания соединения.
- Адрес IP на интерфейсе, который Control-Client будет использовать для соединений.
- IP-адрес удалённого сервера TWAMP.
- Атрибуты проверки подлинности и шифрования, такие как KeyID, Token и вектор инициализации Control-Client (Client-IV), описанные в параграфе 3.1 документа «A One-way Active Measurement Protocol (OWAMP)» [RFC4656] и документе «Randomness Requirements for Security» [RFC4086].

Каждое соединение TWAMP-Control может быть связано с одной или несколькими сессиями TWAMP-Test. Для каждого сеанса тестирования следует указывать перечисленные ниже параметры конфигурации.

- Имя тестовой сессии, которое однозначно указывает конкретный сеанс на уровне Control-Client и Session-Sender. Подобно указанным выше управляющим соединениям это уникальное имя сеанса нужно по причине того, что на момент создания сессии TWAMP-Test такие параметры, как номер порта UDP ещё не известны.
- Адрес IP и номер порта UDP в Session-Sender на пути, тестируемом с помощью TWAMP.
- Адрес IP и номер порта UDP в Session-Reflector на пути, тестируемом с помощью TWAMP.
- Сведения, относящиеся к потоку тестовых пакетов, такие как время запуска теста, применяемая метрика производительности (Performance Metric) в соответствии с «Registry for Performance Metrics» [RFC8911], повторяемость теста.

3.2. Сервер

Каждый сервер TWAMP имеет поле административного статуса, устанавливаемое на уровне устройства, для управления возможностью включения функции TWAMP Server.

Каждый сервер может иметь одно или несколько соединений TWAMP-Control, каждое из которых однозначно указывается кваттетом {IP-адрес Control-Client, номер порта TCP у Control-Client, IP-адрес сервера, номер порта TCP на сервере}. Элементы конфигурации управляющего соединения TWAMP Server доступны лишь для чтения.

3.3. Session-Sender

TWAMP Session-Sender имеет поле административного статуса, устанавливаемое на уровне устройства, для управления возможностью включения функции.

Для каждой сессии TWAMP-Test имеется один экземпляр Session-Sender, иницируемый передающим устройством. Основные поля конфигурации указаны ниже.

- Имя тестовой сессии, которое **должно** совпадать с именем соответствующего сеанса тестирования на TWAMP Control-Client (параграф 3.1).
- Имя управляющего соединения, которое вместе с именем сеанса тестирования однозначно указывает экземпляр TWAMP Session-Sender.
- Сведения, относящиеся к потоку тестовых пакетов, такие как число пакетов и их распределение (см. «Network performance measurement with periodic streams» [RFC3432]).

3.4. Session-Reflector

TWAMP Session-Reflector имеет поле административного статуса, устанавливаемое на уровне устройства, для управления возможностью включения функции.

С каждым Session-Reflector может быть связана одна или несколько сессий TWAMP-Test, для каждой из которых можно настроить тайм-аут REFWAIT, управляющий прерыванием сессии при отсутствии принимаемых пакетов (параграф 4.2 в TWAMP [RFC5357]).

Предусмотрен доступ для чтения других параметров модели данных, таких как IP-адрес отправителя. Каждая тестовая сессия однозначно указывается кваттетом, указанным в параграфе 3.2.

4. Параметры модели данных

В этом разделе определена модель данных TWAMP с использованием UML [UML] и вводятся отдельные параметры, связанные с 4 логическими элементами TWAMP. Полная спецификация модели данных TWAMP в форме модуля YANG представлена в параграфе 5.2.

4.1. Control-Client

Контейнер клиента (рисунок 3) содержит элементы, относящиеся к логическому элементу TWAMP Control-Client (рисунок 1). Этот контейнер включает административный параметр конфигурации (client/admin-state), управляющий возможностью иницировать соединения TWAMP-Control.

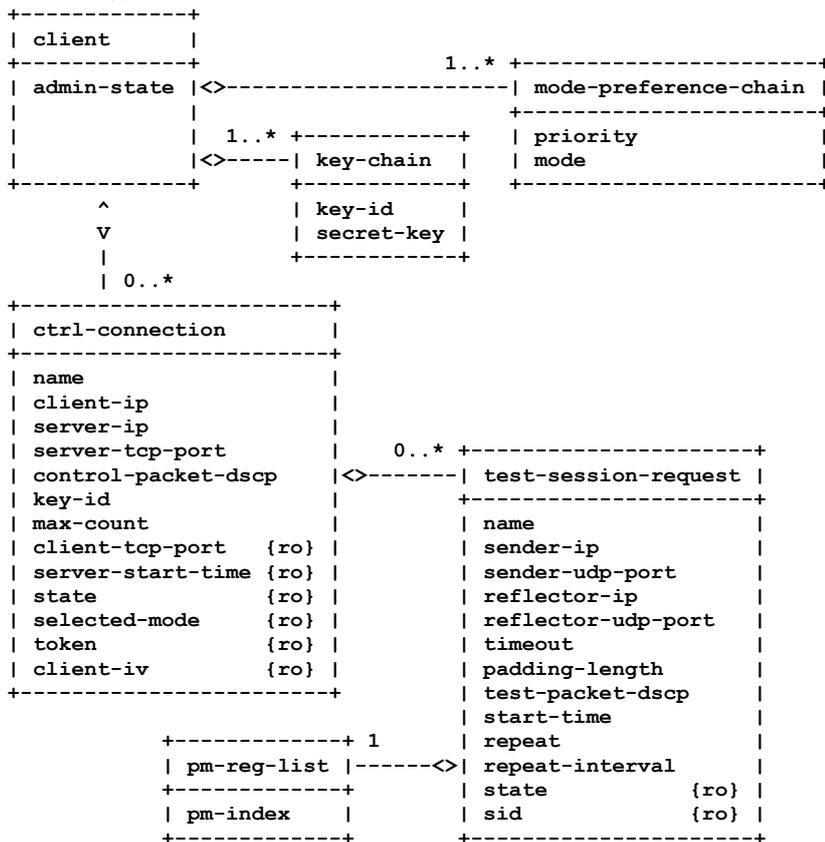


Рисунок 3. Диаграмма классов UML для TWAMP Control-Client.

Контейнер client содержит список (mode-preference-chain), задающий режим значений в соответствии с предпочтительным порядком их использования оператором этого Control-Client, включая режимы проверки подлинности и шифрования. В частности, списки mode-preference-chain содержат режим и соответствующий приоритет в форме 16-битовых целых чисел без знака. Значения приоритета начинаются с 0 (высший) и увеличиваются на 1.

Среди режимов, доступных в Server Greeting, Control-Client **должен** выбрать из списка mode-preference-chain режим с наивысшим приоритетом. В списке предпочтительных режимов может одновременно устанавливаться несколько битовых позиций, например, при обращении к расширенным функциям TWAMP в «Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)» [RFC5618], «Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)» [RFC5938], «Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features» [RFC6038], «IKEv2-Derived Shared Secret Key for the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)» [RFC7717]. Если Control-Client не может определить подходящий режим или комбинация битов не имеет смысла (например, установлены биты authenticated и unauthenticated), он **должен** ответить Set-Up-Response со сброшенными битами Mode, указывающим нежелание продолжать управляющее соединение.

Кроме того в контейнере client содержится список key-chain, связывающий key-id с соответствующим секретным ключом. Сервер и Control-Client используют одно отображение key-id на secret-key (на рисунке 3) и для того, чтобы это работало корректно, значение key-id должно быть уникальным среди всех систем в административном домене. Сервер, подготовленный для сеансов с разными Control-Client, использует key-id для выбора подходящего секретного ключа, а Control-Client обычно использует свой секретный ключ для каждого сервера. Ключ secret-key является общим секретом типа binary и ему **следует** иметь не менее 128 битов энтропии. Представлению key-id и secret-key **следует** соответствовать параграфу 9.8 в YANG [RFC7950]. Размер производного ключа (dkLen в «PKCS #5: Password-Based Cryptography Specification Version 2.1» [RFC8018]) **должен** быть не меньше 16 октетов для AES Session-key при шифровании и 32 октетов для HMAC-SHA1 Session-key при проверке подлинности (см. также параграф 6.10 в OWAMP [RFC4656]).

В каждом контейнере client также содержится список управляющих соединений, каждый элемент которого описывает соединение TWAMP-Control, иницируемое этим Control-Client. **Нужно** указать одно управляющее соединение (ctrl-connection) на соединение TWAMP-Control (TCP), иницируемое с данного устройства.

В каждом ctrl-connection содержится список test-session-request с информацией, связанной с Control-Client для этого сеанса тестирования. Включаются сведения, связанные с обменом сообщениями Request-TW-Session и Accept-Session (см. параграф 3.5 в TWAMP [RFC5357]).

Нужно указать один экземпляр test-session-request для каждой сессии TWAMP-Test, которая должна согласовываться этим соединением TWAMP-Control через обмен сообщениями Request-TW-Session, Accept-Session.

Control-Client также отвечает за планирование сеансов TWAMP-Test, поэтому в test-session-request содержатся сведения об этих действиях (например, pm-index, repeat-interval).

4.2. Сервер

Контейнер server (рисунок 4) содержит элементы, относящиеся к конфигурации логического объекта TWAMP Server (рисунок 1). Контейнер включает административный параметр конфигурации (server/admin-state), управляющий возможностью восприятия устройством соединений TWAMP-Control.

Устройство, работающее в роли сервера (Server Role) не может задавать атрибуты на уровне соединений TWAMP-Control, поскольку оно не знает заранее о входящих соединениях TWAMP-Control. Поэтому все параметры, которые сервер может захотеть применять для входящих соединений, должны настраиваться на уровне сервера и применяются ко всем входящим соединениям TWAMP-Control.

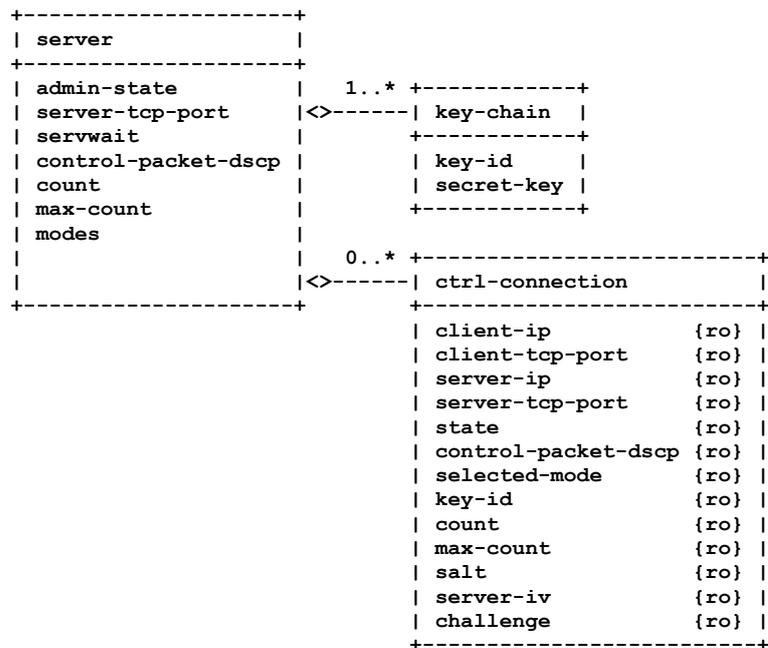


Рисунок 4. Диаграмма классов UML для сервера TWAMP.

