

*Архипов Игорь Юрьевич*  
*студент, 2 курс, факультет “Комплексная безопасность ТЭК”*  
*Российский государственный университет нефти и газа (НИУ)*

*имени И. М. Губкина*

*Россия, г. Москва*

*Муляков Алексей Александрович*  
*студент, 2 курс, факультет “Комплексная безопасность ТЭК”*  
*Российский государственный университет нефти и газа (НИУ)*

*имени И. М. Губкина*

*Россия, г. Москва*

## **ГЕНЕРАТОРЫ СЕТЕВОГО ТРАФИКА WAN / LAN ДЛЯ ТЕСТИРОВАНИЯ ПРОПУСКНОЙ СПОСОБНОСТИ И СТРЕСС- ТЕСТИРОВАНИЯ СЕТИ. РАЗВЕРТЫВАНИЕ И ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ**

**Аннотация:** В данной статье рассмотрены принципы работы, развертывание и функциональные возможности генераторов сетевого трафика WAN/LAN, подчеркивая их критическую важность для объективной оценки производительности и устойчивости современных корпоративных сетей. Генераторы трафика – это специализированные аппаратные или программные решения, предназначенные для имитации реалистичного или экстремального сетевого потока данных с целью тестирования пропускной способности. Актуальность исследования обусловлена необходимостью выявления узких мест, проверки конфигураций оборудования и обеспечения соответствия сетей растущим требованиям к производительности и надежности. Современные генераторы трафика поддерживают моделирование разнообразных протоколов, что позволяет проводить комплексное тестирование как на физическом, так и на виртуальном уровне. Знание методик развертывания, настройки параметров генерации (интенсивность, размер пакетов, шаблоны) и анализа результатов тестирования позволяет сетевым инженерам и

администраторам гарантировать стабильность, производительность и отказоустойчивость инфраструктуры до ее ввода в промышленную эксплуатацию.

**Ключевые слова:** генераторы сетевого трафика, тестирование пропускной способности, стресс-тестирование, производительность сети, функциональное тестирование.

**Abstract:** This article discusses the principles of operation, deployment, and functionality of WAN/LAN network traffic generators, emphasizing their critical importance for an objective assessment of the performance and sustainability of modern corporate networks. Traffic generators are specialized hardware or software solutions designed to simulate realistic or extreme network data flow in order to test throughput. The relevance of the study is due to the need to identify bottlenecks, verify equipment configurations, and ensure that networks meet growing performance and reliability requirements. Modern traffic generators support modeling of various protocols, which allows for comprehensive testing at both the physical and virtual levels. Knowledge of deployment techniques, generation settings (intensity, packet size, templates), and analysis of test results allows network engineers and administrators to ensure the stability, performance, and fault tolerance of the infrastructure before it is put into commercial operation.

**Keywords:** network traffic generators, bandwidth testing, stress testing, network performance, functional testing.

## Введение

В условиях постоянного роста объема передаваемых данных и ужесточения требований к качеству обслуживания (QoS) и отказоустойчивости корпоративных и операторских сетей, задача объективной оценки их реальной пропускной способности и устойчивости к нагрузке выходит на первый план. Планирование модернизации, проверка конфигурации, верификация сервисных соглашений (SLA) и подготовка к пиковым нагрузкам требуют использования специализированных

инструментов, способных детерминировано генерировать, контролировать и анализировать сетевой трафик. Ключевым инструментом для решения этих задач являются генераторы сетевого трафика – аппаратные или программные решения, предназначенные для эмуляции разнородного трафика, характерного для современных сетей передачи данных. В отличие от простых утилит проверки связности, такие генераторы позволяют создавать сложные, настраиваемые потоки данных с заданными параметрами (интенсивность, размер пакетов, типы протоколов, шаблоны нагрузки), что необходимо для проведения стресс-тестирования и определения «узких мест» инфраструктуры. Особую актуальность приобретает разделение тестовых сценариев для сегментов WAN (глобальная сеть) и LAN (локальная сеть). В сегменте LAN критически важна проверка максимальной производительности коммутационного оборудования, эффективности механизмов качества обслуживания (QoS), сегментации (VLAN) и фильтрации (ACL) под высокой нагрузкой. Для WAN-сегментов, часто имеющих ограниченную пропускную способность и подверженных задержкам и джиттеру, фокус смещается на тестирование туннельных решений (IPsec, DMVPN, SD-WAN), механизмов сжатия и оптимизации трафика, а также на проверку стабильности каналов связи в условиях длительной и неравномерной нагрузки.

Стресс-тестирование – это процесс проверки устойчивости системы или приложения при экстремальных или предельных условиях работы, чтобы оценить его способность справляться с нагрузками, превышающими обычные. Цель стресс-тестирования – выявить точки отказа системы, определить её пределы и убедиться, что она продолжает функционировать или корректно восстанавливается после перегрузки.[6]

Функциональное тестирование – процесс проверки, как работает программное обеспечение или система в соответствии с заданными требованиями. Главная цель – убедиться, что все функции программы выполняются так, как ожидалось.[7]

**Объект исследования:** генераторы сетевого трафика.

**Предмет исследования:** методы и инструменты генерации сетевого трафика (LAN/WAN) для тестирования производительности, пропускной способности и устойчивости сетевой инфраструктуры.

**Цель исследования:** установить генераторы сетевого трафика и провести различные тестирования.

### **Литературный обзор**

Мы рассмотрели данную статью “WAN/LAN Network Traffic Generators for Bandwidth & Stress Testing” про генератор сетевого трафика Nping. Nping – инструмент с открытым исходным кодом для анализа ответов, управления временем отклика и генерации сетевых пакетов.

В данной статье рассмотрены основной функционал генератора сетевого трафика Nping. Но отсутствует упоминание о следующих важных аспектах:

1. Поддержка множества протоколов. Он может генерировать пакеты не только для ICMP (ping), но и для TCP, UDP, ARP, SCTP, позволяя тестировать ответы конкретных сервисов (например, отправляя TCP-пакет с флагом SYN).

2. Продвинутая настройка пакетов. Можно вручную задавать флаги TCP (SYN, ACK, RST, FIN и др.). Поля IP (TTL, идентификатор, фрагментацию). Случайные или полезные данные (payload). Скорость отправки и таймауты.[1]

В статье “Qperf - man pages section 1: User commands” мы рассмотрели генератор сетевого трафика Qperf, в ней приведено корректное, но не полное описание данного генератора сетевого трафика, не рассмотрены следующие важные моменты.

1. Богатый набор тестов (метрик) и параметров. Тесты: помимо tcp\_bw (пропускная способность) и tcp\_lat (задержка), есть множество других, например, udp\_bw, udp\_lat. Глубокие метрики: измеряет не только средние значения, но и время установки соединения, размеры окон (window size), загрузку CPU, что помогает находить узкие места.

2. Гибкость настройки тестов. Можно детально управлять параметрами: размером сообщений (--msg\_size), длительностью теста (--t) и другими. [8]

В статье “Утилита iPerf: тестирование пропускной способности и скорости сети” описывается генератор сетевого трафика Iperf3, но в ней не указаны ключевые технические особенности, которые делают Iperf3 стандартом индустрии стресс-тестирования сетей.

1. Детальная аналитика и метрики. Iperf3 предоставляет не просто общую пропускную способность, но и интервальные отчеты с выводом статистики (bandwidth, jitter, loss) каждые N секунд, что позволяет видеть динамику и стабилизацию потока; расширенные TCP-метрики с отчетами о потерях пакетов (retransmits), размере окна (window size).

2. Продвинутое возможности и флаги. Обратный тест (-R): Клиент запрашивает у сервера отправку трафика к себе, что обходит проблемы с асимметричной маршрутизацией или настройками брандмауэра. Контроль битрейта UDP (-b) позволяет точно задать нагрузку (например, -b 100М для 100 Мбит/с) и измерить потери и джиттер. [2]

Таблица 1 – Сравнительная характеристика генераторов сетевого трафика

	Nping	Qperf	Iperf3
Требования к ВМ	ЦП - достаточно 1 ядра; ОЗУ - минимум 512 Мб	ЦП - достаточно 1 ядра; ОЗУ - минимум 1 Гб	ЦП - не менее 2 ядер при скорости в 10 Гбит/с, - не менее 4 ядер при скорости в 25 Гбит/с; ОЗУ - минимум 1 Гб
Уровни модели OSI, на которых работает	L3-4	L4	L4
Максимальная скорость генерируемого трафика	2 Гбит/с	20 Гбит/с	25 Гбит/с

Актуальная версия у производителя	7.98	0.4.11	3.19.1-alt1
Установленная пакетная версия	7.95	0.4.11	3.19.1-alt1

### **Гипотезы исследования**

- Определить наиболее подходящий генератор сетевого трафика для стресс-тестирования сети.

### **Методы исследования**

**Тип исследования:** экспериментальный, с элементами сравнительного анализа.

**Характеристика выборки:** тестовая сеть из коммутатора. А также включает узлы с установленными генераторами трафика (iPerf3, Nping, qperf), выступающие в роли клиентов и серверов.

**Методы сбора и обработки данных:** генерация тестовой нагрузки и анализ трафика в Wireshark.

#### **Процедура эксперимента:**

- Установка программного генератора Iperf3.
- Установка программного генератора Nping.
- Установка программного генератора Qperf.
- Проведение тестов.

