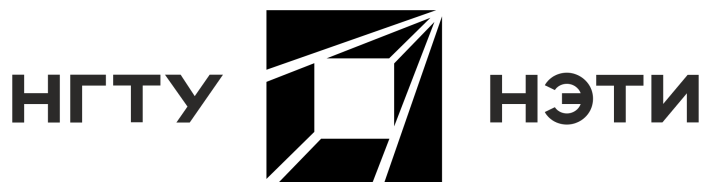


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Кафедра систем сбора и обработки данных



ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине: Сетевые информационные технологии

на тему: Протоколы стека TCP/IP

Вариант №3

Факультет: ФПМИ

Группа: ПММ-21

Выполнили: Сухих А.С., Черненко Д.А.

Проверил: к.т.н., доцент Кобылянский В.Г.

Дата выполнения: 25.10.22

Отметка о защите:

Новосибирск 2022

Цель работы: Изучение структуры передаваемых по сети кадров и пакетов, работающих на канальном и сетевом уровне.

Ход работы:

1. Запустить перехват пакетов в Wireshark.

Через «Захват» и «Опции» были захвачены различные пакеты данных: DNS и UDP, а также пакеты TCP с ошибками получения:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.3	77.88.55.70	TCP	54	57968 → 443 [FIN,
2	0.000262	192.168.0.3	87.250.251.119	TCP	54	57969 → 443 [FIN,
3	0.057482	87.250.251.119	192.168.0.3	TLSv1.2	93	Application Data
4	0.057530	192.168.0.3	87.250.251.119	TCP	54	57969 → 443 [RST,
5	0.058160	87.250.251.119	192.168.0.3	TLSv1.2	78	Application Data
6	0.058160	87.250.251.119	192.168.0.3	TCP	54	443 → 57969 [FIN,
7	0.067680	77.88.55.70	192.168.0.3	TLSv1.2	93	Application Data
8	0.067732	192.168.0.3	77.88.55.70	TCP	54	57968 → 443 [RST,
9	0.068414	77.88.55.70	192.168.0.3	TLSv1.2	78	Application Data
10	0.068414	77.88.55.70	192.168.0.3	TCP	54	443 → 57968 [FIN,

Рисунок 1.1 — захват пакетов данных

2. Определить с помощью утилиты ping доступность заданных узлов sklad-service.ru, eye.moof.ru, gmail.com, wiw.ru, luminator.ru, hotlog.ru.

```
C:\Users\Данил>ping sklad-service.ru

Обмен пакетами с sklad-service.ru [91.189.114.22] с 32 байтами данных:
Ответ от 91.189.114.22: число байт=32 время=45мс TTL=57
Ответ от 91.189.114.22: число байт=32 время=51мс TTL=57
Ответ от 91.189.114.22: число байт=32 время=45мс TTL=57
Ответ от 91.189.114.22: число байт=32 время=51мс TTL=57

Статистика Ping для 91.189.114.22:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 45мсек, Максимальное = 51 мсек, Среднее = 48 мсек

C:\Users\Данил>_
```

Рисунок 2.1 — обмен с sklad-service.ru данными

```
C:\Users\Данил>ping eye.moof.ru

Обмен пакетами с eye.moof.ru [90.156.201.70] с 32 байтами данных:
Ответ от 90.156.201.70: число байт=32 время=46мс TTL=56
Ответ от 90.156.201.70: число байт=32 время=51мс TTL=56
Ответ от 90.156.201.70: число байт=32 время=50мс TTL=56
Ответ от 90.156.201.70: число байт=32 время=49мс TTL=56

Статистика Ping для 90.156.201.70:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 46мсек, Максимальное = 51 мсек, Среднее = 49 мсек

C:\Users\Данил>
```

Рисунок 2.2 — обмен с eye.moof.ru данными

```
C:\Users\Данил>ping gmail.com

Обмен пакетами с gmail.com [64.233.165.18] с 32 байтами данных:
Ответ от 64.233.165.18: число байт=32 время=65мс TTL=58
Ответ от 64.233.165.18: число байт=32 время=68мс TTL=58
Ответ от 64.233.165.18: число байт=32 время=102мс TTL=58
Ответ от 64.233.165.18: число байт=32 время=65мс TTL=58

Статистика Ping для 64.233.165.18:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 65мсек, Максимальное = 102 мсек, Среднее = 75 мсек

C:\Users\Данил>_
```

Рисунок 2.3 — обмен с gmail.com данными

```
C:\Users\Данил>ping wiw.ru

Обмен пакетами с wiw.ru [89.208.206.225] с 32 байтами данных:
Ответ от 89.208.206.225: число байт=32 время=46мс TTL=55
Ответ от 89.208.206.225: число байт=32 время=56мс TTL=55
Ответ от 89.208.206.225: число байт=32 время=51мс TTL=55
Ответ от 89.208.206.225: число байт=32 время=53мс TTL=55

Статистика Ping для 89.208.206.225:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 46мсек, Максимальное = 56 мсек, Среднее = 51 мсек

C:\Users\Данил>
```

Рисунок 2.4 — обмен с wiw.ru данными

```

C:\Users\Данил>ping luminator.ru

Обмен пакетами с luminator.ru [90.156.201.32] с 32 байтами данных:
Ответ от 90.156.201.32: число байт=32 время=48мс TTL=56
Ответ от 90.156.201.32: число байт=32 время=50мс TTL=56
Ответ от 90.156.201.32: число байт=32 время=51мс TTL=56
Ответ от 90.156.201.32: число байт=32 время=51мс TTL=56

Статистика Ping для 90.156.201.32:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 48мсек, Максимальное = 51 мсек, Среднее = 50 мсек

C:\Users\Данил>

