Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №4**

**«Анализ трафика компьютерных сетей с помощью утилиты Wireshark»**

по дисциплине “Компьютерные сети”

Выполнили:

Студенты группы P3334

Баянов Р. Д.

Преподаватель:

Алиев Т. И.

Санкт-Петербург

2025 г.

**Содержание**

[**Задание** 3](#_Toc195369253)

[**Вариант** 4](#_Toc195369254)

[**Анализ трафика утилиты ping** 5](#_Toc195369255)

[**Анализа трафика утилиты tracert (traceroute)** 6](#_Toc195369256)

[**Анализ HTTP-трафика** 7](#_Toc195369257)

[**Анализ ARP-трафика** 8](#_Toc195369258)

[**Вывод** 9](#_Toc195369259)

**Задание**

Цель работы – изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

В процессе выполнения домашнего задания выполняются наблюдения за передаваемым трафиком с компьютера пользователя в Интернет и в обратном направлении. Применение специализированной утилиты Wireshark позволяет наблюдать структуру передаваемых кадров, пакетов и сегментов данных различных сетевых протоколов. При выполнении УИР рекомендуется выполнить анализ последовательности команд и определить назначение служебных данных, используемых для организации обмена данными в протоколах: ARP, DNS, FTP, HTTP, DHCP.

**Вариант**

Для выполнения лабораторной работы будут представлены пункты 4.1, 4.2, 4.3, 4.5.

Сайт для анализа трафика – brd.ru

**Анализ трафика утилиты ping**

В командной строке поочерёдно с увеличением размера будем отправлять пакеты через утилиту ping на сайт *brd.ru.* Формат команды:

***ping -l <размер пакета> -n <кол-во пакетов> brd.ru***

Опция -n нужна для того, чтобы отправлять один пакет, так как сама по себе утилита по умолчанию отправляет 4 пакета.

Применим данную команду несколько раз для случаев, когда размер у пакета будет 100, 200, 500, 1000, 2000, 5000, 10000 байт.

Для начала опишем структуру пакета. Утилита управляет ICMP запросами и ICMP ответами. Структура:

1. Канальный уровень – Ethernet 2

Заголовок содержит:

* + Destination MAC address – MAC адрес получателя.
  + Source MAC address – MAC-адрес отправителя.
  + Type – поле типа протокола.

1. Сетевой уровень – IP-заголовок

Заголовок содержит:

* Version
* Header Length
* Identification – идентификатор фрагмента
* Protocol – тип вложенного протокола
* Flags – указывается DF и MF
* TTL – ограничение на кол-во хопов
* Fragment offset – смещение фрагмента (если пакет был фрагментирован)
* Header Checksum – контрольная сумма заголовка
* Source IP address
* Destination IP address

1. Сетевой протокол ICMP

* Type – request или reply
* Checksum – контрольная сумма ICMP-пакета
* Identifier – уникальный ID запроса
* Seq number – номер последовательности запроса

1. Поле данных (Payload)

Примерно так выглядит структура пакета ICMP:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

**Ответы на вопросы:**

1. Имеет ли место фрагментации исходного пакета, какое поле на это указывает?

Фрагментация происходит, какой размер IP-пакета превышает MTU (maximum transmission unit) (обычно 1480 байт для Ethernet). Признаком фрагментации служат:

* Флаг MF (More Fragments) в IP-заголовке
* Поле Fragment Offset (смещение фрагмента)

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Заметим, что когда пакет фрагментируется, то часть данных отправляется вместе с ICMP заголовком, а остальные фрагменты чисто по протоколу IP.

1. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?

* MF = 1 – промежуточный фрагмент.
* MF = 0 – последний фрагмент.



Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

1. Чему равно количество фрагментов при передаче ping-пакетов?

Учитывая, что один фрагмент по MTU равен примерно 1480 байт.

То кол-во фрагментов будет равно *размер пакета / 1480* и округлить до верхнего целого числа.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. График: размер пакета – кол-во фрагментов.
2. Как изменить поле TTL с помощью утилиты ping?

Изменить это поле можно командой: *ping -l 3000 -n 1 -i 5 brd.ru*



1. Что содержится в поле данных ping-пакета?

