|  |  |
| --- | --- |
| **ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ** | ***СТБ RFC7640/ОР*** |

СРЕДСТВА ЭЛЕКТРОСВЯЗИ МУЛЬТИСЕРВИСНЫХ СЕТЕЙ

Механизмы обеспечения качества обслуживания

Управление передачей данных

Методы испытаний

**СРОДКI ЭЛЕКТРАСУВЯЗI МУЛЬТЫСЭРВIСНЫХ СЕТАК**

Механізмы забеспячэння якасці абслугоўвання

Кіраванне перадачай даных

Метады выпрабаванняў

(IETF RFC 7640:2015, IDT)

*Настоящий проект стандарта не подлежит применению до его утверждения*



######

###### Госстандарт

###### Минск

УДК 621.39 МКС **33.020**; 35.020 IDT

**Ключевые слова:** мультисервисная сеть, средства электросвязи, управление передачей данных, испытания

**Предисловие**

Цели, основные принципы, положения по государственному регулированию и управлению в области технического нормирования и стандартизации установлены Законом Республики Беларусь «О техническом нормировании и стандартизации».

1 ПОДГОТОВЛЕН открытым акционерным обществом «Гипросвязь» на основе собственного перевода на русский язык англоязычной версии документа, указанного в пункте 3

ВНЕСЕН Министерством связи и информатизации Республики Беларусь

2 утвержден и введен в действие постановлением Госстандарта Республики Беларусь

от №

3 Настоящий стандарт идентичен международному документу IETF RFC 7640 (09/2015) «Эталонные испытания управления передачей данных» («Traffic Management Benchmarking», IDT).

Наименование государственного стандарта изменено относительно наименования международного документа в связи с особенностями технического нормирования и стандартизации Республики Беларусь и для увязки с наименованиями, принятыми в существующем комплексе государственных стандартов.

Международный документ разработан Рабочей группой эталонных испытаний (BMWG) области управления и обслуживания Инженерного совета Интернета (IETF).

Сокращения и обозначения, применяемые в настоящем стандарте, приведены в дополнительном приложении ДА.

4 ВВЕДЕН ВПЕРВЫЕ

Настоящий стандарт не может быть воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Госстандарта Республики Беларусь

Издан на русском языке

**Содержание**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | Введение………………………….………………….…………………………........................................... |  |
|  |  |  |
|  | 1.1 Описание управления передачей данных……………………………………………………………. |  |
|  |  |  |
|  | 1.2 Описание конфигурации и испытаний в условиях лаборатории………………………………….. |  |
|  |  |  |
| 2 | Применяемые условные обозначения……………………………………………………………………… |  |
|  |  |  |
| 3 | Область действия и цели….…………………………….……….………………………………………….. |  |
|  |  |  |
| 4 | Метрики трафика для эталонных испытаний……………………………………………………………. |  |
|  |  |  |
|  | 4.1 Метрики для испытаний трафиком без сохранения состояния……………………………………. |  |
|  |  |  |
|  | 4.2 Метрики для испытаний трафиком с сохранением состояния…………………………………….. |  |
|  |  |  |
| 5 | Требования к испытательному оборудованию…………………………………………………………. |  |
|  |  |  |
|  | 5.1 Генерация испытательного трафика без сохранения состояния…………………………………. |  |
|  |  |  |
|  | 5.2 Генерация испытательного шаблона с сохранением состояния………………………………….. |  |
|  |  |  |
| 6 | Эталонные испытания……………………………………………………………………………………….. |  |
|  |  |  |
|  | 6.1 Испытания правил обработки трафика………………………………………………………………… |  |
|  |  |  |
|  | 6.2 Испытания формирователя очереди/диспетчера трафика………………………………………… |  |
|  |  |  |
|  | 6.3 Испытания шейпера……………………………………………………………………………………… |  |
|  |  |  |
|  | 6.4 Нагрузочные испытания параллельной производительности……………………………………… |  |
|  |  |  |
| 7 | Аспекты безопасности………………………………………………………………………………………. |  |
|  |  |  |
| 8 | Ссылки………………...………………..……………………………………………………………………… |  |
|  |  |  |
|  | 8.1 Нормативные ссылки…………………………………………………………………………………….. |  |
|  |  |  |
|  | 8.2 Информационные ссылки……………………..………………………………………………………… |  |
|  |  |  |
| Приложение А | (справочное) Инструменты из открытых источников для проведения испытаний управления передачей данных …………………………………………………………… |  |
|  |  |  |  |  |
| Приложение В | (рекомендуемое) Испытательные шаблоны для ТСР с сохранением состояния….. |  |
|  |  |  |
| Приложение ДА | (справочное) Сокращения и обозначения…………………………………………………. |  |
|  |  |  |

**ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ**

СРЕДСТВА ЭЛЕКТРОСВЯЗИ МУЛЬТИСЕРВИСНЫХ СЕТЕЙ

**Механизмы обеспечения качества обслуживания**

**Управление передачей данных**

**Методы испытаний**

**СРОДКI ЭЛЕКТРАСУВЯЗI МУЛЬТЫСЭРВIСНЫХ СЕТАК**

**Механізмы забеспячэння якасці абслугоўвання**

**Кіраванне перадачай даных**

**Метады выпрабаванняў**

Telecommunication facilities of multiservice networks

Data transmission management

Test methods

**Дата введения \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **1 Введение**

Управление передачей данных (то есть правила обработки трафика, шейпинг и так далее) становится все более важной составляющей при реализации сетевого качества обслуживания (QoS).

В настоящее время отсутствует базовая структура для эталонных испытаний этих функций, несмотря на то, что есть некоторые стандарты, адресованные к определенным областям, как указано в 1.1.

Настоящий стандарт описывает базовую структуру для проведения повторимых эталонных испытаний управления передачей данных для устройств и систем в условиях лаборатории.

Особым образом в данной базовой структуре определены методы характеризации производительности следующих механизмов управления передачей данных в сетевых устройствах: классификация трафика, правила обработки трафика, организация очередей и диспетчеризация и шейпинг трафика.

Данная базовая структура эталонных испытаний, так же может быть использована как испытательная процедура, помогающая настраивать параметры управления передачей данных до активации услуги. Дополнительно к эталонным испытаниям на уровнях 2 и 3 (Ethernet/IP) настоящим стандартом предложены испытательные шаблоны уровня 4 (TCP) для обеспечения более реалистичных эталонных испытаний пользовательского трафика.

# **1.1 Описание управления передачей данных**

В общем случае, устройство с поддержкой управления передачей данных выполняют следующие функции:

– классификация трафика: идентифицирует трафик согласно различным конфигурационным правилам (например, IEEE 802.1Q (VLAN), точка кода дифференцированных услуг (DSCP)) и маркирует этот трафик внутри сетевого устройства. Множество внешних приоритетов может быть привязано к таким же приоритетам в устройстве;

– правила обработки трафика: ограничивает скорость трафика, который поступает в устройство в соответствии с его классификацией. Если трафик превышает установленные пределы, то он либо сбрасывается, либо перемаркируется и пересылается к следующему сетевому устройству;

– диспетчеризация трафика: обеспечивает классификацию трафика внутри сетевого устройства путем направления пакетов в различного типа очереди и применяет алгоритм диспетчеризации для установления пересылаемой последовательности пакетов;

– шейпинг трафика: управляет трафиком активно его буферизируя и сглаживая его выходной уровень, пытаясь таким образом адаптировать всплескоообразный трафик к сконфигурированным пределам;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Проект, окончательная редакция***

– активное управление формированием очереди (AQM): включает в себя мониторинг статуса внутренних очередей и проактивный сброс (или перемаркировку) пакетов, вынуждающих хосты использовать протоколы обнаружения перегрузок для «сброса давления», и, таким образом, снижает перегрузку очереди [RFC 7567]. С другой стороны, классические техники управления передачей данных реактивно сбрасывают (или перемаркируют) пакеты в соответствии с условиями заполнения очереди. Сценарии эталонных испытаний для AQM различны и находятся за рамками применения данной испытательной базовой структуры.

Несмотря на то, что AQM за рамками применения данной испытательной базовой структуры, необходимо отметить, что ТСР метрики и ТСР испытательные шаблоны (определены в 4.2 и 5.2, соответственно) могут быть использованы для испытания новых алгоритмов AQM (нацеленных на избежание ситуаций с «раздуванием» буфера). Примеры таких алгоритмов включают контролируемую задержку [CoDel] и расширенный пропорционально интегральный контроллер [PIE].

Следующий рисунок представляет собой основную модель функционирования управления передачей данных в сетевом устройстве. Схема не предназначена для представления всех создаваемых производителем вариаций функционирования управления передачей данных, но обеспечивает контекст для описываемой испытательной базовой структуры.

**Рисунок 1 – Основные механизмы управления передачей данных в сетевом устройстве**

Воздействие на входе, такое как классификация, описано в [RFC4689] и включает в себя IP-адреса, номера портов и DSCP. Согласно терминологии маркировки [RFC2697] и [RFC2698] определяют одноуровневый трехцветный маркер и двухуровневый трехцветный маркер, соответственно.

Metro Ethernet Forum (MEF) специфицирует правила обработки и моделирования трафика в терминах входной и выходной пользовательской/операторской функции согласования, как определено в MEF 12.2 [MEF-12.2], также, как и атрибуты входного профиля и профиля пропускной способности канала передачи данных, как определено в MEF 10.3 [MEF-10.3] и MEF 26.1 [MEF-26.1].

# **1.2 Описание конфигурации и испытаний в условиях лаборатории**

Следующий рисунок показывает схему лабораторной установки для испытаний управления передачей данных.



**Рисунок 2 – Схема лабораторной установки для испытаний управления передачей данных**

Как показано на схеме лабораторной установки, базовая структура поддерживает испытания для одностороннего управления передачей данных и для двухстороннего управления передачей данных (при котором роли передачи и приема поменяются местами для обратного канала).

Данная базовая структура описывает испытания и метрики для каждого из следующих механизмов управления передачей данных:

– классификация трафика;

– правила обработки трафика;

– организация очередей и диспетчеризация трафика;

– шейпинг.

Испытания разделены на индивидуальные испытания и испытания номинальной производительности. Индивидуальные испытания предназначены для эталонной проверки функций управления передачей данных в соответствии с метриками, определенными в 4. Испытания производительности проверяют функции управления передачей данных под нагрузкой из множества одновременных индивидуальных испытаний и их потоков.

Такая проверка включает в себя одновременные испытания множества интерфейсов с включенными определенными функциями управления передачей данных и увеличение нагрузки до предельной производительности каждого из интерфейсов.

Например, устройство указано, как способное выполнять шейпинг на всех своих интерфейсах. Тогда сначала будет выполнено индивидуальное испытание для эталонной проверки указанной функции шейпинга относительно метрик, определенных в 4. Затем будет выполнено испытание производительности для проверки корректности функционирования функции шейпинга на всех интерфейсах и при максимальной трафиковой нагрузке.

Эмулятор сетевой задержки (NDE) требуется при испытаниях ТСР с фиксацией состояния, для разрешения ТСР использовать ТСР окно значимого размера в своей петле управления.

Необходимо также отметить, что NDE СЛЕДУЕТ быть пассивным по своей природе (например, бухта оптического патчкорда). Это необходимо для исключения возможных эффектов, которые могут иметь место для испытательных потоков при активном задерживающем элементе (то есть, для испытательного генератора негативных воздействий). В случае, когда практически не реализуема бухта оптического патчкорда, требуемая для обеспечения необходимой задержки, активный NDE ДОЛЖЕН быть независимо верифицирован на способность вносить сконфигурированную задержку без потерь. Другими словами, испытуемое устройство (DUT) будет удалено из схемы, а для NDE будет выполнено независимое эталонное испытание производительности.

Необходимо отметить, что NDE СЛЕДУЕТ использовать только для эмуляции задержки. Большинство NDE позволяют выполнять задерживающие действия для каждого потока, имитируя QoS приоритезацию. Для данной базовой структуры единственной целью NDE является простое внесение задержки для всех пакетов (эмуляция сетевого времени задержки). Поэтому, при эталонном испытании производительности NDE, оценивают максимальную предложенную нагрузку относительно следующих размеров кадров: 128, 256, 512, 768, 1024, 1500 и 9600 байт. Полученная погрешность задержки для каждого из этих размеров кадров может быть затем использована для калибровки диапазона ожидаемого значения производной от пропускной способности канала передачи данных и задержки "из конца в конец" в нем (BDP) при испытаниях ТСР с фиксацией состояния.

**2 Применяемые условные обозначения**

Ключевые слова «ДОЛЖЕН», «НЕ ДОЛЖЕН», «НЕОБХОДИМ», «ОБЯЗАН», «НЕ ОБЯЗАН», «СЛЕДУЕТ», «НЕ СЛЕДУЕТ», «РЕКОМЕНДОВАН», «МОЖЕТ» и «ОПЦИОНАЛЬНО» в настоящем стандарте следует интерпретировать так, как описано в [RFC 2119].