Каждый контейнер server содержит список key-chain, связывающий key-id с соответствующим secret-key. Как отмечено в параграфе 4.1, Server и Control-Client используют одно сопоставление key-id с общим ключом secret-key и для корректной работы значение key-id должно быть уникальным для каждой системы в административном домене. Сервер, подготовленный к работе с несколькими Control-Client, использует key-id для выбора подходящего secret-key, а Control-Client обычно используют отдельный секретный ключ для каждого сервера. Значение key-id указывает серверу, какой из ключей secret-key клиент Control-Client желает применять для проверки подлинности и шифрования.

Каждое активное входящее управляющее соединение на сервере представляется объектом ctrl-connection. Нужно указать один объект ctrl-connection на каждое входящее соединение TWAMP-Control (TCP), которое воспринято и

активно на сервере. Каждый объект ctrl-connection может быть однозначно указан квартетом {client-ip, client-tcp-port, server-ip, server-tcp-port}. Все элементы списка ctrl-connection доступны лишь для чтения.

4.3. Session-Sender

Контейнер session-sender, показанный на рисунке 5, содержит элементы, относящиеся к логическому объекту TWAMP Session-Sender. В session-sender помещается административный параметр (session-sender/admin-state) управляющий возможностью устройства инициировать сессии TWAMP-Test.

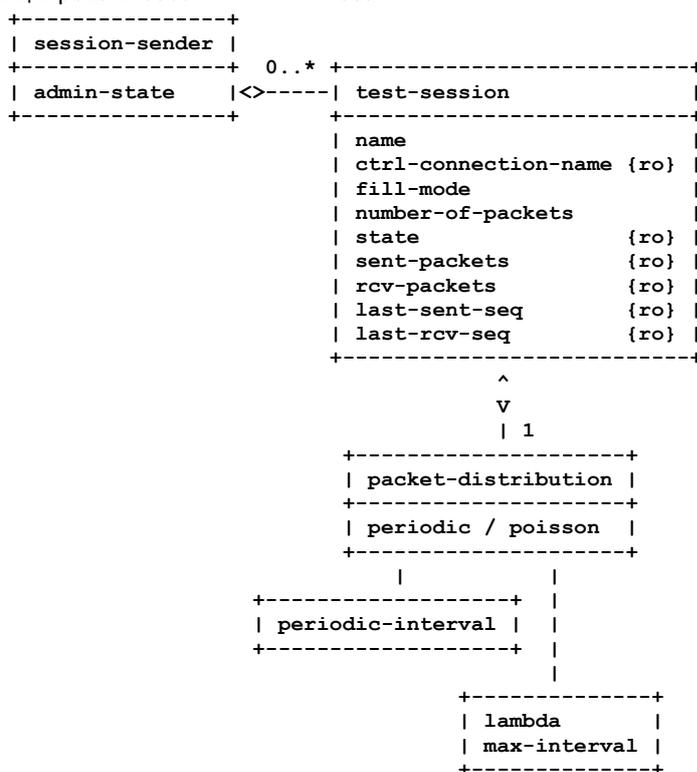


Рисунок 5. Диаграмма классов UML для Session-Sender.

Каждая сессия TWAMP-Test, созданная Session-Sender, будет представляться объектом test-session. **Нужно** создавать 1 объект test-session для каждой сессии TWAMP-Test, в которой будут передаваться пакеты.

4.4. Session-Reflector

Контейнер session-reflector, показанный на рисунке 6, содержит элементы, относящиеся к конфигурации логического объекта TWAMP Session-Reflector. В session-reflector включается административный параметр (session-reflector/admin-state), управляющий возможностью устройства воспринимать входящие сессии TWAMP-Test.

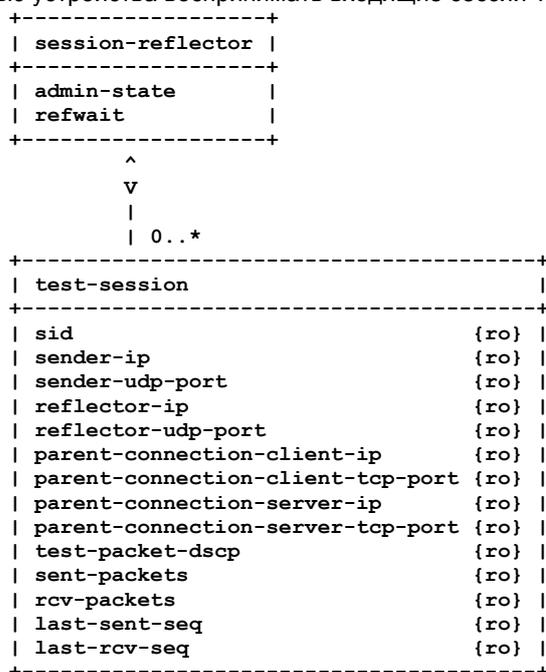


Рисунок 6. Диаграмма классов UML для Session-Reflector.

Устройство в роли Session-Reflector не может настраивать атрибуты на уровне сессии, поскольку заранее не знает о входящих сеансах. Поэтому все параметры, которые Session-Reflector может пожелать применить к входящим сессиям TWAMP-Test, должны настраиваться на уровне Session-Reflector и будут применяться ко всем входящим сессиям. Каждую входящую сессию TWAMP-Test, активную в Session-Reflector, **нужно** представлять 1 экземпляром объекта test-session. Все элементы test-session доступны лишь для чтения.

Экземпляры test-session индексируются идентификатором сессии (Session Identifier или SID) в параметре sid. Значения SID автоматически выделяет TWAMP Server по мере получения запросов на сеансы тестирования и возвращается их клиентам Control-Client в поле SID сообщения Accept-Session, как указано в параграфе 4.3 «Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features» [RFC6038].

При попытке получить рабочие данные для сеансов тестирования от устройства Session-Reflector пользователь не знает, какие сессии в данный момент активны на устройстве и какие SID выделены для них. Если у пользователя есть сетевой доступ к устройству Control-Client, он может прочитать данные для сессии из client/ctrl-connection/test-session-request/sid и получить значение SID (рисунок 3). Это значение SID можно использовать в качестве индекса для получения отдельного экземпляра session-reflector/test-session с устройства Session-Reflector. Если у пользователя нет доступа к Control-Client через сеть, единственным способом является извлечение всех экземпляров test-session с устройства Session-Reflector и поиск среди них нужного пользователю. Это может быть проблематично при большом числе активных сессий на устройстве.

Каждая сессия TWAMP-Test на устройстве Session-Reflector включает квартет {parent-connection-client-ip, parent-connection-client-tcp-port, parent-connection-server-ip, parent-connection-server-tcp-port}, который **должен** соответствовать эквивалентному квартету {client-ip, client-tcp-port, server-ip, server-tcp-port} в server/ctrl-connection. Этот квартет позволяет пользователю отслеживать сессию TWAMP-Test до (родительского) соединения TWAMP-Control, согласовавшего сеанс тестирования.

5. Модель данных

В этом разделе приведена формальная спецификация модели данных TWAMP с использованием YANG.

5.1. Диаграмма дерева YANG

В этом параграфе дано упрощённое графическое представление модели данных TWAMP с использованием дерева YANG. Читателю следует помнить, что ограничение размера строки 72 символами ведёт к искусственному переводу строк в некоторых узлах дерева. В этом документе используются диаграммы с нотацией, заданной в «YANG Tree Diagrams» [RFC8340].

Следует отметить, что символ \ в конце строки применяется в этом документе для разделения длинных строк на несколько (например, [reflector-udp-port] в диаграмме дерева является частью [sender-ip sender-udp-port reflector-ip]).

```

module: ietf-twamp
  +--rw twamp
    +--rw client {control-client}?
      | +--rw admin-state?          boolean
      | +--rw mode-preference-chain* [priority]
      | | +--rw priority            uint16
      | | +--rw mode?               twamp-modes
      | +--rw key-chain* [key-id]
      | | +--rw key-id              string
      | | +--rw secret-key?        binary
      | +--rw ctrl-connection* [name]
      | | +--rw name                 string
      | | +--rw client-ip?          inet:ip-address
      | | +--rw server-ip           inet:ip-address
      | | +--rw server-tcp-port?    inet:port-number
      | | +--rw control-packet-dscp? inet:dscp
      | | +--rw key-id?             string
      | | +--rw max-count-exponent? uint8
      | | +--ro client-tcp-port?    inet:port-number
      | | +--ro server-start-time?  uint64
      | | +--ro repeat-count?       uint64
      | | +--ro state?
      | | | control-client-connection-state
      | | +--ro selected-mode?      twamp-modes
      | | +--ro token?              binary
      | | +--ro client-iv?          binary
      | +--rw test-session-request* [name]
      | | +--rw name                 string
      | | +--rw sender-ip?          inet:ip-address
      | | +--rw sender-udp-port?    union
      | | +--rw reflector-ip        inet:ip-address
      | | +--rw reflector-udp-port? inet:port-number
      | | +--rw timeout?            uint64
      | | +--rw padding-length?     uint32
      | | +--rw test-packet-dscp?   inet:dscp
      | | +--rw start-time?         uint64
      | | +--rw repeat?             uint32
      | | +--rw repeat-interval?    uint32
      | | +--rw pm-reg-list* [pm-index]
      | | | +--rw pm-index          uint16
      | | +--ro state?              test-session-state
      | | +--ro sid?                string
    +--rw server {server}?
      | +--rw admin-state?          boolean
      | +--rw server-tcp-port?      inet:port-number
      | +--rw servwait?             uint32
      | +--rw control-packet-dscp?  inet:dscp
      | +--rw count?                uint8
      | +--rw max-count-exponent?   uint8
      | +--rw modes?                twamp-modes
  