```

Рисунок 2.5 — обмен с luminator.ru данными

```

C:\Users\Данил>ping hotlog.ru

Обмен пакетами с hotlog.ru [89.208.236.251] с 32 байтами данных:
Ответ от 89.208.236.251: число байт=32 время=50мс TTL=55
Ответ от 89.208.236.251: число байт=32 время=54мс TTL=55
Ответ от 89.208.236.251: число байт=32 время=49мс TTL=55
Ответ от 89.208.236.251: число байт=32 время=51мс TTL=55

Статистика Ping для 89.208.236.251:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 49мсек, Максимальное = 54 мсек, Среднее = 51 мсек

C:\Users\Данил>

```

Рисунок 2.6 — обмен с hotlog.ru данными

Трассировка узла:

```
[andrew@manpc ~]$ traceroute gmail.com
```

```

1?: [LOCALHOST] pmtu 1500
1: _gateway 0.345ms
1: _gateway 0.308ms
2: 137-192-51-254.novotelecom.ru 3.069ms
3: 10.245.138.241 2.246ms
4: 10.245.138.242 1.915ms asymm 5
5: 149-128-50.novotelecom.ru 2.529ms asymm 6
6: bbr03.spb.ertelecom.ru 43.639ms asymm 12
7: net131.234.188-159.ertelecom.ru 43.714ms asymm 11
8: no reply

```

После седьмого хопа трассировка не смогла получить ответ. Наиболее вероятно это связано с тем, что магистральные провайдеры блокируют ICMP-ответы по запросам с целью избежания DoS-атак.

4. С помощью клиента WinSCP подключиться по протоколу FTP к серверу fpm2.amn.nstu.ru и выполнить копирование в Ваш домашний каталог текстового файла test2.txt.

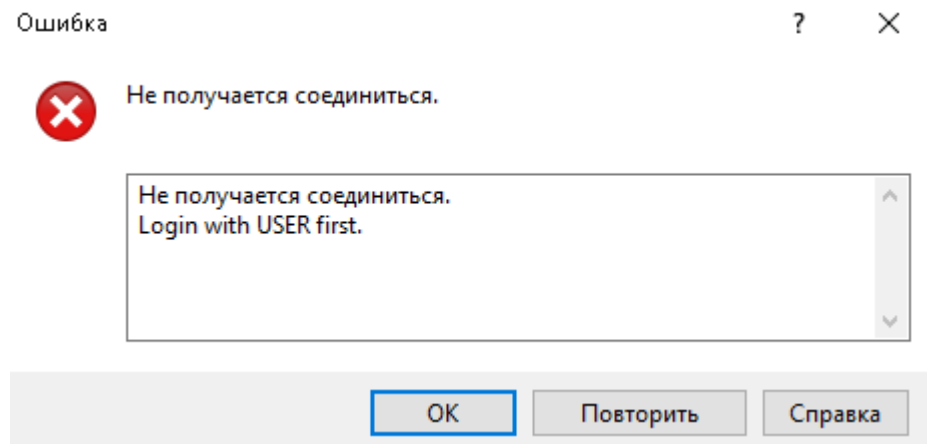


Рисунок 4.1 — невозможно подключиться к серверу

5. Остановить перехват пакетов и сохранить результаты в файл с расширением .pcapng.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.3	77.88.55.70	TCP	54	57968 → 443 [FIN,
2	0.000262	192.168.0.3	87.250.251.119	TCP	54	57969 → 443 [FIN,
3	0.057482	87.250.251.119	192.168.0.3	TLSv1.2	93	Application Data
4	0.057530	192.168.0.3	87.250.251.119	TCP	54	57969 → 443 [RST,
5	0.058160	87.250.251.119	192.168.0.3	TLSv1.2	78	Application Data
6	0.058160	87.250.251.119	192.168.0.3	TCP	54	443 → 57969 [FIN,
7	0.067680	77.88.55.70	192.168.0.3	TLSv1.2	93	Application Data
8	0.067732	192.168.0.3	77.88.55.70	TCP	54	57968 → 443 [RST,
9	0.068414	77.88.55.70	192.168.0.3	TLSv1.2	78	Application Data
10	0.068414	77.88.55.70	192.168.0.3	TCP	54	443 → 57968 [FIN,

Рисунок 5.1 — результаты работы программы

Данный набор пакетов я сохранил в файл с раширением .pcapng.

6. С помощью WireShark определить внутреннюю структуру кадров и пакетов, передаваемых по сети; сравнить ее со структурами, описанными в протоколах Ethernet, IP и TCP.