Использованы следующие сокращения:

AQM – Active Queue Management – активное управление формированием очереди;

BB – минимальная пропускная способность канала передачи данных;

BDP – Bandwidth-Delay Product – производное от пропускной способности (цифрового) канала передачи данных и задержки "из конца в конец" в нем;

BSA – Burst Size Achieved – достигнутый размер всплеска;

CBS – Commited Burst Size – согласованный размер всплеска;

CIR – Commited Information Rate – согласованная информационная скорость;

DUT – Device Under Test – испытуемое устройство;

EBS – Excess Burst Size – расширенный размер всплеска;

EIR – Excess Information Rate – расширенная информационная скорость;

NDE – Network Delay Emulator – эмулятор сетевой задержки;

QL – Queue Length – длина очереди;

QoS – Quality of Service – качество обслуживания;

RTT – Round-Trip Time – время кругового обращения[[1]](#footnote-1);

SBB – Shaper Burst Bytes – байты всплеска шейпера;

SBI – Shaper Burst Interval – интервал всплесков шейпера;

SP – Strict Priority – строгий приоритет;

SR – Shaper Rate – скорость шейпера;

SSB – Send Socket Buffer – буфер сокета передачи;

SUT – System Under Test – испытуемая система;

Ti – Transmission Interval – интервал передачи;

TTP –TCP Test Pattern – TCP испытательный шаблон;

TTPET – TCP Test Pattern Execution Time – время исполнения TCP испытательного шаблона.

**3 Область действия и цели**

Областью действия настоящего стандарта является представление базовой структуры эталонных испытаний и проверки возможностей сетевых устройств по управлению передачей данных в лабораторных условиях. Указанные сетевые устройства могут включать, но не ограничиваться следующими типами:

– коммутаторы (включая устройства 2/3 уровня);

– маршрутизаторы;

– межсетевые экраны;

– основные аппаратные реализации для работы на 4 – 7 уровнях (прокси, WAN-ускорители и т.д.).

По существу, любое сетевое устройство, которое выполняет управление передачей данных, как определено в 1.1, может быть подвергнуто эталонным испытаниям или проверено с использованием этой базовой структуры.

Основной целью является оценка максимальной производительности пересылки, предположительно, находящуюся в границах установленных трафиковых пределов, которую сетевое устройство может поддерживать без отбрасывания или искажения пакетов и без ущерба соответствию множеству реализаций функций управления передачей данных. Данная проверка является стандартной для сравнения между устройствами.

В пределах данной базовой структуры определены метрики для каждого испытания управления передачей данных, но без использования критерия «прошел/не прошел», который находится вне сферы деятельности BMWG. Данная базовая структура предоставляет методы испытаний и метрики для проведения повторяемых испытаний, которые будут обеспечивать получение средств для сравнения измеренной производительности между DUT.

Как упоминалось в 1.2, эти методы описывают индивидуальные испытания и метрики для ряда функций управления. Кроме того, в пределах области действия настоящего стандарта является то, что данная базовая структура будет выполнять эталонное испытание каждой функции в терминах общей оцененной производительности. Такая проверка включает одновременное испытание множества интерфейсов с включенной определенной функцией управления передачей данных, вплоть до предела производительности каждого из интерфейсов.

Вне области действия данной базовой структуры определение процедуры для испытаний множества конфигураций функций управления передачей данных одновременно. Многообразие возможных комбинаций практически неограниченно и способность определить функциональную «точку сбоя» будет практически невозможно.

Тем не менее, 6.4 предлагает варианты для некоторых профилей одновременно работающих функций, которые могут быть использованы для эталонных испытаний. Ключевым требованием для любого конкурентного испытания функции является то, что испытания ДОЛЖНЫ предоставлять достоверные и повторяемые результаты.

Кроме того, вне области действия настоящего стандарта выполнение испытаний на соответствие стандартам. Испытания, определенные в данной базовой структуре, обеспечивают эталонное испытание функций управления передачей данных согласно метрикам, определенным в 4 и не предназначены для проверки соответствия стандартам, связанными с управлением передачей данных.

Текущие спецификации не определяют конкретное поведение или реализацию, однако существующие спецификации (указанные в 1.1) позволяют реализациям вариации относительно точности краткосрочной скорости и других факторов. Основным движущим фактором для данной базовой структуры является предоставление объективных средств для сравнения функций управления передачей с данными различных производителей.

Следующей целью является разработка методов, которые используют потоки с распознающим перегрузки транспортом (ТСР) как часть трафиковой нагрузки, и по-прежнему предоставляют повторяемые результаты в изолированном испытательном окружении. Данная базовая структура будет реализовывать испытательные шаблоны с запоминанием состояния (ТСР или уровень приложений), которые также могут быть использованы для дальнейших эталонных испытаний характеристик соответствующих техник управления передачей данными, таких как организация очередей и диспетчеризация трафика и шейпинг трафика. В случаях, где сетевое устройство является по своей природе запоминающим состояние (т.е. межсетевой экран и т.д.), испытательные шаблоны с запоминанием состояния важны для проверки, наряду с UDP трафиком без запоминания состояния в конкретных испытательных сценариях (т.е. приложения, использующие ТСР транспорт и UDP VoIP, и т.д.).

Как упоминалось ранее, важна повторяемость результатов испытаний, особенно принимая во внимание природу запоминающего состояние ТСР-трафика. С этой стороны испытания с запоминанием состояния будут использовать ТСР испытательные шаблоны, эмулирующие приложения. Данная базовая структура также предоставляет руководство для моделирования приложений и инструменты из открытых источников для обеспечения повторяемых стимулов. И наконец, ТСР-метрики из [RFC6349] ДОЛЖНЫ быть измерены для каждого испытания с запоминанием состояния и предоставлены средства для сравнения каждого из повторенных испытаний.

Хотя данная базовая структура нацелена на проверку ТСР приложений (т.е. веб, электронная почта, база данных и т.д.), она также может быть применена к протоколу передачи с управлением потоками (SCTP) относительно испытательных шаблонов. WebRTC, системы сигнализации 7 (SS7) и 3GPP являются SCTP-ориентированными приложениями, которые могут быть смоделированы с помощью данной базовой структуры для эталонных испытаний эффекта SCTP на характеристики управления передачей трафика.

Необходимо отметить, что базовая структура не касается испытательных шаблонов шифрованного ТСР, хотя метрики, определенные в 4.2 по-прежнему могут быть использованы, так как они базируются на измерениях ТСР повторной передачи и RTT (относительно любой полезной нагрузки). Таким образом, если шифрованный TCP станет популярным, то будет естественным для испытателей обсуждать шаблоны шифрованного ТСР и включать их в испытательные сценарии.

**4 Метрики трафика для эталонных испытаний**

Метрики, измеряемые во время эталонных испытаний разделены на две (2) группы: метрики уровня пакетов, используемые для испытаний трафиком без сохранения состояния, и метрики уровня ТСР, используемые для испытания трафиком с сохранением состояния.

**4.1 Метрики для испытаний трафиком без сохранения состояния**

Измерения трафиком без сохранения состояния требуют, чтобы в полезную нагрузку были вставлены номер последовательности и временная метка для выполнения анализа потери пакетов. Анализ задержки может быть достигнут вставкой временных меток напрямую в пакеты или временных меток, сохраненных в каком-либо другом месте (захват пакетов). Данная базовая структура не определяет формат пакета для переноса номера последовательности или временной информации.

Тем не менее [RFC4737] и [RFC4689] предоставляют рекомендации для отслеживания последовательности, наряду с определением упорядоченных и неупорядоченных пакетов.

Следующие метрики ДОЛЖНЫ быть определены во время эталонных испытаний трафиком без сохранения состояния соответствующих компонентов испытаний:

– достигнутый размер всплеска (BSA): для испытания правил обработки трафика и испытания формирования сетевой очереди, испытательное оборудование должно быть сконфигурировано на посылку всплесков для проверки либо согласованного размера всплеска (CBS), либо расширенного размера всплеска (EBS) установщика правил обработки трафика или для определения сконфигурированного размера очереди/буфера в DUT. Метрика BSA является измерением реального размера всплеска, принятого с выходного порта DUT без потерь пакетов. Например, сконфигурированный CBS в DUT равен 64 кБ, но после всплескового испытания выявлено, что только всплеск равный 63 кБ может быть достигнут без потерь пакетов. Тогда 63 кБ соответствует BSA. Кроме того, следует фиксировать среднюю вариацию задержки пакета (PDV) (см. далее), выявленную при посылке пакетов с размером всплеска равным BSA. Данная метрика ОБЯЗАНА быть зафиксирована в единицах измерения байт, килобайт или мегабайт;

– потерянные пакеты (LP): для всех испытаний управления передачей данных испытательное оборудование будет передавать испытательные пакеты на входной порт DUT и измерять число пакетов, принятых с его выходного порта. Разница между пакетами, переданными на входной порт и принятыми с выходного порта является значением потерянных пакетов на выходном порту. Эти пакеты должны иметь уникальные идентификаторы, чтобы было возможным измерение только испытательных пакетов. Для случаев передачи множества потоков с входного на выходной порты (например, IP-взаимодействие), каждый индивидуальный поток должен иметь номера последовательности внутри общего потока испытательных пакетов. [RFC6703] и [RFC2680] описывают необходимость установления временных границ ожидания, после которых пакет считается потерянным. Данное пороговое значение ДОЛЖНО быть зафиксировано со значением, которое представляется целым неотрицательным числом;

– неупорядоченность (OOS): в дополнение к метрике LP должен отслеживаться порядок следования пакетов. [RFC4689] определяет основную функцию отслеживания порядка следования, а также дает определения для упорядоченных и неупорядоченных пакетов. Неупорядоченные пакеты будут подсчитываться в соответствии с [RFC4737]. Данная метрика ОБЯЗАНА быть зафиксирована со значением, которое представляется целым неотрицательным числом;

– задержка пакета (PD): метрика PD представляет собой разницу между временными отметками, принятых с выходного порта пакетов и пакетов переданных на входной порт, как определено в [RFC1242]. Время передающего и принимающего хостов должно быть синхронизировано по времени (достижимо посредством использования NTP, GPS, и т.д.). Данная метрика ОБЯЗАНА быть зафиксирована, как реальное число секунд, при этом отрицательное значение обычно отражает существование проблем синхронизации по времени между устройствами;

– вариация задержки пакета (PDV): метрика PDV представляет собой вариацию между временными метками принятых с выходного порта пакетов, как указано в [RFC5481]. Необходимо отметить, что согласно [RFC5481] PDV представляет собой вариацию односторонней задержки между множеством пакетов в индивидуальном трафиковом потоке. Согласно формуле измерения, приведенной в [RFC5481], выбирают наибольший процентиль 99 % и единицы измерения будут реальными значениями секунд (отрицательная величина невозможна для PDV и будет указывать на ошибку измерения);

– скорость шейпера (SR): SR представляет собой среднюю выходную скорость DUT (бит в секунду) за интервал испытания. SR применима только для испытаний шейпинга трафика;

– байты всплеска шейпера (SBB): шейпер трафика будет выпускать пакеты сериями различных размеров; эти кадры будут выпускаться друг за другом с учетом обязательного межкадрового интервала. Данная метрика характеризует метод, которым шейпер выпускает трафик. Некоторые шейперы передают больше всплесков за интервал, и всплеск в один пакет будет применяться к наименее распространенному варианту шейпера, посылающему поток одиночных пакетов с постоянной битовой скоростью. Данная метрика ОБЯЗАНА быть зафиксирована в единицах измерения байт, килобайт или мегабайт. SBB метрика применима только для испытаний шейпинга трафика;

– интервал всплесков шейпера (SBI): SBI представляет собой время между всплесками, выпущенными шейпером и измеренное на выходном порту DUT. Данная метрика ОБЯЗАНА быть зафиксирована, как реальное число секунд. SBI метрика применима только для испытаний шейпинга трафика.

**4.2 Метрики для испытаний трафиком с сохранением состояния**

Метрики с сохранением состояния будут базироваться на ТСР-метриках [RFC6349] и ДОЛЖНЫ в себя включать:

– время выполнения ТСР испытательного шаблона (TTPET): [RFC6349] определяет время передачи ТСР для групповой пересылки, которая представляет собой простое измерение времени передачи байт через одно или параллельные ТСР-соединения. ТСР испытательный шаблон, использованный в испытаниях управления передачей данных, будет включать групповую пересылку и интерактивные приложения. Шаблон взаимодействия включает такие варианты, как HTTP бизнес-приложения и приложения баз данных. ТТРЕТ будет представлять собой измерение времени для одиночного выполнения ТСР испытательного шаблона (ТТР). Среднее, минимальное и максимальное время будет либо измерено, либо рассчитано и выражено в реальных значениях секунд.

В качестве примера можно привести интерактивную HTTP TTP сессию, которой требуется 5 с в GigE сети с временем задержки 0,5 мс. В течение десяти (10) выполнений этого ТТР, ТТРЕТ результатами могут быть средние 6,5 с, минимальные 5,0 с и максимальные 7,9 с.