```

```

| +--rw key-chain* [key-id]
| | +--rw key-id      string
| | +--rw secret-key? binary
| +--ro ctrl-connection*
|   [client-ip client-tcp-port server-ip server-tcp-port]
|   +--ro client-ip      inet:ip-address
|   +--ro client-tcp-port inet:port-number
|   +--ro server-ip      inet:ip-address
|   +--ro server-tcp-port inet:port-number
|   +--ro state?         server-ctrl-connection-state
|   +--ro control-packet-dscp? inet:dscp
|   +--ro selected-mode? twamp-modes
|   +--ro key-id?        string
|   +--ro count?         uint8
|   +--ro max-count-exponent? uint8
|   +--ro salt?          binary
|   +--ro server-iv?     binary
|   +--ro challenge?     binary
+--rw session-sender {session-sender}?
| +--rw admin-state?    boolean
| +--rw test-session* [name]
| | +--rw name          string
| | +--ro ctrl-connection-name? string
| | +--rw fill-mode?    padding-fill-mode
| | +--rw number-of-packets uint32
| | +--rw (packet-distribution)?
| | | +--:(periodic)
| | | | +--rw periodic-interval    decimal64
| | | +--:(poisson)
| | | | +--rw lambda                decimal64
| | | | +--rw max-interval?         decimal64
| | +--ro state?                    sender-session-state
| | +--ro sent-packets?              uint32
| | +--ro rcv-packets?              uint32
| | +--ro last-sent-seq?            uint32
| | +--ro last-rcv-seq?            uint32
+--rw session-reflector {session-reflector}?
+--rw admin-state?    boolean
+--rw refwait?        uint32
+--ro test-session*
  [sender-ip sender-udp-port reflector-ip \
  reflector-udp-port]
+--ro sid?                string
+--ro sender-ip           inet:ip-address
+--ro sender-udp-port
|   dynamic-port-number
+--ro reflector-ip        inet:ip-address
+--ro reflector-udp-port  inet:port-number
+--ro parent-connection-client-ip? inet:ip-address
+--ro parent-connection-client-tcp-port? inet:port-number
+--ro parent-connection-server-ip? inet:ip-address
+--ro parent-connection-server-tcp-port? inet:port-number
+--ro test-packet-dscp?   inet:dscp
+--ro sent-packets?      uint32
+--ro rcv-packets?      uint32
+--ro last-sent-seq?    uint32
+--ro last-rcv-seq?    uint32

```

Рисунок 7. Диаграмма дерева YANG.

5.2. Модуль YANG

В этом параграфе представлен модуль YANG для модели данных TWAMP, заданной этим документом. Модуль импортирует определения из «Common YANG Data Types» [RFC6991] и ссылается на документы «Framework for IP Performance Metrics» [RFC2330], «Network performance measurement with periodic streams» [RFC3432], «A One-way Active Measurement Protocol (OWAMP)» [RFC4656], «A Two-Way Active Measurement Protocol (TWAMP)» [RFC5357], «Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)» [RFC5618], «Network Time Protocol Version 4: Protocol and Algorithms Specification» [RFC5905], «Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)» [RFC5938], «Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features» [RFC6038], «Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)» [RFC7312], «IKEv2-Derived Shared Secret Key for the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)» [RFC7717], «Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)» [RFC8545], «Registry for Performance Metrics» [RFC8911].

```

<CODE BEGINS> file "ietf-twamp@2021-11-17.yang"
module ietf-twamp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-twamp";
  prefix ietf-twamp;

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

```

```

organization
  "IETF IPPM (IP Performance Metrics) Working Group";
contact
  "WG Web: <https://datatracker.ietf.org/wg/ippm/documents/>
  WG List: <mailto:ippm@ietf.org>

  Editor: Ruth Civil
         <mailto:ruthcivil@gmail.com>

  Editor: Al Morton
         <mailto:acmorton@att.com>

  Editor: Reshad Rahman
         <mailto:reshad@yahoo.com>

  Editor: Mahesh Jethanandani
         <mailto:mjethanandani@gmail.com>

  Editor: Kostas Pentikousis
         <mailto:kostas.pentikousis@detecon.com>";
description
  "Этот модуль YANG задаёт независимую от производителя модель
  данных для протокола TWAMP (Two-Way Active Measurement Protocol).

  Модель определяет 4 логических элемента TWAMP - Control-Client,
  Server, Session-Sender, Session-Reflector, как показано в
  логической модели TWAMP (рисунок 1 в RFC 8913).

  Модуль YANG использует функции (feature) для индикации логических
  элементов, поддерживаемых реализацией TWAMP.

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

  Авторские права (Copyright (c) 2021) 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 8913, где правовые
  аспекты приведены более полно.";
revision 2021-11-17 {
  description
    "Исходный выпуск.

    Ссылается на RFC 5357, RFC 5618, RFC 5938, RFC 6038, RFC 7717,
    RFC 8911.";
  reference
    "RFC 8913: Two-Way Active Measurement Protocol (TWAMP) YANG
    Data Model";
}

/*
 * Определения типов
 */

typedef twamp-modes {
  type bits {
    bit unauthenticated {
      position 0;
      description
        "Режим без аутентификации, где не применяется проверка
        подлинности и шифрование в TWAMP-Control и TWAMP-Test.
        KeyID, Token, Client-IV не применяются в сообщениях
        Set-Up-Response. См. параграф 3.1 в RFC 4656.";
      reference
        "RFC 4656: A One-way Active Measurement Protocol (OWAMP),
        параграф 3.1";
    }
    bit authenticated {
      position 1;
      description
        "режим с аутентификацией, в котором Control-Client и
        Server владеют общим секретом, не позволяющим 'кражу
        услуг'. В соответствии с разделом 6 RFC 4656, в режиме
        с аутентификацией 'временные метки передаются открыто
        без криптографической защиты, а остальная часть сообщения
        имеет такую же защиту как в режиме с шифрованием. Этот
        режим обеспечивает компромисс между защитой и точностью.'";
    }
  }
}

```

```
reference
  "RFC 4656: A One-way Active Measurement Protocol (OWAMP),
  раздел 6";
}
bit encrypted {
  position 2;
  description
    "режим с шифрованием 'делает невозможным незаметное изменение
    временных меток' (раздел 1 в RFC 4656). См. также раздел 4
    в RFC 7717.";
  reference
    "RFC 4656: A One-way Active Measurement Protocol (OWAMP),
    раздел 6
    RFC 7717: IKEv2-Derived Shared Secret Key for the One-Way
    Active Measurement Protocol (OWAMP) and Two-Way Active
    Measurement Protocol (TWAMP), раздел 4";
}
bit unauth-test-encrypt-control {
  position 3;
  description
    "При использовании смешанного режима защиты протокол TWAMP-Test
    работает в режиме без аутентификации, а TWAMP-Control – в
    режиме с шифрованием.";
  reference
    "RFC 5618: Mixed Security Mode for the Two-Way Active
    Measurement Protocol (TWAMP)";
}
bit individual-session-control {
  position 4;
  description
    "Этот режим разрешает отдельные тестовые сессии, использующие
    Session Identifier.";
  reference
    "RFC 5938: Individual Session Control Feature
    for the Two-Way Active Measurement Protocol (TWAMP)";
}
bit reflect-octets {
  position 5;
  description
    "Этот режим указывает свойство отражения пакетов.";
  reference
    "RFC 6038: Two-Way Active Measurement Protocol (TWAMP)
    Reflect Octets and Symmetrical Size Features";
}
bit symmetrical-size {
  position 6;
  description
    "Этот режим указывает поддержку симметричного размера
    тестовых пакетов отправителя.";
  reference
    "RFC 6038: Two-Way Active Measurement Protocol (TWAMP)
    Reflect Octets and Symmetrical Size Features";
}
bit IKEv2Derived {
  position 7;
  description
    "В этом режиме общий ключ выводится из защищённой связи (SA)
    протокола Internet Key Exchange Protocol Version 2 (IKEv2).";
  reference
    "RFC 7717: IKEv2-Derived Shared Secret Key for
    the One-Way Active Measurement Protocol (OWAMP)
    and Two-Way Active Measurement Protocol (TWAMP)";
}
}
description
  "задаёт настраиваемые режимы TWAMP-Mode, поддерживаемые при
  организации соединения TWAMP-Control между Control-Client
  и Server. В разделе 7 RFC 7717 описан реестр TWAMP-Modes и
  указаны спецификации форматов.";
}

typedef control-client-connection-state {
  type enumeration {
    enum active {
      description
        "Указывает активное соединение TWAMP-Control с сервером.";
    }
    enum idle {
      description
        "Указывает неактивное соединение TWAMP-Control с сервером.";
    }
  }
}
description
  "Указывает состояние соединения TWAMP-Control у Control-Client.";
}
```

```
typedef test-session-state {
    type enumeration {
        enum accepted {
            value 0;
            description
                "Указывает воспринятый запрос сессии TWAMP-Test.";
        }
        enum failed {
            value 1;
            description
                "Указывает отказ для сессии TWAMP-Test по неуказанной
                причине (catch-all).";
        }
        enum internal-error {
            value 2;
            description
                "Указывает отказ для сессии TWAMP-Test в результате
                внутренней ошибки.";
        }
        enum not-supported {
            value 3;
            description
                "Указывает отказ для сессии TWAMP-Test по причине того,
                что некоторые аспекты запроса сессии не поддерживаются.";
        }
        enum permanent-resource-limit {
            value 4;
            description
                "Указывает отказ для сессии TWAMP-Test по причине
                постоянной ограниченности ресурсов.";
        }
        enum temp-resource-limit {
            value 5;
            description
                "Указывает отказ для сессии TWAMP-Test по причине
                временной ограниченности ресурсов.";
        }
    }
    description
        "Указывает состояние сессии TWAMP-Test у Control-Client.";
}

typedef server-ctrl-connection-state {
    type enumeration {
        enum active {
            description
                "Указывает активное соединение TWAMP-Control с
                Control-Client.";
        }
        enum servwait {
            description
                "Указывает, что соединение TWAMP-Control с Control-Client
                находится в состоянии SERVWAIT, как указано в параграфе
                3.1 RFC 5357.";
            reference
                "RFC 5357: A Two-Way Active Measurement Protocol (TWAMP),
                параграф 3.1";
        }
    }
    description
        "Указывает состояние соединения TWAMP-Control на сервере.";
}

typedef sender-session-state {
    type enumeration {
        enum active {
            description
                "Указывает, что сессия TWAMP-Test активна.";
        }
        enum failure {
            description
                "Указывает отказ сессии TWAMP-Test.";
        }
    }
    description
        "Указывает состояние сессии TWAMP-Test у Session-Sender.";
}

typedef padding-fill-mode {
    type enumeration {
        enum zero {
            description
                "Пакеты TWAMP-Test дополняются нулями.";
        }
        enum random {
            description