TCP:

Бит	0 — 3	4 — 9	10 — 15	16 — 31
0	Порт источника, Source Port			Порт назначения, Destination Port
32	Порядковый номер, Sequence Number (SN)			
64	Номер подтверждения, Acknowledgment Number (ACK_SN)			
96	Длина за- головка	Зарезервиро- вано	Флаги	Размер окна
128	Контрольная сумма Опции (необязательное)			Указатель важности
160				
	Данные			

Рисунок 6.1 — структура TCP сегмента

- ▼ Transmission Control Protocol, Src Port: 51976, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
 - Source Port: 51976
 - Destination Port: 443
 - [Stream index: 0]
 - [Conversation completeness: Complete, WITH_DATA (31)]
 - [TCP Segment Len: 0]
 - Sequence Number: 1 (relative sequence number)
 - Sequence Number (raw): 4232329516
 - [Next Sequence Number: 1 (relative sequence number)]
 - Acknowledgment Number: 1 (relative ack number)
 - Acknowledgment number (raw): 3251004379
 - 0101 = Header Length: 20 bytes (5)
 - > Flags: 0x010 (ACK)
 - Window: 64240
 - [Calculated window size: 64240]
 - [Window size scaling factor: -2 (no window scaling used)]
 - Checksum: 0xef65 [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - > [Timestamps]
 - > [SEQ/ACK analysis]

Рисунок 6.2 — структура TCP сегмента в программе Wireshark

IP:

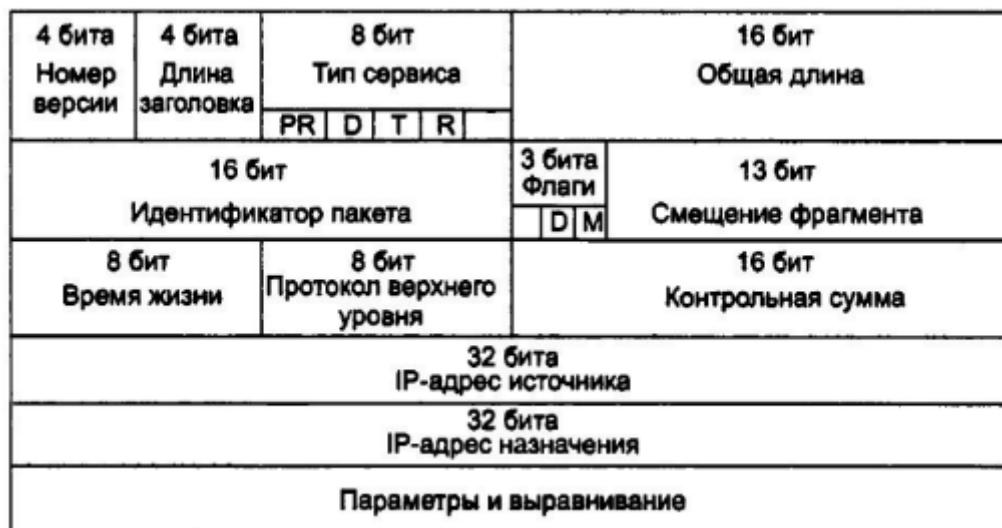


Рисунок 6.3 — структура IP пакета

```

Internet Protocol Version 4, Src: 192.168.198.199, Dst: 87.240.190.70
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0xb0d5 (45269)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0xac53 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.198.199
    Destination Address: 87.240.190.70

```

Рисунок 6.4 — структура IP пакета в программе Wireshark

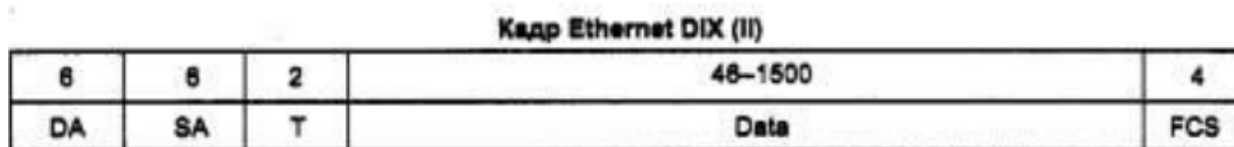


Рисунок 6.5 — структура Ethernet кадра

```

Ethernet II, Src: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31), Dst: ae:0d:9a:00:c4:a3 (ae:0d:9a:00:c4:a3)
  > Destination: ae:0d:9a:00:c4:a3 (ae:0d:9a:00:c4:a3)
  > Source: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
  Type: IPv4 (0x0800)

```

Рисунок 6.6 — структура Ethernet кадра в программе Wireshark

7. Определить последовательность прохождения запросов, реализующих алгоритм трассировки одного из заданных узлов.

```
[andrew@manpc AMI_1st_sem]$ traceroute sklad-service.ru
```

```
1?: [LOCALHOST] pmtu 1500

1: _gateway 0.339ms

1: _gateway 0.298ms

2: 137-192-51-254.novotelecom.ru 1.727ms

3: 10.245.138.241 1.525ms

4: 10.245.138.242 2.094ms asymm 5

5: 10.245.141.14 2.144ms asymm 4

6: no reply

7: ctv-r1.nic.ru 55.098ms asymm 15

8: wcarp.hosting.nic.ru 55.477ms reached
```

Resume: pmtu 1500 hops 8 back 8

Трассировка была выполнена с помощью команды `tracert`. Она использует UDP-порты для отправки пакетов, чтобы отслеживать путь к указанному пункту назначения.

No.	Time	Source	Destination	Protocol	TTL	Length	Info
19	2.227994346	192.168.0.102	91.189.114.22	UDP	1	1514	1514 39879 → 44444 Len=1472
20	2.228311048	192.168.0.1	192.168.0.102	ICMP	64,1	590	Time-to-live exceeded (Time to live exceeded in transit)
21	2.228702004	192.168.0.102	91.189.114.22	UDP	1	1514	1514 39879 → 44445 Len=1472
22	2.228981025	192.168.0.1	192.168.0.102	ICMP	64,1	590	Time-to-live exceeded (Time to live exceeded in transit)
23	2.229108271	192.168.0.102	91.189.114.22	UDP	2	1514	1514 39879 → 44446 Len=1472
24	2.230706825	37.192.51.254	192.168.0.102	ICMP	254,1	70	Time-to-live exceeded (Time to live exceeded in transit)
27	2.282353911	192.168.0.102	91.189.114.22	UDP	3	1514	1514 39879 → 44447 Len=1472
28	2.283852650	10.245.138.241	192.168.0.102	ICMP	253,1	70	Time-to-live exceeded (Time to live exceeded in transit)
36	2.341624586	192.168.0.102	91.189.114.22	WireGuard	4	1514	Transport Data, receiver=0x00000000, counter=15013, datalen=1440
37	2.343696424	10.245.138.242	192.168.0.102	ICMP	251,1	70	Time-to-live exceeded (Time to live exceeded in transit)
46	2.398870093	192.168.0.102	91.189.114.22	UDP	5	1514	1514 39879 → 44449 Len=1472
44	2.392994192	10.245.141.14	192.168.0.102	ICMP	252,1	70	Time-to-live exceeded (Time to live exceeded in transit)
47	2.443828436	192.168.0.102	91.189.114.22	UDP	6	1514	1514 39879 → 44450 Len=1472
58	3.444876393	192.168.0.102	91.189.114.22	UDP	6	1514	1514 39879 → 44451 Len=1472
189	4.445942143	192.168.0.102	91.189.114.22	UDP	6	1514	1514 39879 → 44452 Len=1472
124	5.447618294	192.168.0.102	91.189.114.22	UDP	7	1514	1514 39879 → 44453 Len=1472
128	5.502004365	195.208.200.157	192.168.0.102	ICMP	241,1	70	Time-to-live exceeded (Time to live exceeded in transit)
131	5.570534222	192.168.0.102	91.189.114.22	UDP	8	1514	1514 39879 → 44454 Len=1472
132	5.625908093	91.189.114.22	192.168.0.102	ICMP	57,1	590	Destination unreachable (Port unreachable)

Рисунок 7.1 - трафик ICMP-пакетов в результате работы команды `tracert`

В процессе своей работы программа отправляет дейтаграммы на указанный адрес с параметром Time-to-live (время жизни), увеличивающимся на 1 с каждым хопом. Таким образом, каждый маршрутизатор в сети получает

пакет с истекшим временем жизни и отправляет ICMP-ответ с ошибкой Time-to-live exceeded.

В процессе работы программа отправляет до 3 пакетов. На 6 хопе программа не получила ответ, поэтому видим все 3 отправленных пакета.

8. Восстановить сеанс обмена данными по протоколу HTTP между браузером и сервером при выполнении п.3.

Сеанс обмена был осуществлён исключительно с сайтом nstu.ru, так как на момент изучения данного пункта лабораторной работы сайт moodle был недоступен.

Первым делом, перехватим трафик программой WireShark:

	Time	Source	Destination	Protocol
1	0.000000	192.168.0.3	13.107.4.52	TCP
2	0.131550	13.107.4.52	192.168.0.3	TCP
3	0.131613	192.168.0.3	13.107.4.52	TCP
4	0.131830	192.168.0.3	13.107.4.52	HTTP
5	0.272153	13.107.4.52	192.168.0.3	TCP
6	0.285251	13.107.4.52	192.168.0.3	HTTP
7	0.285363	192.168.0.3	13.107.4.52	TCP
8	0.406106	13.107.4.52	192.168.0.3	TCP
9	0.780957	13.107.4.52	192.168.0.3	TCP
10	0.780996	192.168.0.3	13.107.4.52	TCP

Рисунок 8.1 — перехват данных с сайта nstu.ru

Рассмотрим более детально пакеты протокола HTTP:

```
▼ Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Cache-Control: no-store\r\n
  > Content-Length: 22\r\n
    Content-Type: text/plain; charset=utf-8\r\n
    Last-Modified: Wed, 19 Oct 2022 18:32:13 GMT\r\n
    Accept-Ranges: bytes\r\n
    ETag: 0x8D343F9E96C9DAC\r\n
    Access-Control-Allow-Origin: *\r\n
    Access-Control-Expose-Headers: X-MSEdge-Ref\r\n
    Timing-Allow-Origin: *\r\n
    X-Content-Type-Options: nosniff\r\n
    X-Cache: CONFIG_NOCACHE\r\n
    X-MSEdge-Ref: Ref A: A19D982E937A47359A1282999C71C0DD Ref B: TYAEDGE0808 Ref C: 2022-11-06T10:28:04Z\r\n
    Date: Sun, 06 Nov 2022 10:28:04 GMT\r\n
    Connection: close\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.153421000 seconds]
  [Request in frame: 4]
  [Request URI: http://www.msftconnecttest.com/connecttest.txt]
  File Data: 22 bytes
  > Line-based text data: text/plain (1 lines)
```

Рисунок 8.2 — результат обращения к сайту nstu.ru по протоколу http

```
GET /connecttest.txt HTTP/1.1
Cache-Control: no-cache
Connection: Close
Pragma: no-cache
User-Agent: Microsoft NCSI
Host: www.msftconnecttest.com

HTTP/1.1 200 OK
Cache-Control: no-store
Content-Length: 22
Content-Type: text/plain; charset=utf-8
Last-Modified: Wed, 19 Oct 2022 18:32:13 GMT
Accept-Ranges: bytes
ETag: 0x8D343F9E96C9DAC
Access-Control-Allow-Origin: *
Access-Control-Expose-Headers: X-MSEdge-Ref
Timing-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Cache: CONFIG_NOCACHE
X-MSEdge-Ref: Ref A: A19D982E937A47359A1282999C71C0DD Ref B: TYAEDGE0808 Ref C: 2022-11-06T10:28:04Z
Date: Sun, 06 Nov 2022 10:28:04 GMT
Connection: close

Microsoft Connect Test
```

Рисунок 8.3 — результат обращения к сайту nstu.ru по протоколу http

Как мы можем увидеть, у нас вышло получить доступ к сайту, а также мы смогли зафиксировать тип и размер передаваемых данных.

9. Восстановить сеанс обмена данными по протоколу FTP при выполнении п.4, найти перехваченные логин и пароль, а также восстановить содержимое переданного файла.

Так как п.4 выполнить не удалось из-за отсутствия доступа к серверу fpm2.ami.nstu.ru., данное задание будет выполнено с подключением к общедоступному серверу ftp.dlptest.com, позволяющему загружать файлы, которые будут храниться на сайте не более 10 минут.

Восстановленный сеанс обмена данными:

220 Welcome to the DLP Test FTP Server

USER dlpuser

331 Please specify the password.

PASS rNrKYTX9g7z3RgJRmxWuGHbeu

230 Login successful.

SYST

215 UNIX Type: L8

TYPE I

200 Switching to Binary mode.

PORT 192,168,0,102,169,111

200 PORT command successful. Consider using PASV.

RETR ftp_upload_test.txt

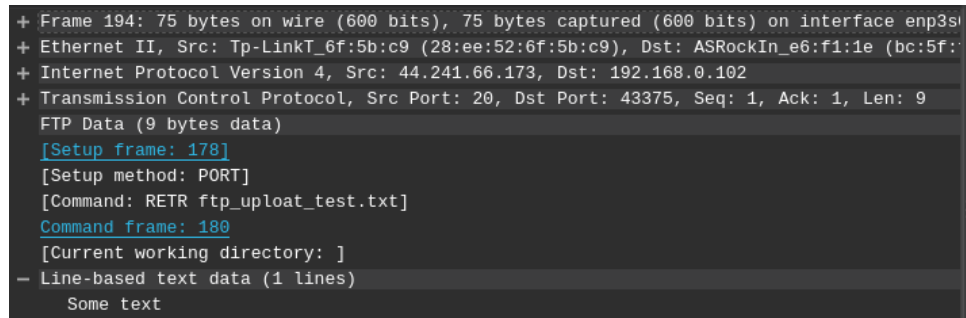
150 Opening BINARY mode data connection for ftp_upload_test.txt (9 bytes).

226 Transfer complete.

QUIT

221 Goodbye.

С сервера был скачан файл ftp_upload_test.txt с содержимым “Some text”, который был ранее на сервер. Восстановить содержимое переданного файла можно по пакету протокола “FTP-DATA”:



```
+ Frame 194: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface enp3s1
+ Ethernet II, Src: Tp-LinkT_6f:5b:c9 (28:ee:52:6f:5b:c9), Dst: ASRockIn_e6:f1:1e (bc:5f:7e:14:5b:00)
+ Internet Protocol Version 4, Src: 44.241.66.173, Dst: 192.168.0.102
+ Transmission Control Protocol, Src Port: 20, Dst Port: 43375, Seq: 1, Ack: 1, Len: 9
  FTP Data (9 bytes data)
    [Setup frame: 178]
    [Setup method: PORT]
    [Command: RETR ftp_upload_test.txt]
    Command frame: 180
    [Current working directory: ]
  - Line-based text data (1 lines)
    Some text
```

Рисунок 9.1 - восстановление с помощью Wireshark содержимого файла, переданного по протоколу FTP

10. Определить последовательность прохождения запросов, реализующих протокол ARP. Построить схему работы протокола и формат пакетов.

Схема работы ARP-протокола следующая:

- 1) Отправляем ARP-запрос с нашего ПК по нужному нам IP на широковещательный MAC-адрес: ff:ff:ff:ff:ff:ff.
- 2) Если в сети есть устройство с нужным нам IP, оно отправляет ARP-ответ, в котором находится MAC-адрес компьютера.
- 3) Отправитель ARP-запроса получает ответ и извлекает из него MAC-адрес, после чего записывает полученные IP и MAC адреса в таблицу соответствия.

Формат ARP-запроса можно увидеть на рисунке 10.1, а также он описан ниже:

- 1) Тип сети
- 2) Тип протокола сетевого уровня
- 3) Длина локального адреса

- 4) Длина глобального адреса
- 5) Тип операции (1 - запрос, 2 - ответ)
- 6) Локальный адрес отправителя
- 7) Глобальный адрес отправителя
- 8) Локальный адрес получателя
- 9) Глобальный адрес получателя

Ethernet II, Src: CiscoInc_c2:e9:00 (00:0d:bd:c2:e9:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)															
Destination: Broadcast (ff:ff:ff:ff:ff:ff)															
Source: CiscoInc_c2:e9:00 (00:0d:bd:c2:e9:00)															
Type: ARP (0x0806)															
Padding: 00000000000000000000000000000000															
Address Resolution Protocol (request)															
Hardware type: Ethernet (1)															
Protocol type: IPv4 (0x0800)															
Hardware size: 6															
Protocol size: 4															
Opcode: request (1)															
Sender MAC address: CiscoInc_c2:e9:00 (00:0d:bd:c2:e9:00)															
Sender IP address: 195.19.132.65															
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)															
Target IP address: 195.19.132.63															

0000	ff	ff	ff	ff	ff	ff	00	0d	bd	c2	e9	00	08	06	00	01
0010	08	00	06	04	00	01	00	0d	bd	c2	e9	00	c3	13	84	41A
0020	00	00	00	00	00	00	00	c3	13	84	3f	00	00	00	00	00?.....
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Рисунок 10.1 — ARP-запрос

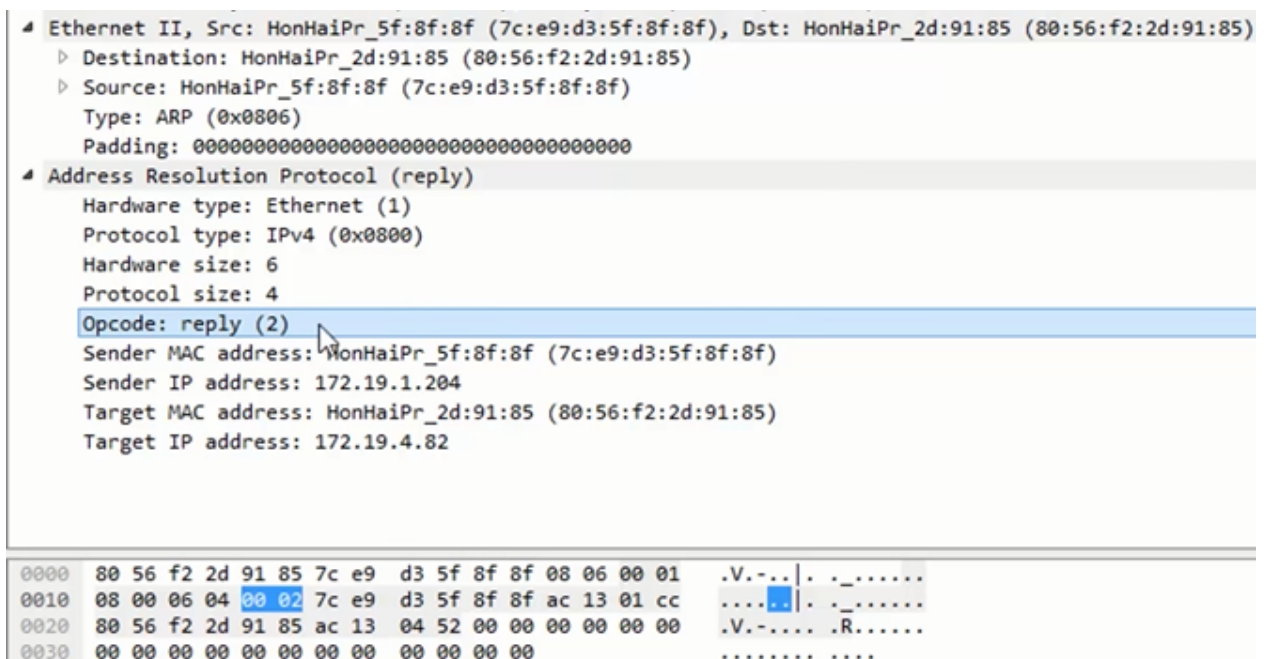


Рисунок 10.2 — ARP-ответ

11. Найти в перехваченном трафике пакеты, передаваемые по протоколу STP, определить назначение данного протокола.

STP (Spanning Tree Protocol) - протокол связующего дерева. Работает на канальном уровне модели OSI и обеспечивает отсутствие петель между коммутаторами (или маршрутизаторами), связанных избыточными физическими соединениями. Примером таких соединений является резервирование канала, когда коммутаторы объединяются двумя или более каналами связи, либо при соединении коммутаторов в кольцо. При отправке broadcast-сообщения сетевым устройством возникает широковещательный шторм.

STP позволяет устанавливать такие избыточные соединения в заблокированное состояние, на которые при падении основного соединения переключается сетевое устройство.

При фильтрации пакетов на интерфейсе сетевой карты с помощью Wireshark не было обнаружено пакетов с протоколом STP, так как в домашней локальной сети нет резервных соединений и иных петель.

12. Найти в перехваченном трафике широковещательные запросы по протоколам DHCP, ARP и ответы на них. Определить структуру передаваемых по этим протоколам кадров.

Начнём с протокола DHCP, так как ARP мы уже рассматривали выше. Для начала зайдём в командную строку и возьмём себе новый IP адрес при помощи команды `inconfig /release` и `inconfig /renew`:

```
Адаптер беспроводной локальной сети Беспроводная сеть:

DNS-суффикс подключения . . . . . :
Локальный IPv6-адрес канала . . . : fe80::8df4:e5c5:7f2f:4a9%3
Основной шлюз. . . . . :
```

Рисунок 12.1 — освобождение IP-адреса

```
Адаптер беспроводной локальной сети Беспроводная сеть:

DNS-суффикс подключения . . . . . :
Локальный IPv6-адрес канала . . . : fe80::8df4:e5c5:7f2f:4a9%3
IPv4-адрес. . . . . : 192.168.0.3
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . : 192.168.0.1
```

Рисунок 12.2 — получение IP-адреса

13.089834	192.168.0.3	192.168.0.1	DHCP	342 DHCP Release
32.025715	0.0.0.0	255.255.255.255	DHCP	344 DHCP Discover
32.082694	192.168.0.1	192.168.0.3	DHCP	328 DHCP Offer
32.083457	0.0.0.0	255.255.255.255	DHCP	370 DHCP Request
32.130367	192.168.0.1	192.168.0.3	DHCP	328 DHCP ACK