– эффективность ТСР: после выполнения ТТР, эффективность ТСР отражает процент байт, которые не были переданы.

 Переданные байты – Повторно переданные байты

Эффективность TCP = х 100.

 Переданные байты

«Переданные байты» представляют собой общее число переданных ТСР байт, включая оригинальные байты и повторно переданные байты. Для избежания путаницы с переупорядоченными пакетами, которые являются повторно переданными пакетами (как может быть в случае с интерпретацией декодирования пакета), данные повторно переданные байты следует фиксировать с позиции ТСР/IP стека отправителя;

– задержка в буфере: задержка в буфере отражает увеличение RTT во время испытания ТСР, относительно базового RTT DUT (внутреннее время задержки без перегрузки). RTT и техника измерения RTT (средний относительно базового) описаны в [RFC6349]. Согласно [RFC6349], средний RTT получают из суммарного значения всех RTT, измеренных во время реального испытания и фиксируемых через каждую секунду, деленного на длительность испытания в секундах.

 Суммарное RTT за время передачи

Среднее RTT за время передачи = ,

 Длительность передачи в секундах

 Среднее RTT за время передачи – Базовое RTT

Задержка в буфере = х 100.

 Базовое RTT

Необходимо отметить, что даже несмотря на то, что это однозначно не указано в [RFC6349], повторно переданные пакеты не следует использовать в измерениях RTT.

Кроме того, в результаты испытания следует записывать средний RTT в миллисекундах по отношению к полной длительности испытания, так же, как и число отсчетов.

**5 Требования к испытательному оборудованию**

Возможности испытательного окружения для испытания управления передачей данных разделены на две группы: испытание трафика без сохранения состояния и испытание трафика с сохранением состояния.

**5.1 Генерация испытательного трафика без сохранения состояния**

Испытательное оборудование ДОЛЖНО быть способно генерировать трафик с уровнем вплоть до значения линейной скорости DUT. Испытательное оборудование должно быть калибровано для подтверждения того, что оно не теряет какие-либо пакеты. Внутренняя PD и PDV испытательного оборудования должны также быть калиброваны и вычтены из метрик PD и PDV. Испытательное оборудование должно поддерживать испытываемую инкапсуляцию, например, IEEE 802.1Q VLAN, IEEE 802.1ad Q-in-Q, многопротокольная коммутация по меткам (MPLS). Кроме того, испытательное оборудование должно разрешать управление техниками классификации, определенными в [RFC4689] (например, IP-адрес, DSCP, классификация по типам обслуживания).

Инструмент из открытых источников iperf может быть использован для генерации UDP трафика без сохранения состояния и рассмотрен в приложении А. Так как iperf программный инструмент, то могут быть ограничения по производительности при высоких канальных скоростях (например, 1 Гбит/с, 10 Гбит/с). Важна тщательная калибровка любого испытательного окружения, использующего iperf. На высоких канальных скоростях рекомендовано использование аппаратных пакетных испытательных инструментов.

**5.1.1 Отслеживание всплесков с трафиком без сохранения состояния**

Центральной темой для испытаний управления передачей данных является эталонная проверка определенного параметра всплеска и функции управления передачей данных, поскольку параметры всплеска, перечисленные в соглашении об уровне обслуживания (SLA) указаны в байтах. Для повышения эффективности, в испытания рекомендовано включение функции отслеживания всплесков, так как эта функция автоматизирует ручной процесс определения максимального размера всплеска, который может быть поддержан функцией управления передачей данных.

Алгоритм отслеживания всплесков следует начинать с искомого размера всплеска (максимального размера всплеска, поддерживаемого функцией управления передачей данных), посылкой одиночных всплесков, до тех пор, пока не будет определен наибольший всплеск, проходящий без потерь. Если всплеск искомого размера проходит, то испытание считается завершенным. Аспект «отслеживания» возникает тогда, когда искомый размер всплеска не достигнут, алгоритм будет сбрасывать сконфигурированный минимальный размер всплеска и выполнять пошаговое увеличение размера всплеска до тех пор, пока не будет найден максимальный всплеск, поддерживаемый DUT. Рекомендованный шаг увеличения размера всплеска соответствует 1 кБ.

Для функции ограничителя трафика, если всплеск определенного размера проходит, то следует увеличивать размер всплеска на 1 кБ для подтверждения того, что ограничитель трафика на самом деле правильно сконфигурирован (либо полностью включен).

**5.2 Генерация испытательного шаблона с сохранением состояния**

Испытательный ТСР-хост имеет множество таких же атрибутов, как и испытательный ТСР-хост, описанный в [RFC6349]. Испытательное ТСР-устройство может быть обычным компьютером, либо отдельным коммуникационным испытательным инструментом. В обоих случаях, оно должно быть способно эмулировать как клиента, так и сервер.

Для любых испытаний, использующих испытательный ТСР-трафик, эмулятор сетевой задержки (функция NDE, как показано на схеме лабораторной установки в 1.2) должен быть использован для получения значимого BDP. Как обсуждалось в 1.2, искомая скорость трафика и сконфигурированный RTT ДОЛЖНЫ быть независимо подтверждены с использованием только NDE для всех испытаний с сохранением состояния (для гарантии того, что NDE может добавлять задержку без внесения каких-либо потерь пакетов).

Испытательный ТСР-хост ДОЛЖЕН быть способен генерировать и принимать испытательный ТСР-трафик с сохранением состояния на полной канальной скорости DUT. Как ранее указывалось, испытание ТСР пропускной способности на скоростях более 500 Мбит/с может потребовать использование аппаратного высокопроизводительного сервера или отдельных аппаратных испытательных инструментов.

Испытательный ТСР-хост ДОЛЖЕН разрешать подстройку как размера буфера сокета передачи, так и размера буфера сокета приема. Буферы сокета должны быть достаточно большими, чтобы BDP мог достигать значения полного объема передачи испытательного ТСР-трафика приложения.

Измерение RTT и повторных передач на соединение будет, главным образом, требовать выделенного коммуникационного испытательного инструмента. При отсутствии выделенного аппаратного испытательного инструмента, такие измерения могут потребовать совместного управления с инструментом захвата пакетов, то есть управлять испытаниями ТСР пропускной способности и анализировать RTT и повторные передачи захватываемых пакетов.

Реализация ТСР, использованная испытательным хостом, ДОЛЖНА быть указана в результатах испытания (например, TCP New Reno, поддержка TCP-опций).

Дополнительно результаты испытания ОБЯЗАНЫ предоставлять конкретные детали алгоритма управления перегрузками, согласно [RFC3148].

Несмотря на то, что [RFC6349] описывает средства управления испытаниями пропускной способности ТСР групповой пересылки, базовая структура управления передачей данных будет расширена выполнением ТСР испытаний внутри интерактивного ТСР-трафика приложений. Примерами являются электронная почта, HTTP и бизнес-приложения. Такой интерактивный трафик является двухсторонним и может быть чато-образным, что означает наличие множества обратных подключений в процессе передачи трафика во время выполнения транзакции (в отличие от относительно однонаправленного индивидуального потока групповой передачи приложений).

Испытательное оборудование должно не только поддерживать групповую ТСР-передачу трафика приложения, но и ДОЛЖНО также поддерживать чато-образный трафик. В валидный стресс-тест СЛЕДУЕТ включать оба типа трафика. Это необходимо из-за неравномерной, всплескообразной природы чато-образных приложений по сравнению с относительно равномерной природой групповой передачи (групповая передача плавно стабилизируется к состоянию баланса в условиях отсутствия потерь).

В то время, как iperf является прекрасным выбором для испытания ТСР-групповой передачи, инструмент из открытых источников netperf предоставляет возможность управления поведением клиентских и серверных запросов/ответов. Инструмент netperf-wrapper представляет собой скрипт на языке Python, который запускает множество одновременных копий netperf и суммирует результаты. Приложение А предоставляет описание netperf/netperf-wrapper, также, как и iperf. Для любого программного инструмента должна быть квалифицирована производительность относительно испытываемой канальной скорости. Аппаратные испытательные инструменты следует проанализированы на надежность результатов при высоких канальных скоростях (например, 1 Гбит/с, 10 Гбит/с).

**5.2.1 Определения ТСР испытательных шаблонов**

Как указывалось, в целях для данной базовой структуры, определены техники спецификации испытательных шаблонов ТСР-трафика для эталонных испытаний методов управления передачей данных и получения повторяемых результатов. Некоторые сетевые устройства, такие как межсетевые экраны, не будут обрабатывать испытательный трафик без сохранения состояния, это является другой причиной почему должен быть использован испытательный ТСР-трафик с сохранением состояния.

Приложение может быть полностью эмулировано вплоть до уровня 7; тем не менее, данная базовая структура предлагает, чтобы были использованы ТСР испытательные шаблоны с сохранением состояния для того, чтобы обеспечить детализированное и повторяемое управление эталонными испытаниями. Следующая диаграмма иллюстрирует простое приложение для веб-браузинга (HTTP).



**Рисунок 3 – Диаграмма простого индивидуального потока для веб-приложения**

В данном примере веб-браузер клиента (клиент) запрашивает URL, и затем веб-сервер отправляет содержимое веб-страницы клиенту (после серверной задержки 100 мс). Такое асинхронное «запрос/ответ» поведение свойственно большинству основанных на ТСР приложений, таких как электронная почта (SMTP), передача файлов (FTP и блок сообщений сервера (SMB)), базы данных (SQL), веб-приложения (SOAP) и передача состояния представления (REST). Влияние на сетевые элементы обусловлено большим количеством клиентов и разнообразием всплескообразного (неравномерного) трафика, которые подвергают перегрузкам функции управления трафиком. Реальная эмуляция протоколов конкретных приложений не требуется, и ТСР испытательные шаблоны могут быть определены для имитации индивидуальных потоков сетевого трафика приложений и получения повторяемых результатов.

Техники моделирования приложений предложены в [3GPP2-C\_R1002-A], который предоставляет примеры для моделирования поведения приложений HTTP, FTP и протокола беспроводного приложения (WAP) на ТСР-уровне. Модели описаны с различными математическими законами распределения для байтов запроса/ответа и времени меж-запросных промежутков. Форматы определения модели, описанные в [3GPP2-C\_R1002-A], являются основой для руководства, представленного в приложении В, а также являются подобными форматам, используемым инструментами сетевого моделирования. Захват пакетов также может быть использован для характеризации трафика приложения и определения некоторых из испытательных шаблонов, перечисленных в приложении В.

Данная базовая структура не определяет фиксированный набор испытательных ТСР-шаблонов, но предоставляет блоки испытаний, которые СЛЕДУТ выполнить, см. приложение В. Некоторые из этих примеров воспроизводят указанные в [CA-Benchmark], где предлагаются смеси трафика для различных образцов профилей приложений. Другим примером является такой простой и хорошо известный тип трафика приложения, как НТТР.

**6 Эталонные испытания**

Методология эталонных трафиковых испытаний использует испытательную схему из 1.2 и метрики, определенные в 4.

В каждом испытании СЛЕДУЕТ сравнивать внутреннюю статистику сетевого устройства (доступную через командную строку управления, SNMP, и т.д.) с измеренными метриками, определенными в 4. Таким образом можно оценить точность внутренних счетчиков управления передачей данных в условиях индивидуального испытания и в условиях испытания производительности, как описано в 4.1 и 4.2. Такое сопоставление не предназначено для сравнения статистики реального времени, а скорее для подтверждения того, что сбор совокупной статистики после испытания завершен и счетчики устройства обновились (для счетчиков устройств характерно обновление через интервал 10 с или более).

С точки зрения конфигурации устройства, функциональность по диспетчеризации и шейпингу может быть применена к логическим портам (например, агрегация каналов (LAG)). Результатом этого будет применение одинаковой конфигурации диспетчеризации и шейпинга ко всем физическим портам-участникам. Настоящий стандарт сфокусирован только на испытаниях уровня физических портов.

Следующие разделы устанавливают цель, процедуру, метрики и форму представления отчетности для каждого испытания. Для всех шагов испытания должны быть указаны следующие глобальные параметры.

Число запусков испытания (Tr): число запусков испытания, необходимое для обеспечения точности и повторяемости результатов. Минимальным рекомендованным значением является 10.

Длительность испытания (Td): длительность испытательной итерации, выраженная в секундах. Минимальным рекомендованным значением является 60 с.

Вариабельность результатов испытания ДОЛЖНА быть измерена между запусками испытаний и, если вариация представляет собой значительную часть измеренных значений, то следующий шаг может потребовать изменения методов, для достижения лучшей сопоставимости.

**6.1 Испытания правил обработки трафика**

Ограничитель трафика определяют, как элемент, выполняющий функцию правил обработки трафика. Смысл испытаний правил обработки трафика заключается в подтверждении производительности ограничителя (то есть параметры CIR/CBS и EIR/EBS). Испытания будут проверять то, что сетевое устройство может обрабатывать CIR с CBS и EIR, и EBS, и будет использовать концепцию пакетного тестирования при предельной нагрузке, как описано в [RFC2544] (но адаптировано к алгоритмам и терминологии размера всплеска). Кроме того, [MEF-14], [MEF-19], и [MEF-37] предлагают некоторые основы для специфических компонентов данного испытания. Алгоритм отслеживания всплеска, определенный в 5.1.1, также может быть использован для автоматизации измерения величины CBS.