```

```
"Пакеты TWAMP-Test дополняются псевдослучайными числами.";
}
}
description
  "Указывает тип заполнения в пакетах TWAMP-Test.";
}

typedef dynamic-port-number {
  type inet:port-number {
    range "49152..65535";
  }
  description
    "Диапазон динамических номеров порта.";
}

/*
 * Функции
 */

feature control-client {
  description
    "Указывает, что устройство поддерживает конфигурацию
    логического элемента TWAMP Control-Client.";
}

feature server {
  description
    "Указывает, что устройство поддерживает конфигурацию
    логического элемента TWAMP Server.";
}

feature session-sender {
  description
    "Указывает, что устройство поддерживает конфигурацию
    логического элемента TWAMP Session-Sender.";
}

feature session-reflector {
  description
    "Указывает, что устройство поддерживает конфигурацию
    логического элемента TWAMP Session-Reflector.";
}

/*
 * Группы узлов с многократным использованием
 */

grouping key-management {
  list key-chain {
    key "key-id";
    leaf key-id {
      type string {
        length "1..80";
      }
    }
    description
      "KeyID применяется для соединения TWAMP-Control. Согласно
      параграфу 3.1 в RFC 4656, KeyID является строкой UTF-8
      размером до 80 октетов и служит для выбора общего секрета,
      который клиент Control-Client хочет применять для
      аутентификации или шифрования'.";
  }
  leaf secret-key {
    type binary;
    description
      "Секретный ключ, соответствующий KeyID для этого соединения
      TWAMP-Control.";
  }
  description
    "Связывает KeyID с секретными ключами в соединении TWAMP-Control.";
}
description
  "Применяется сервером и Control-Client для управления ключами
  TWAMP-Control.";
}

grouping maintenance-statistics {
  leaf sent-packets {
    type uint32;
    config false;
    description
      "Указывает число переданных пакетов.";
  }
  leaf rcv-packets {
    type uint32;
    config false;
    description

```

```

    "Указывает число принятых пакетов .";
}
leaf last-sent-seq {
    type uint32;
    config false;
    description
        "Указывает последний переданный порядковый номер.";
}
leaf last-rcv-seq {
    type uint32;
    config false;
    description
        "Указывает последний принятый порядковый номер .";
}
description
    "Используется для статистики обслуживания TWAMP-Test.";
}

grouping count {
    leaf count {
        type uint8 {
            range "10..31";
        }
        default "15";
        description
            "Параметр передаётся клиенту Control-Client как часть
            сообщения Server Greeting и служит для вывода ключа по
            общему секрету в соответствии с параграфом 3.1 в RFC 4656.
            (это ДОЛЖНА быть степень числа 2 не меньше 1024). Значение
            определяет указанная степень. Например, указание здесь
            числа 20 означает  $2^{20} = 1048576$ . По умолчанию задано 15,
             $2^{15} = 32768$ .";
    }
    description
        "Структуры данных многократного использования для счётчиков,
        применяемые в Server и Control-Client.";
}

grouping max-count-exponent {
    leaf max-count-exponent {
        type uint8 {
            range "10..31";
        }
        default "20";
        description
            "Ограничивает максимальное значение Count, которое ДОЛЖНО
            быть степенью числа 2 и не меньше 1024 в соответствии с
            RFC 5357. Задаётся указанием степени. Например, 10 означает
            максимальное значение счётчика  $2^{10} = 1024$ . По умолчанию
            задано 20,  $2^{20} = 1048576$ .

            TWAMP Server использует это значение в сообщении
            Server Greeting, передаваемом клиенту Control-Client.

            TWAMP Control-Client использует заданное значение для
            предотвращения DoS-атак1) путём разрыва управляющего соединения
            с сервером при получении сообщения Server-Greeting с Count
            больше заданного максимума (раздел 6 в RFC 5357).

            Кроме того, в соответствии с разделом 6 в RFC 5357

            'Если атакующий установит максимальное значение Count
            ( $2^{32}$ ), атакуемая система «остановится» на продолжительное
            время, пытаясь создать ключи. Поэтому совместимым с TWAMP
            системам СЛЕДУЕТ иметь конфигурационную возможность
            ограничивать максимальное значение Count. По умолчанию для
            Count СЛЕДУЕТ задавать максимальное значение 32768.'

            В случае этого документа для max-count-exponent по умолчанию
            СЛЕДУЕТ устанавливать значение 15, соответствующее
            максимальному значению  $2^{15}$  или 32768.

            RFC 5357 не уточняет «продолжительное время», но ясно, что
            оно зависит от доступных вычислительных ресурсов и операторам
            нужно учитывать это при оценке безопасности.";
    }
    description
        "Многократно применяемая структура данных для max-count
        используется в контейнерах сервера и Control-Client.";
}

/*
 * Узлы данных конфигурации

```

¹Denial-of-service - отказ в обслуживании.

```

*/
container twamp {
  description
  "Конфигурация логических элементов TWAMP состоит из 4 моделей,
  соответствующих 4 логическим элементам TWAMP - Control-Client,
  Server, Session-Sender, Session-Reflector на рисунке 1 в
  RFC 8913.";
  container client {
    if-feature "control-client";
    description
    "Конфигурация логического элемента TWAMP Control-Client.";
    leaf admin-state {
      type boolean;
      default "true";
      description
      "Указывает, может ли устройство быть TWAMP Control-Client.";
    }
    list mode-preference-chain {
      key "priority";
      unique "mode";
      leaf priority {
        type uint16;
        description
        "Указывает приоритет режима Control-Client 16-битовым
        целым числом без знака. Значения приоритета начинаются с
        0 (высший) и увеличиваются на 1 для последующих.";
      }
      leaf mode {
        type twamp-modes;
        description
        "Поддерживаемые режимы TWAMP-Mode соответствующие
        приоритету.";
      }
    }
    description
    "Указывает предпочтительный порядок использования клиентом
    Control-Client поддерживаемых TWAMP-Mode.

    Из режимов, указанных в сообщении TWAMP Server Greeting
    (см. рисунок 2 в RFC 7717), Control-Client ДОЛЖЕН выбрать
    наиболее приоритетный в списке mode-preference-chain.";
  }
}
uses key-management;
list ctrl-connection {
  key "name";
  description
  "Список управляющих соединений TWAMP Control-Client.
  Каждая запись списка описывает соединение, которое будет
  инициировано этим Control-Client.";
  leaf name {
    type string;
    description
    "Уникальное имя, служащее ключом для указания отдельного
    соединения TWAMP-Control на устройстве Control-Client.";
  }
  leaf client-ip {
    type inet:ip-address;
    description
    "IP-адрес локального устройства Control-Client для
    включения в поле source IP заголовка IP в пакетах
    TWAMP-Control (TCP), относящихся к этому управляющему
    соединению. Если параметр не задан, устройству НУЖНО
    выбрать один из своих адресов IP.";
  }
  leaf server-ip {
    type inet:ip-address;
    mandatory true;
    description
    "IP-адрес удалённого устройства Server, с которым
    инициируется соединение TWAMP-Control.";
  }
  leaf server-tcp-port {
    type inet:port-number;
    default "862";
    description
    "Этот параметр задаёт номер порта TCP, который будет
    применяться в этом исходящем соединении TWAMP-Control.
    Обычно это общеизвестный порт TWAMP-Control (862),
    заданный в RFC 5357. Однако известны реализации TWAMP,
    созданные до выделения этого номера порта. В таких
    ранних реализациях номер порта можно задавать. Этот
    параметр нужен для совместимости с такими реализациями.";
  }
  leaf control-packet-dscp {
    type inet:dscp;
    default "0";
    description

```

```

"Значение кода дифференцированного обслуживания (DSCP) для
включения в заголовок IP пакетов TWAMP-Control (TCP),
создаваемых этим Control-Client.";
}
leaf key-id {
  type string {
    length "1..80";
  }
  description
    "Указывает значение KeyID выбранное для этого
соединения TWAMP-Control.";
}
uses max-count-exponent;
leaf client-tcp-port {
  type inet:port-number;
  config false;
  description
    "Указывает порт отправителя TCP в пакетах TWAMP-Control,
относящихся к этому управляющему соединению.";
}
leaf server-start-time {
  type uint64;
  config false;
  description
    "Указывает время Start-Time, анонсированное сервером в
сообщении Server-Start (RFC 4656, параграф 3.1), которое
представляет время, когда текущий экземпляр сервера
начал работу. Метка использует формат RFC 5905 в
соответствии с параграфом 4.1.2 RFC 4656.";
  reference
    "RFC 4656: A One-way Active Measurement Protocol (OWAMP),
параграфы 3.1 и 4.1.2
RFC 5905: Network Time Protocol Version 4: Protocol and
Algorithms Specification";
}
leaf repeat-count {
  type uint64;
  config false;
  description
    "Указывает число повторов сеанса тестирования. При работе
теста это значение будет больше 0. Если параметр repeat
не равен 0, это значение будет не больше repeat.";
}
leaf state {
  type control-client-connection-state;
  config false;
  description
    "Указывает текущий статус соединения TWAMP-Control.";
}
leaf selected-mode {
  type twamp-modes;
  config false;
  description
    "режимы TWAMP-Mode, выбранные Control-Client для этого
управляющего соединения в соответствии с полем Mode в
сообщении Set-Up-Response.";
  reference
    "RFC 4656: A One-way Active Measurement Protocol (OWAMP),
параграф 3.1";
}
leaf token {
  type binary {
    length "64";
  }
  config false;
  description
    "64 октета, содержащие конкатенацию 16 октетов Challenge,
16 октетов сеансового ключа шифрования AES и 32 октетов
сеансового ключа аутентификации HMAC-SHA1 (см. последний
абзац параграфа 6.10 в RFC 4656).

Если выбран режим, заданный в RFC 7717 (selected-mode),
размер token ограничен 16 октетами.";
  reference
    "RFC 4656: A One-way Active Measurement Protocol (OWAMP),
параграф 6.10
RFC 7717: IKEv2-Derived Shared Secret Key for the
One-Way Active Measurement Protocol (OWAMP) and
Two-Way Active Measurement Protocol (TWAMP)";
}
leaf client-iv {
  type binary {
    length "16";
  }
  config false;
  description