* Заголовок ICMP
* Идентификатор
* Номер последовательности
* Содержимое

**Анализа трафика утилиты tracert (traceroute)**

Введём в командную строку команду *tracert brd.ru*

Данная утилита также пользуется протоколом ICMP, поэтому разбирать его структуру мы не будем. Но, помимо этого, утилита tracert отправляет DNS-пакеты. DNS – это протокол, который переводит доменные имена в IP-адреса, которые понятны компьютерам. С помощью ключа -d можно сделать так, чтобы DNS пакеты отправлялись уже после построения маршрута, так как они не несут в себе важный функционал.

Вот структура DNS пакета:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

Заголовок размером 12 байт содержит:

* ID – уникальный ID-запроса
* Flags – ошибки, авторитетность, тип запроса/ответа.
* QDCOUNT – кол-во запросов
* ANCOUNT – кол-во ответов
* INSCOUNT – кол-во записей авторитетных серверов
* ARCOUNT – кол-во дополнительных записей

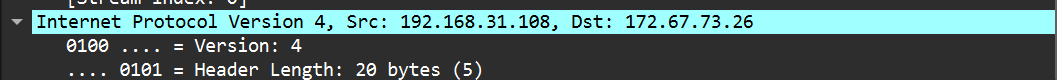
Раздел вопросов, который содержит доменное имя, которое мы запрашиваем.

Раздел ответов, который содержит IP-адрес в ответ на запрос.

**Ответы на вопросы:**

1. Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?

Заголовок IP обычно составляет 20 байт для IPv4.



Поле данных – это содержимое, инкапсулированное в IP-пакете, ICMP-пакета.

У ICMP заголовок равен 8 байт, а сами данные 64 байта. Следовательно, данные 72 байта. IP-заголовок 20 байт.

1. Как и почему изменяется поле TTL в следующих ICMP-пакетах tracert?

Утилита tracert посылает ICMP-пакеты с увеличивающимся TTL, начиная с 1.

Каждый маршрутизатор уменьшает TTL на 1. Когда TTL становится 0 – маршрутизатор отбрасывает пакет и отправляет обратно ICMP Time Exceeded.

Это позволяет tracert определить каждый узел на пути.

TTL изменяется поэтапно, чтобы каждый узел по очереди откликнулся, и таким образом строится маршрут.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

1. Чем отличаются ICMP-пакеты, генерируемые tracert, от ICMP-пакетов ping?

ping всегда шлёт ICMP Echo Request и ждёт Echo Reply.

Tracert использует ICMP Echo Request с разным TTL и анализирует:

* ICMP Time Excedeed от промежуточных маршрутизаторов.
* ICMP Echo Reply от конечного узла.

То есть ping проверяет доступность узла, а tracert строит маршрут до него.

1. Чем отличаются ICMP reply от ICMP error и зачем нужны оба?

ICMP reply – отклик от целевого хоста, подтверждающий, что он доступен.

ICMP error – приходит от маршрутизаторов, когда TTL истекает. Эти пакеты нужны для определения маршрута.

Оба типа позволяют tracert:

* Узнать IP каждого промежуточного маршрутизатора (через error).
* Подтвердить достижение конечного узла (через reply).

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. Что изменится в работе tracert, если убрать ключ -d? Какой трафик будет генерироваться дополнительно?

Ключ -d отключает обратное разрешение IP-адресов в доменные имена.

Без -d tracert будет пытаться разрешить IP-адреса в имена хостов (через DNS). Это приведёт к дополнительному DNS-трафику, так как каждый IP будет запрашиваться у DNS-сервера.

Вот окно wireshark без ключа -d:

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Вот окно wireshark с ключом -d:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

**Анализ HTTP-трафика**

Запустим анализ в wireshark и перейдём на сайт brd.ru. К сожалению, сайт, который подходит нам по варианту, не обладает возможностью принимать условные GET-запросы. Сколько раз не обновляй мы не можем получить ответ 304. Поэтому воспользуемся сайтом, который точно обладает такой возможностью, а именно сайтом *example.com.* Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Сначала просто зайдём на сайт example.com.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Заметим, что всё отработало как надо, мы получаем ответ 200.

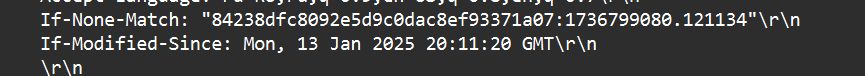
Теперь попробуем обновить страницу и посмотрим, что будет.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

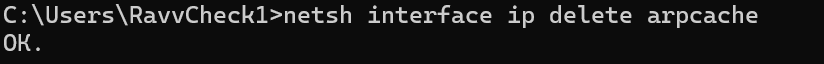


Заметим, что мы получаем совсем другую ситуацию. Здесь у нас получилось отправить условный GET-запрос. И мы получаем ответ 304 от сервера. Это можно понять по появившимся полям Last-Modified и If-Modified-Since.

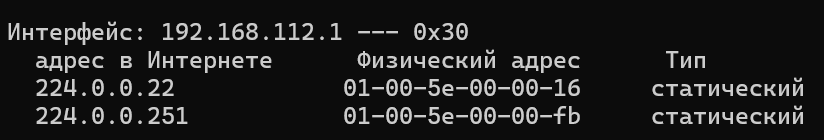
**Анализ ARP-трафика**

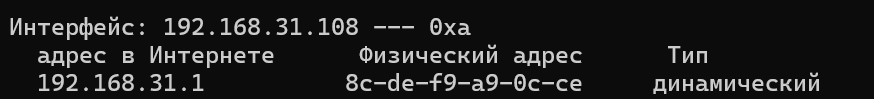
Для начала очистим ARP-таблицу с помощью команды:

*netsh interface ip delete arpcache*



Получим вот такую arp-таблицу:



После удаления кэша браузера отправимся на сайт brd.ru и увидим новую запись, в arp-таблице. 

Заметим, что это вообще не похоже на IP адрес сайта, на который мы перешли. А всё, потому что MAC-адреса используются только в локальной сети. Мы не сможем увидеть ARP-запрос, который узнаёт MAC-адрес нашего сайта, так как его и вовсе нет. Но мы видим IP-адрес 192.168.31.1. Вероятнее всего это IP нашего маршрутизатора, который как раз таки и взялся в дальнейшем уже за поиск того сайта, на который мы перешли.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**Ответы на вопросы:**

* 1. Какие МАС-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют?

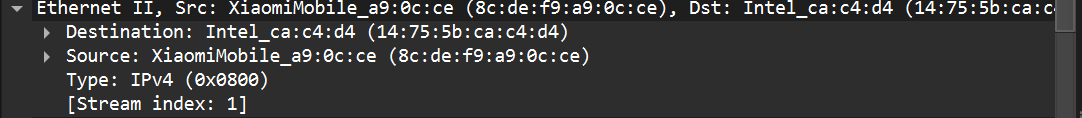
В ARP-пакетах мы увидим два типа MAC-адресов:

* MAC-адрес отправителя запроса – адрес нашего компьютера. Он используется в поле Sender MAC-address
* MAC-адрес искомого устройства:

В ARP-запросе (who-has) поле Target MAC Address будет заполнено нулями, потому что он ещё известен.

В ARP-ответе (is-at) это будет MAC-адрес шлюза/маршрутизатора, провайдера или другого узла локальной сети, связанного с IP, на который отправляется запрос.

* 1. Какие МАС-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Какие устройства они идентифицируют?

HTTP работает поверх TCP/IP и Ethernet. В Ethernet-заголовке каждого HTTP-пакета указывается:

* MAC-адрес источника – это MAC-адрес твоего компьютера
* MAC-адрес назначения – это обычно MAC-адрес ближайшего маршрутизатора/шлюза, через который трафик пойдёт в Интернет.

MAC-адреса веб-сайта, на который мы заходим, мы не увидим, потому что MAC-адреса используются только внутри локальной сети.

* 1. Для чего ARP-запроса содержит IP-адрес источника?

ARP-запроса содержит IP-адрес источника, чтобы:

* Получатель запроса (тот, чей IP адрес запрашивается) мог записать в свою ARP-таблицу соответствие, и тем самым сократить количество ARP-запросов в будущем.
* Получатель понимал, кто запрашивает – это нужно для формирования ARP-запроса-ответа.

IP-адрес источника нужен для обратной связи и корректного построения локальной маршрутизации.

**Вывод**

Выполнив данную лабораторную работу, я с помощью программы wireshark проанализировал передачу пакетов по сети. Мне удалось описать структуры DNS, ICMP, IP, ARP и HTTP протоколов. Выяснил, что передача по сети на самом деле очень сложный механизм, который включает в себя взаимодействие огромного количества протоколов и интерфейсов.