Испытания разделены на две (2) группы: индивидуальные испытания ограничителя и затем испытания полной производительности соблюдения правил обработки трафика. Важно сначала оценить базовую функциональность отдельного ограничителя и затем перейти к полной номинальной производительности устройства. Данная производительность может включать в себя ряд контрольных правил обработки трафика на устройство и ряд ограничителей, одновременно активных для всех портов.

**6.1.1 Индивидуальные испытания ограничителя**

**Цель**

Испытать ограничитель, как определено в [RFC4115] или [MEF-10.3], в зависимости от спецификации оборудования. Дополнительно для гарантии того, что ограничитель разрешает прохождение пачек с указанными CBS и EBS, при испытании ограничителя ДОЛЖНО проверяться то, что он будет отмечать или отбрасывать расширенные пакеты и пропускать трафик с указанными значениями CBS/EBS.

**Аннотация к испытанию**

При испытаниях правил обработки трафика следует использовать трафик без сохранения состояния. Испытательный ТСР трафик с сохранением состояния будет, в основном, подвергаться неблагоприятному воздействию со стороны ограничителя при отсутствии шейпинга трафика. Поэтому, хотя ТСР-трафик и может быть использован, однако более правильным является проверка ограничителя трафиком без сохранения состояния.

В качестве примера ограничителя, определенного в [RFC4115], предположим CBS/EBS равным 64 кБ и CIR/EIR равным 100 Мбит/с на физическом канале 1 Гбит/с (в режиме отсутствия окраски трафика). Всплеск трафика без сохранения состояния в 64 кБ должна быть послана в ограничитель на скорости GigE. Это соответствует временному промежутку всплеска примерно 0,512 мс (64 кБ на 1 GigE). Генератор трафика должен располагать эти всплески с таким интервалом, чтобы суммарная пропускная способность гарантировано не превышала CIR. В данном примере Ti между всплесками будет равен CBS \* 8 / CIR = = 5,12 мс.

**Метрики испытания**

Метрики, определенные в 4.1 (BSA, LP, OOS, PD, и PDV), ОБЯЗАНЫ быть измерены на выходном порту и записаны.

**Процедура**

1. Конфигурируют параметры правил обработки трафика в DUT для требуемых испытываемых значений CIR/EIR и CBS/EBS.

2. Конфигурируют испытательное оборудование для генерации пачек трафика без сохранения состояния эквивалентных CBS, и интервалом между ними эквивалентным Ti (CBS в битах/CIR).

3. Испытание трафиком, соответствующем правилам: генерируют трафик пачек CBS + EBS ко входному порту ограничителя и измеряют метрики, определенные в 4.1 (BSA, LP, OOS, PD, и PDV), на выходном порту в течении всего времени Td (по умолчанию, длительностью 60 с).

4. Испытание расширенным трафиком: генерируют всплески, превышающие CBS + EBS байт ко входному порту ограничителя и проверяют, что ограничитель разрешает на выход только BSA байты. Расширенный всплеск ДОЛЖЕН быть записан, рекомендованным значением является 1000 байт. Дополнительные испытания за пределами простого примера режима отсутствия окраски трафика могут включать режим окраски трафика, конфигурацию EIR превышающую CIR и т.д.

**Форма представления отчетности**

Индивидуальный протокол ограничителя ДОЛЖЕН содержать все результаты каждого запущенного в испытаниях CIR/EIR/CBS/EBS. Рекомендованным является следующий формат:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Краткая конфигурация испытания: Tr, Td

Краткая конфигурация DUT: CIR, EIR, CBS, EBS

Следует, чтобы таблица результатов содержала записи для каждого запуска испытания, как указано далее (от испытания #1 до испытания #Tr):

– испытание трафиком, соответствующем правилам: BSA, LP, OOS, PD, и PDV;

– испытание расширенным трафиком: BSA.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**6.1.2 Испытания полной производительности ограничителя**

**Цель**

Смысл испытаний полной производительности заключается в проверке характеристик ограничителя в масштабируемом окружении с множеством входных пользовательских ограничителей на множестве физических портов. Данное испытание будет оценивать максимальное число активных ограничителей, которое указано изготовителем устройства.

**Аннотация к испытанию**

Определенная полная производительность функции правил обработки трафика в основном выражается значением числа ограничителей активных на каждом индивидуальном физическом порту, также, как и числом уникальных скоростей ограничителя, которые им применяются. Для всех испытаний полной производительности, процедура эталонных испытаний и форма представления отчетности, описанные в 6.1.1 для одиночного ограничителя, ДОЛЖНЫ быть применены ко всем ограничителям физического порта.

Например, для коммутирующего устройства уровня 2 может быть указано в спецификации, что на каждом из 32 физических портов могут быть применены правила обработки трафика с использованием группы правил службы контроля соблюдения правил обработки. Устройство может транспортировать трафик одиночного пользователя на каждый физический порт и один ограничитель является привязанным к физическому порту. Другой возможностью является то, что один физический порт может транспортировать трафик множества пользователей и в этом случае множество индивидуальных пользовательских потоков будет подвержено правилам обработки параллельно на индивидуальном физическом порту (отдельные ограничители для каждого пользователя на индивидуальном порту).

**Метрики испытания**

Метрики, определенные в 4.1 (BSA, LP, OOS, PD, и PDV), ОБЯЗАНЫ быть измерены на выходном порту и записаны.

Следующие разделы устанавливают определенные испытательные сценарии, процедуры и формы представления отчетности для каждого испытания полной производительности ограничителя.

**6.1.2.1 Максимум ограничителей на одиночном физическом порту**

**Аннотация к испытанию**

Первое испытание полной производительности ограничителя будет оценивать один физический порт с максимумом ограничителей на этом физическом порту.

Допускается множество категорий входных ограничителей со скоростями r1, r2, …, rn. На одиночном физическом порту присутствует множество пользователей. Каждый пользователь может быть представлен VLAN с одним тегом, VLAN с двойным тегом, элементом виртуальной частной LAN службы (VPLS) и т.д. Каждый пользователь привязан к отдельному ограничителю. Каждый из ограничителей может использовать скорости r1, r2, …, rn.

Примером конфигурации будет:

– Y1 пользователей, скорость ограничителя r1;

– Y2 пользователей, скорость ограничителя r2;

– Y3 пользователей, скорость ограничителя r3;

…

– Yn пользователей, скорость ограничителя rn.

Некоторая часть пропускной способности физического порта выделена для иного трафика (т.е. иного, чем трафик пользователя), этот трафик включает в себя трафик протокола управления сетью. Для иного трафика присутствует отдельный ограничитель. Типичные реализации имеют три категории ограничителей, могут существовать некоторые реализации с более, чем тремя категориями входных ограничителей.

**Процедура**

1. Конфигурируют параметры правил обработки трафика в DUT для требуемых испытываемых значений CIR/EIR и CBS/EBS для каждой скорости (уровня) ограничителя (r1-rn).

2. Конфигурируют испытательное оборудование для генерации пачек трафика без сохранения состояния эквивалентных CBS и интервалом между ними эквивалентным Ti (CBS в битах/CIR) для каждого пользовательского потока (Y1-Yn). Инкапсуляция для каждого пользователя должна также быть сконфигурирована в соответствии с испытываемой службой (VLAN, VPLS, привязка по IP, и так далее).

3. Испытание трафиком, соответствующим правилам: генерируют трафик пачек CBS + EBS ко входному порту ограничителя для каждого потока пользовательского трафика, и измеряют метрики, определенные в 4.1 (BSA, LP, OOS, PD, и PDV), на выходном порту для каждого потока в течении всего времени Td (по умолчанию, длительностью 30 с).

4. Испытание расширенным трафиком: генерируют всплески, превышающие CBS + EBS байт ко входному порту ограничителя для каждого потока пользовательского трафика, и проверяют, что ограничитель разрешает на выход только BSA-байты для каждого потока. Расширенный всплеск ДОЛЖЕН быть записан, рекомендованным значением является 1000 байт.

**Форма представления отчетности**

Индивидуальный протокол ограничителя ДОЛЖЕН содержать все результаты каждого запущенного в испытаниях CIR/EIR/CBS/EBS для каждого потока пользовательского трафика. Рекомендованным является следующий формат:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Краткая конфигурация испытания: Tr, Td

Инкапсуляция потока пользовательского трафика: привязать каждый поток к VLAN, VPLS, IP-адресу

Краткая конфигурация DUT для каждого потока пользовательского трафика: CIR, EIR, CBS, EBS

Следует, чтобы таблица результатов содержала записи для каждого запуска испытания, как указано далее (от испытания #1 до испытания #Tr):

– пользовательский поток Y1-Yn (см. примечание) испытания трафиком, соответствующим правилам: BSA, LP, OOS, PD, и PDV;

– пользовательский поток Y1-Yn (см. примечание) испытания расширенным трафиком: BSA.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Примечание – Для каждого запуска испытания будет две (2) строки для каждого пользовательского потока: результат испытания трафиком, соответствующем правилам и результат испытания расширенным трафиком.

**6.1.2.2 Один ограничитель на всех физических портах**

**Аннотация к испытанию**

Второе испытание полной производительности ограничителя использует одиночную функцию ограничителя на каждом физическом порту со всеми физическими портами в активном состоянии. В этом испытании на каждом физическом порту один ограничитель. Ограничитель может применять одну из скоростей r1, r2, …, rn. Все физические порты в сетевом устройстве являются активными.

**Процедура**

Процедура для этого испытания идентична процедуре, приведенной в 6.1.1. Сконфигурированные параметры должны быть зафиксированы для каждого порта, а протокол испытания должен содержать результаты для каждого проверенного выходного порта.

**6.1.2.3 Максимум ограничителей на всех физических портах**

**Аннотация к испытанию**

Третье испытание полной производительности ограничителя является комбинацией первого и второго испытаний полной производительности, т.е. максимум ограничителей активно на каждом физическом порту и все физические порты активны.

**Процедура**

Процедура для этого испытания идентична процедуре, приведенной в 6.1.2.1. Сконфигурированные параметры должны быть зафиксированы для каждого порта, а протокол испытания должен содержать результаты для каждого потока и каждого проверенного выходного порта.

**6.2 Испытания формирователя очереди/диспетчера трафика**

Формирование очередей и диспетчеризация трафика находятся тесно взаимосвязанными в части того, что приоритет очереди определяет способ, которым диспетчер трафика передает пакеты наружу с выходного порта.

Поскольку очередь/буфер устройства является главным образом выходной функцией, то данная испытательная базовая структура будет рассматривать испытания на выходе (хотя технически возможно использование очередей с входной стороны).

Подобно испытаниям контроля соблюдения правил обработки трафика, данные испытания разделены на две группы: индивидуальные испытания функции формирователя очереди/диспетчера трафика и затем испытания полной производительности.

**6.2.1 Индивидуальные испытания формирователя очереди/диспетчера трафика**

Различные техники диспетчеризации включают в себя FIFO, организацию очереди со строгим приоритетом (SP) и организацию взвешенной равнодоступной очереди (WFQ), а также другие варианты. Данная базовая испытательная структура проводит испытания с минимум тремя техниками, хотя проверка других алгоритмов диспетчеризации в устройствах остается на усмотрение испытателя.

**6.2.1.1 Испытания формирователя очереди/диспетчера трафика с трафиком без сохранения состояния**

**Цель**

Проверить, что сконфигурированные очередь и используемая техника диспетчеризации способны обрабатывать всплески трафика без сохранения состояния на всю глубину заполнения очереди.

**Аннотация к испытанию**

Очередь в сетевом устройстве базируется на его памяти в отличие от функции контроля соблюдения правил обработки, которая базируется на маркере или разрешении на передачу. Тем не менее, такая же концепция, как и в 6.1, может быть использована для испытаний формирователя очередей в сетевом устройстве.

Следует, чтобы формирователь сетевой очереди в устройстве был сконфигурирован на желаемый размер в килобайтах (т.е. длина очереди (QL)), а затем следует передавать трафик без сохранения состояния для проверки этой QL.

Следует, чтобы формирователь очереди был способен обрабатывать периодически повторяющиеся всплески с прерываниями передачи пропорциональными минимальной пропускной способности канала передачи данных (BB). Вышеупомянутое прерывание передачи используется здесь, как интервал передачи (Ti). Ti может быть определен для пачек трафика в зависимости от QL и BB выходного интерфейса.

Ti = QL \* 8 / BB

Необходимо отметить, что данная формула подобна Ti, требуемом для передачи в ограничитель (QL = CBS, BB = CIR). Отметим также, что алгоритм отслеживания всплеска, описанный в 5.1.1, также может быть использован для автоматизации измерения значения очереди.