```

```

"Указывает вектор инициализации Control-Client (Client-IV),
создаваемый клиентом случайным образом (RFC 4656):

'Значение Client-IV просто должно быть уникальным (т. е.
ДОЛЖНО различаться в каждой сессии, использующей тот же
секретный ключ. Простым способом решения этой задачи без
использования громоздкого состояния является создание
Client-IV с использованием криптографически защищённого
источника случайных чисел.'

Если выбран режим, заданный в RFC 7717 (selected-mode),
размер client-iv ограничен 12 октетами.";
reference
"RFC 4656: A One-way Active Measurement Protocol (OWAMP)
RFC 7717: IKEv2-Derived Shared Secret Key for the
One-Way Active Measurement Protocol (OWAMP) and
Two-Way Active Measurement Protocol (TWAMP)";
}
list test-session-request {
  key "name";
  description
    "Информация, связанная с Control-Client для этой
    сессии тестирования.";
  leaf name {
    type string;
    description
      "Уникальное имя для указания этой сессии TWAMP-Test
      на Control-Client.";
  }
  leaf sender-ip {
    type inet:ip-address;
    description
      "IP-адрес устройства Session-Sender, помещаемый в поле
      source IP заголовка IP в пакетах TWAMP-Test (UDP),
      связанных с этой сессией тестирования. Это значение
      помещается в поле Sender Address сообщения
      Request-TW-Session.

      Если значение не задано, устройству НУЖНО выбрать один
      из своих адресов IP.";
  }
  leaf sender-udp-port {
    type union {
      type dynamic-port-number;
      type enumeration {
        enum autoallocate {
          description
            "Указывает, что Control-Client будет автоматически
            выделять номер порта TWAMP-Test (UDP) из
            диапазона динамических портов.";
        }
      }
    }
    default "autoallocate";
    description
      "Номер порта UDP, используемый Session-Sender для этой
      сессии TWAMP-Test. Номер должен браться из
      диапазона динамических номеров.

      По умолчанию устройству Control-Client НУЖНО выделять
      номер порта UDP для сессии TWAMP-Test автоматически.

      Заданное (или выделенное автоматически) значение
      анонсируется в поле Sender Port сообщения
      Request-TW-Session (см. параграф 3.5 в RFC 5357).
      Отметим, что при автоматическом выделении номера порта
      для сессии и параметре repeat, указывающем повтор
      теста устройство может выделить другой номер при
      согласовании следующей (повторной) итерации сессии.";
  }
  leaf reflector-ip {
    type inet:ip-address;
    mandatory true;
    description
      "Адрес IP, относящийся к удалённому устройству
      Session-Reflector, с которым организован сеанс
      TWAMP-Test. Это значение помещается в поле
      Receiver Address сообщения Request-TW-Session.";
  }
  leaf reflector-udp-port {
    type inet:port-number {
      range "862 | 49152..65535";
    }
    description
      "Этот параметр задаёт номер порта UDP, применяемый
      Session-Reflector для этой сессии TWAMP-Test. По

```

```

    умолчанию номер берётся из динамического диапазона
    и помещается в поле Receiver Port сообщения Request-
    TW-Session. МОЖНО указать общеизвестный порт (862).";
reference
"RFC 8545: Well-Known Port Assignments for the One-Way
Active Measurement Protocol (OWAMP) and the Two-Way
Active Measurement Protocol (TWAMP)";
}
leaf timeout {
  type uint64;
  units "seconds";
  default "2";
  description
    "Время (в секундах), в течение которого устройству
    Session-Reflector следует продолжать отвечать на
    пакеты, относящиеся к этой сессии TWAMP-Test, после
    приёма сообщения Stop-Sessions TWAMP-Control.

    Это значение помещается в поле Timeout сообщения
    Request-TW-Session.";
reference
"RFC 5357: A Two-Way Active Measurement Protocol
(TWAMP), параграф 3.5";
}
leaf padding-length {
  type uint32 {
    range "64..4096";
  }
  description
    "Число байтов заполнения, добавляемых в пакеты
    TWAMP-Test (UDP), создаваемые Session-Sender.

    Это значение помещается в поле Padding Length
    сообщения Request-TW-Session.";
reference
"RFC 4656: A One-way Active Measurement Protocol
(OWAMP), параграф 3.5";
}
leaf test-packet-dscp {
  type inet:dscp;
  default "0";
  description
    "Значение DSCP, помещаемое в заголовок IP пакетов
    TWAMP-Test, создаваемых Session-Sender, и в заголовок
    UDP пакетов отклика TWAMP-Test, создаваемых устройством
    Session-Reflector, для этого сеанса тестирования.

    Это значение помещается в поле Type-P Descriptor
    сообщения Request-TW-Session.";
reference
"RFC 5357: A Two-Way Active Measurement Protocol
(TWAMP)";
}
leaf start-time {
  type uint64;
  default "0";
  description
    "Время начала сессии (но не раньше времени ввода
    команды TWAMP Start-Sessions, параграф 3.4 RFC 5357).

    Значение start-time помещается в поле Start Time
    сообщения Request-TW-Session.

    Временная метка соответствует формату RFC 5905, как
    указано в параграфе 3.5 RFC 4656.

    Принятое по умолчанию значение 0 указывает, что сессия
    начнётся сразу при получении сообщения Start-Sessions.";
}
leaf repeat {
  type uint32 {
    range "0..4294967295";
  }
  default "0";
  description
    "Это значение управляет повтором сессии TWAMP-Test. По
    завершении сеанса проверяется значение этого параметра.

    Принятое по умолчанию значение 0 указывает, что повтор
    сессии НЕДОПУСТИМ.

    Если repeat имеет значение от 1 до 4294967294, сеанс
    тестирования НУЖНО повторять с использованием значения
    repeat-interval и родительское соединение TWAMP-Control
    для этой сессии перезапускается для согласования нового
    экземпляра сессии TWAMP-Test.

```

```

Значение 4294967295 указывает, что сеанс НУЖНО
повторять безостановочно с использованием параметра
repeat-interval, не уменьшая значения счётчика.";
}
leaf repeat-interval {
  when "../repeat!='0'" {
    description
      "Этот параметр управляет временем повтора сессий
      TWAMP-Test при значении repeat > 0.

      При repeat-interval = 0 согласование новой сессии
      НУЖНО начинать сразу после завершения прежней. В
      иных случаях Control-Client будет ждать в течение
      заданного repeat-interval числа секунд перед началом
      согласования нового экземпляра сессии TWAMP-Test.";
  }
  type uint32;
  units "seconds";
  default "0";
  description
    "Интервал повторения в секундах.";
}
list pm-reg-list {
  key "pm-index";
  leaf pm-index {
    type uint16;
    description
      "Значение числового индекса Registered Metric в
      Performance Metrics Registry (см. RFC 8911).
      Выходная статистика задаётся соответствующей
      записью реестра.";
  }
  description
    "Список индексов Performance Metrics Registry,
    передающих характеристики потока пакетов с одним
    или несколькими измеряемыми показателями.

    Все элементы pm-reg-list ДОЛЖНЫ иметь общие
    характеристики потока, чтобы их можно было
    комбинировать для указания всех параметров, измеряемых
    в одном потоке.";
  reference
    "RFC 8911: Registry for Performance Metrics";
}
leaf state {
  type test-session-state;
  config false;
  description
    "Указывает состояние сессии TWAMP-Test - принятый
    запрос или ошибка.";
  reference
    "RFC 5357: A Two-Way Active Measurement Protocol
    (TWAMP), параграф 3.5";
}
leaf sid {
  type string;
  config false;
  description
    "Идентификатор сессии (SID), выделенный сервером для
    этого сеанса TWAMP-Test и возвращаемый клиенту
    Control-Client в поле SID сообщения Accept-Session.";
  reference
    "RFC 6038: Two-Way Active Measurement Protocol (TWAMP)
    Reflect Octets and Symmetrical Size
    Features, параграф 4.3";
}
}
}
}
container server {
  if-feature "server";
  description
    "Конфигурация логического элемента TWAMP Server.";
  leaf admin-state {
    type boolean;
    default "true";
    description
      "Указывает, может ли устройство работать как TWAMP Server.";
  }
  leaf server-tcp-port {
    type inet:port-number;
    default "862";
    description
      "Задаёт общеизвестный порт TCP, используемый TWAMP-Control.
      Сервер прослушивает на этом порту входящие соединения

```

```

TWAMP-Control. Хотя порт задан фиксированным значением
(862) в RFC 5357, имеются реализации TWAMP, созданные
до выделения этого номера, которые могут указывать порт
в конфигурации. Параметр предназначен для совместимости
с такими реализациями.";
}
leaf servwait {
  type uint32 {
    range "1..604800";
  }
  units "seconds";
  default "900";
  description
    "Тайм-аут сессии TWAMP-Control (TCP) в секундах.
    (параграф 3.1 в RFC 5357)

    'Server МОЖЕТ прервать любое организованное управляющее
    при отсутствии приёма связанных с ним пакетов в течение
    SERWAIT секунд.'";
}
leaf control-packet-dscp {
  type inet:dscp;
  description
    "Значение DSCP, помещаемое в заголовок IP пакетов
    TWAMP-Control (TCP), генерируемых сервером.

    В параграфе 3.1 RFC 5357 указано, что серверу СЛЕДУЕТ
    применять значение DSCP из пакета TCP SYN от клиента
    Control-Client. Однако на практике TWAMP обычно
    реализуется с использованием стека TCP общего назначения
    в базовой операционной системе, который может не давать
    пользователю таких сведений. Поэтому не всегда возможно
    реализовать поведение, описанное в RFC 5357, в переносимой
    между ОС версии TWAMP.

    По умолчанию для этого элемента не применяется установка
    DSCP из TCP SYN от Control-Client.";
  reference
    "RFC 5357: A Two-Way Active Measurement Protocol (TWAMP),
    параграф 3.1";
}
uses count;
uses max-count-exponent;
leaf modes {
  type twamp-modes;
  description
    "Битовая маска режимов TWAMP-Mode, которые этот экземпляр
    сервера готов поддерживать (реестр IANA TWAMP-Modes).";
}
uses key-management;
list ctrl-connection {
  key "client-ip client-tcp-port server-ip server-tcp-port";
  config false;
  description
    "Список всех входящих соединений TWAMP-Control (TCP).";
  leaf client-ip {
    type inet:ip-address;
    description
      "IP-адрес удалённого устройства Control-Client, который
      указан в поле source IP пакетов TWAMP-Control (TCP),
      относящихся к этому управляющему соединению.";
  }
  leaf client-tcp-port {
    type inet:port-number;
    description
      "Номер порта отправителя TCP в пакетах TWAMP-Control
      (TCP) этого управляющего соединения.";
  }
  leaf server-ip {
    type inet:ip-address;
    description
      "IP-адрес локального устройства Server, который указан
      в поле destination IP пакетов TWAMP-Control (TCP) в
      этом управляющем соединении.";
  }
  leaf server-tcp-port {
    type inet:port-number;
    description
      "Порт получателя TCP в пакетах TWAMP-Control (TCP) этого
      управляющего соединения. Обычно это совпадает с
      server-tcp-port в контейнере twamp/server. Однако если
      пользователь изменит server/server-tcp-port после
      организации управляющего соединения, это значение будет
      указывать фактический номер server-tcp-port.";
  }
}
leaf state {