### **Проведение эксперимента**

После того, как мы ознакомились с принципами работы и установили программные генераторы сетевого трафика, перейдем к их функциональному

тестированию. Для этого развернем тестовый стенд, состоящий из коммутатора Cisco2960 и двух виртуальных машин, выполняющих роль клиента и сервера.

При выполнении эксперимента с использовался коммутатор Cisco2960. Топология сети указана на рисунке 1.

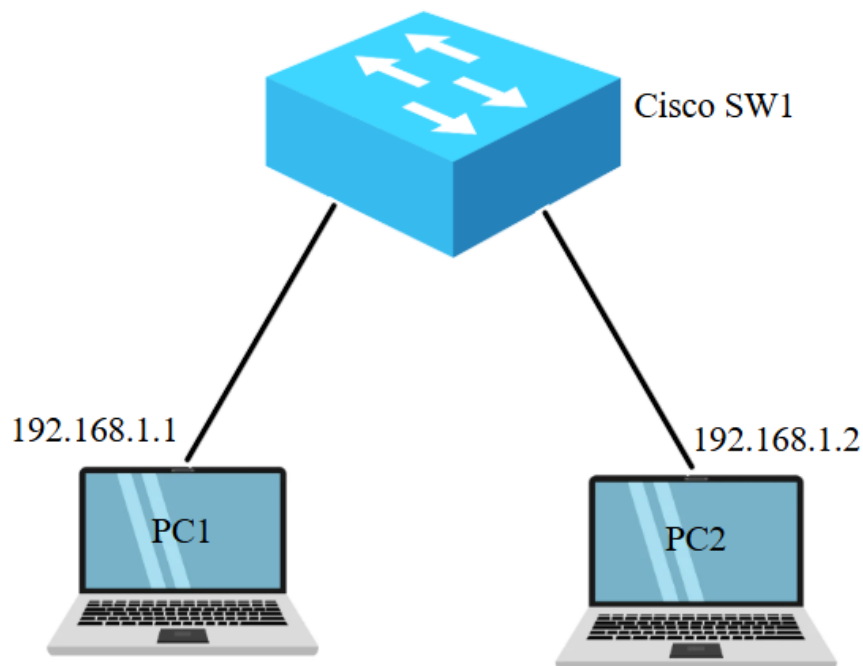


Рисунок 1 – Топология сети

Сначала установили генератор сетевого трафика `qperf`. Далее на сервере, а именно на принимающей стороне ввели `qperf`, а на тестирующей стороне ввели `qperf <ip_сервера> <тесты>`.

Для начала провели базовый тест TCP на задержку и измерение пропускной способности. На клиенте ввели команду `qperf 192.168.1.1 tcp_lat` и `qperf 192.168.1.1 tcp_bw`. Клиент и сервер обмениваются короткими TCP-пакетами, чтобы измерить время круговой задержки (RTT) между ними. Результат представлен на рисунке 2 и 3.

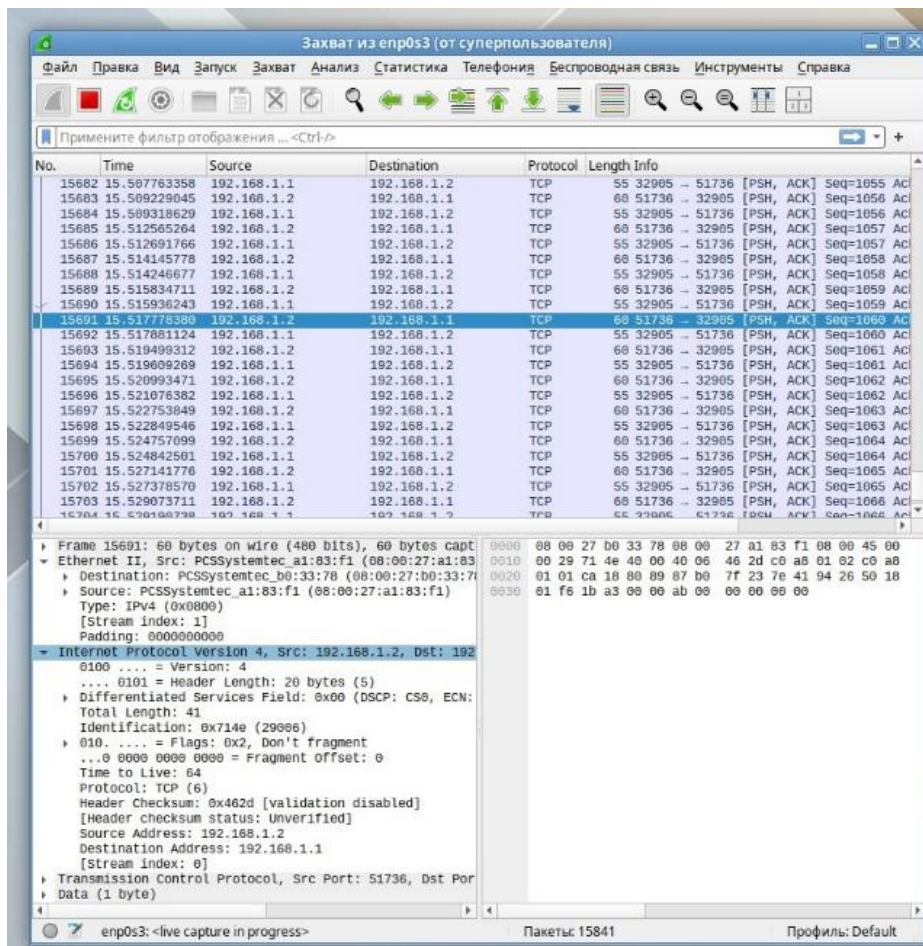


Рисунок 2 – Базовое тестирование curlf

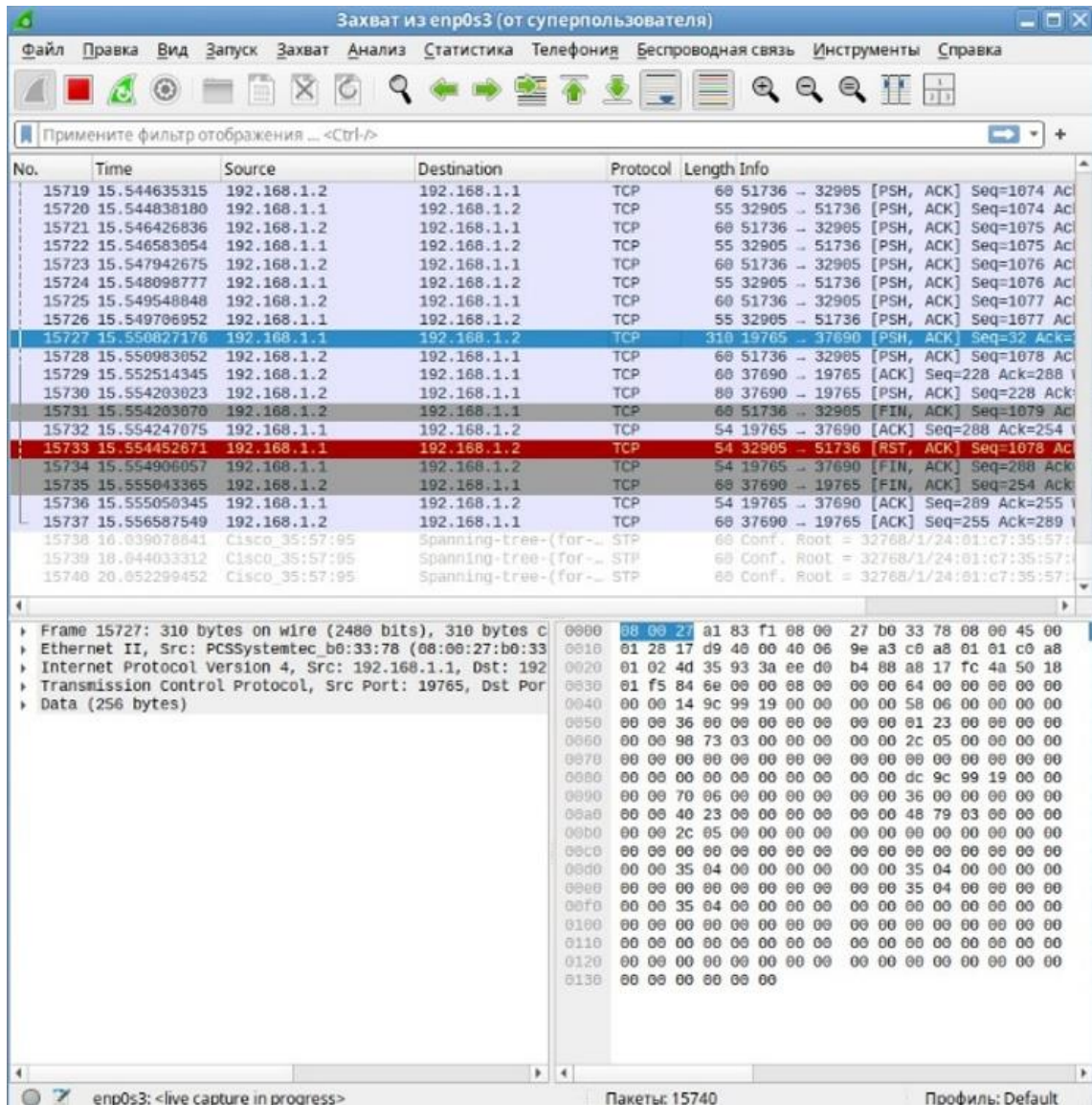


Рисунок 3 – Тестирование для измерения пропускной способности

На рисунке 4 зафиксирован момент пиковой нагрузки. По каналу 41023 - 51842 идут гигантские пакеты по 4 КБ данных каждый, в обоих направлениях, с минимальными паузами. Это и есть стресс-тест канала на максимальную скорость (tcp\_bw).