Рисунок 12.3 — DORA: получение от сервера IP-адреса клиентом

- ▼ Ethernet II, Src: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - > Source: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
 - Type: IPv4 (0x0800)
- > Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
- > User Datagram Protocol, Src Port: 68, Dst Port: 67
- ▼ Dynamic Host Configuration Protocol (Discover)
 - Message type: Boot Request (1)
 - Hardware type: Ethernet (0x01)
 - Hardware address length: 6
 - Hops: 0
 - Transaction ID: 0x1d9b52ab
 - Seconds elapsed: 0
 - > Bootp flags: 0x0000 (Unicast)
 - Client IP address: 0.0.0.0
 - Your (client) IP address: 0.0.0.0
 - Next server IP address: 0.0.0.0
 - Relay agent IP address: 0.0.0.0
 - Client MAC address: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
 - Client hardware address padding: 00000000000000000000
 - Server host name not given
 - Boot file name not given
 - Magic cookie: DHCP

Рисунок 12.4 — DISCOVER: отправка серверу запроса от клиента с целью получения IP

- ▼ Ethernet II, Src: DwnetTec_0f:04:b0 (e4:26:86:0f:04:b0), Dst: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
 - > Destination: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
 - > Source: DwnetTec_0f:04:b0 (e4:26:86:0f:04:b0)
 - Type: IPv4 (0x0800)
- > Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.3
- > User Datagram Protocol, Src Port: 67, Dst Port: 68
- ▼ Dynamic Host Configuration Protocol (Offer)
 - Message type: Boot Reply (2)
 - Hardware type: Ethernet (0x01)
 - Hardware address length: 6
 - Hops: 0
 - Transaction ID: 0x1d9b52ab
 - Seconds elapsed: 0
 - > Bootp flags: 0x0000 (Unicast)
 - Client IP address: 0.0.0.0
 - Your (client) IP address: 192.168.0.3
 - Next server IP address: 0.0.0.0
 - Relay agent IP address: 0.0.0.0
 - Client MAC address: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
 - Client hardware address padding: 00000000000000000000
 - Server host name not given
 - Boot file name not given
 - Magic cookie: DHCP

Рисунок 12.5 — OFFER: отправка пакета с назначенным IP-адресом клиенту


```

▼ Ethernet II, Src: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
▼ Dynamic Host Configuration Protocol (Request)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x1d9b52ab
  Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP

```

Рисунок 12.6 — REQUEST: отправка пакета клиентом серверу с IP клиента для подтверждения присвоения ему этого адреса

```

▼ Ethernet II, Src: DwnetTec_0f:04:b0 (e4:26:86:0f:04:b0), Dst: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
  > Destination: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
  > Source: DwnetTec_0f:04:b0 (e4:26:86:0f:04:b0)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.3
> User Datagram Protocol, Src Port: 67, Dst Port: 68
▼ Dynamic Host Configuration Protocol (ACK)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x1d9b52ab
  Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 192.168.0.3
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: HonHaiPr_e7:1b:31 (60:6d:c7:e7:1b:31)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP

```

Рисунок 12.6 — ACK: получение пакета от сервера с информацией о присвоении клиенту IP-адреса

Ethernet II, Src: CiscoInc_c2:e9:00 (00:0d:bd:c2:e9:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: CiscoInc_c2:e9:00 (00:0d:bd:c2:e9:00)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: CiscoInc_c2:e9:00 (00:0d:bd:c2:e9:00)
Sender IP address: 195.19.132.65
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 195.19.132.63

0000	ff ff ff ff ff 00 0d bd c2 e9 00 08 06 00 01
0010	08 00 06 04 00 01 00 0d bd c2 e9 00 c3 13 84 41A
0020	00 00 00 00 00 00 c3 13 84 3f 00 00 00 00 00 00?.....
0030	00 00 00 00 00 00 00 00 00 00 00 00

Рисунок 12.7 — ARP-запрос

Ethernet II, Src: HonHaiPr_5f:8f:8f (7c:e9:d3:5f:8f:8f), Dst: HonHaiPr_2d:91:85 (80:56:f2:2d:91:85)
Destination: HonHaiPr_2d:91:85 (80:56:f2:2d:91:85)
Source: HonHaiPr_5f:8f:8f (7c:e9:d3:5f:8f:8f)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: HonHaiPr_5f:8f:8f (7c:e9:d3:5f:8f:8f)
Sender IP address: 172.19.1.204
Target MAC address: HonHaiPr_2d:91:85 (80:56:f2:2d:91:85)
Target IP address: 172.19.4.82

0000	80 56 f2 2d 91 85 7c e9 d3 5f 8f 8f 08 06 00 01	.V.-.. . _.....
0010	08 00 06 04 00 02 7c e9 d3 5f 8f 8f ac 13 01 cc _.....
0020	80 56 f2 2d 91 85 ac 13 04 52 00 00 00 00 00 00	.V.-... .R.....
0030	00 00 00 00 00 00 00 00 00 00 00 00

Рисунок 12.8 — ARP-ответ

13. Определить значение поля «Тип данных» для кадра Ethernet при передаче пакетов IP, ARP, ICMP, DNS, DHCP.

У всех протоколов кроме ARP тип данных в кадре Ethernet указан как IPv4(6), так как это поле указывает какой протокол используется в протоколе третьего уровня. Протокол ARP можно отнести к протоколам третьего

уровня, так как он взаимодействует с IP-адресами. Остальные протоколы используют протокол IP для маршрутизации, поэтому он указан в типе данных.

Протокол	“Тип данных” в кадре Ethernet
IP	IPv4 (0x0800)
ARP	ARP (0x0806)
ICMP	IPv4 (0x0800)
DNS	IPv4 (0x0800)
DHCP	IPv4 (0x0800)

14. Построить статистику по используемым за время сеанса протоколам.

Протокол	Процент Пакетов	Пакеты	Процент Байтов	Байты	Бит/с	Конечные Пакеты
▼ Frame	100.0	43114	100.0	33734913	1090 k	0
▼ Ethernet	100.0	43114	1.8	603596	19 k	0
▼ Internet Protocol Version 6	0.1	24	0.0	960	31	0
▼ User Datagram Protocol	0.1	23	0.0	184	5	0
Multicast Domain Name System	0.1	23	0.0	2982	96	23
Internet Control Message Protocol v6	0.0	1	0.0	64	2	1
▼ Internet Protocol Version 4	99.9	43090	2.6	861836	27 k	0
▼ User Datagram Protocol	21.0	9033	0.2	72264	2336	0
Simple Service Discovery Protocol	0.0	13	0.0	2944	95	13
QUIC IETF	1.6	693	1.0	348894	11 k	693
Multicast Domain Name System	0.1	23	0.0	2982	96	23
▼ ISO 8602/X.234 CLTP ConnectionLess Transport Protocol	0.0	1	0.0	71	2	0
Malformed Packet	0.0	1	0.0	0	0	1
Domain Name System	0.9	370	0.1	24422	789	370
Data	18.4	7933	19.8	6680867	215 k	7933
▼ Transmission Control Protocol	79.0	34040	74.5	25145770	812 k	28360
Transport Layer Security	13.1	5660	71.6	24150008	780 k	5660
Malformed Packet	0.0	1	0.0	0	0	1
Data	0.0	19	0.1	27546	890	19
Internet Group Management Protocol	0.0	9	0.0	72	2	9
Internet Control Message Protocol	0.0	8	0.0	320	10	8

Рисунок 14.1 — статистика по протоколам

15. Изучить процесс установления соединения по протоколу TCP.

TCP (Transfer Control Protocol) - протокол контроля передачи данных. Протокол транспортного уровня, обеспечивающий гарантию доставки данных, а также гарантию сохранения порядка следования байтов. При установке соединения использует алгоритм “тройного рукопожатия”. Перед

непосредственной передачей данных одной из сторон (как правило клиентом) инициируется процесс обмена отправкой сегмента SYN с некоторым номером передаваемого байта Sequence Number (SeqN), в ответ другая сторона (как правило сервер) отправляет сегмент готовности к установке соединения SYN-ACK с номером передаваемого байта SeqN из своей последовательности, а также следующим номером байта Acknowledgement Number (AckN) ожидаемого от инициатора. Инициатор отправляет сегмент подтверждения ACK с номером передаваемого байта SeqN, соответствующим AckN из полученного ответа и ожидаемым номером следующего байта AckN из последовательности байтов сервера. Соединение считается установленным.

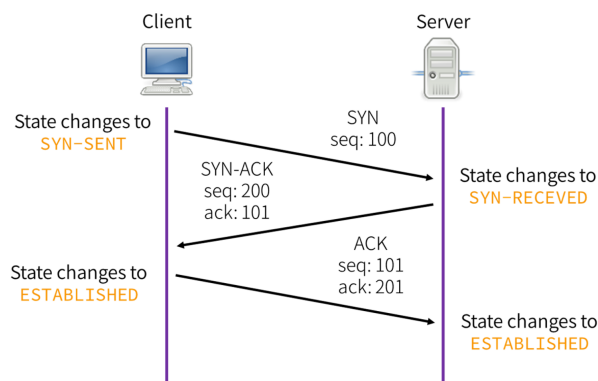


Рисунок 15.1 — схема “тройного рукопожатия”

Пример работы этого протокола рассмотрим с помощью Wireshark:

Source	Destination	Protocol	Length	Info	Sequence number	Acknowledgement number
192.168.0.102	142.250.150.132	TCP	74	41672 → 443 [SYN] Seq=...	121795064	0
142.250.150.132	192.168.0.102	TCP	74	443 → 41672 [SYN, ACK]...	3864220037	121795065
192.168.0.102	142.250.150.132	TCP	66	41672 → 443 [ACK] Seq=...	121795065	3864220038
192.168.0.102	142.250.150.132	TLSv1.3	736	Client Hello	121795065	3864220038
192.168.0.102	142.250.150.132	TLSv1.3	72	Change Cipher Spec	121795735	3864220038
192.168.0.102	142.250.150.132	TLSv1.3	236	Application Data	121795741	3864220038

Рисунок 15.2 — анализ трафика установки TCP-соединения

С адреса 192.168.0.102 инициализируется соединение с сервером 142.250.150.132 отправкой сегмента с флагом SYN и SeqN = 121795064. Сервер принимает соединение, отправляя сегмент с флагами SYN, ACK и

значениями SeqN = 3864220037 и AckN = 121795065, что на 1 больше, чем полученный ранее от инициатора SeqN.

Хост подтверждает соединение отправляя сегмент с флагом ACK и SeqN, равным полученному AckN от сервера, а также AckN больше полученного SeqN от сервера на 1.

При дальнейшей передаче данных от хоста в различных сегментах AckN остается прежним, а номера передаваемых байтов SeqN изменяются.

Вывод: В ходе лабораторной работы нашей бригадой были детально изучены протоколы UDP, TCP, ARP, HTTP, DNS, DHCP и т.д. Благодаря этому удалось выявить различия между этими протоколами.