Всплеск трафика без сохранения состояния ОБЯЗАНА быть передана на скорости канала внутри интервала передачи (Ti). Метрики, определенные в 4.1 ОБЯЗАНЫ быть измерены на выходном порту и записаны, основной целью является подтверждение BSA и проверка того, что пакеты не отбрасываются.

Функция диспетчеризации должна также быть характеризована для испытания способности устройства распределять очереди согласно приоритету. Примером может быть двухуровневый приоритет, который включает SP и FIFO формирование очередей. При нагрузке индивидуального потока превышающей скорость выходного порта, пакеты с наивысшим приоритетом должны будут передаваться без потерь (а также, с поддержкой малого времени задержки), в то время, как очередь с более низким приоритетом (или best-effort) может быть отброшена.

**Метрики испытания**

Метрики, определенные в 4.1 (BSA, LP, OOS, PD, и PDV), ОБЯЗАНЫ быть измерены на выходном порту и записаны.

**Процедура**

1. Конфигурируют параметры QL и техники диспетчеризации в DUT (FIFO, SP и так далее).

2. Конфигурируют испытательное оборудование для генерации всплеска трафика без сохранения состояния эквивалентной QL и интервалом эквивалентным Ti (QL в битах/BB).

3.Генерируют QL всплески в направлении DUT и измеряют метрики, определенные в 4.1 (LP, OOS, PD, и PDV), на выходном порту в течении полного времени Td (по умолчанию, длительностью 30 с).

**Форма представления отчетности**

Индивидуальный протокол для формирователя очереди/диспетчера трафика без сохранения состояния ДОЛЖЕН содержать все результаты для каждого запущенного в испытаниях QL/BB. Рекомендованным является следующий формат:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Краткая конфигурация испытания: Tr, Td

Краткая конфигурация DUT: техника диспетчеризации (например, FIFO, SP, WFQ, и так далее), BB и QL

Следует, чтобы таблица результатов содержала записи для каждого запуска испытания, как указано далее (от испытания #1 до испытания #Tr):

– LP, OOS, PD и PDV.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**6.2.1.2 Испытания формирователя очереди/диспетчера трафика с трафиком с сохранением**

**состояния**

**Цель**

Проверить, что сконфигурированный формирователь очереди и используемая техника диспетчеризации способны обрабатывать всплески трафика с сохранением состояния на всю глубину заполнения очереди.

**Аннотация к испытанию**

Для обеспечения большей реалистичности эталонного испытания и для проверки формирователя очередей в устройствах уровня 4, таких, как межсетевые экраны, испытания с трафиком с сохранением состояния являются рекомендованными для проверки формирователя очередей. Испытания с трафиком с сохранением состояния также будут использовать эмулятор сетевой задержки (NDE) из конфигурации испытательной установки, описанной в 1.2.

BDP для испытательного ТСР трафика должно быть калибровано для значения QL формирователя очереди устройства. Согласно рекомендациям [RFC6349], BDP эквивалентен следующему выражению:

BB \* RTT / 8 (в байтах)

NDE должен быть сконфигурирован на значение RTT, достаточное для разрешения BDP быть большим, чем QL. Пример испытательного сценария приведен ниже:

– входной канал = GigE;

– выходной канал = 100 Мбит/с (BB);

– QL = 32 кБ

RTT (min) = QL \* 8 / BB и будет эквивалентно 2,56 мс

 (и BDP = 32 кБ)

В данном примере одно (1) ТСР-соединение с размером окна/SSB, равным 32 кБ, будет требоваться для испытания QL равной 32 кБ. Данное испытание ТСР-групповой передачи может быть выполнено с использованием iperf, как описано в приложении А.

Два типа ТСР-испытаний ДОЛЖНЫ быть выполнены: испытание ТСР-групповой передачи и испытательный шаблон микро-пачек, как задокументировано в приложении В. В испытании ТСР-групповой передачи всплески присутствуют только во время состояния медленного старта ТСР (или предотвращения перегрузок), в то время, как испытательный шаблон микро-всплесков эмулирует всплескообразность уровня приложений, которая может возникать в любое время в течении ТСР-соединения.

Другими типами испытаний, которые СЛЕДУЕТ включать, являются следующие: простые веб-сайты, сложные веб-сайты, бизнес-приложения, электронная почта и SMB/CIFS (основная система интернет файлов) копирование файла (все из которых также задокументированы в приложении В).

**Метрики испытания**

Результаты испытания будут фиксироваться для метрик с сохранением состояния, определенных в 4.2 – главным образом, это время выполнения ТСР испытательного шаблона (TTPET), Эффективность ТСР и задержка в буфере.

**Процедура**

1. Конфигурируют параметры QL и техники диспетчеризации в DUT (FIFO, SP и т.д.).

2. Конфигурируют испытательный генератор\* с профилем смеси трафика эмулированного приложения.

– Смесь приложений ДОЛЖНА быть определена в значениях процента от общей испытываемой пропускной способности канала передачи данных.

– Скорость передачи для каждого приложения внутри смеси ДОЛЖНА также быть конфигурируемой.

\* Для гарантии повторяемости результатов, испытательный генератор ДОЛЖЕН быть способен генерировать точный испытательный ТСР шаблон для каждого указанного приложения.

3. Генерируют трафик приложения между входными (сторона клиента) и выходными (сторона сервера) портами DUT и измеряют метрики (TTPET, эффективность ТСР и задержку в буфере) для каждого потока приложения на входных и выходных портах в течении полного времени Td (по умолчанию, длительностью 60 с).

Ряд пунктов требуют прояснения касательно измерений приложения: сессия приложения может состоять из одиночного ТСР-соединения, либо из множества ТСР-соединений.

Если сессия приложения использует одиночное ТСР-соединение, то пропускная способность/метрики приложения имеют 1-1 взаимосвязь с измерениями ТСР-соединения.

Если сессия приложения (например, приложения на основе НТТР) использует множество ТСР-соединений, то все ТСР-соединения суммируются в части пропускной способности/метрик для этого приложения.

Кроме того, есть случай многократных реализаций сессии приложения (т.е., множество FTP, эмулирующих множество клиентов). В такой ситуации в испытании следует измерять/фиксировать каждую сессию приложения FTP отдельно, указывая в табличной форме минимум, максимум и среднее для всех FTP-сессий.

И, наконец, измерения пропускной способности приложения основаны на ТСР пропускной способности уровня 4 и не включают повторно переданные байты. Метрика эффективности ТСР ДОЛЖНА быть измерена в течение испытания, так как благодаря этому обеспечивается измерение «goodput» в течение каждого испытания.

**Форма представления отчетности**

Индивидуальный протокол формирователя очереди/диспетчера трафика с сохранением состояния ДОЛЖЕН содержать все результаты для каждого запущенного в испытаниях диспетчера трафика и QL/BB. Рекомендованным является следующий формат:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Краткая конфигурация испытания: Tr, Td

Краткая конфигурация DUT: техника диспетчеризации (например, FIFO, SP, WFQ и т.д.), BB и QL

Смесь приложения и интенсивность: проценты, сконфигурированные для каждого типа приложения

Следует, чтобы таблица результатов содержала записи для каждого запуска испытания, с максимумом, минимумом и средним для каждой сессии приложения, как указано далее (от испытания #1 до испытания #Tr):

– пропускная способность (бит/с) и ТТРЕТ для каждой сессии приложения;

– байты на входе и байты на выходе для каждой сессии приложения;

– эффективность ТСР и задержка в буфере для каждой сессии приложения

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**6.2.2 Испытания полной производительности формирователя очереди/диспетчера трафика**

**Цель**

Цель данных испытаний полной производительности в эталонной проверке характеристик производительности формирователя очереди/диспетчера трафика в масштабируемом окружении с множеством формирователей очереди/диспетчеров трафика, активных на множестве выходных физических портах. Данные испытания будут проверять максимальное количество формирователей очереди/диспетчеров трафика, указанное изготовителем устройства. Каждый приоритет в системе будет привязываться к отдельной очереди.

**Метрики испытания**

Метрики, определенные в 4.1 (BSA, LP, OOS, PD и PDV), ОБЯЗАНЫ быть измерены на выходном порту и записаны.

Следующие разделы устанавливают определенные испытательные сценарии, процедуры и формы представления отчетности для каждого испытания полной производительности формирователя очереди/диспетчера трафика.

**6.2.2.1 Множество очередей, активен один порт**

Для первого испытания полной производительности формирователя очереди/диспетчера трафика будет проверено множество очередей на порт на одиночном физическом порту. В этом случае, все очереди (обычно их восемь) являются активными на одиночном физическом порту. Трафик от множества входных физических портов направляют на один и тот же выходной физический порт. Это приведет к возникновению превышения предела подписки на выходном физическом порту.

Существует множество типов схем приоритезации и комбинаций приоритетов, которыми управляет диспетчер трафика. Следующие разделы определяют схемы приоритезации, которые следует испытывать.

**6.2.2.1.1 Строгий приоритет на выходном порту**

**Аннотация к испытанию**

Для данного испытания, следует проверять SP-диспетчеризацию на выходном физическом порту, при этом следует использовать методологии эталонных испытаний, определенные в 6.2.1.1 (без сохранения состояния) и 6.2.1.2 (с сохранением состояния) (процедура, метрики и форма представления отчетности). Для заданного приоритета следует, чтобы каждый входной физический порт получал подходящую часть пропускной способности выходного физического интерфейса.

Так как данные испытания проверяют полную производительность, то конфигурация и форма представления отчетности (см. 6.2.1.1 и 6.2.1.2) ДОЛЖНЫ также содержать:

**Конфигурация**

– Число физических входных портов, активных во время испытания.

– Классификация маркировки (DSCP, VLAN и т.д.) для каждого физического входного порта.

– Скорость трафика для трафика без сохранения состояния и скорость трафика/смесь для трафика с сохранением состояния для каждого физического входного порта.

**Форма представления отчетности**

Для каждого потока трафика входного порта достигнутый уровень пропускной способности и метрики на выходном порту.

**6.2.2.1.2 Строгий приоритет плюс WFQ на выходном порту**

**Аннотация к испытанию**

Для данного испытания, следует чтобы SP и WFQ были включены одновременно в диспетчере трафика, но на одиночном выходном порту. Здесь следует использовать методологии эталонных испытаний, определенные в 6.2.1.1 (без сохранения состояния) и 6.2.1.2 (с сохранением состояния) (процедура, метрики и форма представления отчетности). Дополнительно, следует чтобы распределение пропускной способности выходного порта между взвешенными очередями было пропорциональным, согласно назначенным весам. Для заданного приоритета следует, чтобы каждый входной физический порт получал достаточную часть пропускной способности выходного физического интерфейса.

Так как данные испытания проверяют полную производительность, то конфигурация и форма представления отчетности (см. 6.2.1.1 и 6.2.1.2) ДОЛЖНЫ также содержать:

**Конфигурация**

– Число физических входных портов, активных во время испытания.

– Классификация маркировки (DSCP, VLAN, и т.д.) для каждого физического входного порта.

– Скорость трафика для трафика без сохранения состояния и скорость трафика/смесь для трафика с сохранением состояния для каждого физического входного порта.

**Форма представления отчетности**

Для каждого потока трафика входного порта достигнутый уровень пропускной способности и метрики для каждой очереди на выходном порту (для SP и WFQ).

Пример:

– SP очередь на выходном порту: пропускная способность и метрики для входных потоков 1-n;

– WFQ на выходном порту: пропускная способность и метрики для входных потоков 1-n.

**6.2.2.2 Одиночная очередь на порт, все порты активны**

**Аннотация к испытанию**

Трафик от множества входных физических портов направляют на один и тот же выходной физический порт. Это приведет к возникновению превышения предела подписки на выходном физическом порту. Кроме того, такой же объем трафика направляют к каждому физическому выходному порту.

Здесь следует использовать методологии эталонных испытаний, определенные в 6.2.1.1 (без сохранения состояния) и 6.2.1.2 (с сохранением состояния) (процедура, метрики и форма представления отчетности). Каждый входной физический порт получал подходящую часть пропускной способности выходного физического интерфейса. Дополнительно, следует чтобы каждый выходной физический порт принимал одинаковый объем трафика.

Так как данные испытания проверяют полную производительность, то конфигурация и форма представления отчетности (см. 6.2.1.1 и 6.2.1.2) ДОЛЖНЫ также содержать:

**Конфигурация**

– Число входных портов, активных во время испытания.

– Число выходных портов, активных во время испытания.

– Классификация маркировки (DSCP, VLAN и т.д.) для каждого физического входного порта.

– Скорость трафика для трафика без сохранения состояния и скорость трафика/смесь для трафика с сохранением состояния для каждого физического входного порта.

**Форма представления отчетности**

Для каждого выходного порта достигнутый уровень пропускной способности и метрики каждой очереди выходного порта для каждого потока от входного порта

Пример:

– выходной порт 1: пропускная способность и метрики для входных потоков 1-n;

– выходной порт n: пропускная способность и метрики для входных потоков 1-n.