```

```
type server-ctrl-connection-state;
description
  "Указывает статус соединения TWAMP-Control на сервере.";
}
leaf control-packet-dscp {
  type inet:dscp;
  description
    "Значение DSCP в заголовке IP пакетов TWAMP-Control (TCP),
    передаваемых сервером в этом управляющем соединении.
    Обычно это совпадает со значением параметра
    control-packet-dscp в контейнере twamp/server. Однако
    если пользователь поменяет server/dscp уже после
    организации соединения это доступное лишь для чтения
    значение будет показывать фактическое поле DSCP в этом
    соединении TWAMP-Control.";
}
leaf selected-mode {
  type twamp-modes;
  description
    "Режим, выбранный для этого соединения TWAMP-Control
    установкой поля Mode в сообщении Set-Up-Response.";
}
leaf key-id {
  type string {
    length "1..80";
  }
  description
    "Значение KeyID для данного соединения TWAMP-Control,
    которое выбрал Control-Client.";
}
uses count {
  description
    "Значение Count в данном соединении TWAMP-Control.
    Обычно оно совпадает со значением в контейнере
    twamp/server. Однако если пользователь изменит
    server/count после организации соединения, это
    доступное лишь для чтения поле будет указывать
    фактическое значение счётчика в данном соединении
    TWAMP-Control.";
}
uses max-count-exponent {
  description
    "Доступное лишь для чтения поле, указывающее фактическое
    значение max-count в этом управляющем соединении. Обычно
    оно совпадает со значением в контейнере twamp/server.";
}
leaf salt {
  type binary {
    length "16";
  }
  description
    "Параметр, используемый для вывода ключа из общего
    секрета, как указано в параграфе 3.1 RFC 4656.
    Передаётся клиенту Control-Client как часть сообщения
    Server Greeting.";
}
leaf server-iv {
  type binary {
    length "16";
  }
  description
    "Вектор инициализации сервера (Server-IV), создаваемый
    сервером случайным образом.";
}
leaf challenge {
  type binary {
    length "16";
  }
  description
    "Случайная последовательность октетов, создаваемая
    сервером. Как описано в client/token, Challenge
    применяется Control-Client для подтверждения
    владения общим секретом.";
}
}
}
container session-sender {
  if-feature "session-sender";
  description
    "Конфигурация логического объекта TWAMP Session-Sender.";
  leaf admin-state {
    type boolean;
    default "true";
    description
      "Указывает, может ли устройство работать как
      TWAMP Session-Sender.";
  }
}
```

```

}
list test-session {
  key "name";
  description
    "Список тестовых сессий TWAMP Session-Sender.";
  leaf name {
    type string;
    description
      "Уникальное имя этой сессии TWAMP-Test, используемое для
      её идентификации логическим объектом Session-Sender.";
  }
  leaf ctrl-connection-name {
    type string;
    config false;
    description
      "Имя родительского соединения TWAMP-Control, отвечающего
      за согласование этой сессии TWAMP-Test.";
  }
}
leaf fill-mode {
  type padding-fill-mode;
  default "zero";
  description
    "Указывает применение для заполнения пакетов TWAMP-Test
    (UDP) (1) псевдослучайных чисел или (2) нулей, как
    указано в параграфе 4.2.1 RFC 5357.";
}
leaf number-of-packets {
  type uint32;
  mandatory true;
  description
    "Общее число пакетов TWAMP-Test (UDP), передаваемых
    устройством Session-Sender в этом сеансе тестирования.";
}
choice packet-distribution {
  description
    "Указывает распределение, используемое для передачи
    пакетов TWAMP-Test (UDP).";
  case periodic {
    leaf periodic-interval {
      type decimal64 {
        fraction-digits 5;
      }
      units "seconds";
      mandatory true;
      description
        "Указывает интервал (в секундах) между первыми
        битами пакетов TWAMP-Test (UDP) в этой сессии.";
      reference
        "RFC 3432: Network performance measurement with
        periodic streams";
    }
  }
  case poisson {
    leaf lambda {
      type decimal64 {
        fraction-digits 5;
      }
      units "seconds";
      mandatory true;
      description
        "Указывает средний интервал (в секундах) между
        пакетами с распределением Пуассона. При расчёте
        используется величина, обратная lambda и размер
        пакета TWAMP-Test (зависит от режима и заполнения).";
      reference
        "RFC 2330: Framework for IP Performance Metrics";
    }
    leaf max-interval {
      type decimal64 {
        fraction-digits 5;
      }
      units "seconds";
      description
        "Задаёт максимальное время (в секундах) между
        передачей пакетов.";
      reference
        "RFC 7312: Advanced Stream and Sampling Framework
        for IP Performance Metrics (IPPM)";
    }
  }
}
leaf state {
  type sender-session-state;
  config false;
  description
    "Указывает состояние тестовой сессии Session-Sender.";
}

```

```
    }
    uses maintenance-statistics;
  }
}
container session-reflector {
  if-feature "session-reflector";
  description
    "Конфигурация логического объекта TWAMP
    Session-Reflector.";
  leaf admin-state {
    type boolean;
    default "true";
    description
      "Указывает, может ли устройство быть
      TWAMP Session-Reflector.";
  }
  leaf refwait {
    type uint32 {
      range "1..604800";
    }
    units "seconds";
    default "900";
    description
      "Session-Reflector МОЖЕТ прервать любую сессию при
      отсутствии связанных с ней пакетов в течение REFWAIT
      REFWAIT секунд. Как указано в параграфе 3.1 5357,
      этот тайм-аут позволяет Session-Reflector освободить
      ресурсы в случае отказа.";
  }
}
list test-session {
  key "sender-ip sender-udp-port
    reflector-ip reflector-udp-port";
  config false;
  description
    "Тестовые сессии TWAMP Session-Reflector.";
  leaf sid {
    type string;
    description
      "Автоматически выделенный идентификатор для этой сессии
      TWAMP-Test, уникальный лишь в контексте устройства
      Server/Session-Reflector. Это значение передаётся
      клиенту Control-Client, запросившему сеанс тестирования,
      в поле SID сообщения Accept-Session.";
  }
  leaf sender-ip {
    type inet:ip-address;
    description
      "IP-адрес удалённого устройства, который служит source IP
      в пакетах TWAMP-Test (UDP) этой тестовой сессии.";
  }
  leaf sender-udp-port {
    type dynamic-port-number;
    description
      "Порт отправителя UDP в пакетах TWAMP-Test этой сессии.";
  }
  leaf reflector-ip {
    type inet:ip-address;
    description
      "IP-адрес локального устройства Session-Reflector,
      служащий destination IP в пакетах TWAMP-Test (UDP)
      данного сеанса тестирования.";
  }
  leaf reflector-udp-port {
    type inet:port-number {
      range "862 | 49152..65535";
    }
    description
      "Порт получателя UDP в пакетах TWAMP-Test (UDP)
      этой сессии.";
  }
  leaf parent-connection-client-ip {
    type inet:ip-address;
    description
      "IP-адрес устройства Control-Client, который служит
      source IP в пакетах TWAMP-Control (TCP) родительского
      управляющего соединения, согласовавшего эту сессию.";
  }
  leaf parent-connection-client-tcp-port {
    type inet:port-number;
    description
      "Порт отправителя TCP в пакетах TWAMP-Control (TCP)
      родительского соединения, согласовавшего эту сессию.";
  }
  leaf parent-connection-server-ip {
    type inet:ip-address;
    description

```



```

<twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
  <client>
    <admin-state>true</admin-state>
    <ctrl-connection>
      <name>RouterA</name>
      <client-ip>2001:db8:203:0:113::1</client-ip>
      <server-ip>2001:db8:203:0:113::2</server-ip>
      <test-session-request>
        <name>Test1</name>
        <sender-ip>2001:db8:203:1:113::3</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>2001:db8:203:1:113::4</reflector-ip>
        <reflector-udp-port>55000</reflector-udp-port>
        <start-time>0</start-time>
      </test-session-request>
      <test-session-request>
        <name>Test2</name>
        <sender-ip>2001:db8:203:0:113::1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>2001:db8:203:0:113::2</reflector-ip>
        <reflector-udp-port>55001</reflector-udp-port>
        <start-time>0</start-time>
      </test-session-request>
    </ctrl-connection>
    <ctrl-connection>
      <name>RouterB</name>
      <client-ip>2001:db8:203:0:113::1</client-ip>
      <server-ip>2001:db8:203:0:113::3</server-ip>
    </ctrl-connection>
  </client>
</twamp>
</config>

```

6.2. Сервер

На рисунке 9 показан пример конфигурации Server со включённым server/admin-state, что позволяет устройству, следующему рисунку 2, отвечать на соединения TWAMP-Control и сессии TWAMP-Test.

```

<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
    </server>
  </twamp>
</config>

```

Рисунок 9. Экземпляр XML, разрешающий работу сервера.

Приведённый ниже пример представляет Server с соединением TWAMP-Control, соответствующим имени (client/ctrl-connection/name) RouterA, представленному в параграфе 6.1.

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
      <ctrl-connection>
        <client-ip>203.0.113.1</client-ip>
        <client-tcp-port>16341</client-tcp-port>
        <server-ip>203.0.113.2</server-ip>
        <server-tcp-port>862</server-tcp-port>
        <state>active</state>
      </ctrl-connection>
    </server>
  </twamp>
</data>

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
      <ctrl-connection>
        <client-ip>2001:db8:203:0:113::1</client-ip>
        <client-tcp-port>16341</client-tcp-port>
        <server-ip>2001:db8:203:0:113::2</server-ip>
        <server-tcp-port>862</server-tcp-port>
        <state>active</state>
      </ctrl-connection>
    </server>
  </twamp>
</data>

```

6.3. Session-Sender

На рисунке 10 показан пример конфигурации Session-Sender со включённым session-sender/admin-state, что позволяет устройству, следующему рисунку 2, инициировать соединения TWAMP-Control и сессии TWAMP-Test.