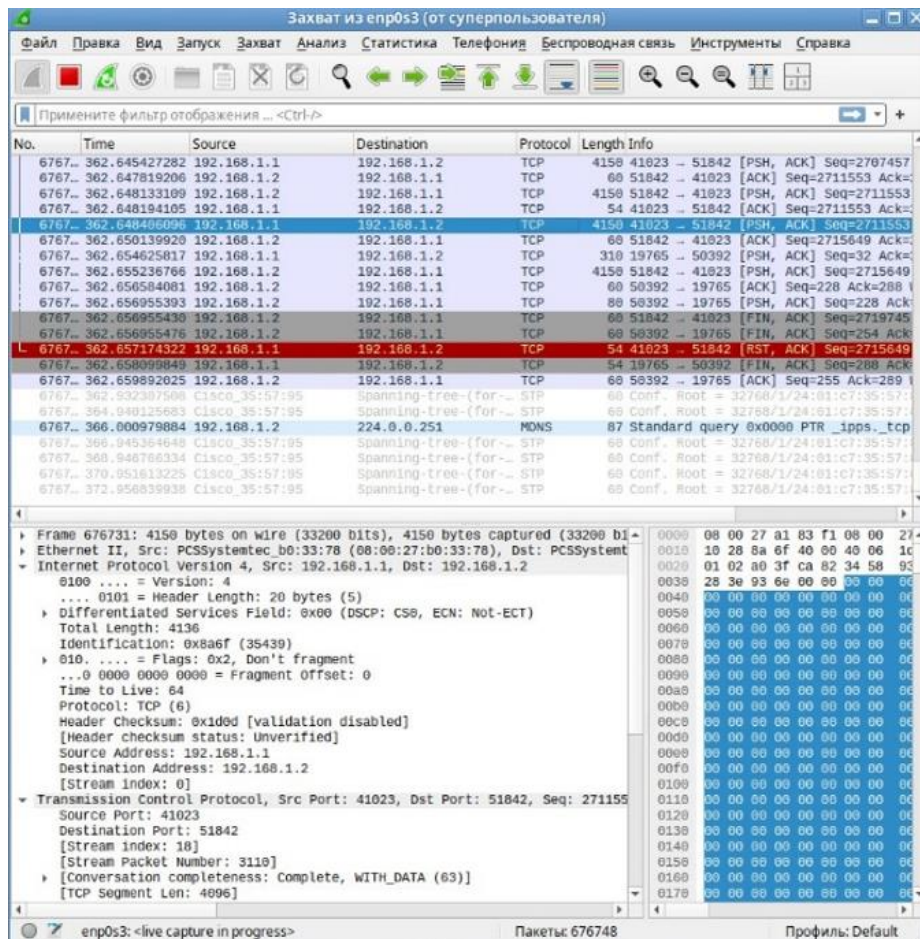


Рисунок 4 – Момент пиковой нагрузки с использованием qperf

Далее провели агрессивное нагрузочное стресс-тестирование UDP-канала на максимальную пропускную способность с использованием экстремально больших датаграмм, чтобы определить максимальную пропускную способность канала. Ввели команду `qperf 192.168.1.1 udp_lat`. Результат представлен на рисунке 5.

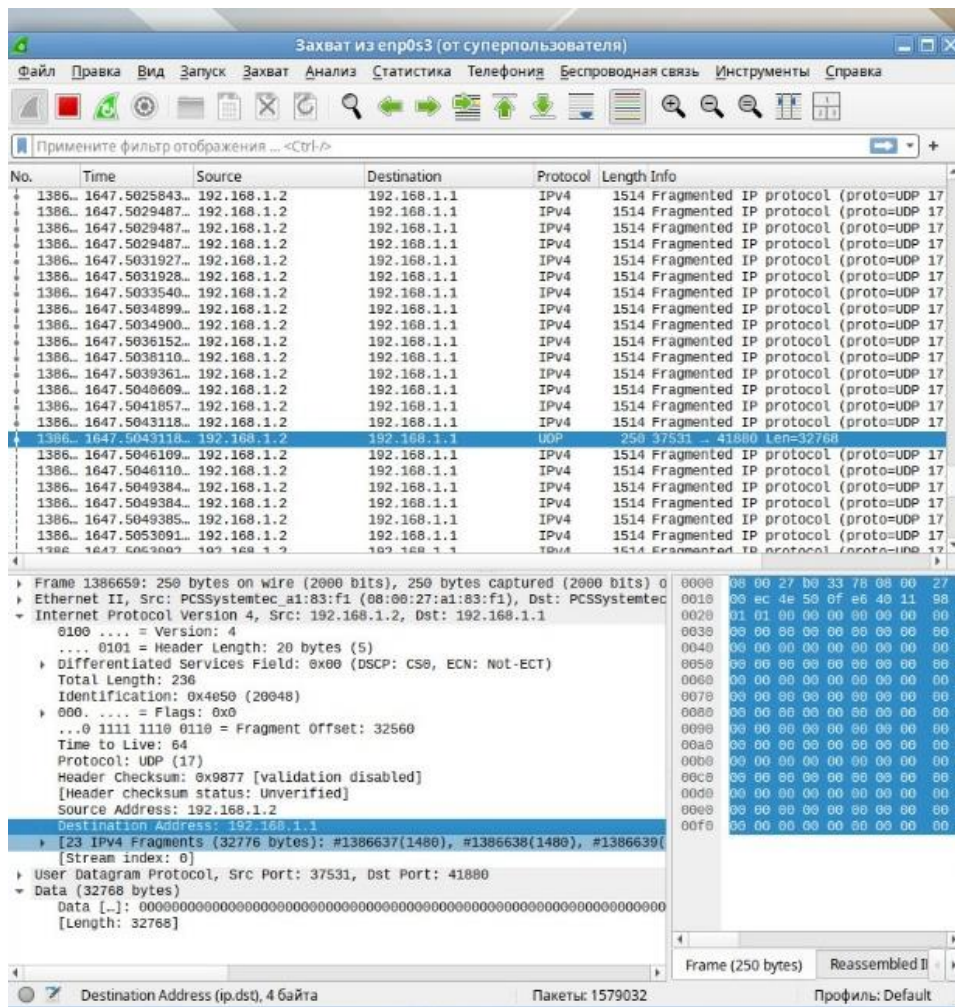


Рисунок 5 – Тестирование UDP-канала на пропускную способность

Далее командой `iperf 192.168.1.100 -t 30 tcp_bw tcp_lat` провели тест продолжительностью 30 секунд. На рисунке 6 представлен захват трафика во время проведения теста. В отличие от нагрузочных тестов, здесь наблюдается интенсивный обмен крайне малыми TCP-пакетами (55-69 байт).

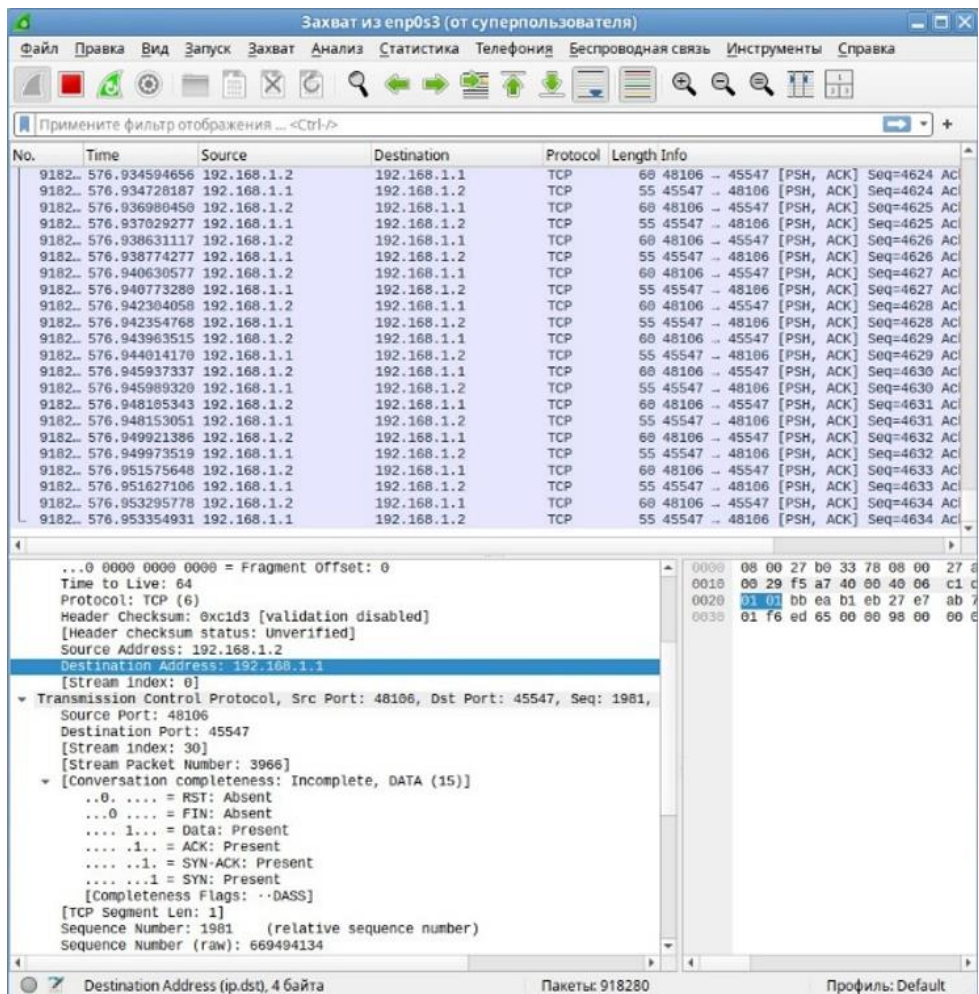


Рисунок 6 – Захват трафика во время проведения длительного теста

Далее установили другой генератор сетевого трафика iperf3.

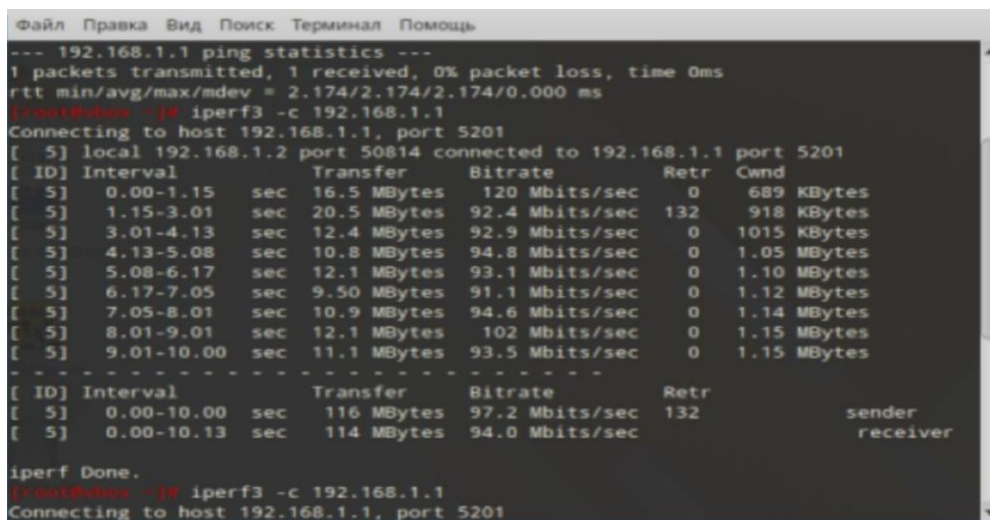


Рисунок 7 – Установленный iperf3

Iperf3. Для начала провели базовый тест на 10 секунд.

```

[root@ubun ~]# iperf3 -c 192.168.1.1
Connecting to host 192.168.1.1, port 5201
[ 5] local 192.168.1.2 port 49002 connected to 192.168.1.1 port 5201
[ ID] Interval           Transfer             Bitrate             Retr    Cwnd
[ 5]  0.00-1.00      sec   14.0 MBytes        117 Mb/s             0      583 KBytes
[ 5]  1.00-2.22      sec   13.4 MBytes        92.2 Mb/s            201     867 KBytes
[ 5]  2.22-3.07      sec   9.88 MBytes        97.7 Mb/s            0      948 KBytes
[ 5]  3.07-4.12      sec  10.8 MBytes        85.9 Mb/s            0     1.02 MBytes
[ 5]  4.12-5.02      sec  11.0 MBytes        102 Mb/s             117     783 KBytes
[ 5]  5.02-6.02      sec  10.9 MBytes        91.8 Mb/s            0      831 KBytes
[ 5]  6.02-7.58      sec  17.6 MBytes        94.8 Mb/s            9      612 KBytes
[ 5]  7.58-8.04      sec   5.50 MBytes        100 Mb/s             0      634 KBytes
[ 5]  8.04-9.04      sec  11.2 MBytes        94.1 Mb/s            0      683 KBytes
[ 5]  9.04-10.09     sec  11.0 MBytes        88.2 Mb/s            1     520 KBytes
-----
[ ID] Interval           Transfer             Bitrate             Retr
[ 5]  0.00-10.09     sec   115 MBytes        95.9 Mb/s            328
[ 5]  0.00-10.12     sec   113 MBytes        93.5 Mb/s
iperf Done.

```

Рисунок 8 – Клиент, базовый тест

No.	Time	Source	Destination	Protocol	Length	Info
17689	15.976147262	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168276558
17690	15.976307296	192.168.1.2	192.168.1.1	TCP	4434	49002 → 5201 [ACK] Seq=168276558 Acks=1
17692	15.976491331	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168286938
17693	15.976655512	192.168.1.2	192.168.1.1	TCP	4434	49002 → 5201 [ACK] Seq=168286938 Acks=1
17694	15.976794839	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168285318
17695	15.976951364	192.168.1.2	192.168.1.1	TCP	4434	49002 → 5201 [ACK] Seq=168285318 Acks=1
17696	15.977048675	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168296998
17697	15.9771322655	192.168.1.2	192.168.1.1	TCP	1514	49002 → 5201 [PSH, ACK] Seq=168296998 Acks=1
17698	15.977322889	192.168.1.2	192.168.1.1	TCP	2974	49002 → 5201 [ACK] Seq=168291159 Acks=1
17699	15.977518023	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168294678
17699	15.9771743604	192.168.1.2	192.168.1.1	TCP	4434	49002 → 5201 [ACK] Seq=168294678 Acks=1
17691	15.9771857280	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168298458
17692	15.9772015596	192.168.1.2	192.168.1.1	TCP	2974	49002 → 5201 [ACK] Seq=168298458 Acks=1
17693	15.9772192999	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168301376
17694	15.972464133	192.168.1.2	192.168.1.1	TCP	5894	49002 → 5201 [ACK] Seq=168301376 Acks=1
17695	15.972691428	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168307216
17696	15.973515519	192.168.1.2	192.168.1.1	TCP	11734	49002 → 5201 [ACK] Seq=168307216 Acks=1
17697	15.973790719	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168316896
17698	15.975711773	192.168.1.2	192.168.1.1	TCP	2974	49002 → 5201 [PSH, ACK] Seq=168318896 Acks=1
17699	15.975711806	192.168.1.2	192.168.1.1	TCP	11734	49002 → 5201 [ACK] Seq=168321816 Acks=1
17700	15.975914813	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168333496
17701	15.977446996	192.168.1.2	192.168.1.1	TCP	14654	49002 → 5201 [ACK] Seq=168333496 Acks=1
17702	15.977727683	192.168.1.1	192.168.1.2	TCP	54	5201 → 49002 [ACK] Seq=1 Acks=168348096

Рисунок 9 – Сервер, базовый тест (wireshark)

Далее же более длительный тест сети, на 30 секунд.

```

[root@ubun ~]# iperf3 -c 192.168.1.1 -t 30
Connecting to host 192.168.1.1, port 5201
[ 5] local 192.168.1.2 port 36840 connected to 192.168.1.1 port 5201
[ ID] Interval           Transfer             Bitrate             Retr    Cwnd
[ 5]  0.00-1.06      sec   14.6 MBytes        116 Mb/s             0      632 KBytes
[ 5]  1.06-2.24      sec   13.4 MBytes        94.5 Mb/s            0     1.29 MBytes
[ 5]  2.24-3.26      sec   11.1 MBytes        92.0 Mb/s            0     1.68 MBytes
[ 5]  3.26-4.18      sec   10.8 MBytes        97.8 Mb/s            0     1.72 MBytes
[ 5]  4.18-5.16      sec   9.88 MBytes        84.8 Mb/s            701     1.21 MBytes
[ 5]  5.16-6.00      sec   9.50 MBytes        94.6 Mb/s            0     1.33 MBytes
[ 5]  6.00-7.08      sec  12.4 MBytes        96.5 Mb/s            108     1.00 MBytes
[ 5]  7.08-8.07      sec  10.1 MBytes        85.8 Mb/s            115     697 KBytes
[ 5]  8.07-9.03      sec  11.1 MBytes        97.2 Mb/s            8      570 KBytes
[ 5]  9.03-10.00     sec   9.50 MBytes        81.8 Mb/s            0      606 KBytes
[ 5]  10.00-11.00    sec  12.1 MBytes        102 Mb/s             0      627 KBytes
[ 5]  11.00-12.04    sec  10.9 MBytes        87.7 Mb/s            0      639 KBytes
[ 5]  12.04-13.19    sec  12.2 MBytes        89.8 Mb/s            0      643 KBytes
[ 5]  13.19-14.02    sec   9.50 MBytes        95.7 Mb/s            0      643 KBytes
[ 5]  14.02-15.01    sec   9.38 MBytes        79.3 Mb/s            0      643 KBytes
[ 5]  15.01-16.83    sec  20.4 MBytes        94.0 Mb/s            0      656 KBytes
[ 5]  16.83-17.16    sec   4.00 MBytes        100 Mb/s            0      660 KBytes

```

Рисунок 10 – Клиент, тест на 30 секунд

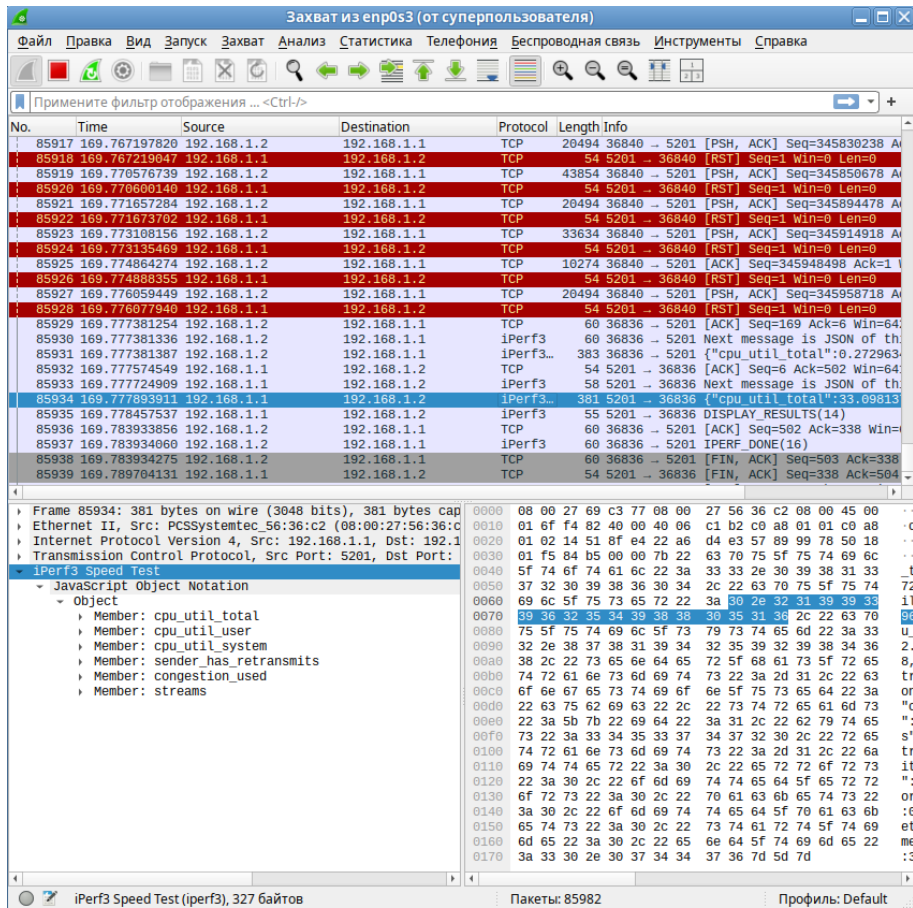


Рисунок 11 – Сервер, тест на 30 секунд (wireshark)

Поменяли местами роли, проведя тест в обратном направлении (сервер отправляет данные клиенту).

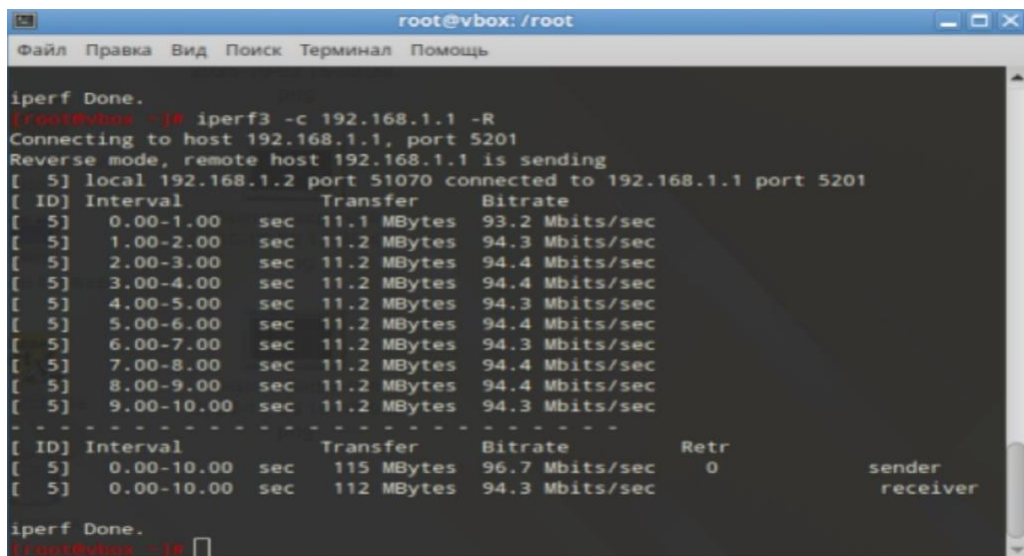


Рисунок 12 – Клиент, тест в обратном направлении

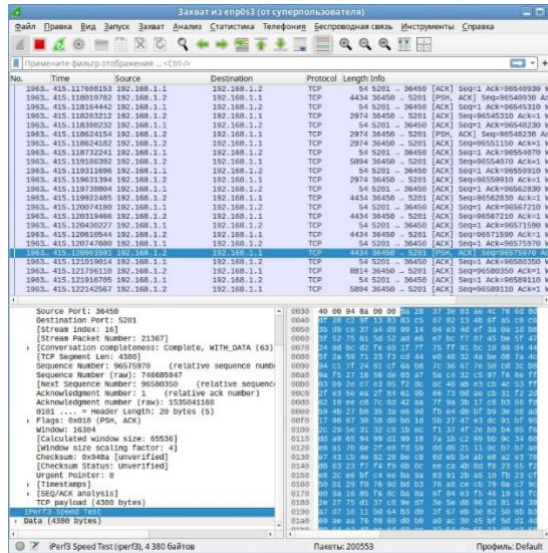


Рисунок 13 – Сервер, тест в обратном направлении (wireshark)

Далее попробовали провести тестирование на 60 секунд с параллельными потоками для ПОЛНОГО ИСПОЛЬЗОВАНИЯ КАНАЛА.

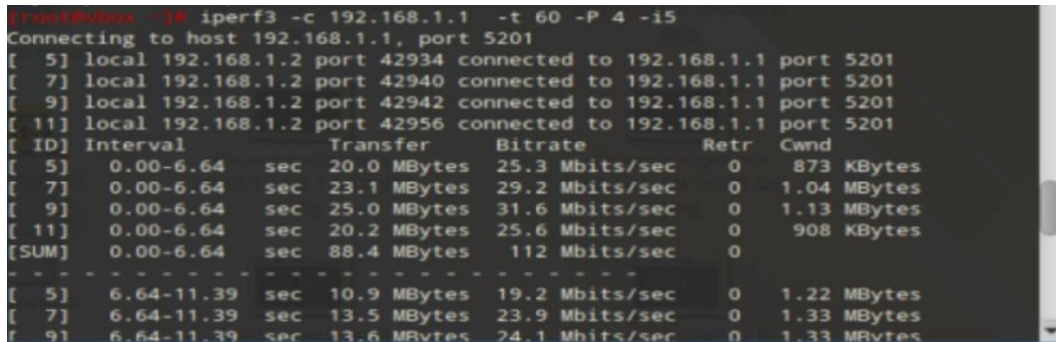


Рисунок 14 – Клиент, тест на 60 секунд с параллельными потоками

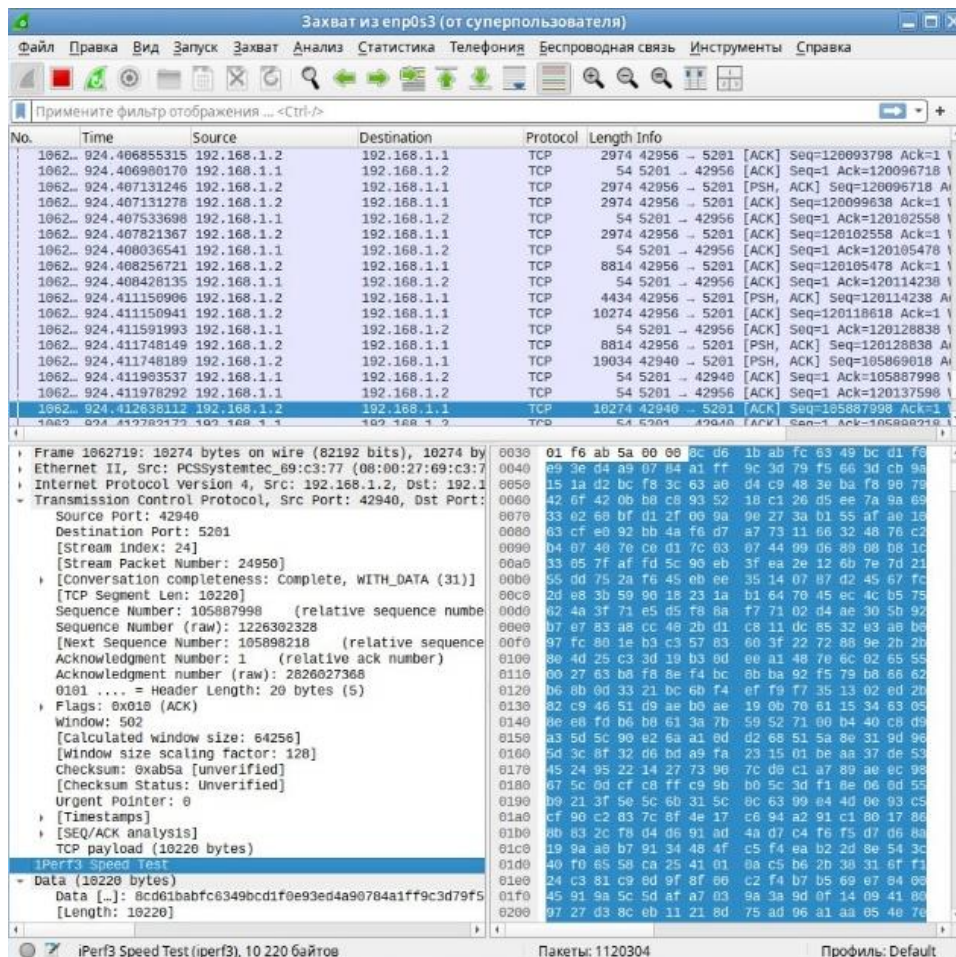


Рисунок 15 – Сервер, тест на 60 секунд с параллельными потоками (wireshark)

Также провели длительное стресс-тестирование сети на 300 секунд.

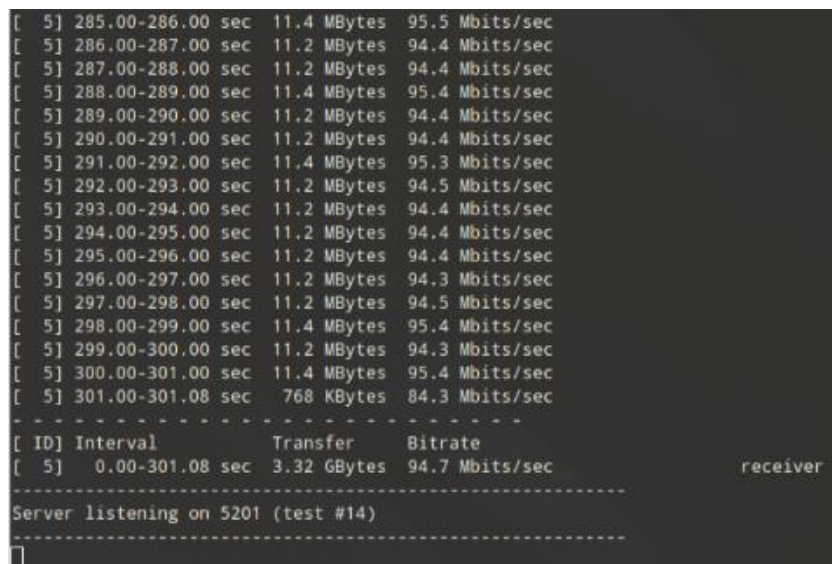


Рисунок 16 – Стресс-тест на 300 секунд

Проверили пропускную способность при изменении размера окна TCP.

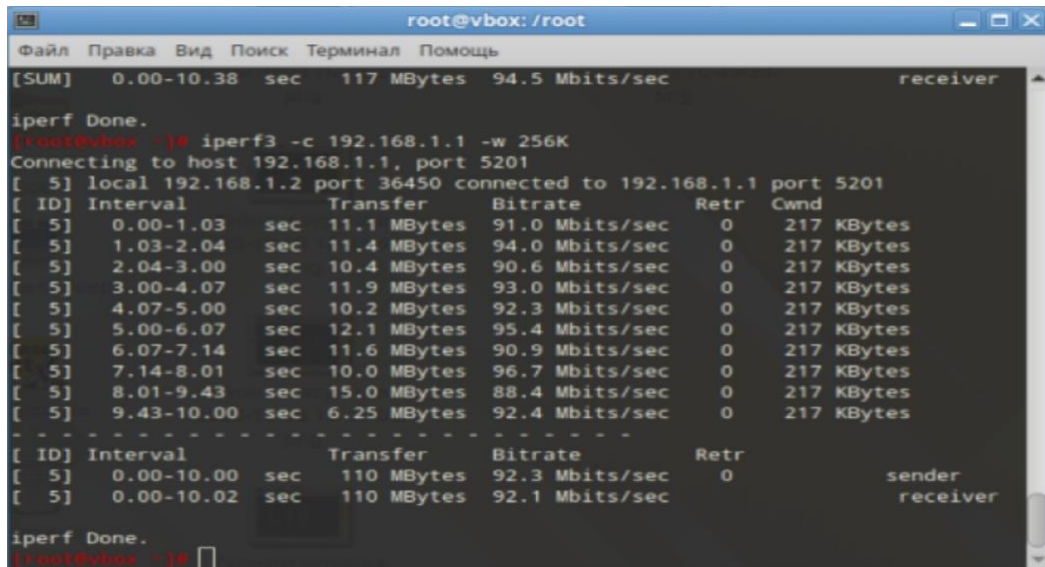


Рисунок 17 – Клиент, тест с изменением окна TCP

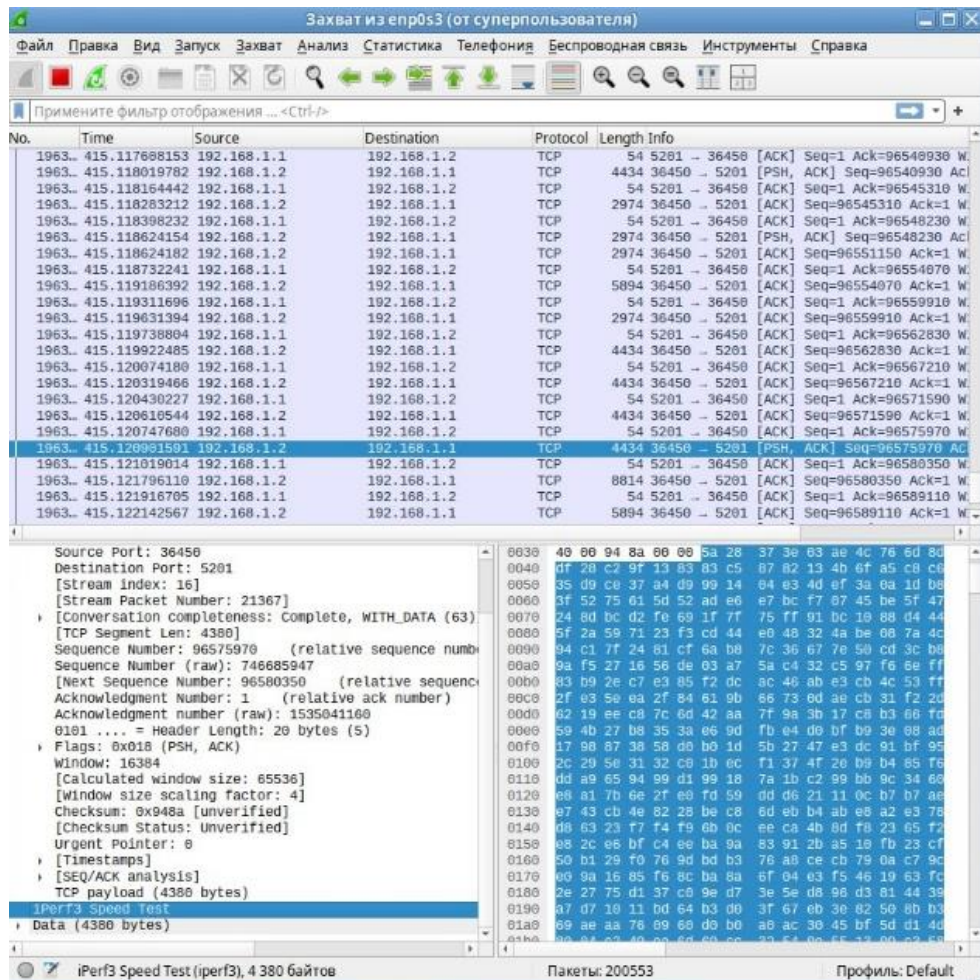


Рисунок 18 – Сервер, тест с изменением окна TCP (wireshark)

Далее провели тестирование сети с максимальной скоростью UDP-трафика. Результат представлен на рисунках 19-20.

```

root@vbox: /root
Файл Правка Вид Поиск Терминал Помощь

iperf Done.
[root@vbox ~]# iperf3 -c 192.168.1.1 -u -b 0
Connecting to host 192.168.1.1, port 5201
[ 5] local 192.168.1.2 port 58239 connected to 192.168.1.1 port 5201
[ ID] Interval            Transfer            Bitrate            Total Datagrams
[ 5] 0.00-1.24          sec 62.0 MBytes       420 Mbits/sec      44558
[ 5] 1.24-2.72          sec 74.7 MBytes       424 Mbits/sec      53641
[ 5] 2.72-3.35          sec 31.6 MBytes       422 Mbits/sec      22731
[ 5] 3.35-4.08          sec 36.4 MBytes       419 Mbits/sec      26174
[ 5] 4.08-6.11          sec 102 MBytes        420 Mbits/sec      73328
[ 5] 6.11-6.11          sec 2.85 KBytes       349 Mbits/sec      3
[ 5] 6.11-7.84          sec 85.7 MBytes       416 Mbits/sec      61527
[ 5] 7.84-8.14          sec 14.4 MBytes       409 Mbits/sec      10348
[ 5] 8.14-9.22          sec 53.5 MBytes       416 Mbits/sec      38442
^C[ 5] 10.55-44.23      sec 0.00 Bytes       0.00 bits/sec      0
-----
[ ID] Interval            Transfer            Bitrate            Jitter            Lost/Total Datagrams
[ 5] 0.00-44.23          sec 527 MBytes        100 Mbits/sec      0.000 ms          0/378650 (0%) sender
[ 5] 0.00-44.23          sec 0.00 Bytes        0.00 bits/sec      0.000 ms          0/0 (0%) receiver
iperf3: interrupt - the client has terminated by signal Interrupt(2)
[root@vbox ~]#

```

Рисунок 19 – Клиент, тест с максимальной скоростью UDP-трафика

```

Захват из epr0s3 (от суперпользователя)
Файл Правка Вид Запуск Захват Анализ Статистика Телефония Беспроводная связь Инструменты Справка
Примените фильтр отображения ... <Ctrl-F>
No. Time Source Destination Protocol Length Info
6071.. 763.807893221 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365809] Time sent=811,567
6071.. 763.807893276 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365810] Time sent=811,567
6071.. 763.807893300 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365811] Time sent=811,567
6071.. 763.807893324 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365812] Time sent=811,567
6071.. 763.807893348 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365813] Time sent=811,567
6071.. 763.807893372 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365814] Time sent=811,567
6071.. 763.807893396 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365815] Time sent=811,567
6071.. 763.807893421 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365816] Time sent=811,567
6071.. 763.808020490 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365817] Time sent=811,567
6071.. 763.808020568 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365818] Time sent=811,567
6071.. 763.809325488 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365819] Time sent=811,567
6071.. 763.809325535 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365820] Time sent=811,567
6071.. 763.809325559 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365821] Time sent=811,567
6071.. 763.809325583 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365822] Time sent=811,567
6071.. 763.809325608 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365823] Time sent=811,567
6071.. 763.809325632 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365824] Time sent=811,567
6071.. 763.809325656 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365825] Time sent=811,567
6071.. 763.809325680 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365826] Time sent=811,567
6071.. 763.809451349 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365827] Time sent=811,567
6071.. 763.809451402 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365828] Time sent=811,567
6071.. 763.810401866 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365829] Time sent=811,567
6071.. 763.810401899 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365830] Time sent=811,567
6071.. 763.810401917 192.168.1.2 192.168.1.1 UDP 1502 58239 - 5201 [365831] Time sent=811,567
-----
Frame 607163: 1502 bytes on wire (12016 bits), 1502 bytes captured (12016 bytes) on interface epr0s3
Ethernet II, Src: PCSystemtec_69:c3:77 (08:00:27:69:c3:77), Dst: 192.168.1.1 (08:00:27:69:c3:77)
Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 58239, Dst Port: 5201
Source Port: 58239
Destination Port: 5201
Length: 1468
Checksum: 0x5264 [unverified]
[Checksum Status: Unverified]
[Stream Index: 88]
[Stream Packet Number: 365718]
[Timestamps]
UDP payload (1460 bytes)
iperf3 sequence: 365826
Time Sent: 811,5674876 seconds
iperf3 sequence: 365826
Data [..]: e4e15260ddfa4453548d3e7638d5d5d22b970e4eb714 [Length: 1448]
-----
iperf3 Speed Test (iperf3), 1 460 байтов
Пакеты: 619915
Профиль: Default

```

Рисунок 20 – Сервер, тест с максимальной скоростью UDP-трафика (wireshark)

Nping. Провели стресс-тестирование TCP трафиком, использовали команду nping --tcp -p 80 --flags syn --rate 5000 --count 0 192.168.1.1.

```

root@vbox: /root
Файл Правка Вид Поиск Терминал Помощь
SENT (51.5034s) TCP 192.168.1.2:50256 > 192.168.1.1:80 A ttl=64 id=61177 iplen=40 seq=846530988 w
in=1480
SENT (51.5037s) TCP 192.168.1.2:50256 > 192.168.1.1:80 A ttl=64 id=61177 iplen=40 seq=846530988 w
in=1480
RCVD (51.5039s) TCP 192.168.1.1:80 > 192.168.1.2:50256 R ttl=64 id=0 iplen=40 seq=3242584494 win=
0
RCVD (51.5039s) TCP 192.168.1.1:80 > 192.168.1.2:50256 R ttl=64 id=0 iplen=40 seq=3242584494 win=
0
SENT (51.5039s) TCP 192.168.1.2:50256 > 192.168.1.1:80 A ttl=64 id=61177 iplen=40 seq=846530988 w
in=1480
RCVD (51.5041s) TCP 192.168.1.1:80 > 192.168.1.2:50256 R ttl=64 id=0 iplen=40 seq=3242584494 win=
0
SENT (51.5041s) TCP 192.168.1.2:50256 > 192.168.1.1:80 A ttl=64 id=61177 iplen=40 seq=846530988 w
in=1480
RCVD (51.5044s) TCP 192.168.1.1:80 > 192.168.1.2:50256 R ttl=64 id=0 iplen=40 seq=3242584494 win=
0
RCVD (51.5044s) TCP 192.168.1.1:80 > 192.168.1.2:50256 R ttl=64 id=0 iplen=40 seq=3242584494 win=
0
SENT (51.5044s) TCP 192.168.1.2:50256 > 192.168.1.1:80 A ttl=64 id=61177 iplen=40 seq=846530988 w
in=1480
SENT (51.5047s) TCP 192.168.1.2:50256 > 192.168.1.1:80 A ttl=64 id=61177 iplen=40 seq=846530988 w
in=1480
RCVD (51.5048s) TCP 192.168.1.1:80 > 192.168.1.2:50256 R ttl=64 id=0 iplen=40 seq=3242584494 win=
0
SENT (51.5048s) TCP 192.168.1.2:50256 > 192.168.1.1:80 A ttl=64 id=61177 iplen=40 seq=846530988 w
in=1480^C
Max rtt: 48.890ms | Min rtt: 0.004ms | Avg rtt: 0.004ms
Raw packets sent: 286984 (11.479MB) | Rcvd: 282725 (13.005MB) | Lost: 4259 (1.48%)
Nping done: 1 IP address pinged in 51.51 seconds

```

Рисунок 21 – Клиент, тест

The screenshot shows a Wireshark interface with a packet list and a packet details pane. The packet list shows a series of TCP segments from 192.168.1.2 to 192.168.1.1. The details pane for packet 3266734 shows a TCP segment with the following fields:

- Source Port: 50256
- Destination Port: 80
- [Stream index: 124843]
- [Stream Packet Number: 558413]
- [Conversation completeness: Incomplete (36)]
- [TCP Segment Len: 0]
- Sequence Number: 1 (relative sequence number)
- Sequence Number (raw): 846530988
- [Next Sequence Number: 1 (relative sequence number)]
- Acknowledgment Number: 1 (relative ack number)
- Acknowledgment number (raw): 3242584494
- 0101 .... = Header Length: 20 bytes (5)
- Flags: 0x010 (ACK)
- Window: 1480
- [Calculated window size: 1480]
- [Window size scaling factor: -1 (unknown)]
- Checksum: 0x7b02 [unverified]
- [Checksum status: unverified]
- Urgent Pointer: 0
- [Timestamps]

Рисунок 22 – Сервер, стресс-тестирование TCP трафиком

Далее сменили флаг на ack, использовали команду `nping --tcp -p 80 --flags ack --rate 500 --count 10000 192.168.1.1`.

```

root@vbox: /root
RCVD (21.2297s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
SENT (21.2305s) TCP 192.168.1.2:32845 > 192.168.1.1:80 S ttl=64 id=9002 iplen=40 seq=1957393403 w
In=1480
RCVD (21.2317s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
SENT (21.2326s) TCP 192.168.1.2:32845 > 192.168.1.1:80 S ttl=64 id=9002 iplen=40 seq=1957393403 w
In=1480
RCVD (21.2338s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
SENT (21.2348s) TCP 192.168.1.2:32845 > 192.168.1.1:80 S ttl=64 id=9002 iplen=40 seq=1957393403 w
In=1480
RCVD (21.2358s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
SENT (21.2375s) TCP 192.168.1.2:32845 > 192.168.1.1:80 S ttl=64 id=9002 iplen=40 seq=1957393403 w
In=1480
RCVD (21.2387s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
SENT (21.2397s) TCP 192.168.1.2:32845 > 192.168.1.1:80 S ttl=64 id=9002 iplen=40 seq=1957393403 w
In=1480
RCVD (21.2409s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
SENT (21.2417s) TCP 192.168.1.2:32845 > 192.168.1.1:80 S ttl=64 id=9002 iplen=40 seq=1957393403 w
In=1480
RCVD (21.2428s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
SENT (21.2437s) TCP 192.168.1.2:32845 > 192.168.1.1:80 S ttl=64 id=9002 iplen=40 seq=1957393403 w
In=1480
RCVD (21.2455s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
SENT (21.2452s) TCP 192.168.1.2:32845 > 192.168.1.1:80 S ttl=64 id=9002 iplen=40 seq=1957393403 w
In=1480
RCVD (21.2469s) TCP 192.168.1.1:80 > 192.168.1.2:32845 RA ttl=64 id=0 iplen=40 seq=0 win=0
Max rtt: 5.658ms | Min rtt: 0.011ms | Avg rtt: 0.325ms
Raw packets sent: 10000 (400.000KB) | Rcvd: 10000 (460.000KB) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 21.29 seconds

```

Рисунок 23 – Клиент, тест с измененным на ask флагом

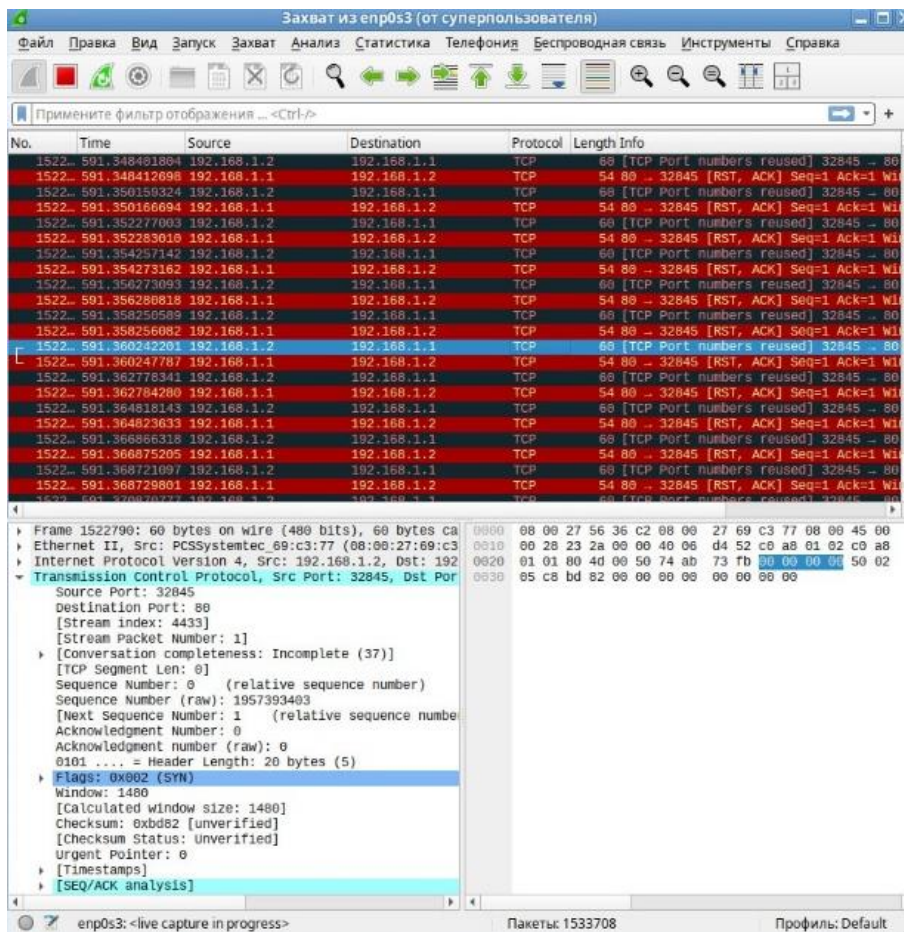


Рисунок 24 – Сервер, тест с измененным на ask флагом (wireshark)

Так как Nping работает и на L3, попробуем генерировать напрямую пакеты, используя команду `nping --icmp --rate 1000 --count 1000 192.168.1.1`.

```

root@vbox: /root
Файл Правка Вид Поиск Терминал Помощь
ttl=64 id=8486 iplen=1052 ]
SENT (26.2244s) ICMP [192.168.1.2 > 192.168.1.1 Echo request (type=8/code=0) id=8247 seq=25136] IP [
  [ttl=64 id=60939 iplen=1052 ]
RCVD (26.2251s) ICMP [192.168.1.1 > 192.168.1.2 Echo reply (type=0/code=0) id=8247 seq=25135] IP [
  [ttl=64 id=8487 iplen=1052 ]
SENT (26.2254s) ICMP [192.168.1.2 > 192.168.1.1 Echo request (type=8/code=0) id=8247 seq=25137] IP [
  [ttl=64 id=60939 iplen=1052 ]
RCVD (26.2260s) ICMP [192.168.1.1 > 192.168.1.2 Echo reply (type=0/code=0) id=8247 seq=25136] IP [
  [ttl=64 id=8488 iplen=1052 ]
SENT (26.2264s) ICMP [192.168.1.2 > 192.168.1.1 Echo request (type=8/code=0) id=8247 seq=25138] IP [
  [ttl=64 id=60939 iplen=1052 ]
RCVD (26.2272s) ICMP [192.168.1.1 > 192.168.1.2 Echo reply (type=0/code=0) id=8247 seq=25137] IP [
  [ttl=64 id=8491 iplen=1052 ]
SENT (26.2274s) ICMP [192.168.1.2 > 192.168.1.1 Echo request (type=8/code=0) id=8247 seq=25139] IP [
  [ttl=64 id=60939 iplen=1052 ]
RCVD (26.2282s) ICMP [192.168.1.1 > 192.168.1.2 Echo reply (type=0/code=0) id=8247 seq=25138] IP [
  [ttl=64 id=8492 iplen=1052 ]
SENT (26.2284s) ICMP [192.168.1.2 > 192.168.1.1 Echo request (type=8/code=0) id=8247 seq=25139] IP [
  [ttl=64 id=60939 iplen=1052 ]
RCVD (26.2292s) ICMP [192.168.1.1 > 192.168.1.2 Echo reply (type=0/code=0) id=8247 seq=25139] IP [
  [ttl=64 id=8493 iplen=1052 ]
SENT (26.2295s) ICMP [192.168.1.2 > 192.168.1.1 Echo request (type=8/code=0) id=8247 seq=25141] IP [
  [ttl=64 id=60939 iplen=1052 ]
RCVD (26.2303s) ICMP [192.168.1.1 > 192.168.1.2 Echo reply (type=0/code=0) id=8247 seq=25139] IP [
  [ttl=64 id=8497 iplen=1052 ]
^C
Max rtt: 1.812ms | Min rtt: 0.007ms | Avg rtt: 0.014ms
Raw packets sent: 25140 (26.447MB) | Rcvd: 25139 (26.446MB) | Lost: 1 (0.00%)
Nping done: 1 IP address pinged in 26.23 seconds

```

Рисунок 25 – Клиент, генерация пакетов

The screenshot shows a Wireshark capture of network traffic. The main window displays a list of captured packets, with the following columns: No., Time, Source, Destination, Protocol, Length, and Info. The packets are ICMP Echo (ping) requests and replies between 192.168.1.1 and 192.168.1.2. The details pane for the selected packet (No. 56041) shows the following information:

- Internet Control Message Protocol
  - Type: 8 (Echo (ping) request)
  - Code: 0
  - Checksum: 0x9045 [correct]
  - [Checksum Status: Good]
  - Identifier (BE): 18857 (0x49a9)
  - Identifier (LE): 43337 (0xa949)
  - Sequence Number (BE): 7697 (0x1e11)
  - Sequence Number (LE): 4382 (0x111e)
  - [Response frame: 56042]

The packet bytes pane shows the raw data of the selected packet, including the Ethernet II header, Internet Protocol Version 4 header, and the ICMP Echo (ping) request data.

Рисунок 26 – Сервер, тест с генерацией пакетов (wireshark)

## Заключение

Таблица 2 – Результаты тестирования пропускной способности

Iperf		Nping		Qperf	
Время тестирования, с	Пропускная способность, МБит/с	Время тестирования, с	Пропускная способность, МБит/с	Время тестирования, с	Пропускная способность, МБит/с
10	97	10	2,3	10	86
60	96	60	2,0	60	83
300	93	90	1,8	300	81
Стресс-тестирование сети					
+		-		+	

В ходе исследования были рассмотрены и применены инструменты для генерации сетевого трафика: Nping, Iperf3 и Qperf. Эксперименты по развертыванию и функциональному тестированию подтвердили их эффективность для оценки пропускной способности как в сегментах LAN, так и в условиях эмуляции WAN-соединений. Nping продемонстрировал возможности по формированию низкоуровневого трафика и проверке доступности, Iperf3 зарекомендовал себя как универсальное и мощное средство для измерения пропускной способности TCP/UDP, а qperf предоставил детальную метрику задержек и производительности.

Выдвинутая гипотеза подтвердилась: наилучший генератор для тестирования пропускной способности – Iperf3 и Qperf, потому что, они обладают высокой мощностью, что подходит для стресс-тестирования, а Nping не подходит, потому что он обладает недостаточно высокой скоростью генерируемого трафика.

Таким образом, сравнительный анализ рассмотренных генераторов трафика и их применение является необходимым этапом для валидации конфигурации сети, планирования её масштабирования и обеспечения отказоустойчивости.

## Литература

1. WAN/LAN Network Traffic Generators for Bandwidth & Stress Testing  
URL: <https://www.netadmintools.com/wan-and-lan-network-traffic-generators/>
2. Утилита iPerf: тестирование пропускной способности и скорости сети  
URL: <https://winitpro.ru/index.php/2014/11/05/testirovanie-propusknoj-sposobnosti-seti-s-iperf/>
3. Инструмент оценки производительности сети Linux (IPERF, QPERF)  
URL: <https://russianblogs.com/article/66693950724/>
4. Введение в iperf URL: <https://yamadharm.github.io/ru/teaching/simulation-networks/lab/mininet-iperf-intro/>
5. Утилита nping: тестирование пропускной способности и скорости сети  
URL: <https://infosecportal.ru/soft/nping/>
6. Стресс-тестирование URL: <https://serverspace.ru/support/glossary/stress-testirovanie/>
7. Функциональное тестирование URL: <https://ibs-qa.ru/media/chto-takoe-funktsionalnoe-testirovanie/>
8. Qperf - man pages section 1: User commands URL: [https://docs.oracle.com/cd/E88353\\_01/html/E37839/qperf-1.html](https://docs.oracle.com/cd/E88353_01/html/E37839/qperf-1.html)
9. НОРМИРОВАНИЕ ПОКАЗАТЕЛЕЙ ПРИ ОРГАНИЗАЦИИ КИБЕРПОЛИГОНА ПО СЕТЕВЫМ ТЕХНОЛОГИЯМ Уймин А.Г. В книге: Нефть и газ - 2024. Тезисы докладов 78-ой Международной молодежной научной конференции. Москва, 2024. С. 1253-1254.