**6.2.2.3 Множество очередей на порт, все порты активны**

**Аннотация к испытанию**

Трафик от множества входных физических портов направляют во все очереди каждого выходного физического порта. Это приведет к возникновению превышения предела подписки на выходном физическом порту. Кроме того, такой же объем трафика направляют к каждому физическому выходному порту.

Здесь следует использовать методологии эталонных испытаний, определенные в 6.2.1.1 (без сохранения состояния) и 6.2.1.2 (с сохранением состояния) (процедура, метрики и форма представления отчетности). Для заданного приоритета следует, чтобы каждый входной физический порт получал подходящую часть пропускной способности выходного физического интерфейса. Дополнительно, следует чтобы каждый выходной физический порт принимал одинаковый объем трафика.

Так как данные испытания проверяют полную производительность, то конфигурация и форма представления отчетности (см. 6.2.1.1 и 6.2.1.2) ДОЛЖНЫ также содержать:

**Конфигурация**

– Число входных физических портов, активных во время испытания.

– Классификация маркировки (DSCP, VLAN и т.д.) для каждого физического входного порта.

– Скорость трафика для трафика без сохранения состояния и скорость трафика/смесь для трафика с сохранением состояния для каждого физического входного порта.

**Форма представления отчетности**

Для каждого выходного порта достигнутый уровень пропускной способности и метрики каждой очереди выходного порта для каждого потока от входного порта.

Пример:

– выходной порт 1, SP очередь: пропускная способность и метрики для входных потоков 1-n;

– выходной порт 2, WFQ: пропускная способность и метрики для входных потоков 1-n;

– выходной порт n, SP очередь: пропускная способность и метрики для входных потоков 1-n;

– выходной порт 1, WFQ: пропускная способность и метрики для входных потоков 1-n.

**6.3 Испытания шейпера**

Подобно формирователю очереди, шейпер трафика базируется на памяти, но с добавленным интеллектом к активному диспетчеру трафика. Концепция, аналогичная описанной в 6.2 (испытания формирователя очереди), может быть применена для испытания шейпера сетевого устройства.

Испытания снова разделяют на две группы: индивидуальные испытания шейпера и затем испытания полной производительности шейпера.

**6.3.1 Индивидуальные испытания шейпера**

Шейпер, главным образом, имеет три компоненты, которые могут быть сконфигурированы:

– байты входной очереди;

– скорость шейпера (SR), бит/с;

– согласованный всплеск (Bc) и расширенный всплеск пачки (Be) в байтах.

Входная очередь удерживает всплескообразный трафик, а затем шейпер выполняет дозированную передачу трафика на выход в соответствии с параметрами SR и Вс/Ве. Шейпер, главным образом, выполняет передачу в сторону ограничителей, смыслом чего является соответствие эмитируемого трафика пределам ограничителя.

**6.3.1.1 Испытания шейпера с трафиком без сохранения состояния**

**Цель**

Испытание шейпера передачей пачек трафика без сохранения состояния в сторону шейпера входного порта и проверка того, что выходной трафик сформирован в соответствии с трафиковым профилем шейпера.

**Аннотация к испытанию**

Трафик без сохранения состояния должен быть выброшен в сторону входного порта DUT и не должен превышать значение входной очереди. Выброс может представлять собой одиночный всплеск или множество всплесков. Если передано множество всплесков, то интервал передачи (Ti) должен быть достаточным, чтобы SR не была превышена. В примере будут объяснены случаи с одиночным всплеском и множеством всплесков.

В данном примере входной и выходной порты шейпера являются полнодуплексными Gigabit Ethernet. Входная очередь сконфигурирована на значение 512 000 байт, SR = 50 Мбит/с, а Bc и Ве оба сконфигурированы на значение 32 000 байт. Для испытания одиночным всплеском передающее испытательное оборудование будет выбрасывать максимум 512 000 байт в сторону входного порта и затем прекращать передачу.

Если предполагается испытание с множеством всплесков, то выброс байт, разделенный интервалом передачи между всплесками 512 000 байт, не должен превышать SR. Интервал передачи (Ti) должен соответствовать формуле, подобной формуле, описанной в 6.2.1.1 для формирователя очередей, а именно:

Ti = Ingress Queue \* 8 / SR

Для предыдущего примера Ti между всплесками должен быть больше чем 82 мс (512 000 байт \* 8 / 50 000 000 бит/с). Это обеспечивает среднюю скорость около 50 Мбит/с, поэтому входная очередь не будет переполнена.

**Метрики испытания**

Метрики, определенные в 4.1 (LP, OOS, PDV, SR, SBB и SBI), ОБЯЗАНЫ быть измерены на выходном порту и записаны.

**Процедура**

1. Конфигурируют в шейпере DUT параметры входной QL и выходной скорости шейпера (SR, Bc, Be).

2. Конфигурируют генератор на генерацию всплеска трафика без сохранения состояния, эквивалентную QL и интервал эквивалентный Ti (QL в биты/ВВ).

3. Генерируют трафик QL-пачек в сторону DUT и измеряют метрики, определенные в 4.1 (LP, OOS, PDV, SR, SBB и SBI) на выходном порту в течении полного времени Td (по умолчанию, длительностью 30 с).

**Форма представления отчетности**

Индивидуальный протокол шейпера с трафиком без сохранения состояния ДОЛЖЕН содержать все результаты для каждого запущенного в испытаниях QL/SR. Рекомендованным является следующий формат:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Краткая конфигурация испытания: Tr, Td

Краткая конфигурация DUT: входная скорость всплеска, QL, SR

Следует, чтобы таблица результатов содержала записи для каждого запуска испытания, как указано далее (от испытания #1 до испытания #Tr):

– LP, OOS, PDV, SR, SBB и SBI

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**6.3.1.2 Испытания шейпера с трафиком c сохранением состояния**

**Цель**

Испытание шейпера передачей пачек трафика с сохранением состояния в сторону шейпера входного порта и проверка того, что выходной трафик сформирован в соответствии с трафиковым профилем шейпера.

**Аннотация к испытанию**

Для обеспечения большей реалистичности эталонного испытания и для проверки формирователя очередей в устройствах уровня 4, таких, как межсетевые экраны, испытания с трафиком с сохранением состояния являются также рекомендованными для проверки шейпера. Испытания с трафиком с сохранением состояния также будут использовать эмулятор сетевой задержки (NDE) из конфигурации испытательной установки, описанной в 1.2.

BDP для испытательного ТСР-трафика должно быть рассчитано так, как описано в 6.2.1.2. Для правильной нагрузки сетевых буферов и функции шейпинга трафика следует, чтобы размер окна ТСР (который является минимумом из значений TCP RWND и сокета отправителя) был больше, чем BDP, которым будет нагружаться шейпер. BDP коэффициенты от 1,1 до 1,5 являются рекомендованными, но значения, оставшиеся на усмотрение испытателя, следует задокументировать.

Совокупный размер ТСР-окна\* (RWND на принимающем конце и CWND на передающем конце) устанавливается равным размеру ТСР-окна для каждого соединения умноженному на число соединений.

\* Как описано в разделе 3 [RFC6349], SSB ДОЛЖЕН быть достаточно большим, чтобы заполнить BDP.

Например, если BDP эквивалентен 256 кБ, а для каждого соединения используется размер окна соединения 64 кБ, то потребуется четыре (4) соединения для заполнения BDP и 5-6 соединений (превышение предела подписки BDP) для стресс-теста функции шейпинга трафика.

ДОЛЖНЫ быть выполнены два типа ТСР-испытаний: испытание ТСР-групповой передачи и испытательный шаблон микро-пачек, как задокументировано в приложении В. В испытании ТСР-групповой передачи, всплески присутствуют только во время состояния медленного старта ТСР (или предотвращения перегрузок), в то время, как испытательный шаблон микро-пачек эмулирует всплескообразность уровня приложений, которая может возникать в любое время в течении ТСР-соединения.

Другими типами испытаний, которые СЛЕДУЕТ включать, являются следующие: простые веб-сайты, сложные веб-сайты, бизнес-приложения, электронная почта и SMB/CIFS (основная система интернет файлов) копирование файла (все из которых также задокументированы в приложении В).

**Метрики испытания**

Результаты испытания будут фиксироваться для метрик с сохранением состояния, определенных в 4.2, главным образом, это время выполнения ТСР испытательного шаблона (TTPET), эффективность ТСР и задержка в буфере.

**Процедура**

1. Конфигурируют в DUT параметры входной QL и выходной скорости шейпера (SR, Bc, Be).

2. Конфигурируют испытательный генератор\* с профилем смеси трафика эмулированного приложения.

Смесь приложений ДОЛЖНА быть определена в значениях процента от общей испытываемой пропускной способности.

Скорость передачи для каждого приложения внутри смеси ДОЛЖНА также быть конфигурируемой.

\* Для гарантии повторяемости результатов, испытательный генератор ДОЛЖЕН быть способен генерировать точный испытательный ТСР-шаблон для каждого указанного приложения.

3. Генерируют трафик приложения между входными (сторона клиента) и выходными (сторона сервера) портами DUT и измеряют метрики (TTPET, эффективность ТСР и задержку в буфере) для каждого потока приложения на входных и выходных портах в течении полного времени Td (по умолчанию, длительностью 30 с).

**Форма представления отчетности**

Индивидуальный протокол шейпера с трафиком с сохранением состояния ДОЛЖЕН содержать все результаты для каждого запущенного в испытаниях диспетчера трафика и QL/SR. Рекомендованным является следующий формат:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Краткая конфигурация испытания: Tr, Td

Краткая конфигурация DUT: входная скорость всплеска, QL, SR

Смесь приложения и интенсивность: проценты, сконфигурированные для каждого типа приложения

Следует, чтобы таблица результатов содержала записи для каждого запуска испытания, с максимумом, минимумом и средним для каждой сессии приложения, как указано далее (от испытания #1 до испытания #Tr):

– пропускная способность (бит/с) и ТТРЕТ для каждой сессии приложения;

– байты на входе и байты на выходе для каждой сессии приложения;

– эффективность ТСР и задержка в буфере для каждой сессии приложения.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**6.3.2 Испытания полной производительности шейпера**

**Цель**

Цель данных испытаний масштабируемости в проверке характеристик производительности шейпера в масштабируемом окружении с множеством шейперов, активных на множестве очередей множества выходных физических портов. Данные испытания будут проверять максимальное количество шейперов, указанное изготовителем устройства.

Следующие разделы устанавливают определенные испытательные сценарии, процедуры и форматы протоколирования для каждого испытания полной производительности шейпера.

**6.3.2.1 Один формирователь очереди с шейпингом, все физические порты активны**

**Аннотация к испытанию**

Первое испытание полной производительности шейпера включает в себя шейпинг на каждом порту со всеми физическими портами в активном состоянии. Трафик от множества входных физических портов направлен к одному и тому же выходному физическому порту. Это будет вызывать превышение предела подписки на выходном физическом порту. Кроме того, такой же объем трафика направлен к каждому выходному физическому порту.

Здесь следует использовать методологии эталонных испытаний, описанные в 6.3.1.1 (без сохранения состояния) и 6.3.1.2 (с сохранением состояния) (процедура, метрики и форма представления отчетности). Так как данное испытание проверяет полную производительность, то конфигурация и протокол результатов ДОЛЖНЫ также содержать:

**Конфигурация**

– Число физических входных портов, активных во время испытания.

– Классификация маркировки (DSCP, VLAN и т.д.) для каждого физического входного порта.

– Скорость трафика для трафика без сохранения состояния и скорость трафика/смесь для трафика с сохранением состояния для каждого физического входного порта.

– Параметры шейпера выходного порта с шейпингом (QL, SR, Bc, Be).

**Форма представления отчетности**

Для каждого активного выходного порта, достигнутый уровень пропускной способности и метрики шейпера для потока трафика каждого входного порта

Пример:

– выходной порт 1: пропускная способность и метрики для входных потоков 1-n;

– выходной порт n: пропускная способность и метрики для входных потоков 1-n.

**6.3.2.2 Все формирователи очередей с шейпингом, один порт активен**

**Аннотация к испытанию**

Второе испытание полной производительности шейпера выполняют со всеми формирователями очередей с активным шейпингом на одном физическом порту. Методология эталонных испытаний, описанная в испытании шейпинга на каждом порту (см. 6.3.2.1), служит основой для данного испытания.

Дополнительно, каждая из SP очередей на выходном физическом порту сконфигурирована с формирователем трафика. Для очереди с высоким приоритетом, максимальный размер доступной пропускной способности канала передачи данных ограничен пропускной способностью формирователя трафика. Для очередей с низким приоритетом, максимальный размер доступной пропускной способности канала передачи данных ограничен пропускной способностью формирователя трафика и трафиком в очередях с высоким приоритетом.

Здесь следует использовать методологии эталонных испытаний, описанные в 6.3.1.1 (без сохранения состояния) и 6.3.1.2 (с сохранением состояния) (процедура, метрики и форма представления отчетности). Так как данное испытание проверяет полную производительность, то конфигурация и протокол результатов (см. 6.3.1) ДОЛЖНЫ также содержать:

**Конфигурация**

– Число физических входных портов, активных во время испытания.