```

<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-sender>
      <admin-state>true</admin-state>
    </session-sender>
  </twamp>
</config>

```

Рисунок 10. Экземпляр XML, разрешающий работу Session-Sender.

Приведённый ниже пример конфигурации показывает Session-Sender с 2 сессиями TWAMP-Test, представленными в параграфе 6.1.

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-sender>
      <admin-state>true</admin-state>
      <test-session>
        <name>Test1</name>
        <ctrl-connection-name>RouterA</ctrl-connection-name>
        <number-of-packets>900</number-of-packets>
        <periodic-interval>1</periodic-interval>
      </test-session>
      <test-session>
        <name>Test2</name>
        <ctrl-connection-name>RouterA</ctrl-connection-name>
        <number-of-packets>900</number-of-packets>
        <lambda>1</lambda>
        <max-interval>2</max-interval>
      </test-session>
    </session-sender>
  </twamp>
</data>

```

6.4. Session-Reflector

На рисунке 11 показан пример конфигурации Session-Reflector со включённым session-reflector/admin-state, что позволяет устройству, следующему рисунку 2, отвечать на сессии TWAMP-Test.

```

<?xml version="1.0" encoding="utf-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
    </session-reflector>
  </twamp>
</config>

```

Рисунок 11. Экземпляр XML, разрешающий работу Session-Reflector.

Приведённый ниже пример показывает 2 сессии Session-Reflector TWAMP-Test, соответствующие представленным в параграфе 6.3.

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
      <test-session>
        <sender-ip>203.0.113.3</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>203.0.113.4</reflector-ip>
        <reflector-udp-port>50001</reflector-udp-port>
        <sid>1232</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se
rver-tcp-port>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <sender-ip>203.0.113.1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>192.0.2.2</reflector-ip>
        <reflector-udp-port>50001</reflector-udp-port>
        <sid>178943</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-

```

```

server-ip>
  <parent-connection-server-tcp-port>862</parent-connection-se\
rver-tcp-port>
  <sent-packets>21</sent-packets>
  <rcv-packets>21</rcv-packets>
  <last-sent-seq>20</last-sent-seq>
  <last-rcv-seq>20</last-rcv-seq>
</test-session>
</session-reflector>
</twamp>
</data>

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
      <test-session>
        <sender-ip>203.0.113.3</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>203.0.113.4</reflector-ip>
        <reflector-udp-port>54001</reflector-udp-port>
        <sid>1232</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-\
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-\
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-\
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se\
rver-tcp-port>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <sender-ip>203.0.113.1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>192.0.2.2</reflector-ip>
        <reflector-udp-port>55001</reflector-udp-port>
        <sid>178943</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-\
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-\
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-\
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se\
rver-tcp-port>
        <sent-packets>21</sent-packets>
        <rcv-packets>21</rcv-packets>
        <last-sent-seq>20</last-sent-seq>
        <last-rcv-seq>20</last-rcv-seq>
      </test-session>
    </session-reflector>
  </twamp>
</data>

```

7. Вопросы безопасности

Фактически все имеющиеся измерительные системы с использованием TWAMP [RFC5357] администрируются одним сетевым оператором. Атаки на измерительную инфраструктуру могут быть организованы третьей стороной для захвата возможности генерации пакетов, искажения измерений или выполнения иных злонамеренных действий.

Заданные этим документом модули YANG определяют схему для данных, предназначенную для доступа через сеть с использованием протоколов управления, таких как NETCONF [RFC6241] или RESTCONF [RFC8040]. Нижним уровнем NETCONF служит защищённый транспорт с обязательной поддержкой SSH (Secure Shell) [RFC6242]. Нижним уровнем RESTCONF служит протокол HTTPS с обязательной поддержкой защиты на транспортном уровне (TLS) [RFC8446].

Модель доступа к конфигурации сети (NACM – Network Configuration Access Control Model) [RFC8341] обеспечивает возможность разрешить доступ лишь определённым пользователям NETCONF или RESTCONF к заранее заданному подмножеству операций NETCONF или RESTCONF и содержимого.

В этом модуле данных YANG определено множество узлов данных, которые разрешают запись, создание и удаление (т. е. config true, как принято по умолчанию). Эти узлы могут быть конфиденциальными или уязвимыми в некоторых сетевых средах. Запись в такие узлы (например, edit-config) без должной защиты может негативно влиять на работу сети. Ниже перечислены субдеревья и узлы, которые могут быть конфиденциальны или уязвимы.

- Запись в admin-state может приводить к созданию непредусмотренных тестовых сессий.
- При записи в узел number-of-packets, определяющий число пакетов, передаваемых в любой конкретной сессии тестирования, большого значения тестовая сессия будет длиться дольше ожидаемого времени.
- Особо уязвимые узлы включают несколько тайм-аутов, заданных протоколом, для защиты от сессий, которые не активны, но потребляют ресурсы. Параметр REFWAIT управляет прерыванием сессий при отсутствии

пакетов, узлы count и max-count-exponent управляют итерациями функции вывода ключей по паролю (Password-Based Key Derivation Function 2 или PBKDF2).

- Узел dscp с разными маркерами DSCP может вызывать искажение тестового трафика в сети и манипулирование результатом.
- Узлы в mode-preference-chain, задающие значения mode и priority, которые указывают предпочтительный порядок использования оператором, могут служить для манипуляций с отправкой неаутентифицированного или нешифрованного трафика, что позволяет организовать атаки в пути.

Ограничение доступа к таким узлам позволит сократить возможности организации атак в сетевой среде.

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 поддеревья и узлы, которые могут быть конфиденциальны или уязвимы.

- Заданный в модели узел token с конкатенацией Challenge, AES Session-key (шифрование) и HMAC-SHA1 Session-key (аутентификация) чувствителен в плане конфиденциальности и может использоваться для нарушения работы тестовых сессий. Возможность считывания этого поля следует предоставлять лишь ограниченному числу администраторов тестовой сети.

Модель TWAMP YANG на задаёт операций RPC, как указано в Приложении В, и оставляет определение операций NETCONF RPC за реализациями. При определении таких операций RPC они могут оказаться уязвимыми в некоторых сетевых средах, поэтому важно контролировать доступ к этим операциям.

8. Взаимодействие с IANA

Агентство IANA внесло приведённый ниже идентификатор URI в реестр IETF XML Registry [RFC3688].

```
URI: urn:ietf:params:xml:ns:yang:ietf-twamp
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

Агентство IANA внесло указанный ниже модуль YANG в реестр YANG Module Names [RFC6020].

```
Name: ietf-twamp
Namespace: urn:ietf:params:xml:ns:yang:ietf-twamp
Prefix: twamp
Reference: RFC 8913
```

9. Литература

9.1. 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>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, DOI 10.17487/RFC3432, November 2002, <<https://www.rfc-editor.org/info/rfc3432>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, [RFC 4086](#), DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/info/rfc4086>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", [RFC 4656](#), DOI 10.17487/RFC4656, September 2006, <<https://www.rfc-editor.org/info/rfc4656>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", [RFC 5357](#), DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [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>>.
- [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", [RFC 6038](#), DOI 10.17487/RFC6038, October 2010, <<https://www.rfc-editor.org/info/rfc6038>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.

- [RFC7717] Pentikousis, K., Ed., Zhang, E., and Y. Cui, "IKEv2-Derived Shared Secret Key for the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7717, DOI 10.17487/RFC7717, December 2015, <<https://www.rfc-editor.org/info/rfc7717>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [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>>.
- [RFC8545] Morton, A., Ed. and G. Mirsky, Ed., "Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)", RFC 8545, DOI 10.17487/RFC8545, March 2019, <<https://www.rfc-editor.org/info/rfc8545>>.
- [RFC8911] Bagnulo, M., Claise, B., Eardley, P., Morton, A., and A. Akhter, "Registry for Performance Metrics", RFC 8911, DOI 10.17487/RFC8911, November 2021, <<https://www.rfc-editor.org/info/rfc8911>>.
- [UML] ISO/IEC, "Information technology - Open Distributed Processing - Unified Modeling Language (UML) Version 1.4.2", ISO/IEC 19501:2005, OMG-UML VER 1.3, April 2005.
- [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>>.

9.2. Дополнительная литература

- [NSC] John, W., Pentikousis, K., Agapiou, G., Jacob, E., Kind, M., Manzalini, A., Risso, F., Staessens, D., Steinert, R., and C. Meirosu, "Research directions in network service chaining", 2013 IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, DOI 10.1109/SDN4FNS.2013.6702549, November 2013, <<https://doi.org/10.1109/SDN4FNS.2013.6702549>>.
- [PERF-METRICS] IANA, "Performance Metrics", <<https://www.iana.org/assignments/performance-metrics>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5618, DOI 10.17487/RFC5618, August 2009, <<https://www.rfc-editor.org/info/rfc5618>>.
- [RFC5938] Morton, A. and M. Chiba, "Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5938, DOI 10.17487/RFC5938, August 2010, <<https://www.rfc-editor.org/info/rfc5938>>.
- [RFC7312] Fabini, J. and A. Morton, "Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)", RFC 7312, DOI 10.17487/RFC7312, August 2014, <<https://www.rfc-editor.org/info/rfc7312>>.
- [RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <<https://www.rfc-editor.org/info/rfc7426>>.
- [RFC8018] Moriarty, K., Ed., Kaliski, B., and A. Rusch, "PKCS #5: Password-Based Cryptography Specification Version 2.1", RFC 8018, DOI 10.17487/RFC8018, January 2017, <<https://www.rfc-editor.org/info/rfc8018>>.
- [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>>.
- [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>>.

Приложение А. Подробные примеры моделей данных

В этом приложении представлены расширения примеров из раздела 6 с конфигурацией дополнительных полей, таких как параметры аутентификации, значения DSCP и т. п.

