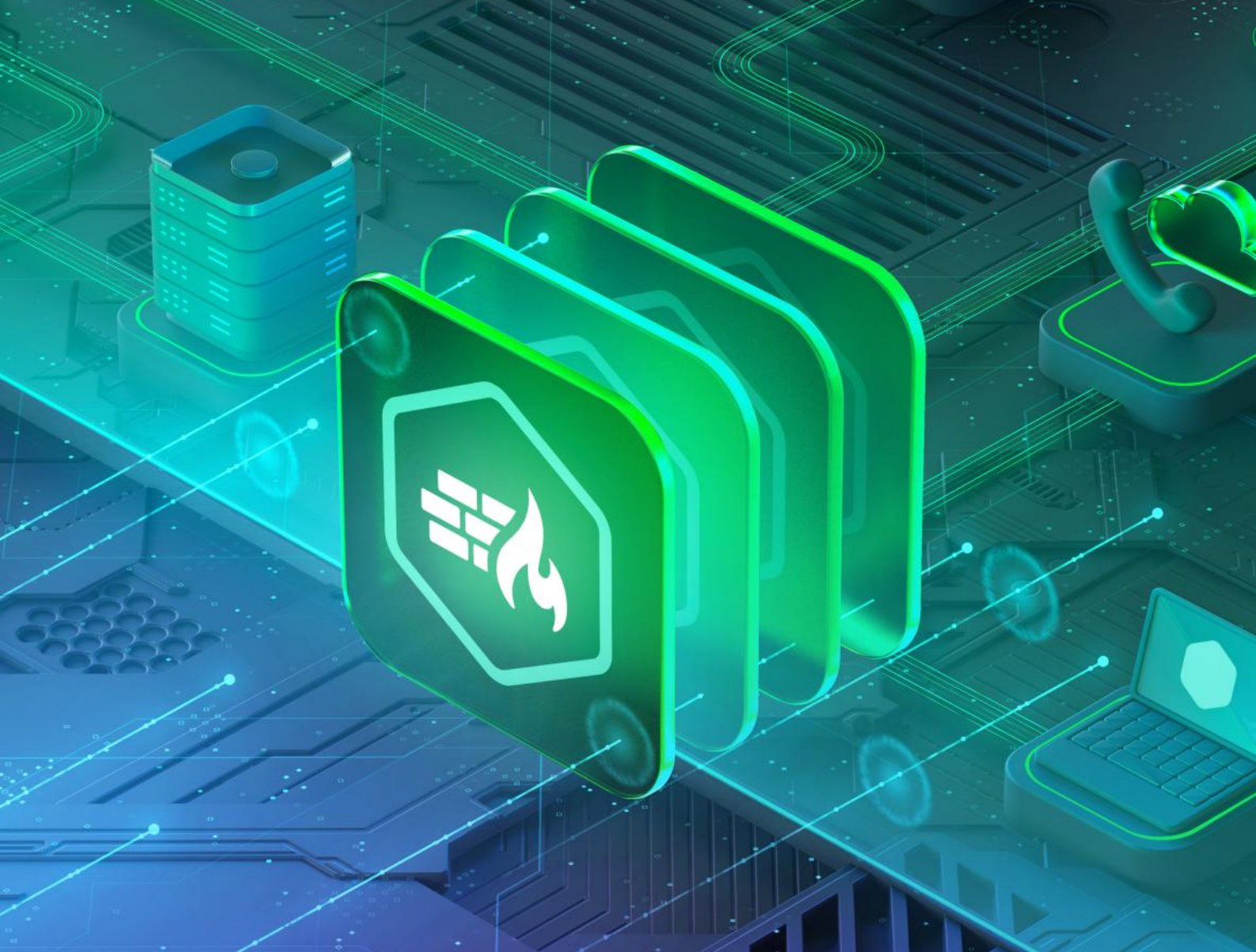




Kaspersky NGFW

Performance Test Guide

03.09.2026

Версия документа:
1.0

Руководство по проведению нагрузочных испытаний
Kaspersky NGFW

kaspersky активируй
будущее

Центр экспертного сопровождения
предпродажной подготовки

RGB

0 168 142	29 29 27	51 194 255	51 92 255	112 51 255	255 51 92	239 237 238	239 255 252
-----------	----------	------------	-----------	------------	-----------	-------------	-------------

Изменения

Дата	Изменения
17.08.2026	Первая версия документа
03.09.2026	Добавлены подразделы «Описание основных счетчиков TRex» и «Отчет о нагрузочном тестировании NGFW»

Содержание

1. Назначение документа	6
1.1. Состав стенда.....	6
1.2. Целевая конфигурация стенда.....	7
2. Быстрый запуск	8
2.1. Развертывание	8
2.2. Проверка готовности.....	9
2.3. Параметры передачи метрик.....	10
2.4. Прогоны для KX-100.....	10
2.5. Прогоны для KX-400	10
2.6. Где смотреть результат	11
2.7. Признаки недостоверного результата.....	12
3. Развертывание хоста TRex	13
3.1. Подготовка операционной системы.....	13
3.2. Установка TRex	13
3.3. Настройка сервиса TRex.....	14
3.4. Проверка работоспособности	16
4. Развертывание средств тестирования.....	17
4.1. Программное обеспечение	17
4.2. Размещение профилей трафика.....	17
4.3. Файлы захвата профиля EMIX.....	18
4.4. Развертывание системы сбора и визуализации метрик.....	18
5. Настройка тестируемого устройства	20
5.1. Общие настройки NGFW	20
5.2. Конфигурация политики безопасности NGFW	20
5.3. Настройка лимитов памяти генератора.....	24
6. Характеристики трафика испытаний.....	25
6.1. Механизм воспроизведения	25
6.2. Профиль EMIX.....	25
6.3. Профиль максимального числа соединений	27
6.4. Профиль соединений в секунду	28
6.5. Сопоставление профилей	29
7. Порядок проведения испытаний.....	30
7.1. Общие условия.....	30
7.2. Логические проверки параметров прогона	33
7.3. Методика определения максимума.....	34
7.4. Описание основных счетчиков TRex.....	37

8. Программа и методика испытаний	41
8.1. Испытание 1. Максимальная производительность на EMIX	41
8.2. Испытание 2. Максимальное количество одновременных соединений	44
8.3. Испытание 3. Максимальное количество соединений в секунду	47
9. Отчет о нагрузочном тестировании NGFW	49
9.1. Общие сведения	49
9.2. Сведения о конфигурации стенда	50
9.3. Результаты испытаний	51
9.4. Заключение	53
9.5. Подписи	53
10. Приложение А. Развертывание генератора на виртуальной машине ..	54
10.1. Ограничения виртуализованного стенда	54
10.2. Отключение аппаратного расчета контрольных сумм	54
10.3. Настройка очередей virtio	55
10.4. Настройка моста ядра Linux	55
10.5. Диагностика прохождения трафика	56
10.6. Размещение виртуальных машин	56

1. Назначение документа

Документ определяет порядок развертывания стенда нагрузочного тестирования межсетевого экрана Kaspersky NGFW с применением генератора трафика TRex, а также программу и методику испытаний.

Документ предназначен для инженеров, выполняющих нагрузочное тестирование в рамках пресейл-активностей, пилотных проектов и приемочных испытаний.

1.1. Состав стенда

Стенд включает следующие компоненты:

- Хост генератора трафика TRex – физический сервер.
- Тестируемое устройство (DUT – Device Under Test) – Kaspersky NGFW.
- Подсистема сбора и визуализации метрик: InfluxDB и Grafana.

Генератор трафика подключается к DUT двумя интерфейсами: порт 0 (Trusted) эмулирует клиентскую сеть, порт 1 (DMZ) – серверную. Трафик проходит через DUT в обоих направлениях. Метрики прогона передаются в InfluxDB и отображаются на дашборде Grafana. Третий интерфейс (MGMT) не находится под управлением TRex и предназначен для управления генератором трафика.

Вариант логической схемы демонстрационного нагрузочного стенда Kaspersky NGFW представлен на рисунке 1. На схеме изображен один узел межсетевого экрана NGFW и один узел сервера для генерации сетевого трафика TRex, соединенные двумя линками, один в зоне LAN и один в зоне DMZ. Test_net_1 и Test_net_2 – логические эмулируемые сети клиентов и серверов TRex. На хосте MGMT-AIO развернута система управления Kaspersky NGFW.

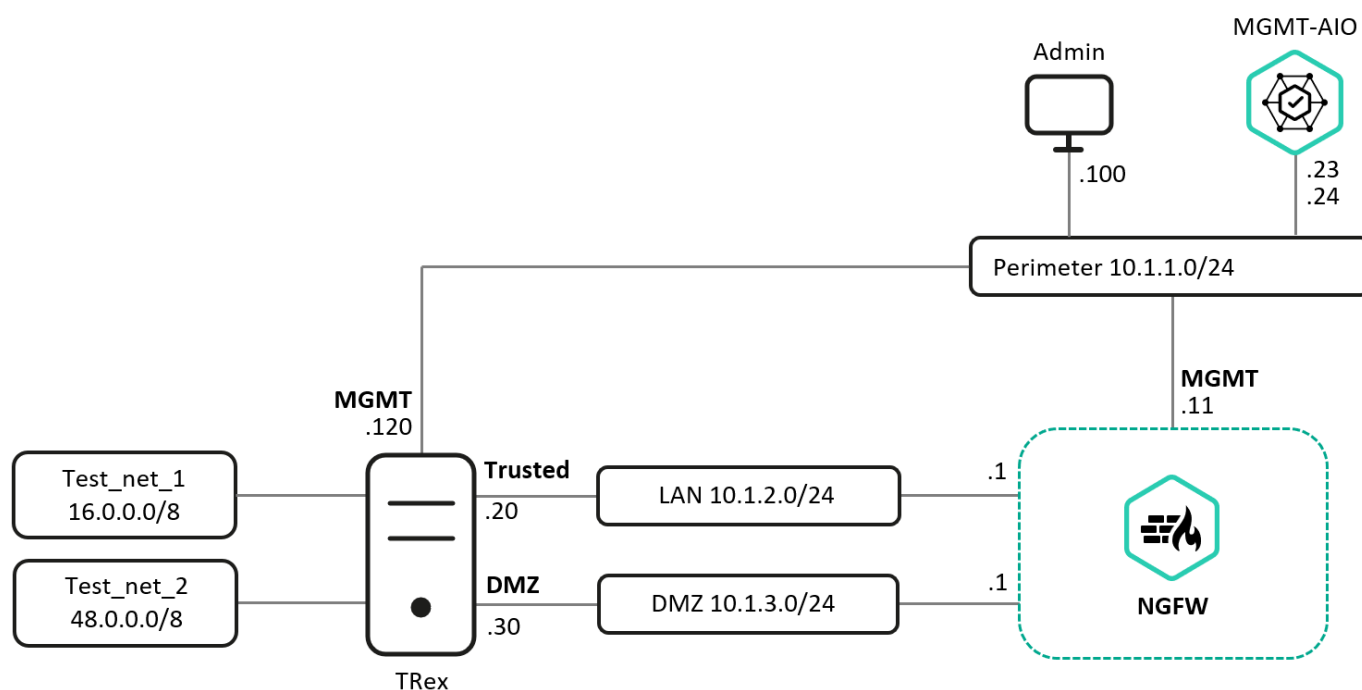


Рисунок 1 – Стенд нагрузочного тестирования

1.2. Целевая конфигурация стенда

Настоящий документ описывает целевую конфигурацию: генератор трафика развернут на физическом сервере, тестируемое устройство – аппаратная платформа NGFW, соединение выполнено физическими линками напрямую либо через коммутатор.

Компонент	Целевая конфигурация
Хост TRex	Физический сервер (bare-metal)
Сетевые карты	Intel E810 QSFP
CPU	Intel Xeon Silver 4314
RAM	256 GB
Подключение к DUT	Физические линки, прямое соединение
Тестируемое устройство	Аппаратная платформа NGFW
Аппаратный расчет контрольных сумм	Включен (по умолчанию)

Note: Развертывание генератора трафика на виртуальной машине допускается для отладки и предварительных проверок, но не является целевой конфигурацией: программный датапас (Datapath) гипервизора существенно ограничивает достижимые показатели. Особенности такого развертывания приведены в приложении А.

2. Быстрый запуск

Раздел содержит краткую последовательность действий для развертывания стенда и проведения испытаний платформ KX-100 и KX-400.

Note: В данном разделе приведены только команды. Настройка, развертывание, обоснование параметров, методика поиска максимума и порядок оформления результатов изложены в разделах 3-9. Значения адресов, паролей и путей приведены как пример и подлежат замене на фактические. При отсутствии опыта работы с TRex рекомендуется предварительно ознакомиться с разделами 3-9.

2.1. Развертывание

Настройка хоста генератора трафика:

```
sudo timedatectl set-timezone Europe/Moscow
sudo apt-get update && sudo apt-get upgrade -y
sudo systemctl stop ufw && sudo systemctl disable ufw
```

Скачать и распаковать архив с генератором трафика:

```
wget https://trex-tgn.cisco.com/trex/release/v3.06.tar.gz
sudo tar -xzf v3.06.tar.gz -C /opt/ && sudo mv /opt/v3.06 /opt/trex
```

Установка генератора и настройка интерфейсов в интерактивном режиме в соответствии с п. 3.2 (конфигурация с IP-адресами без использования MAC-режима):

```
cd /opt/trex/ && sudo ./dpdk_setup_ports.py -i
```

Задать IP-адреса и шлюзы интерфейсов генерации трафика в **/etc/trex_cfg.yaml**.

Настройка больших страниц памяти:

```
sudo sysctl -w vm.nr_hugepages=8192
sudo nano /etc/systemd/system/trex.service
```

Создать файл сервиса TRex. Файл заполнить следующим содержимым, значение параметра **-c** задается согласно п. 3.3.3:

File: /etc/systemd/system/trex.service

```
[Unit]
Description=TREX Service

[Service]
StandardOutput=journal
StandardError=journal
WorkingDirectory=/opt/trex/
ExecStart=/opt/trex/t-rex-64 -i --astf -c 6

[Install]
WantedBy=multi-user.target
```

Запустить сервис trex:

```
sudo systemctl daemon-reload && sudo systemctl enable --now trex.service
```

```
sudo systemctl status trex.service
```

Загрузить архив с профилями, генератором трафика и файлами захвата. Распаковать архивы, создать папки и установить нужные пакеты:

```
mkdir -p /home/ngfw/perf_test && cd /home/ngfw/perf_test
```

```
tar -xzf trex_perf_stend.tar.gz && chmod +x trex_run_perf_influx.py
```

```
sudo apt-get install -y python3-pip && sudo pip3 install influxdb
```

```
sudo mkdir -p /opt/dumps
```

```
sudo tar -xzf dumps.tar.gz -C /opt/dumps/
```

Запустить систему сбора и визуализации метрик:

```
sudo apt-get install -y docker.io docker-compose
```

```
sudo systemctl enable --now docker
```

```
sudo usermod -aG docker $USER && newgrp docker
```

```
mkdir -p grafana && sudo chown -R 472:472 grafana && docker-compose up -d
```

Источник данных и дашборд настраиваются в веб-интерфейсе Grafana согласно п. 4.4.1.

Настроить политику безопасности на тестируемом устройстве. Для подключения скрипта к оркестратору KNBE требуются имя пользователя, пароль и файл корневого сертификата. Сертификат разместить в каталоге генератора. В команде значения PASSWORD и ca.pem заменить на фактические. Правила в политике безопасности NGFW настраиваются согласно п. 5.2.3.

```
cd /home/ngfw/perf_test/policy_generator
```

```
sudo pip3 install -r requirements.txt
```

```
python3 knbe_policy_generator.py --host 10.1.1.23:8090 --user admin \
--password PASSWORD --ca ca.pem --rules 10000
```

Note: Для платформ KX-400 и старше число правил составляет **20000**. Маршруты до тестовых сетей настраиваются согласно п. 5.1.

2.2. Проверка готовности

Проверить состояние сервиса генератора:

```
sudo systemctl status trex.service
```

Проверить состояние портов и разрешение адресов шлюзов:

```
cd /opt/trex/ && ./trex-console
```

```
trex> arp
```

```
trex> stats --ps
```

Проверить доступность базы данных метрик:

```
curl -i http://127.0.0.1:8086/ping
```

Note: Команда **arp** выполняется перед каждым прогоном. При значении **ARP Resolution: unresolved** трафик не передается.

2.3. Параметры передачи метрик

Задать переменные профиля трафика. Набор применяется во всех командах прогона. Значение **--dut-platform** приводится в соответствии с испытываемой платформой:

```
STATS="--send_stats True --influx_addr 127.0.0.1 --influx_port 8086 \
--influx_db trex --influx_interval 5 --latency_pps 1000 \
--ngfw-version 1.3 --ngfw-config pf_dpi --dut-platform KX-100"
```

2.4. Прогоны для KX-100

Максимальная производительность на смешанном трафике:

```
./trex_run_perf_influx.py -s 127.0.0.1 -f emix_fstack_ftp.py \
-m 6.8 -d 600 --rampup 90 --drp 0.001 \
--test-id emix_$(date +%Y%m%d_%H%M) $STATS
```

Максимальное количество одновременных соединений:

```
./trex_run_perf_influx.py -s 127.0.0.1 -f trex_cc_tune_timouts.py \
-m 60000 -d 600 --limit 36000000 --delay 5 \
--rampup 90 --drp 0.001 --test-id cc_$(date +%Y%m%d_%H%M) $STATS
```

Максимальное количество новых соединений в секунду:

```
./trex_run_perf_influx.py -s 127.0.0.1 -f trex_cps.py \
-m 64000 -d 600 -t size=1 \
--rampup 90 --drp 0.001 --test-id cps_$(date +%Y%m%d_%H%M) $STATS
```

2.5. Прогоны для KX-400

Отличаются значениями множителя нагрузки и задержки, остальные параметры совпадают с приведенными для KX-100:

Максимальная производительность на смешанном трафике:

```
./trex_run_perf_influx.py -s 127.0.0.1 -f emix_fstack_ftp.py \
-m 27 -d 600 --rampup 90 --drp 0.001 \
--test-id emix_$(date +%Y%m%d_%H%M) $STATS
```

Максимальное количество одновременных соединений:

```
./trex_run_perf_influx.py -s 127.0.0.1 -f trex_cc_tune_timouts.py \
-m 119048 -d 600 --limit 72000000 --delay 21 \
--rampup 90 --drp 0.001 --test-id cc_$(date +%Y%m%d_%H%M) $STATS
```

Максимальное количество новых соединений в секунду:

```
./trex_run_perf_influx.py -s 127.0.0.1 -f trex_cps.py \
-m 370000 -d 600 -t size=1 \
--rampup 90 --drp 0.001 --test-id cps_$(date +%Y%m%d_%H%M) $STATS
```

Note: Для KX-400 значение **dp_flows** файла **/etc/trex_cfg.yaml** увеличивается до 30 000 000 с соответствующим увеличением объема больших страниц памяти.

Приведенные значения множителя являются начальной точкой поиска. Максимум определяется методом раздела 7.3.

2.6. Где смотреть результат

Показатель	Источник
Пропускная способность, Гбит/с	Grafana, график Total throughput (bit/s), значение Max
Одновременные соединения	Grafana, график Concurrent connections, значение Max
Новых соединений в секунду	Grafana, график Connections/s, значение Max
Доля потерь соединений, %	Вывод скрипта, строка Total TCP/UDP drops percentage during test

Прогон засчитывается при доле потерь не выше 0,001 %.

Дашборд Grafana доступен по адресу <http://<IP-адрес хоста>:3000>

Наблюдение за прогоном в консоли генератора. Открыть консоль и перейти в режим Text User Interface (TUI):

```
cd /opt/trex/ && ./trex-console
```

```
trex> tui
```

Для перехода к статистике испытания нажать **ESC**, затем **t**. Для прокрутки параметров использовать стрелки. Выход из режима TUI: клавиша **q**, прогон при этом не прерывается.

Требуется терминал размером не менее 111×45 символов. По последовательной консоли режим TUI недоступен.

2.7. Признаки недостоверного результата

Результат характеризует тестируемое устройство только при отсутствии ограничения со стороны генератора. Прогон повторяется при любом из следующих признаков:

- Рост показателя **Max Trex TX Queue length, packets** на дашборде.
- Ненулевое значение **queue_full** в консоли генератора.
- Значение **Avg Trex CPU utilization** выше **70 %**.
- Ненулевое значение **oerrors** в выводе команды **stats -p**.

3. Развертывание хоста TRex

3.1. Подготовка операционной системы

Операционная система – Ubuntu Server 22.04. Настройка часового пояса обязательна: метрики передаются с временными метками, и расхождение часовых поясов приводит к отсутствию данных на дашборде.

```
sudo timedatectl set-timezone Europe/Moscow  
  
sudo apt-get update && sudo apt-get upgrade -y  
  
sudo systemctl stop ufw && sudo systemctl disable ufw
```

3.2. Установка TRex

Применяется TRex версии **3.06**. Загрузить дистрибутив с сайта производителя:

```
wget https://trex-tgn.cisco.com/trex/release/v3.06.tar.gz
```

Распаковать архив в каталог **/opt** и переименовать каталог дистрибутива:

```
sudo tar -xzf v3.06.tar.gz -C /opt/  
  
sudo mv /opt/v3.06 /opt/trex
```

Запустить настройку сетевых интерфейсов в интерактивном режиме:

```
cd /opt/trex/  
  
sudo ./dpdk_setup_ports.py -i
```

Скрипт **dpdk_setup_ports.py** выполняется в интерактивном режиме. На вопрос об использовании MAC-режима следует ответить отрицательно: требуется конфигурация с IP-адресами (L3-режим), иначе команды ping и разрешения ARP из консоли TRex работать не будут.

Задаются адреса и шлюзы двух трафиковых интерфейсов, например **10.10.2.20** со шлюзом **10.10.2.1** и **10.10.3.20** со шлюзом **10.10.3.1**. Результат сохраняется в **/etc/trex_cfg.yaml**.

Note: Нумерация портов начинается с нуля. Интерфейс управления в конфигурацию TRex не включается.

3.3. Настройка сервиса TRex

Запуск TRex оформляется в виде системного сервиса, что обеспечивает автоматический старт генератора при загрузке хоста.

Создать файл описания сервиса `/etc/systemd/system/trex.service`:

```
sudo nano /etc/systemd/system/trex.service
```

Файл заполнить следующим содержимым:

File: `/etc/systemd/system/trex.service`

```
[Unit]
Description=TREX Service
[Service]
StandardOutput=journal
StandardError=journal
WorkingDirectory=/opt/trex/
ExecStart=/opt/trex/t-rex-64 -i --astf -c 6
[Install]
WantedBy=multi-user.target
```

Перечитать описание сервисов, включить автозапуск и запустить сервис TRex:

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart trex.service
```

```
sudo systemctl status trex.service
```

3.3.1. Выделение больших страниц памяти.

Note: Linux обычно использует страницы памяти размером 4 КБ. Для высокопроизводительных приложений, таких как TRex, этого недостаточно, поэтому применяются большие страницы памяти (Huge Pages). Они используются DPDK для обеспечения непрерывного доступа к памяти, минимальных задержек, снижения нагрузки на процессор и уменьшения накладных расходов при обработке сетевого трафика.

Зарезервировать большие страницы памяти:

```
sudo sysctl -w vm.nr_hugepages=8192
```

Note: Приведенная команда резервирует 8192 страницы размером 2 МБ каждая, что составляет 16 ГБ оперативной памяти для DPDK-драйвера, с которым работает TRex: $8192 \times 2 \text{ МБ} = 16\,384 \text{ МБ} = 16 \text{ ГБ}$. Обычно для TRex выделяется около 50 % доступной оперативной памяти.

Проверить размер одной большой страницы и фактически выделенный объем:

```
grep Hugepagesize /proc/meminfo
```

```
grep -i huge /proc/meminfo
```

Значение **HugePages_Total** должно соответствовать запрошенному. Настройка, выполненная приведенной командой, действует до перезагрузки.

Для сохранения настройки между перезагрузками зарезервировать страницы на этапе загрузки ядра. Открыть файл параметров загрузчика:

```
sudo nano /etc/default/grub
```

В параметр **GRUB_CMDLINE_LINUX_DEFAULT** добавить:

```
default_hugepagesz=2M hugepagesz=2M hugepages=8192
```

Применить изменения и перезагрузить хост:

```
sudo update-grub && sudo reboot
```

Note: Резервирование на этапе загрузки предпочтительнее: на работающей системе память фрагментирована, и запрошенный объем может быть выделен не полностью.

3.3.2. Распределение процессорных ядер.

Генератор использует три категории потоков: управляющий поток (master), поток измерения задержки (latency) и рабочие потоки обработки трафика (dataplane). Число рабочих потоков задается параметром “-c” в команде запуска сервиса, а закрепление за конкретными ядрами: в разделе **platform** файла **/etc/trex_cfg.yaml**.

Открыть файл конфигурации:

```
sudo nano /etc/trex_cfg.yaml
```

Раздел **platform** заполнить следующим образом:

```
platform:
  master_thread_id: 0
  latency_thread_id: 1
  dual_if:
    - socket: 0
  threads: [2,3,4,5,6,7]
```

Параметр	Назначение
master_thread_id	Ядро управляющего потока: обработка команд консоли, сбор статистики. Трафик не обрабатывает
latency_thread_id	Ядро потока измерения задержки. Задействуется при запуске с параметром --latency_pps
socket	Номер NUMA-ноды, к которой относятся указанные ядра и сетевые карты
threads	Перечень ядер под рабочие потоки. Число ядер в перечне должно быть не меньше значения параметра -c

3.3.3. Выбор числа рабочих потоков.

Увеличение числа рабочих потоков повышает достижимую пакетную скорость, но имеет три ограничения:

- Число аппаратных очередей сетевой карты. Значение **-c** не может превышать число очередей, поддерживаемых картой, при превышении сервис завершается с ошибкой вида **driver maximum tx queues is (N) required (M)**. Карты Intel E810 поддерживают достаточное число очередей и этого ограничения не создают.

- Число физических ядер NUMA-ноды за вычетом ядер, отведенных под управляющий поток и поток измерения задержки. Задействовать логические ядра технологии Hyper-Threading не следует: рабочие потоки DPDK выполняются в режиме непрерывного опроса и конкурируют за исполнительные устройства физического ядра.
- Фактическая потребность. Нарастивать число потоков имеет смысл до тех пор, пока растет достигаемая пакетная скорость. Если при увеличении **-c** показатель **Current PPS** не растет, а загрузка процессора генератора остается ниже **70 %**, ограничение находится не в числе потоков.

Note: Все рабочие потоки и сетевые карты должны находиться в одной NUMA-ноде. Обращение к памяти через межпроцессорную шину существенно снижает производительность генератора. Соответствие сетевой карты NUMA-ноде определяется командой:

```
cat /sys/class/net/<интерфейс>/device/numa_node
```

Порядок подбора: начать с **6** рабочих потоков, выполнить прогон, зафиксировать пакетную скорость и загрузку процессора генератора. При загрузке выше 70 % увеличить число потоков и повторить. Значение фиксируется, когда дальнейшее увеличение прироста не дает.

3.3.4. Число дескрипторов.

Для сетевых карт Intel E810 при высоких нагрузках увеличить число дескрипторов приема и передачи. Добавить в `/etc/trex_cfg.yaml`:

```
rx_desc: 4096
```

```
tx_desc: 4096
```

После изменения конфигурации перезапустить сервис:

```
sudo systemctl restart trex.service
```

3.4. Проверка работоспособности

Запустить консоль TRex и проверить состояние портов, разрешить MAC-адреса шлюзов и проверить связность с NGFW:

```
cd /opt/trex/ && ./trex-console
```

```
trex> stats --ps
```

```
trex> arp
```

```
trex> ping -p 0 -d 10.10.2.1
```

```
trex> ping -p 1 -d 10.10.3.1
```

Команда **stats --ps** отображает состояние портов, MAC-адреса и результат разрешения ARP.

Note: Выполнение команды **arp** обязательно перед проверкой связности и перед каждым прогоном. При значении **ARP Resolution: unresolved** трафик не передается, и команда **ping** завершается по таймауту. Отсутствие ответа на **ping** при успешном разрешении ARP может быть вызвано запретом обработки ICMP на интерфейсах NGFW и не препятствует проведению испытаний: нагрузочным профилям достаточно разрешения ARP.

Консоль TRex в режиме TUI требует терминала размером не менее **111×45** символов. По последовательной консоли режим TUI недоступен.

4. Развертывание средств тестирования

4.1. Программное обеспечение

Создать папку для скриптов, загрузить и распаковать архив с комплектом средств тестирования:

```
mkdir -p /home/ngfw/perf_test && cd /home/ngfw/perf_test
wget <ссылка на архив> -O trex_perf_stend.tar.gz
tar -xzf trex_perf_stend.tar.gz
chmod +x trex_run_perf_influx.py
```

Установить зависимости:

```
sudo apt-get install -y python3-pip
sudo pip3 install influxdb
```

Note: В операционных системах, где действует внешнее управление пакетами Python (Ubuntu 24.04 и новее), к команде добавляется параметр **--break-system-packages**. В Ubuntu 22.04 параметр не поддерживается установленной версией pip и не требуется.

4.2. Размещение профилей трафика

Note: Скрипт запуска определяет путь к клиентской библиотеке TRex автоматически. При необходимости путь задается явно: **export TREX_PATH=/opt/trex**

Профили нагрузки и вспомогательные файлы размещаются в рабочем каталоге:

Файл	Назначение
trex_run_perf_influx.py	Скрипт запуска испытаний
emix_fstack_ftp.py	Профиль смешанного трафика EMIX
trex_cc_tune_timouts.py	Профиль максимального числа соединений
trex_cps.py	Профиль соединений в секунду
http_response_payload.html	Файл полезной нагрузки для trex_cps.py
/opt/dumps/*.pcap	Файлы захвата для профиля EMIX
docker-compose.yml	Файл docker-compose для запуска контейнеров InfluxDB и Grafana.
dashboard_perf_portable.json	Файл с дашбордами для Grafana.

Note: Пакет **trex**, доступный в репозитории PyPI, является сторонним проектом и не относится к генератору трафика. При его наличии в системе импорт библиотеки TRex завершается ошибкой: пакет подлежит удалению: **pip3 uninstall -y trex**

4.3. Файлы захвата профиля EMIX

Профиль EMIX воспроизводит трафик на основе файлов захвата. Комплект подготовленных файлов входит в состав архива средств тестирования и дополнительной обработки не требует.

Разместить файлы захвата в каталоге **/opt/dumps**:

```
sudo mkdir -p /opt/dumps
sudo tar -xzf /home/ngfw/perf_test/dumps.tar.gz -C /opt/dumps/
```

4.4. Развертывание системы сбора и визуализации метрик

Метрики прогона записываются скриптом запуска непосредственно в **InfluxDB**, откуда считываются **Grafana** для отображения на дашборде.

Установить **Docker** и **Docker Compose**:

```
sudo apt-get install -y docker.io docker-compose
sudo systemctl enable --now docker
sudo usermod -aG docker $USER && newgrp docker
```

Создать файл **docker-compose.yml** для запуска **InfluxDB** и **Grafana** со следующим содержимым:

File: /home/ngfw/perf_test/docker-compose.yml

```
services:
  influxdb:
    image: influxdb:1.8
    container_name: influxdb
    restart: unless-stopped
    ports:
      - "8086:8086"
    environment:
      - INFLUXDB_DB=trex
      - INFLUXDB_USER=trex
      - INFLUXDB_USER_PASSWORD=trex
      - INFLUXDB_HTTP_AUTH_ENABLED=false
    volumes:
      - influx_data:/var/lib/influxdb

  grafana:
    image: grafana/grafana:10.4.1
    container_name: grafana
    restart: unless-stopped
    ports:
      - "3000:3000"
    volumes:
      - ./grafana:/var/lib/grafana
    depends_on:
      - influxdb

volumes:
  influx_data:
```

Note: Требуется InfluxDB версии 1.x: дашборд построен на языке запросов InfluxQL. Версия Grafana 10.4.1 соответствует версии, для которой подготовлен дашборд.

Создать каталог для данных Grafana и назначить владельца, под которым работает процесс в контейнере:

```
mkdir -p grafana && sudo chown -R 472:472 grafana
```

Запустить контейнеры **InfluxDB** и **Grafana**:

```
docker-compose up -d
```

Проверить, что оба контейнера запущены:

```
docker ps
```

4.4.1. Настройка Grafana.

Открыть веб-интерфейс Grafana по адресу <http://<IP-адрес хоста>:3000>. Учетные данные по умолчанию: **admin** / **admin**, при первом входе предлагается сменить пароль.

Добавить источник данных: в левом меню выбрать **Connections** → **Data sources** → **Add new data source** → **InfluxDB**. Заполнить поля:

- **Query language** (Язык запросов): **InfluxQL**
- **URL** (Адрес): **http://influxdb:8086**
- **Database** (База данных): **trex**
- **User** (Пользователь): **trex**
- **Password** (Пароль): **trex**

Нажать **Save & test**. При успешной проверке выводится сообщение о доступности источника данных.

Note: Адрес **http://influxdb:8086** указывается по имени контейнера, поскольку Grafana обращается к InfluxDB внутри сети Docker. Адрес localhost в этом поле работать не будет.

Импортировать дашборд: в левом меню выбрать **Dashboards** → **New** → **Import**. Нажать **Upload dashboard JSON file** и указать файл **dashboard_perf_portable.json** из комплекта. В поле выбора источника данных указать созданный ранее источник **InfluxDB** и нажать **Import**.

Note: Две панели раздела **Trex health: Memory** и **Memory Distribution** используют источник данных **Prometheus** и отображают объем памяти хоста через **node_exporter**. При отсутствии **Prometheus** панели остаются пустыми, на метрики испытаний это не влияет.

После импорта в верхней части дашборда становятся доступны фильтры **NGFW Hardware Platform**, **NGFW Config**, **Test ID** и **Version**. Списки заполняются значениями из метрик после первого прогона.

5. Настройка тестируемого устройства

5.1. Общие настройки NGFW

На NGFW должны быть настроены интерфейсы, подключенные к генератору трафика: должны быть заданы IP-адреса и назначены зоны по умолчанию **Trusted** для интерфейса, подключенного к подсети **10.10.2.0** и **DMZ** для интерфейса, подключенного к подсети **10.10.3.0**.

NGFW должен быть подключен к KSC и KNBE, В плагине для устройства должен быть статус **Registered**.

Должна быть настроена отправка событий безопасности в SIEM KUMA.

Также необходимо настроить на NGFW маршруты до тестовых сетей, указывающие на адреса интерфейсов генератора трафика, и сохранить конфигурацию. Пример:

```
ip route 16.0.0.0 netmask 255.0.0.0 vrf Main gateway 10.10.2.20
```

```
ip route 48.0.0.0 netmask 255.0.0.0 vrf Main gateway 10.10.3.20
```

Сеть **16.0.0.0/8** соответствует клиентской стороне, а **48.0.0.0/8** серверной. Адреса шлюзов приводятся в соответствии с адресацией стенда. Настраивается разрешающее правило между зонами для тестового трафика.

5.2. Конфигурация политики безопасности NGFW

Конфигурация должна оставаться неизменной между прогонами и релизами. Число правил фильтрации определяется платформой.

Платформа	Число правил фильтрации
vKX-8	10 000
KX-100	10 000
KX-400	20 000
KX-1000	20 000
KX-3500	20 000

Общие требования к политике:

- Правила содержат квалификаторы уровня приложений (L7).
- Логирование окончания сессии на UDP-порт включено.
- NAT отключен.
- При использовании кластера это указывается в наименовании или описании испытания.

Note: Логирование окончания сессий и обработка правил с квалификаторами L7 создают дополнительную нагрузку на NGFW. Результаты испытаний, проведенных при иной конфигурации, с приведенными несопоставимы.

5.2.1. Формирование политики фильтрации.

Требуемое число правил фильтрации создается автоматически при помощи генератора политики, входящего в комплект средств тестирования. Генератор обращается к серверу управления по программному интерфейсу, создает сетевые объекты, сервисы и политику заданного объема.

Политика формируется из двух частей:

- Фоновые правила с действием «запретить», обеспечивающие требуемую нагрузку на подсистему фильтрации.
- Четыре итоговых разрешающих правила, пропускающих тестовый трафик генератора TRex.

Note: Фоновые правила построены на адресах 192.168.0.0/16 и с тестовыми сетями не пересекаются. Итоговые разрешающие правила размещаются в конце политики, ниже всех фоновых, вследствие чего тестовый трафик обрабатывается именно ими.

Итоговые правила создаются для двух пар тестовых сетей, соответствующих двум парам портов генератора трафика. Для каждой пары формируется прямое и обратное правило.

Наименование правила	Сеть источника	Сеть назначения	Зона источника	Зона назначения
first_forward_final_app_protocol_match	16.0.0.0/8	48.0.0.0/8	Trusted	DMZ
first_reverse_final_app_protocol_match	48.0.0.0/8	16.0.0.0/8	DMZ	Trusted
second_forward_final_app_protocol_match	17.0.0.0/8	49.0.0.0/8	Trusted	DMZ
second_reverse_final_app_protocol_match	49.0.0.0/8	17.0.0.0/8	DMZ	Trusted

Параметры итоговых правил:

- Действие: «разрешить» (Allow).
- Логирование окончания сессии включено.
- Сервис: allow-any-any-tcp-udp (протоколы TCP и UDP, порты с 0 по 65535).
- Зоны источника и назначения: Trusted и DMZ, в обратном правиле зоны меняются местами вместе с сетями.
- Прикладные протоколы: перечислены все поддерживаемые модулем DPI.

Note: На интерфейсы необходимо назначать зоны **Trusted** и **DMZ**, созданные по умолчанию. Идентификаторы зон запрашиваются скриптом у KNBE по их наименованиям, при отсутствии любой из зон генератор завершает работу сообщением о том, какие зоны доступны.

5.2.2. Запуск скрипта для генерации политики безопасности.

Установить зависимости генератора:

```
cd /home/ngfw/perf_test/policy_generator
sudo pip3 install -r requirements.txt
```

Note: В операционных системах, где действует внешнее управление пакетами Python (Ubuntu 24.04 и новее), к команде добавляется параметр --break-system-packages. В Ubuntu 22.04 параметр не поддерживается установленной версией pip и не требуется.

Файл корневого сертификата, используемого KNBE, разместить на хосте генератора трафика в папку **policy_generator**.

Генератор запускается одной командой, параметры передаются в командной строке. Значения, выделенные цветом, приводятся в соответствии с параметрами стенда:

Параметр	По умолчанию	Назначение
--host	–	Адрес и порт оркестратора KNBE
--user	–	Имя пользователя
--password	–	Пароль
--ca	–	Путь к файлу корневого сертификата
--rules	–	Общее число правил, включая четыре итоговых
--policy-name	perf_test_policy	Имя создаваемой политики
--threads	1	Число одновременных запросов при загрузке
--keep-existing	не задан	Не удалять объекты предыдущих запусков
--debug	не задан	Подробный протокол работы

Note: Значение параметра **--rules** задает полное число правил в политике. Четыре итоговых разрешающих правила входят в это число, остальные распределяются поровну между тремя типами фоновых правил: с квалификаторами уровня L4, с квалификаторами уровня L7 и со смешанными. Минимальное допустимое значение – 4.

Число правил задается в соответствии с испытываемой платформой согласно таблице п. 5.2.

Для vKX-8, KX-100:

```
python3 knbe_policy_generator.py \
--host 10.1.1.23:8090 \
--user admin \
--password PASSWORD \
--ca ca.pem \
--rules 10000
```

Для KX-400, KX-1000, KX-3500:

```
python3 knbe_policy_generator.py \
--host 10.1.1.23:8090 \
--user admin \
--password PASSWORD \
--ca ca.pem \
--rules 20000
```

По завершении работы генератор выводит число загруженных правил и время загрузки. Созданную политику следует проверить в веб-интерфейсе сервера управления:

- Имя политики соответствует заданному параметром **--policy-name**.
- Общее число правил соответствует заданному параметром **--rules**.
- Четыре правила с наименованиями, оканчивающимися на **final_app_protocol_match**, расположены в конце списка правил.
- Действие этих правил – «разрешить» (Allow).

После проверки политика назначается тестируемому устройству в системе управления, и выполняется контроль прохождения тестового трафика согласно п. 7.1.

5.2.3. Конфигурации, подлежащие испытанию.

На каждой аппаратной платформе испытанию подлежат следующие конфигурации NGFW:

Конфигурация	Состав	Сценарии
Packet Filter + DPI	L4 FW и Application Control, действие permit	CPS, CC, EMIX
IDPS	Сигнатурный движок IDPS	EMIX
ATP1	DPI, IDPS, Web Control, DNS Security, потоковый антивирус	EMIX
ATP2	DPI, IDPS, Web Control, DNS Security, URL AV, объектный антивирус	EMIX

Для изменения конфигурации в правилах **final_app_protocol_match** назначается действие типа **Inspect** и выбирается групповой профиль безопасности с необходимыми включенными движками безопасности.

В конфигурации **ATP1** в профиле антивируса необходимо включить потоковый антивирус и выключить проверку почтовых протоколов **SMTP**, **POP3**, **IMAP**.

В конфигурации **ATP2** в профиле антивируса необходимо включить объектный антивирус и проверку почтовых протоколов. Для его работы требуется выделение ядер CPU для выполнения асинхронных задач. Количество ядер CPU, выделяемых для асинхронных операций, приведено в п. 8.1.4.

Пример выделения дополнительных ядер для работы объектного антивируса:

```
system linux-cores
```

```
system linux-cores> cores 4
```

```
exit
```

```
commit
```

```
config save
```

```
system reboot
```

5.3. Настройка лимитов памяти генератора

При настройках по умолчанию генератор создает не более 2 000 000 одновременных соединений. Для проведения испытания на максимальное число соединений этот предел увеличивается.

Открыть файл конфигурации:

```
sudo nano /etc/trex_cfg.yaml
```

Добавить раздел **memory**, задав требуемое число потоков:

```
memory :
    dp_flows      : 4000000
    active_flows  : 4000000
    dp_max_flows  : 4000000
```

Значение подбирается исходя из целевого числа одновременных соединений и объема выделенных больших страниц памяти. Ориентировочный расход составляет от 4 до 8 КБ на поток. Часть больших страниц расходуется на пулы пакетных буферов, объем которых зависит от параметров **rx_desc**, **tx_desc** и числа рабочих потоков.

Большие страницы памяти	dp_flows (ориентировочно)	Достижимое число соединений
16 ГБ (8192 × 2 МБ)	4 000 000	до 3,5 млн
32 ГБ (16 384 × 2 МБ)	8 000 000	до 7 млн
64 ГБ	16 000 000	до 14 млн
160 ГБ	40 000 000	до 35 млн

Порядок определения значения:

- Определить целевое число одновременных соединений для испытания.
- Задать **dp_flows** с запасом от 15 до 20 % от целевого значения.
- Выделить большие страницы памяти из расчета от 4 до 8 КБ на поток, но не более половины объема оперативной памяти хоста.
- Перезапустить сервис и убедиться в успешном запуске.
- Проверить фактическое потребление по значению **HugePages_Total**.

Применить изменения:

```
sudo systemctl restart trex.service
```

Note: При недостаточном объеме больших страниц памяти генератор завершается аварийно на этапе выделения памяти, а при загрузке профиля наблюдается длительное ожидание. Целевое число соединений не должно превышать значение **dp_flows**.

6. Характеристики трафика испытаний

6.1. Механизм воспроизведения

Генератор в режиме Advanced Stateful Mode (ASTF) не воспроизводит захваченные пакеты дословно. Из файла захвата извлекается прикладной поток – последовательность и размеры сообщений уровня приложений, который передается через собственную реализацию стека TCP/IP генератора. Размеры кадров определяются параметрами стека TRex, а не размерами кадров в захвате.

Достоверными характеристиками профиля являются количество и порядок прикладных сообщений, объем прикладных данных на сессию, транспортный протокол и номер порта. Фактический средний размер кадра контролируется в ходе прогона по счетчику **m_avg_size**.

Note: Часть файлов захвата снята при включенной аппаратной сборке сегментов, вследствие чего в них присутствуют кадры размером до 65 549 байт, физически не передававшиеся. Распределение размеров кадров в файлах захвата носит справочный характер.

6.2. Профиль EMIX

Профиль **emix_fstack_ftp.py** включает 13 приложений. Частота создания сессий каждого приложения задана в профиле и масштабируется множителем нагрузки линейно.

Испытание на смешанном трафике определяет пропускную способность межсетевого экрана в условиях, приближенных к реальной корпоративной сети. Одновременно передается трафик тринадцати приложений различных типов: шифрованный, веб, почтовый, файловый, служебный и терминальный. Нагрузка создается по всем подсистемам одновременно: разбору протоколов, инспекции содержимого, обработке таблицы сессий.

Результат испытания характеризует производительность устройства при работе с трафиком реального состава и является основным показателем для оценки применимости платформы в заданном сегменте сети.

Приложение	L4	Порт	Сессий/с при -m 1	Прикладных байт на сессию	Доля сессий, %	Доля объема, %
TLS 1.2 (HTTPS через прокси)	TCP	8080	80,84	798 566	28,5	38,2
TLS 1.3	TCP	443	80,36	775 270	28,3	36,9
HTTP GET	TCP	80	16,61	911 795	5,9	9,0
SMTP	TCP	25	10,25	1 026 176	3,6	6,2
HTTP POST	TCP	8081	4,89	1 278 864	1,7	3,7
DNS	UDP	53	58,64	49 896	20,7	1,7
SMB	TCP	445	2,27	938 216	0,8	1,3
SNMP	UDP	161	2,40	790 653	0,8	1,1
FTP data (download)	TCP	20	0,56	3 000 000	0,2	1,0

Приложение	L4	Порт	Сессий/с при -m 1	Прикладных байт на сессию	Доля сессий, %	Доля объема, %
RDP	TCP	3389	14,79	85 535	5,2	0,7
POP3	TCP	110	5,40	19 789	1,9	0,1
SSH	TCP	22	5,40	18 498	1,9	0,1
FTP control	TCP	21	1,11	723	0,0	0,0
Итого			283,5		100	100

Каждый файл захвата содержит одну сессию.

6.2.1. Соотношение протоколов.

Протокол	Доля сессий, %	Доля прикладного объема, %
TCP	78,5	97,2
UDP	21,5	2,8

Прикладной объем формируется преимущественно шифрованным трафиком: на TLS 1.2 и TLS 1.3 суммарно приходится 75,1 % данных при 56,8 % сессий. Протокол DNS формирует пятую часть всех сессий при доле объема 1,7 % и нагружает NGFW по числу обрабатываемых потоков, а не по полосе.

6.2.2. Расчетная нагрузка.

Пропускная способность рассчитывается на уровне L2.

Множитель -m	Сессий/с	L7, Гбит/с	L2, Гбит/с	Пакетов, Mpps
1	284	1,35	1,46	0,26
2	567	2,70	2,93	0,52
4	1 134	5,40	5,86	1,04
6	1 701	8,11	8,79	1,56
10	2 835	13,51	14,64	2,59

6.3. Профиль максимального числа соединений

Профиль **trex_cc_tune_timouts.py** формирует синтетический трафик HTTP без использования файлов захвата.

Испытание на максимальное количество одновременных соединений определяет емкость таблицы сессий межсетевого экрана. Соединения устанавливаются и удерживаются открытыми, объем передаваемых данных при этом минимален.

Нагрузка создается на подсистему учета состояний: каждое соединение занимает запись в таблице сессий и потребляет память устройства. Пропускная способность и пакетная скорость в этом испытании второстепенны.

Результат характеризует предельное число одновременно обслуживаемых пользовательских сеансов и применяется при расчете размера сети, которую способна обслуживать платформа.

Характеристика	Значение
Протокол	TCP, HTTP/1.1, Connection: Keep-Alive
Транзакций на сессию	10 (параметр pipeline)
Запрос	GET /index.html, 247 байт
Ответ	HTTP 200 с телом размером size, КБ
Задержка ответа сервера	delay, секунд перед каждой транзакцией
Клиентские адреса	16.0.0.0 – 16.40.255.255 (около 2,6 млн)
Серверные адреса	48.0.41.0 – 48.81.255.255 (около 5,3 млн)
Размер буферов TCP	65 535 байт

Size, КБ	Ответ, байт	На транзакцию, байт	На сессию, байт
1	1 117	1 364	13 640
2	2 141	2 388	23 880
4	4 189	4 436	44 360

Профиль предназначен для накопления сессий, а не для создания полосы: при задержке 21 секунда сессия существует около 210 секунд, передавая 13,6 КБ прикладных данных, что соответствует примерно 65 байт/с на сессию.

6.4. Профиль соединений в секунду

Профиль **trex_cps.py** формирует синтетический трафик HTTP: установление соединения, передача запроса, получение ответа, закрытие соединения.

Испытание на количество новых соединений в секунду определяет скорость обработки установки и завершения сеансов. Каждое соединение выполняет одну короткую транзакцию и закрывается.

Нагрузка создается на подсистему обработки новых сессий: для каждого соединения выполняется поиск по правилам фильтрации, создание записи в таблице сессий и ее последующее удаление. Число одновременных соединений при этом остается небольшим.

Результат характеризует поведение устройства при массовом обращении пользователей и устойчивость к нагрузкам, при которых преобладает установление новых сеансов, а не передача данных.

Характеристика	Значение
Протокол	TCP, HTTP/1.1
Транзакций на сессию	1
Запрос	GET /3384, 241 байт
Ответ	HTTP 200 с телом размером size, КБ
Клиентские адреса	16.0.0.0 – 16.0.255.255 (65 536 адресов)
Серверные адреса	48.0.0.0 – 48.0.255.255 (около 65 тыс.)
Распределение адресов	последовательное

size, КБ	Ответ, байт	На сессию, байт
1	1 117	1 358
2	2 141	2 382
4	4 189	4 430

Note: Каждое TCP-соединение однозначно определяется набором из четырех значений: адрес и порт источника, адрес и порт назначения. Такой набор называется кортежем. Пока соединение существует, его кортеж занят и повторно использован быть не может; после закрытия соединения кортеж остается недоступным еще некоторое время. Диапазон адресов профиля обеспечивает порядка 4,2 млрд кортежей, что достаточно для всех испытываемых платформ.

6.5. Сопоставление профилей

Характеристика	EMIX	Максимум соединений	Соединений в секунду
Источник данных	13 файлов захвата	синтетический	синтетический
Протоколы	TCP и UDP, 13 приложений	TCP, HTTP	TCP, HTTP
Транзакций на сессию	по файлу захвата	10	1
Прикладных байт на сессию	723 – 3 000 000	13 640	1 358
Основная нагрузка на NGFW	инспекция разнородного трафика, полоса	емкость таблицы сессий	скорость создания и удаления сессий
Основной параметр настройки	-m	delay, затем -m	-m, size

7. Порядок проведения испытаний

7.1. Общие условия

Стенд развернут в соответствии с разделами 3-4. Дашборд импортирован, источник данных настроен. Длительность каждого зачетного прогона составляет 600 секунд, включая фазу набора нагрузки.

7.1.1. Предварительная проверка стенда.

До проведения испытаний NGFW определяются предельные характеристики самого стенда без тестируемого устройства: порты генератора замыкаются напрямую либо через коммутатор, после чего выполняются три прогона в п. 8.1-8.3. Полученные значения являются предельными для стенда.

Note: Проблемы производительности могут возникать на стороне генератора трафика, сетевых карт, шины или коммутационного оборудования. Результат испытания NGFW достоверен только при условии, что он с запасом ниже предельных характеристик стенда. При развертывании генератора на виртуальной машине предварительная проверка обязательна: см. приложение А.

Перед испытанием и между итерациями следует убедиться в отсутствии на NGFW открытых сессий от предыдущих прогонов.

7.1.2. Эталонные конфигурации нагрузочных стендов.

В таблице приведены конфигурации нагрузочных стендов, применяемые для получения эталонных показателей, и достигаемые на них предельные значения. Сетевые карты – Intel E810 QSFP.

Процессор	NUMA	ОЗУ	Полоса	CPS	Соединений
Intel Xeon Silver 4314	2	256 ГБ	около 200 Гбит/с	1,5 млн	58 млн
Intel Xeon Silver 4316	1	128 ГБ	около 100 Гбит/с	750 тыс.	29 млн
Intel Xeon Gold 6240R	2	360 ГБ	около 80 Гбит/с	750 тыс.	29 млн

Приведенные значения являются предельными для генератора трафика и не относятся к тестируемому устройству. Они используются для двух целей: оценки пригодности имеющегося стенда для планируемого испытания и интерпретации результата.

Если предельные показатели стенда сопоставимы с ожидаемыми показателями NGFW, результат испытания будет характеризовать связку стенда и тестируемого устройства, а не устройство. Для получения достоверного результата предельные показатели стенда должны превышать ожидаемые показатели NGFW с запасом не менее 30 процентов.

Фактические предельные показатели применяемого стенда определяются предварительной проверкой согласно п. 7.1.1 и вносятся в протокол испытаний.

7.1.3. Подготовка к прогону.

Перезапустить сервис TRex для сброса счетчиков, открыть консоль, разрешить MAC-адреса шлюзов и запустить режим TUI для наблюдения за прогоном:

```
sudo systemctl restart trex.service
cd /opt/trex && ./trex-console
trex> arp
trex> stats --ps
trex> clear
trex> tui
```

Для просмотра статистики в TUI нажать **ESC** → **t**. Для прокрутки параметров использовать стрелки.

Запуск испытания выполняется из второй сессии подключения.

Note: Досрочная остановка прогона сочетанием клавиш Ctrl+C недопустима: генератор принудительно обрывает активные сессии, они учитываются в счетчике **tcps_testdrops**, вследствие чего результаты измерения потерь становятся непригодными. Для управляемой досрочной остановки применяется параметр **--early-drops-threshold**.

7.1.4. Параметры передачи метрик.

Приведенный набор параметров применяется во всех испытаниях. Значения, выделенные цветом, являются примером и приводятся в соответствии с параметрами конкретного испытания:

```
STATS="--send_stats True --influx_addr 127.0.0.1 --influx_port 8086 \
--influx_db trex --influx_interval 5 --latency_pps 1000 \
--ngfw-version 1.3 --ngfw-config pf_dpi --dut-platform KX-100"
```

Скрипт запуска формирует набор tunables профиля самостоятельно и перезаписывает в нем значения ключей **rampup**, **limit** и **delay** значениями одноименных параметров командной строки **--rampup**, **--limit** и **--delay**. Дублировать эти ключи через параметр **-t** не требуется: заданные таким образом значения будут перезаписаны.

Параметр **--rampup** применяется всегда, в том числе при его отсутствии в команде запуска: значение по умолчанию равно 90 секундам. Фаза набора нагрузки продолжительностью 90 секунд выполняется во всех испытаниях настоящего документа.

Ключи **client_network** и **server_network** скрипт запуска также задает самостоятельно, исходя из адресации тестовых сетей. Задавать их вручную не следует.

Через параметр **-t** передаются только ключи, не формируемые скриптом запуска. В настоящем документе это ключ **size** профиля **trex_cps.py**, задающий размер прикладной транзакции.

Значение **--ngfw-config** обозначает испытываемую конфигурацию NGFW согласно п. 5.2.3 и попадает в метрики как тег, по которому результаты разделяются на дашборде и в отчете. Применяются обозначения: **pf_dpi**, **idps**, **atp1**, **atp2**. Значение задается без пробелов и символа «плюс»: оно подставляется в регулярное выражение фильтра Grafana.

Значение **--dut-platform** указывает испытываемую платформу и изменяется при переходе к следующей платформе: vKX-8, KX-100, KX-400, KX-1000, KX-3500.

Параметр	По умолчанию	Назначение
--send_stats	True	Передача метрик
--influx_addr	127.0.0.1	Адрес InfluxDB
--influx_port	8086	Порт InfluxDB
--influx_db	telegraf	Имя базы данных
--influx_interval	5	Период передачи метрик, с
--latency_pps	0	Частота зондирования задержки
--test-id	test_<метка времени>	Идентификатор прогона
--dut-platform	unknown	Платформа NGFW
--ngfw-config	PF + DPI	Конфигурация NGFW
--ngfw-version	dev	Версия NGFW

7.1.5. Фаза набора нагрузки.

В начале прогона генератор наращивает частоту создания сессий линейно от нуля до целевого значения. Продолжительность нарастания задается параметром **--rampup** и по умолчанию составляет 90 секунд.

Параметр выполняет две независимые функции. В профиле трафика он задает продолжительность нарастания через параметр планировщика scheduler.rampup_sec. В расчете результата он определяет период, потери и потоки которого исключаются из вердикта.

Скрипт запуска передает значение в профиль самостоятельно, поэтому параметр применяется всегда, в том числе при его отсутствии в команде запуска. Оба назначения используют одно и то же значение, рассогласование исключено.

Нарастание нагрузки предотвращает выброс потерь в первые секунды прогона: при мгновенном выходе на целевую нагрузку таблица сессий тестируемого устройства заполняется скачком, что приводит к отказам, не отражающим его установившиеся характеристики.

Увеличение значения замедляет нарастание. Продолжительность нарастания увеличивается при наблюдении в начале прогона выбросов потерь, роста очереди передачи генератора или ошибок выделения памяти. Увеличение требует соответствующего увеличения общей длительности прогона согласно п. 7.2.1.

Note: Продолжительность фазы набора нагрузки не влияет на потери в установившейся фазе. Если потери фиксируются на всем протяжении прогона, их причина не связана с темпом нарастания, и увеличение параметра результата не даст.

Частота создания сессий в установившейся фазе определяется профилем трафика и множителем нагрузки. Для профиля EMIX частота создания сессий и пропускная способность связаны жестко: множитель масштабирует оба показателя пропорционально, снизить один из них независимо от другого параметрами запуска невозможно.

Note: Параметр --latency_pps по умолчанию равен нулю, при этом панели раздела Latency остаются пустыми. Для их наполнения параметр указывается явно.

Значение `--ngfw-config` задается без пробелов и символа «плюс»: значение подставляется в регулярное выражение фильтра Grafana, при котором значение по умолчанию совпадения не дает.

7.2. Логические проверки параметров прогона

Проверки выполняются до запуска. Нарушение любого из условий делает результат недостоверным независимо от поведения NGFW.

7.2.1. Длительность прогона.

Параметр длительности задает полное время прогона, включая фазу набора нагрузки. Требование о длительности активной фазы не менее 300 секунд относится к периоду после ее окончания.

Для профилей EMIX и соединений в секунду:

-d >= rampup + 300

Для профиля максимального числа соединений дополнительно учитывается время накопления сессий:

-d >= rampup + delay * 10 + 300

rampup, с	delay, с	Минимальное -d, с	При -d 600
90	3	420	достаточно
90	5	440	достаточно
90	10	490	достаточно
90	21	600	на границе
90	30	690	недостаточно

7.2.2. Предел общего числа соединений.

Параметр **limit** ограничивает суммарное число клиентских соединений за прогон. По его исчерпанию создание новых соединений прекращается, темп установления снижается до нуля, а показанное число одновременных соединений перестает отражать установившийся режим.

limit >= CPS * длительность прогона

Note: Признак исчерпания лимита: значение **Total connections** в точности равно заданному значению **limit** при существенно сниженном значении Current CPS. В этом случае показатель Current CC отражает лишь число еще не закрытых соединений и не является результатом испытания.

Пример: при 100 000 соединений в секунду и длительности 600 секунд требуется значение limit не менее 60 000 000.

7.2.3. Прочие проверки.

Условие	Следствие нарушения
Целевое CC <= dp_flows	Зависание генератора на выделении памяти, аварийное завершение
keep_timeout > delay * 1000	Обрыв соединений по таймауту генератора, потери относятся на NGFW

7.3. Методика определения максимума

Максимум определяется методом деления отрезка по множителю нагрузки:

- Выполнить прогон с заведомо проходимым значением множителя.
- Зафиксировать долю потерь по метрике Connection drops.
- При доле потерь не выше порога увеличить множитель и повторить.
- При превышении порога принять значение верхней границей и продолжить деление интервала.
- Максимумом считать наибольшее значение, выдержанное 600 секунд без превышения порога.

7.3.1. Досрочное прерывание неуспешного прогона.

Параметр **--early-drops-threshold** задает долю потерь в процентах, при превышении которой прогон прекращается, не дожидаясь истечения заданной длительности. Параметр применяется для сокращения времени поиска максимума: заведомо неуспешные прогоны прекращаются в течение нескольких секунд вместо 600.

Проверка выполняется на каждом интервале передачи метрик по двум условиям, срабатывание любого из которых приводит к прекращению прогона:

- Доля потерь за все время прогона, включая фазу набора нагрузки, превысила заданное значение.
- Фаза набора нагрузки завершена, и доля потерь, рассчитанная без учета потерь и потоков этой фазы, превысила заданное значение.

Note: Без указания параметра прогон выполняется до конца независимо от доли потерь. Проверка отключена, значение по умолчанию отсутствует.

При прекращении прогона выводится сообщение **This test will stop as drops are too big!**, а в файл **result_for_mult.txt** записываются два значения множителя нагрузки через запятую:

Значение	Содержание
Первое	Нижняя граница: множитель, пересчитанный на момент последнего успешного замера
Второе	Верхняя граница: множитель, пересчитанный на момент первого превышения порога

Границы вычисляются пропорционально времени: чем позже в ходе прогона произошло превышение, тем ближе оценка к заданному значению множителя. Полученный диапазон используется как начальный интервал для следующей итерации поиска.

7.3.2. Выбор значения параметра.

При успешном завершении прогона достигнутые значения пропускной способности и числа соединений в секунду записываются в файл `max_results.txt`. Значения являются мгновенными максимумами по замерам, поэтому для отчетности предпочтительны средние значения установившейся фазы, приводимые в легенде графиков.

Значение задается выше порога приемки, но ниже уровня, при котором результат заведомо непригоден. Слишком близкое к порогу приемки значение приводит к прекращению прогонов из-за кратковременных выбросов, а завышенное сводит эффект от параметра к нулю.

Этап поиска	Рекомендуемое значение
Грубое определение диапазона	от 0,1 до 1
Уточнение вблизи максимума	от 0,01 до 0,05
Зачетный прогон	не задается

Note: В зачетном прогоне параметр не указывается: прогон должен отработать полную длительность, а вердикт выносится по значению из вывода скрипта относительно порога, заданного параметром `--drp`.

7.3.3. Расчет доли потерь.

Доля потерь соединений на дашборде и в выводе скрипта рассчитывается по-разному. Панель **Connection drops** делит потери на полное число потоков за прогон, без вычета фазы набора нагрузки. Скрипт по завершении выводит значение, из которого потери и потоки фазы набора нагрузки исключены. Зачетным является значение скрипта: именно оно сопоставляется с порогом, заданным параметром `--drp`. Панель применяется для наблюдения за ходом прогона.

Панели, наименование которых начинается со слова `Current`, отображают последнее полученное значение, а не максимальное за прогон. Максимальные и средние значения приводятся в легенде графиков соответствующего показателя.

По завершении прогона скрипт выводит раздельную статистику клиентской и серверной сторон и блок оценки результата. В блоке оценки приводятся: доля потерь за прогон, суммарные потери UDP, потери TCP по каждой стороне, а также потери и число потоков за фазу набора нагрузки.

При успешном завершении прогона достигнутые значения пропускной способности и числа соединений в секунду записываются в файл `max_results.txt`.

Доля потерь определяется по счетчикам `tcps_drops` и `tcps_conndrops`, а также по счетчику `udps_keepdrops`, значение которого берется как максимум из клиентской и серверной сторон. Сессии, устанавливаемые в фазе набора нагрузки, в расчете не участвуют.

Панель **Connection drops** реализует расчет по указанным счетчикам: суммы отдельно по клиентской и серверной стороне, максимум из двух значений, деление на число открытых потоков. Вычет фазы набора нагрузки в панели не выполняется, поэтому ее значение выше зачетного и применяется для наблюдения за ходом прогона.

Note: Панель **Total connection drops** дополнительно включает счетчики `tcps_keepdrops`, `tcps_timeoutdrop`, `tcps_testdrops` и `udps_keepdrops`, вследствие чего дает завышенное значение. Панель применяется как вспомогательная для выявления обрывов по таймауту и принудительных обрывов при завершении прогона, в вердикт не входит.

7.3.4. Условие достоверности измерения.

Результат характеризует NGFW только при отсутствии ограничения со стороны генератора. Признаки ограничения:

Признак	Источник
Рост показателя Max Trex TX Queue length, packets	Grafana, раздел Evaluation
Ненулевое значение queue_full	TUI
Avg Trex CPU utilization выше 70 %	Grafana
Ненулевое значение oerrors	Команда stats -p

При выявлении любого из признаков измеренное значение характеризует стенд и не может быть принято как показатель NGFW. Заполнение очереди передачи обычно вызвано превышением максимальной пропускной способности сетевого адаптера, в этом случае проверяются параметры PCI.

7.4. Описание основных счетчиков TRex

Перед началом активной фазы и во время разгона необходимо проверить, что генератор не является узким местом. Для этого используются счетчики TRex, выводимые командой `stats -a` в консоли генератора и в режиме TUI.

В таблице ниже перечислены счетчики, которые необходимо контролировать во время прогона. Ненулевые или аномальные значения могут указывать на ограничения со стороны генератора или проблемы в сети/конфигурации, из-за чего результаты испытаний становятся недостоверными.

Счетчики делятся на три группы:

- Общие – быстрая оценка активности генератора.
- ASTF (TCP/UDP) – состояние соединений и качество передачи.
- Flow-table – работа таблицы потоков и использование ресурсов.

7.4.1. Общие счетчики.

Счетчик	Описание	Показатель
active_flows	Активные потоки	Должно расти с нагрузкой
est_flows	Установленные потоки	Если близко к 0 при активном тесте – соединения не устанавливаются
tx_bw_l7	Подтвержденная исходящая L7 пропускная способность (bps)	Должна быть близка к целевой нагрузке
rx_bw_l7	Подтвержденная входящая L7 пропускная способность (bps)	Аналогично
tx_ratio	Отношение подтвержденных данных к переданным (100 % = нет ретрансмиссий)	Падение ниже 100 % указывает на ретрансмиссии/потери
queue_full	Переполнение внутренних очередей TRex	Недостаточно CPU ядер или пропускной способности шины/интерфейсов
oerrors	Ошибки на выходных портах	Проблемы с драйвером, физическим соединением или перегрузка

7.4.2. Счетчики ASTF (TCP/UDP).

Счетчик	Описание	Показатель
tcps_connattempt	Число попыток установить TCP-соединение	Применяется для контроля симметрии сторон: сопоставляется с tcps_accepts серверной стороны. В расчет доли потерь не входит
tcps_connects	Число успешно установленных соединений	Если tcps_connects $\approx$ 0 при ненулевом tcps_connattempt, трафик не проходит (проблема сети, политики, ARP)
tcps_accepts	Число принятых соединений (серверная сторона)	Должно быть равно tcps_connects и tcps_closed при успешном прогоне
tcps_closed	Закрытые соединения (включая сброшенные)	Используется для проверки потерь. На клиенте tcps_connattempt == tcps_closed
tcps_conndrops	Соединения, разорванные на этапе установления, до перехода в установленное состояние (embryonic)	Нехватка ресурсов генератора (dp_flows, память), ошибки сети/маршрутизации, блокировка SYN на NGFW, слишком короткие таймауты. Входит в зачетную формулу доли потерь совместно с tcps_drops
tcps_drops	Разорванные установленные соединения. С tcps_conndrops не пересекается, поэтому в расчете значения складываются	Потери сверх порога
tcps_timeoutdrop	Соединения, разорванные по таймауту ретрансляции	Задержки/потери в сети, DUT не отвечает
tcps_keepprops	Соединения, разорванные из-за сбоя keepalive	Проблемы с долгими сессиями (тест CC)
tcps_sndrexmitpack / tcps_sndrexmitbyte	Ретранслированные пакеты/байты	Высокий уровень ретрансляций – потери или перегрузка
tcps_rcvoopackdrop / tcps_rcvoobytesdrop	Out-of-order пакеты/байты, отброшенные из-за переполнения очереди	Нарушение порядка пакетов. Необходимо проверить сетевые пути
tcps_nombuf	Пакеты TCP, отброшенные из-за нехватки mbuf	Недостаточно буферов памяти. Необходимо увеличить объем больших страниц памяти или снизить нагрузку

Счетчик	Описание	Показатель
udps_keepdrops	Потеря UDP-датаграммы в тракте: ответ не получен, сессия закрыта по таймауту	Входит в зачетную формулу доли потерь: берется максимум значений клиентской и серверной сторон
udps_connects / udps_closed	Установленные и закрытые UDP-сессии	При успешном прогоне udps_closed равно udps_connects. Расхождение указывает на незавершенные сессии
udps_nombuf	UDP-пакеты, отброшенные из-за нехватки mbuf	Недостаточно буферов памяти. Необходимо увеличить объем больших страниц памяти

7.4.3. Счетчики Flow-table.

Счетчик	Описание	Показатель
err_c_nf_throttled	Клиентские потоки не открыты из-за переполнения таблицы потоков	dp_flows слишком мал. Необходимо увеличить dp_flows в конфигурации
err_s_nf_throttled	Аналогично для серверных потоков	dp_flows слишком мал. Необходимо увеличить dp_flows в конфигурации
err_flow_overflow	События переполнения от maintenance-потока	dp_flows слишком мал. Необходимо увеличить dp_flows в конфигурации
err_no_memory	Нехватка динамической памяти для создания потока	Недостаточно оперативной памяти или больших страниц памяти. Необходимо увеличить объем больших страниц памяти
err_dct	Дублирующиеся потоки из-за проблем с aging и больших задержек	Возможны ложные срабатывания. Необходимо проверить задержки, настройки aging
err_c_tuple_err	Потоки не открыты на клиентской стороне из-за нехватки клиентов в пуле	Исчерпан пул клиентских IP/портов. Необходимо увеличить количество клиентов
err_rx_throttled	Rx-поток был приторможен из-за переполнения очереди NIC	Перегрузка приема. Необходимо проверить CPU/нагрузку, возможно, увеличить ядра

Счетчик	Описание	Показатель
rss_redirect_drops	RSS-пакеты, отброшенные из-за переполнения буферов других ядер	Недостаточно мощности/буферов. Необходимо увеличить ресурсы или снизить нагрузку
err_cwf	Клиентский пакет, не соответствующий потоку (в loopback не встречается)	Может быть из-за DUT, отправляющего пакеты после закрытия потока. Необходимо проверить DUT.

7.4.4. Проверка счетчиков TRex перед началом активной фазы.

Ключевые счетчики выводятся командой `stats -a` в консоли генератора и в режиме TUI.

Правила интерпретации счетчиков:

- На клиентской стороне: `tcps_connattempt == tcps_closed` (каждая попытка в итоге закрывается, включая сброшенные).
- На серверной стороне: `tcps_accepts == tcps_connects == tcps_closed`.
- Если `tcps_drops == 0` с обеих сторон, но трафик не идет (мало `tcps_connects`), это означает, что SYN-пакеты не доходят или отбрасываются до установления соединения.
- Если `tcps_conndrops ≈ tcps_connattempt`, это означает, что практически все попытки установить соединение завершаются неудачей. Причинами могут быть: неверная настройка шлюзов или ARP, блокирующее правило NGFW, переполнение таблицы потоков (при этом также растет `err_flow_overflow`) или нехватка памяти.
- Успешная передача данных подразумевает: в пределах одной стороны `tcps_sndbyte == tcps_rcvackbyte`, между сторонами `tcps_sndbyte` клиента == `tcps_rcvbyte` сервера и `tcps_sndbyte` сервера == `tcps_rcvbyte` клиента.
- Падение `tx_ratio` ниже 100 % указывает на ретрансмиссии и возможные потери.

7.4.5. Порядок проверки во время прогона.

Перед основным прогоном и после завершения фазы набора нагрузки (`rampup`) открыть консоль TRex (`./trex-console`), выполнить команду `stats -a` и убедиться, что:

- `tcps_conndrops` не растет (в идеале равно 0 или незначительно);
- `tcps_drops` не растет;
- `err_c_nf_throttled`, `err_s_nf_throttled`, `err_flow_overflow`, `err_no_memory` равны 0 или не увеличиваются;
- `tcps_nombuf`, `udps_nombuf` равны 0;
- `queue_full` (системный) равен 0;
- `oerrors` (системный) равен 0;
- `active_flows` и `est_flows` соответствуют ожидаемой нагрузке;
- `tcps_connects` увеличивается пропорционально нагрузке (трафик реально устанавливается).

Если в процессе разгона или в первые минуты активной фазы наблюдается рост `tcps_conndrops` или любого из перечисленных счетчиков ошибок, необходимо остановить прогон и устранить причину (увеличить `dp_flows`, проверить сетевую связность, снять излишнюю нагрузку с генератора). Прогон с такими признаками не может считаться достоверным.

8. Программа и методика испытаний

Настоящий раздел определяет порядок проведения трех испытаний:

- Максимальная производительность на смешанном трафике (EMIX).
- Максимальное количество одновременных соединений.
- Максимальное количество новых соединений в секунду.

8.1. Испытание 1. Максимальная производительность на EMIX

8.1.1. Цель.

Определить максимальную пропускную способность NGFW на смешанном профиле трафика при доле потерь не выше установленного порога.

Испытание проводится на каждой из четырех конфигураций NGFW, определенных в п. 5.2.3: **Packet Filter + DPI, IDPS, ATP1** и **ATP2**. Конфигурации различаются составом включенных модулей проверки и дают различную пропускную способность на одной и той же платформе.

Для каждой конфигурации выполняется отдельный поиск максимума. Значение параметра **--ngfw-config** приводится в соответствии с испытываемой конфигурацией, что обеспечивает разделение результатов на дашборде и в отчете.

Испытания на максимальное количество одновременных соединений и на количество новых соединений в секунду проводятся только на конфигурации **Packet Filter + DPI**.

8.1.2. Профиль трафика.

Профиль **emix_fstack_ftp.py**, 13 приложений. Характеристики приведены в разделе 6.2. Требуется наличие файлов захвата в каталоге **/opt/dumps**.

8.1.3. Параметры, подлежащие настройке.

Параметр	Назначение	Влияние на результат
-m	Множитель профиля	Линейно масштабирует частоту создания сессий всех приложений. Основной параметр поиска максимума
--rampup N	Фаза набора нагрузки	Задаёт продолжительность нарастания в профиле и исключает потери и потоки первых N секунд из вердикта. Значение по умолчанию 90
--drp X	Порог доли потерь, %	Критерий прохождения. Указывается явно: значение по умолчанию равно 1, что в тысячу раз мягче принятого критерия
--early-drops-threshold X	Порог досрочного прерывания, %	Прерывает прогон, не дожидаясь его окончания, если доля потерь превысила X. Применяется при поиске максимума – описано в разделе 7.3.1
-d	Длительность, с	Для зачетного прогона 600

Параметр	Назначение	Влияние на результат
--latency_pps	Частота зондирования задержки	Требуется для наполнения панелей раздела Latency

На достижимый максимум влияет параметр **-c** сервиса TRex, задающий число рабочих потоков DPDK. Порядок подбора значения приведен в п. 3.3.3.

8.1.4. Команда запуска.

```
./trex_run_perf_influx.py -s 127.0.0.1 -f emix_fstack_ftp.py \
-m 6.8 -d 600 \
--rampup 90 --drp 0.001 \
--test-id emix_$(date +%Y%m%d_%H%M) $STATS
```

Ориентировочные значения множителя рассчитаны от ожидаемой полосы исходя из того, что при -m 1 профиль создает 1,46 Гбит/с на уровне L2. Значения приведены для каждой из четырех испытываемых конфигураций NGFW отдельно, поскольку состав включенных модулей проверки определяет достижимую пропускную способность.

Платформа	Конфигурация NGFW	Ожидаемая полоса, Гбит/с	Ориентировочный, -m
vKX-8	Packet Filter + DPI	5	3,4
	IDPS	4	2,7
	ATP1	2,5	1,7
	ATP2	1,4 (выделено 4 ядра)	1
KX-100	Packet Filter + DPI	10	6,8
	IDPS	3	2,1
	ATP1	2	1,4
	ATP2	1 (выделено 4 ядра)	0,7
KX-400	Packet Filter + DPI	40	27
	IDPS	15	10
	ATP1	12	8,2
	ATP2	5,9 (выделено 7 ядер)	4
KX-1000	Packet Filter + DPI	100	68
	IDPS	40	27
	ATP1	40	27
	ATP2	14 (выделено 14 ядер)	9,6

Платформа	Конфигурация NGFW	Ожидаемая полоса, Гбит/с	Ориентировочный, -m
KX-3500	Packet Filter + DPI	200	137
	IDPS	140	96
	ATP1	60	41
	ATP2	37,6 (выделено 31 ядро)	26

Приведенные значения являются начальной точкой поиска. Фактический максимум определяется методом п. 7.3.

Note: Профиль EMIX асимметричен: 89 процентов объема передается от серверной стороны к клиентской. С портами 100 Гбит/с суммарная пропускная способность ограничена значением около 113 Гбит/с, поскольку порт серверной стороны достигает предела раньше. Ожидаемые результаты для KX-1000 при конфигурации Packet Filter + DPI (100 Гбит/с) достигаются на пределе, а для KX-3500 (200 и 140 Гбит/с) требуют второй пары портов генератора.

8.1.5. Контролируемые параметры.

Параметр	Панель Grafana	Счетчик TUI
Пропускная способность, Гбит/с	Current TP, Total throughput (bit/s)	total_tx_L2, total_rx
Пакетная скорость	Current PPS, Total Packets rate (pps)	total_pps
Доля потерь соединений, %	Connection drops	–
Абсолютные потери соединений	Total connection drops	tcps_conndrops, tcps_drops
Потери в байтах	Total drops, bytes; Drops (by bytes)	–
Пакетные потери	–	drop_rate
Очередь передачи	Max Trex TX Queue length, packets	queue_full
Загрузка процессора генератора, %	Avg Trex CPU utilization	cpu_util
Задержка, мкс	Latency by NIC port	–
Средний размер кадра, байт	Avg packet size, bytes	m_avg_size

Для просмотра статистики в TUI нажать **ESC → t**. Для прокрутки параметров использовать стрелки. Детализация причин потерь выводится командой **stats -a**.

8.2. Испытание 2. Максимальное количество одновременных соединений

8.2.1. Цель.

Определить максимальное количество одновременно поддерживаемых NGFW TCP-сессий при доле потерь не выше установленного порога.

8.2.2. Профиль трафика.

Профиль **trex_cc_tune_timeouts.py**. Серверная сторона удерживает соединение паузой перед отправкой ответа, цикл повторяется 10 раз, за счет чего сессии накапливаются. Характеристики приведены в разделе 6.3.

8.2.3. Расчетные соотношения.

Клиент выполняет 10 запросов по 1 КБ ответа в рамках одного TCP-соединения. Задержка между транзакциями определяется выражением:

$$\text{delay} = (\text{Целевое_CC} / \text{CPS}) / 10$$

Обратные соотношения:

$$\text{Время жизни сессии} = \text{delay} * 10$$

$$\text{CC} = \text{CPS} * \text{Время жизни сессии}$$

Пример: при целевом числе соединений 1 000 000 и темпе установления 20 000 соединений в секунду требуется задержка 5 секунд.

Note: Целевое число соединений не должно превышать значение **dp_flows** файла **/etc/trex_cfg.yaml**. При превышении генератор завершается аварийно на этапе выделения памяти.

8.2.4. Параметры, подлежащие настройке.

Параметр	Назначение	Влияние на результат
--delay N	Пауза серверной стороны, с	Основной параметр наращивания числа соединений. Увеличивает время жизни сессии линейно, не увеличивая полосу и пакетную скорость
-m	Множитель профиля	Второй параметр наращивания. Увеличивает темп установления и вместе с ним число соединений, полосу и нагрузку на генератор
--limit N	Предел общего числа соединений за прогон	Не является числом одновременных соединений. Задается не ниже произведения темпа установления на длительность прогона
-t keep_timeout=N	Таймауты TCP, мс	Должен превышать значение delay в миллисекундах. По умолчанию вычисляется как delay * 1000 + 5000
-d	Длительность, с	Не менее суммы времени набора нагрузки, времени жизни сессии и 300 секунд активной фазы
--drp X	Порог доли потерь, %	Критерий прохождения

Note: Нарастивание числа соединений за счет задержки предпочтительнее наращивания за счет множителя: первое не увеличивает нагрузку на датапас стенда и позволяет достичь целевого значения при меньшей полосе. Значение множителя подбирается вторым, после фиксации задержки.

8.2.5. Команда запуска.

```
./trex_run_perf_influx.py -s 127.0.0.1 -f trex_cc_tune_timouts.py \
-m 60000 -d 600 \
--limit 36000000 --delay 5 --rampup 90 --drp 0.001 \
--test-id cc_$(date +%Y%m%d_%H%M) $STATS
```

Ориентировочные значения рассчитаны от ожидаемых результатов по соотношениям п. 8.2.3. Темп установления соединений численно равен множителю нагрузки.

Платформа	Целевое СС	delay, с	Ориентировочный -m	limit
vKX-8	512 000	5	10 240	6 200 000
KX-100	3 000 000	5	60 000	36 000 000
KX-400	25 000 000	21	119 048	72 000 000
KX-1000	40 000 000	21	190 476	115 000 000
KX-3500	100 000 000	21	476 190	286 000 000

Note: Значение **dp_flows** файла **/etc/trex_cfg.yaml** должно превышать целевое число соединений. Значение 4 000 000, приведенное в п. 5.3, достаточно только для платформ vKX-8 и KX-100. Для остальных платформ требуется соответствующее увеличение **dp_flows** и объема больших страниц памяти (hugepages).

Ожидаемые результаты для KX-1000 и KX-3500 (40 и 100 млн соединений) превышают предельные значения эталонных конфигураций нагрузочных стендов, приведенные в п. 7.1.2 (от 29 до 58 млн). Достижение этих значений требует второго экземпляра генератора либо сервера с большим объемом оперативной памяти.

8.2.6. Контролируемые параметры.

Параметр	Панель Grafana	Счетчик TUI
Одновременные соединения	Current CC, Concurrent connections	m_active_flows, m_est_flows
Всего соединений за прогон	Total connections	tcps_connattempt
Темп установления, соединений/с	Current CPS, Connections/s	total_cps
Доля потерь соединений, %	Connection drops	–
Абсолютные потери соединений	Total connection drops	tcps_drops, tcps_conndrops
Потери по таймауту	–	tcps_timeoutdrop, tcps_keepdrops
Пропускная способность, Гбит/с	Current TP	total_tx_L2
Загрузка процессора генератора, %	Avg Trex CPU utilization	cpu_util

Для просмотра статистики в TUI нажать **ESC → t**. Для прокрутки параметров использовать стрелки.

Детализация причин потерь выводится командой **stats -a**.

8.3. Испытание 3. Максимальное количество соединений в секунду

8.3.1. Цель.

Определить максимальную скорость установления новых TCP-сессий, обрабатываемую NGFW при доле потерь не выше установленного порога.

8.3.2. Профиль трафика.

Профиль **trex_cps.py**. Характеристики приведены в разделе 6.4. В каталоге с профилем должен находиться файл **http_response_payload.html**.

8.3.3. Параметры, подлежащие настройке.

Параметр	Назначение	Влияние на результат
-m	Множитель профиля	Основной параметр поиска максимума. Прямо задает частоту установления соединений
-t size=N	Размер прикладной транзакции, КБ	Уменьшение снижает объем данных на соединение и высвобождает полосу. Рекомендуемые значения 1, 2, 4, 16, 64 КБ. Для испытания обычно 1 КБ
--rampup N	Фаза набора нагрузки	Задаёт продолжительность нарастания в профиле и исключает потери и потоки первых N секунд из вердикта. Значение по умолчанию 90
--drp X	Порог доли потерь, %	Критерий прохождения
-d	Длительность, с	Для зачетного прогона 600

8.3.4. Команда запуска.

```
./trex_run_perf_influx.py -s 127.0.0.1 -f trex_cps.py \
-m 64000 -d 600 -t size=1 \
--rampup 90 --drp 0.001 \
--test-id cps_$(date +%Y%m%d_%H%M) $STATS
```

Число новых соединений в секунду численно равно множителю нагрузки, поэтому ориентировочное значение множителя совпадает с ожидаемым результатом.

Платформа	Целевое число соединений в секунду	Ориентировочный -m
vKX-8	79 000	79 000
KX-100	64 000	64 000
KX-400	370 000	370 000
KX-1000	800 000	800 000
KX-3500	1 300 000	1 300 000

Note: Ожидаемый результат для КХ-3500 (1,3 млн соединений в секунду) превышает предельное значение одного экземпляра генератора, приведенное в п. 7.1.2. Достижение этого значения требует второго экземпляра генератора.

8.3.5. Контролируемые параметры.

Параметр	Панель Grafana	Счетчик TUI
Новых соединений в секунду	Current CPS, Connections/s	total_cps
Инициировано соединений	–	tcps_connattempt
Установлено соединений	–	tcps_connects
Принято сервером	–	tcps_accepts
Одновременные соединения	Current CC	m_active_flows
Доля потерь соединений, %	Connection drops	–
Абсолютные потери соединений	Total connection drops	tcps_drops, tcps_conndrops
Пропускная способность, Гбит/с	Current TP	total_tx_L2
Задержка, мкс	Latency by NIC port	–
Загрузка процессора генератора, %	Avg Trex CPU utilization	cpu_util

Для просмотра статистики в TUI нажать **ESC → t**. Для прокрутки параметров использовать стрелки.

Детализация причин потерь выводится командой **stats -a**.

Симметрия сторон контролируется соотношением счетчиков **tcps_connattempt** клиента и **tcps_accepts** сервера: расхождение указывает на потери при установлении соединений.

8.3.6. Результаты испытания.

Параметр	Значение	Источник
Множитель -m		Параметр запуска
Размер ответа size, КБ		Параметр запуска
Максимум новых соединений в секунду		Grafana, график Connections/s, значение Max в легенде
Доля потерь соединений, %		Вывод скрипта, строка Total TCP/UDP drops percentage during test
Средняя задержка, мкс		Grafana, график Latency by NIC port, значение Mean в легенде
Длительность прогона, с		Вывод скрипта, строка workload duration (sec)

9. Отчет о нагрузочном тестировании NGFW

9.1. Общие сведения

Поле	Значение
Дата и время проведения испытаний	
Организация-исполнитель	
Организация-заказчик	
Платформа NGFW	
Версия программного обеспечения NGFW	
Количество правил фильтрации	
Логирование окончания сессии на UDP-порт	
Версия TRex	
Хост TRex: процессор (модель, ядра)	
Хост TRex: объем памяти	
Хост TRex: тип подключения интерфейсов	
Параметры запуска сервиса TRex	
Значение dp_flows в /etc/trex_cfg.yaml	
Объем выделенных больших страниц памяти	
Идентификаторы прогонов (--test-id)	
Примененный порог потерь, %	0,001
Результаты предварительной проверки стенда	

9.2. Сведения о конфигурации стенда

Схема подключения: описать или приложить рисунок.

Интерфейс NGFW	Зона безопасности	IP-адрес/сеть	Подключение к TRex

Параметры NGFW, влияющие на производительность (например, выделенные ядра, режим DPI, настройки логирования):

9.3. Результаты испытаний

9.3.1. Максимальная производительность на EMIX.

Показатель – пропускная способность, Гбит/с (L2).

Для платформы: _____

Параметр	Packet Filter + DPI	IDPS	ATP1	ATP2	Источник
Ожидаемый результат, Гбит/с					
Фактический результат, Гбит/с					
Множитель -m					Параметр запуска
Пропускная способность, Гбит/с (L2)					Grafana, график Total throughput (bit/s), значение Max в легенде
Пакетная скорость, Mpps					Grafana, график Total Packets rate (pps), значение Max в легенде
Доля потерь соединений, %					Вывод скрипта, строка Total TCP/UDP drops percentage during test
Средняя задержка, мкс					Grafana, график Latency by NIC port, значение Mean в легенде
Длительность прогона, с					Вывод скрипта, строка workload duration (sec)

Выделенное количество ядер CPU для асинхронных задач (для ATP2): _____

9.3.2. Максимальное количество одновременных соединений.

Для платформы: _____

Параметр	Packet Filter + DPI	Источник
Ожидаемый результат, соединений		
Множитель -m		Параметр запуска
Задержка delay, с		Параметр запуска
Максимум одновременных соединений		Grafana, график Concurrent connections, значение Max в легенде
Темп установления на пике, соединений/с		Grafana, график Connections/s, значение Max в легенде
Доля потерь соединений, %		Вывод скрипта, строка Total TCP/UDP drops percentage during test
Длительность удержания пика, с		Grafana, соответствующий график, по оси времени

9.3.3. Максимальное количество соединений в секунду.

Для платформы: _____

Параметр	Значение	Источник
Ожидаемый результат, соединений/с		
Множитель -m		Параметр запуска
Размер ответа size, КБ		Параметр запуска
Максимум новых соединений в секунду		Grafana, график Connections/s, значение Max в легенде
Доля потерь соединений, %		Вывод скрипта, строка Total TCP/UDP drops percentage during test
Средняя задержка, мкс		Grafana, график Latency by NIC port, значение Mean в легенде
Длительность прогона, с		Вывод скрипта, строка workload duration (sec)

9.4. Заключение

Общий результат испытаний:

☐ Соответствует критериям приемки ☐ Не соответствует критериям приемки

Комментарии, замечания и рекомендации:

9.5. Подписи

Должность	ФИО	Подпись	Дата

Примечания по заполнению:

- В колонке «Фактический результат» для каждого испытания указывается значение, полученное при соблюдении критерия (потери $\leq 0,001\%$). Если порог был превышен, то указывается наибольшее значение, выдержанное без превышения порога.
- В разделе «Общие сведения» необходимо указать точные параметры TRex (команду запуска, объем больших страниц памяти и т.п.), так как они могут влиять на воспроизводимость результатов.
- Схема стенда должна быть приложена в виде рисунка или подробного описания, включая соответствие интерфейсов TRex и NGFW, а также зоны безопасности.
- Допускается добавлять дополнительные страницы с графиками из Grafana, логами прогонов и другими материалами, подтверждающими результаты.
- Зачетным является значение, выводимое скриптом по завершении прогона в строке Total TCP/UDP drops percentage during test. Оно рассчитывается по счетчикам tcps_drops и tcps_conndrops обеих сторон и счетчику udps_keepdrops (максимум из двух сторон) с вычетом потерь и потоков фазы набора нагрузки. Панель Connection drops дашборда Grafana применяется для наблюдения за ходом прогона и показывает значение без вычета фазы набора нагрузки.
- Порог задается параметром --drp во всех командах запуска и во всех испытаниях настоящего документа равен 0.001.

10. Приложение A. Развертывание генератора на виртуальной машине

Приложение описывает особенности развертывания генератора трафика на виртуальной машине. Такая конфигурация применяется для отладки, предварительных проверок и демонстраций. Для проведения приемочных испытаний применяется целевая конфигурация согласно разделу 1.2.

10.1. Ограничения виртуализованного стенда

Генератор трафика, развернутый на виртуальной машине, ограничен программным датапасом гипервизора. Достижимые показатели существенно ниже, чем на физическом сервере с картами Intel E810.

Способ подключения интерфейсов	Ограничение
Проброс физического устройства (PCI passthrough)	Практически отсутствует, требуются отдельные физические порты
SR-IOV (виртуальные функции)	Незначительное, коммутация выполняется аппаратно
OVS-DPDK с vhost-user	Умеренное, требуются выделенные ядра под потоки обработки
virtio через мост ядра Linux	Существенное, пакетная скорость ограничена сотнями тысяч пакетов в секунду

Note: Результаты, полученные на виртуализованном стенде, не сопоставимы с результатами целевой конфигурации и не могут применяться для определения характеристик NGFW.

10.2. Отключение аппаратного расчета контрольных сумм

При использовании паравиртуализированных интерфейсов virtio аппаратный расчет контрольных сумм подлежит отключению. По умолчанию TRex включает аппаратный расчет контрольных сумм IP, TCP и UDP. Связка virtio и vhost эту функцию корректно не обрабатывает, вследствие чего пакеты отбрасываются на выходе из виртуальной машины.

Добавить в конфигурацию сервиса TRex дополнительные параметры.

```
ExecStart=/opt/trex/t-rex-64 -i --astf -c 6 --checksum-offload-disable --tso-disable --lro-disable
```

Признаки, наблюдаемые при невыполнении данного требования:

- Команда arp в консоли TRex выполняется успешно, поскольку протокол ARP не использует контрольную сумму.
- Команда ping завершается по таймауту при возрастающем счетчике opackets и нулевом значении oerrors.
- На tap-интерфейсе гипервизора пакеты от генератора отсутствуют.
- Трафик в обратном направлении, от NGFW к генератору, проходит без ошибок.
- Нагрузочные профили трафик не передают.

Note: Наличие в стартовом журнале TRex строки driver capability : TSO LRO указывает на включенную аппаратную разгрузку.

Программный расчет контрольных сумм увеличивает нагрузку на процессорные ядра. При переходе на SR-IOV или проброс физических устройств аппаратную разгрузку следует включить, исключив указанные параметры из команды запуска.

10.3. Настройка очередей virtio

При запуске генератора с числом рабочих потоков, превышающим число очередей интерфейса, сервис завершается с ошибкой вида:

ERROR: driver maximum tx queues is (1) required (6) reduce number of cores to support it

Интерфейс virtio по умолчанию предоставляется виртуальной машине с одной очередью. Для увеличения их числа на гипервизоре в описание каждого трафикового интерфейса добавляется:

```
<driver name='vhost' queues='8'/>
```

Число очередей задается не меньше требуемого значения параметра -с и не больше числа виртуальных процессоров.

Note: После изменения требуется полная остановка и повторный запуск виртуальной машины (**virsh shutdown**, затем **virsh start**). Перезагрузка изнутри гостевой системы устройство не пересоздает.

Каждая очередь создает на гипервизоре отдельный поток обработки, которому требуются вычислительные ресурсы. При исчерпании ресурсов гипервизора увеличение числа рабочих потоков генератора результата не дает.

10.4. Настройка моста ядра Linux

При подключении через мост ядра следует отключить обработку проходящего трафика подсистемой межсетевого экранирования:

```
sudo sysctl -w net.bridge.bridge-nf-call-iptables=0
```

```
sudo sysctl -w net.bridge.bridge-nf-call-ip6tables=0
```

```
sudo sysctl -w net.bridge.bridge-nf-call-arptables=0
```

Также следует отключить протокол связующего дерева, поскольку после подъема интерфейса порт до 30 секунд находится в состоянии блокировки:

```
sudo ip link set br0 type bridge stp_state 0
```

10.5. Диагностика прохождения трафика

При отсутствии трафика проверка выполняется в следующем порядке.

Состояние портов и результат разрешения ARP:

```
trex> stats --ps
```

Счетчики переданных и принятых пакетов до и после проверки связности:

```
trex> stats -p
```

```
trex> ping -p 0 -d 10.10.2.1
```

```
trex> stats -p
```

Наблюдение	Вероятная причина
Значение opackets не возрастает	Генератор не передает: несоответствие числа очередей и рабочих потоков либо неработоспособность интерфейса
Значение opackets возрастает, на tap-интерфейсе пакетов нет	Включен аппаратный расчет контрольных сумм либо действует фильтрация на гипервизоре
ARP разрешается, ICMP не проходит	Обработка ICMP запрещена на интерфейсах NGFW, проведению испытаний не препятствует
Значение oerrors возрастает	Превышение пропускной способности интерфейса

Захват трафика на гипервизоре выполняется на tap-интерфейсе виртуальной машины, а не на мосту:

```
virsh domiflist trex-vm
```

```
sudo tcpdump -ni <tap-интерфейс> -e arp or icmp
```

Note: При работающем генераторе сетевой интерфейс передан в управление DPDK и недоступен из ядра гостевой системы. Захват трафика внутри виртуальной машины средствами tcpdump результата не дает.

10.6. Размещение виртуальных машин

При размещении генератора и тестируемого устройства на виртуальных машинах применяются следующие требования:

- Виртуальные процессоры закрепляются за физическими ядрами без переподписки.
- Виртуальные процессоры и сетевые карты размещаются в одной NUMA-ноде.
- Память виртуальной машины обеспечивается страницами большого размера на стороне гипервизора.
- Режим управления частотой процессора устанавливается в значение максимальной производительности.

Note: При размещении клиентской и серверной частей на разных виртуальных машинах расположение их в одной NUMA-ноде дает до 50 тыс. дополнительных соединений в секунду.



Задать вопросы о продукте
в нашем официальном
Telegram-чате

kaspersky

www.kaspersky.ru

www.securelist.ru