– Классификацию маркировки (DSCP, VLAN и т.д.) для каждого физического входного порта.

– Скорость трафика для трафика без сохранения состояния и скорость трафика/смесь для трафика с сохранением состояния для каждого физического входного порта.

– Параметры очереди шейпера для активного выходного порта: QL, SR, Bc, Be.

**Форма представления отчетности**

Для каждой очереди активного выходного порта, достигнутый уровень пропускной способности и метрики шейпера для потока трафика каждого входного порта.

Пример:

– выходной порт, очередь с высоким приоритетом: пропускная способность и метрики для входных потоков 1-n;

– выходной порт, очередь с низким приоритетом: пропускная способность и метрики для входных потоков 1-n.

**6.3.2.3 Все формирователи очередей с шейпингом, все порты активны**

**Аннотация к испытанию**

Для третьего испытания полной производительности шейпера (которое является комбинацией испытаний, описанных в 6.3.2.1 и 6.3.2.2), все формирователи очередей будут активно выполнять шейпинг и все физические порты будут активны.

Здесь следует использовать методологии эталонных испытаний, описанные в 6.3.1.1 (без сохранения состояния) и 6.3.1.2 (с сохранением состояния) (процедура, метрики и форма представления отчетности). Так как данное испытание проверяет полную производительность, то конфигурация и протокола результатов (см. 6.3.1) ДОЛЖНЫ также содержать:

**Конфигурация**

– Число физических входных портов, активных во время испытания.

– Классификация маркировки (DSCP, VLAN и т.д.) для каждого физического входного порта.

– Скорость трафика для трафика без сохранения состояния и скорость трафика/смесь для трафика с сохранением состояния для каждого физического входного порта.

– Для каждого активного выходного порта: параметры шейпера порта и параметры каждого формирователя очереди (QL, SR, Bc, Be).

**Форма представления отчетности**

Для каждой очереди активного выходного порта, достигнутый уровень пропускной способности и метрики шейпера для потока трафика каждого входного порта.

Пример:

– выходной порт 1, очередь с высоким приоритетом: пропускная способность и метрики для входных потоков 1-n;

– выходной порт 1, очередь с низким приоритетом: пропускная способность и метрики для входных потоков 1-n;

…

– выходной порт n, очередь с высоким приоритетом: пропускная способность и метрики для входных потоков 1-n;

– выходной порт n, очередь с низким приоритетом: пропускная способность и метрики для входных потоков 1-n.

**6.4 Нагрузочные испытания параллельной производительности**

Как упоминалось ранее в 3, существует возможность указания различных комбинаций параллельно работающих функций управления передачей данных, которые следует проверить при испытании полной производительности. Тем не менее, некоторые профили, указанные далее, могут быть использованы для испытания множества конфигураций функций управления передачей данных:

– ограничители на входе и формирователи очередей на выходе;

– ограничители на входе и шейперы на выходе (индивидуальный поток не должен быть сначала ограничен согласно правил обработки трафика, а затем подвержен шейпингу, это должно быть два различных индивидуальных потока, испытываемых в одно и то же время).

Процедуры испытаний и формы представления отчетности из 6.1, 6.2 и 6.3 могут быть модифицированы для согласования с профилем испытания полной производительности.

**7 Аспекты безопасности**

Настоящий стандарт не оказывает прямого влияния на безопасность Интернет или корпоративных сетей до тех пор, пока эталонные проверки не выполняются на устройствах или системах, подключенных к производственным сетям.

В дальнейшем эталонные проверки выполняют на базе «черного ящика», опираясь исключительно на измерения, выполняемые вне DUT/SUT.

НЕ СЛЕДУЕТ, чтобы в DUT/SUT имелись специальные возможности для целей эталонных проверок. СЛЕДУЕТ, чтобы любые последствия для сетевой безопасности, возникающие от DUT/SUT были одинаковы как в лаборатории, так и в производственных сетях.

**8 Ссылки**

**8.1 Нормативные ссылки**

|  |  |
| --- | --- |
| [3GPP2-C\_R1002-A] | 3rd Generation Partnership Project 2, "cdma2000 EvaluationMethodology", Version 1.0, Revision A, May 2009,<http://www.3gpp2.org/public\_html/specs/C.R1002-A\_v1.0\_Evaluation\_Methodology.pdf>. |
|  |  |
| [RFC 1242] | Bradner, S., "Benchmarking Terminology for NetworkInterconnection Devices", RFC 1242, DOI 10.17487/RFC1242,July 1991, <http://www.rfc-editor.org/info/rfc1242>. |
|  |  |
| [RFC 2119] | Bradner, S., "Key words for use in RFCs to IndicateRequirement Levels", BCP 14, RFC 2119,DOI 10.17487/RFC2119, March 1997,<http://www.rfc-editor.org/info/rfc2119>. |
|  |  |
| [RFC 2544] | Bradner, S. and J. McQuaid, "Benchmarking Methodology forNetwork Interconnect Devices", RFC 2544,DOI 10.17487/RFC2544, March 1999,<http://www.rfc-editor.org/info/rfc2544>. |
|  |  |
| [RFC 2680] | Almes, G., Kalidindi, S., and M. Zekauskas, "A One-wayPacket Loss Metric for IPPM", RFC 2680,DOI 10.17487/RFC2680, September 1999,<http://www.rfc-editor.org/info/rfc2680>. |
|  |  |
| [RFC 3148] | Mathis, M. and M. Allman, "A Framework for DefiningEmpirical Bulk Transfer Capacity Metrics", RFC 3148,DOI 10.17487/RFC3148, July 2001,<http://www.rfc-editor.org/info/rfc3148>. |
|  |  |
| [RFC 4115] | Aboul-Magd, O. and S. Rabie, "A Differentiated ServiceTwo-Rate, Three-Color Marker with Efficient Handling ofin-Profile Traffic", RFC 4115, DOI 10.17487/RFC 4115,July 2005, <http://www.rfc-editor.org/info/rfc4115>. |
|  |  |
| [RFC 4689] | Poretsky, S., Perser, J., Erramilli, S., and S. Khurana,"Terminology for Benchmarking Network-layer TrafficControl Mechanisms", RFC 4689, DOI 10.17487/RFC 4689,October 2006, <http://www.rfc-editor.org/info/rfc4689>. |
|  |  |
| [RFC 4737] | Morton, A., Ciavattone, L., Ramachandran, G., Shalunov,S., and J. Perser, "Packet Reordering Metrics", RFC 4737,DOI 10.17487/RFC4737, November 2006,<http://www.rfc-editor.org/info/rfc4737>. |
|  |  |
| [RFC 5481] | Morton, A. and B. Claise, "Packet Delay VariationApplicability Statement", RFC 5481, DOI 10.17487/RFC 5481,March 2009, <http://www.rfc-editor.org/info/rfc5481>. |
|  |  |
| [RFC 6349] | Constantine, B., Forget, G., Geib, R., and R. Schrage,"Framework for TCP Throughput Testing", RFC 6349,DOI 10.17487/RFC6349, August 2011,<http://www.rfc-editor.org/info/rfc6349>. |
|  |  |
| [RFC 6703] | Morton, A., Ramachandran, G., and G. Maguluri, "ReportingIP Network Performance Metrics: Different Points of View",RFC 6703, DOI 10.17487/RFC6703, August 2012,<http://www.rfc-editor.org/info/rfc6703>. |
|  |  |
| [SPECweb2009] | Standard Performance Evaluation Corporation (SPEC),"SPECweb2009 Release 1.20 Benchmark Design Document",April 2010, <https://www.spec.org/web2009/docs/design/SPECweb2009\_Design.html>. |

**8.2 Информационные ссылки**

|  |  |
| --- | --- |
| [CA-Benchmark] | Hamilton, M. and S. Banks, "Benchmarking Methodology forContent-Aware Network Devices", Work in Progress,draft-ietf-bmwg-ca-bench-meth-04, February 2013. |
|  |  |
| [CoDel] | Nichols, K., Jacobson, V., McGregor, A., and J. Iyengar,"Controlled Delay Active Queue Management", Work inProgress, draft-ietf-aqm-codel-01, Ap |
|  |  |
| [MEF-10.3] | Metro Ethernet Forum, "Ethernet Services AttributesPhase 3", MEF 10.3, October 2013,<https://www.mef.net/Assets/Technical\_Specifications/PDF/MEF\_10.3.pdf>. |
|  |  |
| [MEF-12.2] | Metro Ethernet Forum, "Carrier Ethernet NetworkArchitecture Framework -- Part 2: Ethernet ServicesLayer", MEF 12.2, May 2014,<https://www.mef.net/Assets/Technical\_Specifications/PDF/MEF\_12.2.pdf>. |
|  |  |
| [MEF-14] | Metro Ethernet Forum, "Abstract Test Suite for TrafficManagement Phase 1", MEF 14, November 2005,<https://www.mef.net/Assets/Technical\_Specifications/PDF/MEF\_14.pdf>. |
|  |  |
| [MEF-19] | Metro Ethernet Forum, "Abstract Test Suite for UNIType 1", MEF 19, April 2007, <https://www.mef.net/Assets/Technical\_Specifications/PDF/MEF\_19.pdf>. |
|  |  |
| [MEF-26.1] | Metro Ethernet Forum, "External Network Network Interface(ENNI) - Phase 2", MEF 26.1, January 2012,<http://www.mef.net/Assets/Technical\_Specifications/PDF/MEF\_26.1.pdf>. |
|  |  |
| [MEF-37] | Metro Ethernet Forum, "Abstract Test Suite for ENNI",MEF 37, January 2012, <https://www.mef.net/Assets/Technical\_Specifications/PDF/MEF\_37.pdf>. |
|  |  |
| [PIE] | Pan, R., Natarajan, P., Baker, F., White, G., VerSteeg,B., Prabhu, M., Piglione, C., and V. Subramanian, "PIE: ALightweight Control Scheme To Address the BufferbloatProblem", Work in Progress, draft-ietf-aqm-pie-02,August 2015. |
|  |  |
| [RFC 2697] | Heinanen, J. and R. Guerin, "A Single Rate Three ColorMarker", RFC 2697, DOI 10.17487/RFC2697, September 1999,<http://www.rfc-editor.org/info/rfc2697>. |
|  |  |
| [RFC 2698] | Heinanen, J. and R. Guerin, "A Two Rate Three ColorMarker", RFC 2698, DOI 10.17487/RFC2698, September 1999,<http://www.rfc-editor.org/info/rfc2698>. |
|  |  |
| [RFC 7567] | Baker, F., Ed., and G. Fairhurst, Ed., "IETFRecommendations Regarding Active Queue Management",BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015,<http://www.rfc-editor.org/info/rfc7567>. |

**Приложение А**

**(справочное)**

**Инструменты из открытых источников для проведения испытаний**

**управления передачей данных**

Базовая структура эталонных испытаний устанавливает, что СЛЕДУЕТ выполнять проверки для поведения без сохранения состояния и поведения с сохранением состояния. Некоторыми инструментами, доступными из открытых источников, которые могут быть использованы для выполнения большинства испытаний, предлагаемых данной базовой структурой, являются iperf, netperf (с netperf-wrapper), инструмент uperf, Tmix, TCP-incast-generator, и D-ITG (генератор распределенного интернет трафика).

Iperf может генерировать трафик, базирующийся как на UDP, так и на ТСР, клиент и сервер должны оба быть запущенными с помощью программного обеспечения iperf в одном и том же трафиковом режиме. Сервер устанавливается в режим ожидания, испытательный трафик управляется со стороны клиента. Поддерживается как одностороннее тестирование, так и параллельное двухстороннее тестирование.

Режим UDP может быть использован для испытания трафиком без сохранения состояния. Могут настраиваться целевая пропускная способность канала передачи данных, размер пакета, порт UDP и длительность испытания. Отчет о переданных байтах, потере пакетов и изменении задержки формируется iperf-приемником.

Iperf (режим ТСР), ТСР-incast-generator и D-ITG могут быть использованы для испытания трафиком с сохранением состояния для поверки групповой передачи трафика. Размер ТСР окна (который фактически является SSB), число соединений, размер пакета, порт ТСР и длительность испытания могут регулироваться. Отчет о переданных байтах и достигнутой пропускной способности формируется iperf-передатчиком, в то время как ТСР-incast-generator и D-ITG предоставляют больший объем статистики.

Netperf представляет собой программное приложение, которое обеспечивает проверку пропускной способности сети между двумя хостами на сети. Оно поддерживает UNIX-сокеты домена, ТСР, SCTP и UDP через BSD сокеты. Netperf имеет определенное число предустановленных испытаний, например, измерение характеристик групповой (однонаправленной) передачи данных или с использованием запроса/ответа (http://en.wikipedia.org/wiki/Netperf). Netperf-wrapper представляет собой скрипт на языке Python, который запускает множество одновременных netperf реализаций и объединяет результаты.

Uperf использует описание (или модель) смеси приложений. Данный инструмент генерирует нагрузку в соответствии с описанием модели. Uperf является более гибким, чем netperf в своей способности генерировать поведение приложения типа запрос/ответ внутри одного ТСР-соединения. Описание модели приложения может базироваться на эмпирических данных, однако импорт захваченных пакетов напрямую не поддерживается.

Tmix представляет собой еще один инструмент эмуляции трафика. Он напрямую использует захваченные пакеты для создания профиля трафика. Пакет трассировки является «обратно скомпилированной» описательной характеристикой на уровне исходных кодов, называемой «вектором соединения» каждого ТСР-соединения, представленного в трассировке. В то время, как большинство инструментов широко используются в ns2 моделируемом окружении, Tmix также работает и на Linux-хостах.

Возможности генерации трафика данных инструментов из открытых источников помогают эмулировать ТСР испытательные шаблоны, обсуждаемые в приложении В.

**Приложение В**

**(рекомендуемое)**

**Испытательные шаблоны для ТСР с сохранением состояния**

Базовая структура эталонных испытаний рекомендует, как минимум, следующие ТСР испытательные шаблоны, т.к. они представляют трафик реально существующих приложений (в 5.2.1 описываются некоторые методы создания других, базирующихся на приложениях, ТСР испытательных шаблонов).

– Групповая передача: генерируются одновременные ТСР-соединения, суммарное число байт данных в реальном времени которых, будет заполнять BDP. Для создания данного шаблона ТСР-трафика использовано руководство из [RFC6349].

– Микро-всплеск: генерируется прецизионный шаблон всплеска внутри одного ТСР-соединения или множества ТСР-соединений. Идея заключается в том, что для ТСР устанавливается равновесное состояние, а затем посылается всплеск байт-приложения заданного размера. Испытательное оборудование должно позволять конфигурировать размер всплеска и временной интервал всплеска.

– Шаблоны веб-сайта: модель HTTP-трафика, показанная в таблице 4.1.3-1 [3GPP2-C\_R1002-A] демонстрирует путь разработки таких ТСР испытательных шаблонов. В упрощенном виде, модель НТТР-трафика содержит следующие параметры:

– размер основного объекта (Sm);

– размер встроенного объекта (Se);

– число встроенных объектов на страницу (Nd);

– время обработки клиентом (Tcp);

– время обработки сервером (Tsp).

Шаблоны веб-сайта демонстрируются следующими примерами:

– простой веб-сайт: имитация поведения запрос/ответ и загрузка объекта для базового веб-сайта (маленькая компания);

– сложный веб-сайт: имитация поведения запрос/ответ и загрузка объекта для сложного веб-сайта (сайт для электронной торговли).

Привязываясь к параметрам модели НТТР-трафика, была выведена следующая таблица (посредством анализа и экспериментов) для ТСР испытательных шаблонов простого веб-сайта и сложного веб-сайта:

|  |  |  |
| --- | --- | --- |
| Параметр | Простой веб-сайт | Сложный веб-сайт |
| Размер основного объекта (Sm) | Средний = 10 кБМин. = 100 БМакс. = 500 кБ | Средний = 300 кБМин. = 50 кБМакс. = 2 МБ |
| Размер встроенного объекта (Se) | Средний = 7 кБМин. = 50 БМакс. = 350 кБ | Средний = 10 кБМин. = 100 БМакс. = 1 МБ |
| Число встроенных объектов на страницу (Nd) | Среднее = 5Мин. = 2Макс. = 10 | Среднее = 25Мин. = 10Макс. = 50 |
| Время обработки клиентом (Tcp)\* | Среднее = 3 сМин. = 1 сМакс. = 10 с | Среднее = 10 сМин. = 3 сМакс. = 30 с |
| Время обработки сервером (Tsp)\* | Среднее = 5 сМин. = 1 сМакс. = 15 с | Среднее = 8 сМин. = 2 сМакс. = 30 с |

\* Время обработки клиентом и сервером является распределенным между передачей/приемом всех основных и встроенных объектов.

Параметры в этой таблице являются разумным руководством для генерации ТСР испытательных шаблонов. Испытательное оборудование может использовать фиксированные параметры для упрощенных испытаний и математические распределения для более сложных испытаний. Тем не менее, испытательный шаблон должен быть повторяем для обеспечения того, что результаты эталонных испытаний будут гарантировано сравнимы.

– Интерактивные шаблоны: в то время как шаблоны веб-сайта являются интерактивными по характеру работы, они в основном эмулируют загрузку веб-сайтов различной сложности. Интерактивные шаблоны являются более чато-образными по своей природе, т.к. в них присутствует большое число пользовательских взаимодействий с серверами. К примерам относятся бизнес-приложения, такие как PeopleSoft и Oracle и клиентские приложения, такие как Facebook и IM. Для интерактивных шаблонов была использована техника захвата пакетов для характеризации некоторых бизнес-приложений, а также приложений электронной почты.

В упрощенном виде, интерактивное приложение может быть описано следующими параметрами:

– размер сообщения клиента (Scm);

– число сообщений клиента (Nc);

– размер ответа сервера (Srs);

– число сообщений сервера (Ns);

– время обработки клиентом (Tcp);

– время обработки сервером (Tsp);

– размер закачиваемого файла (Su)\*;

– размер скачиваемого файла (Sd)\*.

\* Параметры размера файла относятся к закачиваемым или скачиваемым вложениям и могут не присутствовать во всех интерактивных приложениях.

Полученная с использованием захвата пакетов и как средство характеризации следующая таблица, представляет руководство для простых бизнес-приложений, сложных бизнес-приложений, eCommerce и передачи/приема электронной почты:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Простоебизнес-приложение | Сложноебизнес-приложение | eCommerce\* | Электронная почта |
| Размер сообщения клиента (Scm) | Средний = 450 БМин. = 100 БМакс. = 1,5 кБ | Средний = 2 кБМин. = 500 БМакс. = 100 кБ | Средний = 1 кБМин. = 100 БМакс. = 50 кБ | Средний = 200 БМин. = 100 БМакс. = 1 кБ |
| Число сообщений клиента (Nc) | Среднее = 10Мин. = 5Макс. = 25 | Среднее = 100Мин. = 50Макс. = 250 | Среднее = 20Мин. = 10Макс. = 100 | Среднее = 10Мин. = 5Макс. = 25 |
| Время обработки клиентом (Tcp)\*\* | Среднее = 10 сМин. = 3 сМакс. = 30 с | Среднее = 30 сМин. = 3 сМакс. = 60 с | Среднее = 15 сМин. = 5 сМакс. = 120 с | Среднее = 5 сМин. = 3 сМакс. = 45 с |
| Размер ответа сервера (Srs) | Средний = 2 кБМин. = 500 БМакс. = 100 кБ | Средний = 5 кБМин. = 1 кБМакс. = 1 МБ | Средний = 8 кБМин. = 100 БМакс. = 50 кБ | Средний = 200 БМин. = 150 БМакс. = 750 кБ |
| Число сообщений сервера (Ns) | Среднее = 50Мин. = 100Макс. = 200 | Среднее = 200Мин. = 25Макс. = 1000 | Среднее = 100Мин. = 15Макс. = 500 | Среднее = 15Мин. = 5Макс. = 40 |
| Время обработки сервером (Tsp)\*\* | Среднее = 0,5 сМин. = 0,1 сМакс. = 5 с | Среднее = 1 сМин. = 0,5 сМакс. = 20 с | Среднее = 2 сМин. = 1 сМакс. = 10 с | Среднее = 4 сМин. = 0,5 сМакс. = 15 с |
| Размер закачиваемого файла (Su) | Средний = 50 кБМин. = 2 кБМакс. = 200 кБ | Средний = 100 кБМин. = 10 кБМакс. = 2 МБ | Средний = N/AМин. = N/AМакс. = N/A | Средний = 100 кБМин. = 20 кБМакс. = 10 МБ |
| Размер скачиваемого файла (Sd) | Средний = 50 кБМин. = 2 кБМакс. = 200 кБ | Средний = 100 кБМин. = 10 кБМакс. = 2 МБ | Средний = N/AМин. = N/AМакс. = N/A | Средний = 100 кБМин. = 20 кБМакс. = 10 МБ |

\* Для eCommerce использована комбинация техник захвата пакетов и рекомендованных индивидуальных потоков трафика, описанных в [SPECweb2009].

\*\* Время обработки клиентом и сервером является распределенным между передачей/приемом всех сообщений. Время обработки клиентом состоит, в основном, из задержек между пользовательскими взаимодействиями (не машинной обработки).

Параметры в данной таблице являются разумным руководством для генерации трафика ТСР испытательных шаблонов. Испытательное оборудование может использовать фиксированные параметры для упрощенных испытаний и математические распределения для более сложных испытаний. Тем не менее, испытательный шаблон должен быть повторяем для обеспечения того, что результаты эталонных испытаний будут гарантировано сравнимы.

– SMB/CIFS копирование файла: имитация сетевого копирования файла, как чтения, так и записи. В противоположность FTP, который обеспечивает групповую передачу и является только потоко-управляемым через ТСР, SMB/CIFS разделяет файл на блоки приложений и использует подтверждение установления связи на уровне приложений в дополнение к ТСР-управлению потоком.

В упрощенном виде SMB/CIFS копирование файла может быть описано следующими параметрами:

– размер сообщения клиента (Scm);

– число сообщений клиента (Nc);

– размер ответа сервера (Srs);

– число сообщений сервера (Ns);

– время обработки клиентом (Tcp);

– время обработки сервером (Tsp);

– размер блока (Sb).

Сообщения клиента и сервера являются сообщениями управления SMB. Размер блока представляет собой порцию данных передаваемого файла.

Полученная с использованием захвата пакетов и как средство характеризации следующая таблица, представляет руководство для SMB/CIFS копирования файла:

|  |  |
| --- | --- |
| Параметр | SMB/CIFS копированиефайла |
| Размер сообщения клиента (Scm) | Средний = 450 БМин. = 100 БМакс. = 1,5 кБ |
| Число сообщений клиента (Nc) | Среднее = 10Мин. = 5Макс. = 25 |
| Время обработки клиентом (Tcp) | Среднее = 1 мсМин. = 0,5 мсМакс. = 2 мс |
| Размер ответа сервера (Srs) | Средний = 2 кБМин. = 500 БМакс. = 100 кБ |
| Число сообщений сервера (Ns) | Среднее = 10Мин. = 10Макс. = 200 |
| Время обработки сервером (Tsp) | Среднее = 1 мсМин. = 0,5 мсМакс. = 2 мс |
| Размер блока (Sb)\* | Средний = N/AМин. = 16 кБМакс. = 128 кБ |

\* В зависимости от испытываемого размера файла, блок определенного размера будет передан «n» количество раз для завершения примера. Например, будем испытывать файл 10 МБ и размер блока будет 64 кБ. В этом случае, будет передано 160 блоков после того, как откроется канал управления между клиентом и сервером.

**Приложение ДА**

**(справочное)**

**Сокращения и обозначения**

В настоящем стандарте кроме приведенных в 2 применены следующие сокращения и обозначения:

FIFO – First In, First Out – первым пришел, первым ушел (способ организации и управления данными относительно времени и приоритетов);

FTP – File Transfer Protocol – протокол передачи файлов;

GPS – Global Positioning System – система глобального позиционирования;

HTTP – Hyper Text Transfer Protocol – протокол передачи гипертекста;

iperf – кроссплатформенная консольная клиент-серверная программа (генератор TCP и UDP трафика для тестирования пропускной способности сети);

LP – Lost Packets – потерянные пакеты;

netperf – программное приложение для тестирования пропускной способности сети между двумя хостами;

NTP – Network Time Protocol – протокол сетевого времени;

OOS – Out-of-Sequence – неупорядоченность;

PD – Packet Delay – задержка пакета;

PDV – Packet Delay Variation – вариация задержки пакеты;

SNMP – Simple Network Management Protocol – простой протокол сетевого управления;

TCP – Transmission Control Protocol – протокол управления передачей;

UDP – User Datagram Protocol – протокол пользовательских дейтаграмм;

URL – Uniform Resource Locator – единый указатель ресурса;

VLAN – Virtual Local Area Network – виртуальная локальная сеть;

VoIP – Voice over IP – голос через IP (общее название коммуникационных протоколов, технологий и методов, обеспечивающих передачу речевого сигнала с использованием IP);

VPLS – Virtual Private LAN Service – сервис частной виртуальной локальной сети.

Директор ОАО «Гипросвязь» А.И.Караим

Начальник НИИЛ ТО НИИЦ ОАО «Гипросвязь» А.И.Воронов

Ведущий инженер

НИИЛ ТО НИИЦ ОАО «Гипросвязь» П.С.Янков

Инженер 1 категории

НИИЛ ТО НИИЦ ОАО «Гипросвязь» О.А.Плякина

1. Известно также как двухсторонняя задержка [↑](#footnote-ref-1)