A.1. Control-Client

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <client>
      <admin-state>true</admin-state>
      <mode-preference-chain>
        <priority>0</priority>
        <mode>authenticated</mode>
      </mode-preference-chain>
      <mode-preference-chain>
        <priority>1</priority>

```

```

    <mode>unauthenticated</mode>
  </mode-preference-chain>
<key-chain>
  <key-id>KeyClient1ToRouterA</key-id>
  <secret-key>c2VjcmVOMQ==</secret-key>
</key-chain>
<key-chain>
  <key-id>KeyForRouterB</key-id>
  <secret-key>c2VjcmVOMg0K</secret-key>
</key-chain>
<ctrl-connection>
  <name>RouterA</name>
  <client-ip>203.0.113.1</client-ip>
  <server-ip>203.0.113.2</server-ip>
  <control-packet-dscp>32</control-packet-dscp>
  <key-id>KeyClient1ToRouterA</key-id>
  <test-session-request>
    <name>Test1</name>
    <sender-ip>203.0.113.3</sender-ip>
    <sender-udp-port>54000</sender-udp-port>
    <reflector-ip>203.0.113.4</reflector-ip>
    <reflector-udp-port>55000</reflector-udp-port>
    <padding-length>64</padding-length>
    <start-time>0</start-time>
  </test-session-request>
  <test-session-request>
    <name>Test2</name>
    <sender-ip>203.0.113.1</sender-ip>
    <sender-udp-port>54001</sender-udp-port>
    <reflector-ip>203.0.113.2</reflector-ip>
    <reflector-udp-port>55001</reflector-udp-port>
    <padding-length>128</padding-length>
    <start-time>0</start-time>
  </test-session-request>
</ctrl-connection>
</client>
</twamp>
</data>

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <client>
      <admin-state>true</admin-state>
      <mode-preference-chain>
        <priority>0</priority>
        <mode>authenticated</mode>
      </mode-preference-chain>
      <mode-preference-chain>
        <priority>1</priority>
        <mode>unauthenticated</mode>
      </mode-preference-chain>
      <key-chain>
        <key-id>KeyClient1ToRouterA</key-id>
        <secret-key>c2VjcmVOMQ==</secret-key>
      </key-chain>
      <key-chain>
        <key-id>KeyForRouterB</key-id>
        <secret-key>c2VjcmVOMg0K</secret-key>
      </key-chain>
      <ctrl-connection>
        <name>RouterA</name>
        <client-ip>2001:db8:203:0:113::1</client-ip>
        <server-ip>2001:db8:203:0:113::2</server-ip>
        <control-packet-dscp>32</control-packet-dscp>
        <key-id>KeyClient1ToRouterA</key-id>
        <test-session-request>
          <name>Test1</name>
          <sender-ip>2001:db8:10:1:1::1</sender-ip>
          <sender-udp-port>54000</sender-udp-port>
          <reflector-ip>2001:db8:10:1:1::2</reflector-ip>
          <reflector-udp-port>55000</reflector-udp-port>
          <padding-length>64</padding-length>
          <start-time>0</start-time>
        </test-session-request>
        <test-session-request>
          <name>Test2</name>
          <sender-ip>2001:db8:203:0:113::1</sender-ip>
          <sender-udp-port>54001</sender-udp-port>
          <reflector-ip>2001:db8:203:0:113::2</reflector-ip>
          <reflector-udp-port>55001</reflector-udp-port>
          <padding-length>128</padding-length>
          <start-time>0</start-time>
        </test-session-request>
      </ctrl-connection>
    </client>
  </twamp>
</data>

```

```
</twamp>
</data>
```

A.2. Сервер

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
      <servwait>1800</servwait>
      <control-packet-dscp>32</control-packet-dscp>
      <modes>authenticated unauthenticated</modes>
      <count>15</count>
      <key-chain>
        <key-id>KeyClient1ToRouterA</key-id>
        <secret-key>c2VjcmVOMQ==</secret-key>
      </key-chain>
      <key-chain>
        <key-id>KeyClient10ToRouterA</key-id>
        <secret-key>c2VjcmVOMTANCg==</secret-key>
      </key-chain>
      <ctrl-connection>
        <client-ip>203.0.113.1</client-ip>
        <client-tcp-port>16341</client-tcp-port>
        <server-ip>203.0.113.2</server-ip>
        <server-tcp-port>862</server-tcp-port>
        <control-packet-dscp>32</control-packet-dscp>
        <selected-mode>unauthenticated</selected-mode>
        <key-id>KeyClient1ToRouterA</key-id>
        <count>15</count>
      </ctrl-connection>
    </server>
  </twamp>
</data>
```

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <server>
      <admin-state>true</admin-state>
      <servwait>1800</servwait>
      <control-packet-dscp>32</control-packet-dscp>
      <modes>authenticated unauthenticated</modes>
      <count>15</count>
      <key-chain>
        <key-id>KeyClient1ToRouterA</key-id>
        <secret-key>c2VjcmVOMQ==</secret-key>
      </key-chain>
      <key-chain>
        <key-id>KeyClient10ToRouterA</key-id>
        <secret-key>c2VjcmVOMTANCg==</secret-key>
      </key-chain>
      <ctrl-connection>
        <client-ip>2001:db8:203:0:113::1</client-ip>
        <client-tcp-port>16341</client-tcp-port>
        <server-ip>2001:db8:203:0:113::2</server-ip>
        <server-tcp-port>862</server-tcp-port>
        <control-packet-dscp>32</control-packet-dscp>
        <selected-mode>unauthenticated</selected-mode>
        <key-id>KeyClient1ToRouterA</key-id>
        <count>15</count>
      </ctrl-connection>
    </server>
  </twamp>
</data>
```

A.3. Session-Sender

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-sender>
      <admin-state>true</admin-state>
      <test-session>
        <name>Test1</name>
        <ctrl-connection-name>RouterA</ctrl-connection-name>
        <fill-mode>zero</fill-mode>
        <number-of-packets>900</number-of-packets>
        <periodic-interval>1</periodic-interval>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <name>Test2</name>
```

```

<ctrl-connection-name>RouterA</ctrl-connection-name>
<fill-mode>random</fill-mode>
<number-of-packets>900</number-of-packets>
<lambda>1</lambda>
<max-interval>2</max-interval>
<sent-packets>21</sent-packets>
<rcv-packets>21</rcv-packets>
<last-sent-seq>20</last-sent-seq>
<last-rcv-seq>20</last-rcv-seq>
</test-session>
</session-sender>
</twamp>
</data>

```

A.4. Session-Reflector

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
      <test-session>
        <sender-ip>203.0.113.3</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>203.0.113.4</reflector-ip>
        <reflector-udp-port>55000</reflector-udp-port>
        <sid>1232</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se
rver-tcp-port>
        <test-packet-dscp>32</test-packet-dscp>
        <sent-packets>2</sent-packets>
        <rcv-packets>2</rcv-packets>
        <last-sent-seq>1</last-sent-seq>
        <last-rcv-seq>1</last-rcv-seq>
      </test-session>
      <test-session>
        <sender-ip>203.0.113.1</sender-ip>
        <sender-udp-port>54001</sender-udp-port>
        <reflector-ip>192.0.2.2</reflector-ip>
        <reflector-udp-port>55001</reflector-udp-port>
        <sid>178943</sid>
        <parent-connection-client-ip>203.0.113.1</parent-connection-
client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
        <parent-connection-server-ip>203.0.113.2</parent-connection-
server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se
rver-tcp-port>
        <test-packet-dscp>32</test-packet-dscp>
        <sent-packets>21</sent-packets>
        <rcv-packets>21</rcv-packets>
        <last-sent-seq>20</last-sent-seq>
        <last-rcv-seq>20</last-rcv-seq>
      </test-session>
    </session-reflector>
  </twamp>
</data>

```

```

<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <twamp xmlns="urn:ietf:params:xml:ns:yang:ietf-twamp">
    <session-reflector>
      <admin-state>true</admin-state>
      <test-session>
        <sender-ip>2001:db8:10:1:1::1</sender-ip>
        <sender-udp-port>54000</sender-udp-port>
        <reflector-ip>2001:db8:10:1:1::2</reflector-ip>
        <reflector-udp-port>55000</reflector-udp-port>
        <sid>1232</sid>
        <parent-connection-client-ip>2001:db8:203:0:113::1</parent-c
onnection-client-ip>
        <parent-connection-client-tcp-port>16341</parent-connection-
client-tcp-port>
        <parent-connection-server-ip>2001:db8:203:0:113::2</parent-c
onnection-server-ip>
        <parent-connection-server-tcp-port>862</parent-connection-se
rver-tcp-port>
        <test-packet-dscp>32</test-packet-dscp>
        <sent-packets>2</sent-packets>

```

```

<rcv-packets>2</rcv-packets>
<last-sent-seq>1</last-sent-seq>
<last-rcv-seq>1</last-rcv-seq>
</test-session>
<test-session>
  <sender-ip>2001:db8:203:0:113::1</sender-ip>
  <sender-udp-port>54001</sender-udp-port>
  <reflector-ip>2001:db8:192:68::2</reflector-ip>
  <reflector-udp-port>55001</reflector-udp-port>
  <sid>178943</sid>
  <parent-connection-client-ip>2001:db8:203:0:113::1</parent-c\
onnection-client-ip>
  <parent-connection-client-tcp-port>16341</parent-connection-\
client-tcp-port>
  <parent-connection-server-ip>2001:db8:203:0:113::2</parent-c\
onnection-server-ip>
  <parent-connection-server-tcp-port>862</parent-connection-se\
rver-tcp-port>
  <test-packet-dscp>32</test-packet-dscp>
  <sent-packets>21</sent-packets>
  <rcv-packets>21</rcv-packets>
  <last-sent-seq>20</last-sent-seq>
  <last-rcv-seq>20</last-rcv-seq>
</test-session>
</session-reflector>
</twamp>
</data>

```

Приложение В. Рабочие команды

Рабочие команды TWAMP могут выполняться программами или вручную, например через командный интерфейс (command-line interface или CLI).

В части программируемости YANG можно использовать для определения вызовов NETCONF Remote RPC, поэтому можно было бы определить операции TWAMP RPC для таких действий, как запуск и остановка управляющих соединений, тестовых сессий или их групп, очистки сохранённых результатов и т. п. Однако TWAMP [RFC5357] не описывает такие операции (см. также раздел 2 и соединения без меток на рисунке 1). В развёрнутых системах разные реализации TWAMP могут поддерживать отличающиеся наборы рабочих команд с разными ограничениями, поэтому данный документ оставляет за реализациями ответственность за определение моделей данных для рабочих команд.

Благодарности

Спасибо Fred Baker, Kevin D'Souza, Gregory Mirsky, Brian Trammell, Robert Sherman, Marius Georgescu за их подробные и конструктивные рецензии, комментарии и предложения по тексту документа.

Naohing Shen внёс вклад в определение модуля YANG (раздел 5). Jan Lindblad и Ladislav Lhotka провели тщательный анализ модуля YANG и примеров из приложения А.

Kostas Pentikousis получил частичную поддержку в рамках исследовательского проекта FP7 UNIFY, финансируемого Европейской комиссией в рамках программы Seventh Framework (грант 619609). Выраженные здесь токи зрения принадлежат лишь авторам и Еврокомиссия не отвечает за любое использование сведений из этого документа.

Участник работы

Lianshu Zheng

Адреса авторов

Ruth Civil

Ciena Corporation
307 Legget Drive
Kanata ON K2K 3C8
Canada
Email: ruthcivil@gmail.com
URI: www.ciena.com

Al Morton

AT&T Labs
200 Laurel Avenue South
Middletown, NJ 07748
United States of America
Phone: +1 732 420 1571
Email: acmorton@att.com

Reshad Rahman

Canada

Email: reshad@yahoo.com

Mahesh Jethanandani

Xoriant Corporation
1248 Reamwood Avenue
Sunnyvale, CA 94089
United States of America
Email: mjethanandani@gmail.com

Kostas Pentikousis (editor)

Detecon
Winterfeldtstrasse 21
10781 Berlin
Germany
Email: kostas.pentikousis@detecon.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru