Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Новосибирский государственный технический университет»

НГТУ НЭТИ

Кафедра теоретической и прикладной информатики

Практическая работа № 3
по дисциплине «Сетевые информационные технологии»

Протоколы стека TCP/IP

Факультет:	ПМИ
Группа:	ПМИМ-01
Студенты:	Наи Сора Орлов М. В.
Бригада:	3
Преподаватель:	Кобылянский В.Г.

Новосибирск
2021

1. Цель работы

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

2. Данные согласно варианту

Вариант	Номер пункта и задание
3	2. Утилита ping: sklad-service.ru, eye.moof.ru, gmail.com, wiw.ru, luminator.ru, hotlog.ru.
	4. test2.txt
	10. ARP
	11. STP

3. Ход работы

1. – 2. Запустить перехват пакетов в WireShark. Определить с помощью утилиты ping доступность заданных узлов в соответствии с вариантом задания, выполнить трассировку к одному из узлов.

```
ping sklad-service.ru

Обмен пакетами с sklad-service.ru [178.210.83.73] с 32 байтами данны
Ответ от 178.210.83.73: число байт=32 время=51мс TTL=48
Ответ от 178.210.83.73: число байт=32 время=51мс TTL=48
Ответ от 178.210.83.73: число байт=32 время=51мс TTL=48
Ответ от 178.210.83.73: число байт=32 время=51мс TTL=48

Статистика Ping для 178.210.83.73:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 51мсек, Максимальное = 51 мсек, Среднее = 51 мсек
```

```
ping eye.moof.ru

Обмен пакетами с eye.moof.ru [90.156.201.99] с 32 байтами данных:
Ответ от 90.156.201.99: число байт=32 время=51мс TTL=51
Ответ от 90.156.201.99: число байт=32 время=51мс TTL=51
Ответ от 90.156.201.99: число байт=32 время=51мс TTL=51
Ответ от 90.156.201.99: число байт=32 время=51мс TTL=51

Статистика Ping для 90.156.201.99:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 51мсек, Максимальное = 51 мсек, Среднее = 51 мсек
```

```
ping gmail.com

Обмен пакетами с gmail.com [64.233.162.18] с 32 байтами данных:
Ответ от 64.233.162.18: число байт=32 время=67мс TTL=101
Ответ от 64.233.162.18: число байт=32 время=67мс TTL=101
Ответ от 64.233.162.18: число байт=32 время=67мс TTL=101
Ответ от 64.233.162.18: число байт=32 время=67мс TTL=101

Статистика Ping для 64.233.162.18:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 67мсек, Максимальное = 67 мсек, Среднее = 67 мсек
```

```
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=51мс TTL=49
Ответ от 89.208.206.225: число байт=32 время=51мс TTL=49
Ответ от 89.208.206.225: число байт=32 время=51мс TTL=49
Ответ от 89.208.206.225: число байт=32 время=51мс TTL=49

Статистика Ping для 89.208.206.225:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
```

```
Приблизительное время приема-передачи в мс:
    Минимальное = 51мсек, Максимальное = 51 мсек, Среднее = 51 мсек
```

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=52мс TTL=51
Ответ от 90.156.201.32: число байт=32 время=52мс TTL=51
Ответ от 90.156.201.32: число байт=32 время=52мс TTL=51
Ответ от 90.156.201.32: число байт=32 время=52мс TTL=51

Статистика Ping для 90.156.201.32:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 52мсек, Максимальное = 52 мсек, Среднее = 52 мсек
```

tracert gmail.com

```
Трассировка маршрута к gmail.com [64.233.162.18]
с максимальным числом прыжков 30:

  1    <1 мс    <1 мс    <1 мс  192.168.0.1
  2     1 ms     1 ms     1 ms  host-109-174-12-1.bb-nsk.sib.mts.ru [109.174.12.1]
  3     1 ms     1 ms     1 ms  78.40.80.50
  4     2 ms     1 ms     1 ms  stn-cr03-be20.10.nsk.mts-internet.net [195.34.36.57]
  5     2 ms     2 ms     2 ms  stn-cr01-be3.54.nsk.mts-internet.net [195.34.50.188]
  6     2 ms     2 ms     2 ms  bhm-cr03-ae2.54.nsk.mts-internet.net [195.34.50.129]
  7     2 ms     2 ms     2 ms  bhm-cr01-ae8.54.nsk.mts-internet.net [212.188.28.226]
  8     1 ms     1 ms     1 ms  bhm-cr02-ae0.16.nsk.mts-internet.net [195.34.50.10]
  9     2 ms     1 ms     1 ms  bhm-cr02-ae11.0.nsk.mts-internet.net [195.34.50.33]
 10     2 ms     2 ms     2 ms  bhm-cr01-ae1.10.nsk.mts-internet.net [195.34.50.14]
 11     3 ms     3 ms     2 ms  bhm-cr03-ae8.54.nsk.mts-internet.net [212.188.28.227]
 12    21 ms    21 ms    22 ms  psshag-cr01-ae12.74.chel.mts-internet.net [195.34.50.153]
 13    53 ms    57 ms    52 ms  che-cr02-ae10.63.sam.mts-internet.net [212.188.42.129]
 14     *       56 ms    53 ms  a197-cr01-ae1.63.msk.mts-internet.net [212.188.29.25]
 15    52 ms    52 ms    52 ms  mag9-cr02-be10.77.msk.mts-internet.net [195.34.50.74]
 16    52 ms    52 ms    51 ms  mag9-cr01-be16.77.msk.mts-internet.net [212.188.29.82]
 17    52 ms    52 ms    52 ms  72.14.223.72
 18    53 ms    53 ms    52 ms  108.170.250.34
 19    69 ms    69 ms    69 ms  172.253.66.116
 20    68 ms    68 ms    68 ms  72.14.235.69
 21    67 ms    67 ms    67 ms  172.253.79.115
 22     *        *        *     Превышен интервал ожидания для запроса.
 23     *        *        *     Превышен интервал ожидания для запроса.
 24     *        *        *     Превышен интервал ожидания для запроса.
 25     *        *        *     Превышен интервал ожидания для запроса.
 26     *        *        *     Превышен интервал ожидания для запроса.
 27     *        *        *     Превышен интервал ожидания для запроса.
 28     *        *        *     Превышен интервал ожидания для запроса.
 29     *        *        *     Превышен интервал ожидания для запроса.
 30     *        *        *     Превышен интервал ожидания для запроса.

Трассировка завершена.
```

tracert hotlog.ru

```
Трассировка маршрута к hotlog.ru [89.208.236.251]
с максимальным числом прыжков 30:

  1    <1 мс    <1 мс    <1 мс  192.168.0.1
  2     1 ms     1 ms     1 ms  host-109-174-12-1.bb-nsk.sib.mts.ru [109.174.12.1]
  3     2 ms     1 ms     1 ms  78.40.80.50
  4     1 ms     1 ms     1 ms  stn-cr03-be20.10.nsk.mts-internet.net [195.34.36.57]
  5     2 ms     2 ms     2 ms  stn-cr01-be3.54.nsk.mts-internet.net [195.34.50.188]
  6    23 ms    23 ms    23 ms  zoo-cr03-be8.66.ekt.mts-internet.net [212.188.42.149]
  7    39 ms    39 ms    39 ms  vish-cr01-be7.66.kaz.mts-internet.net [212.188.29.85]
  8     *        *        *     Превышен интервал ожидания для запроса.
  9    51 ms    51 ms    51 ms  m9-cr04-be8.77.msk.mts-internet.net [212.188.54.213]
 10    50 ms    50 ms    50 ms  m9-cr03-ae13.77.msk.mts-internet.net [212.188.42.106]
 11    50 ms    51 ms    50 ms  212.188.44.170
 12    51 ms    51 ms    50 ms  vl2000.sr3.msk6.ip.di-net.ru [213.248.3.37]
 13    51 ms    50 ms    50 ms  79.137.189.186
 14    54 ms    53 ms    56 ms  89.208.236.251
 15    51 ms    51 ms    51 ms  89.208.236.251

Трассировка завершена.
```

3. С помощью браузера просмотреть несколько страниц на сайте nstu.ru; подключиться к системе Moodle и просмотреть файлы с календарным планом выполнения лабораторных работ и рейтинговой системой по курсу «Сетевые информационные технологии».

Просмотрели страницу расписания занятий группы и главную страницу.

4. С помощью клиента WinSCP подключиться по протоколу FTP к серверу fpm2.ami.nstu.ru и выполнить копирование в Ваш домашний каталог текстового файла согласно варианта из таблицы. Архив с файлами можно скачать из системы Moodle.

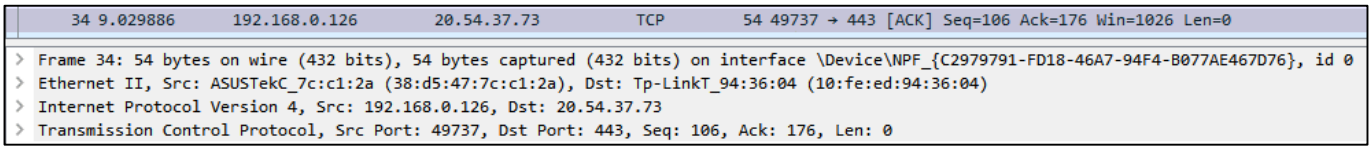
Скопировали файл test2.txt в домашнюю папку.

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

Результат захвата сохранен в файл.

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

Выберем из списка пакетов один пакет TCP и проанализируем его:



- На транспортном уровне – протокол TCP

Структура заголовка				
Бит	**0 — 3**	**4 — 6**	**7 — 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 заголовка, (**Data offset**)	Зарезервировано	Флаги	Размер Окна, **Window size**
128	Контрольная сумма, **Checksum**			Указатель важности, **Urgent Point**
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

```
▼ Transmission Control Protocol, Src Port: 49737, Dst Port: 443, Seq: 106, Ack: 176, Len: 0
    Source Port: 49737
    Destination Port: 443
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence Number: 106    (relative sequence number)
    Sequence Number (raw): 1016745035
    [Next Sequence Number: 106    (relative sequence number)]
    Acknowledgment Number: 176    (relative ack number)
    Acknowledgment number (raw): 455641650
    0101 .... = Header Length: 20 bytes (5)
  ▼ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·····A····]
    Window: 1026
    [Calculated window size: 1026]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xfabf [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▼ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 33]
      [The RTT to ACK the segment was: 0.040722000 seconds]
  ▼ [Timestamps]
      [Time since first frame in this TCP stream: 0.147735000 seconds]
      [Time since previous frame in this TCP stream: 0.040722000 seconds]
```

Структуры TCP-сегментов совпадает.

- На сетевом уровне – протокол IPv4

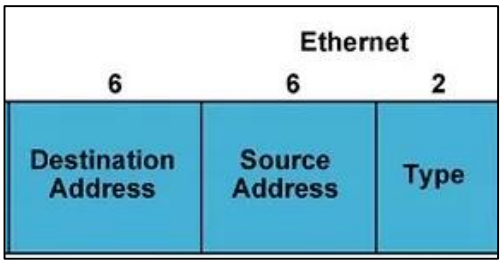
IPv4 Header Format

Октет	0								1								2								3							
Бит	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	Версия				Размер заголовка				Differentiated Services Code Point						Explicit Congestion Notification		Размер пакета (полный)																	
32	Идентификатор																Флаги				Смещение фрагмента											
64	Время жизни								Протокол								Контрольная сумма заголовка															
96	IP-адрес источника																															
128	IP-адрес назначения																															
160	Опции (если размер заголовка > 5)																															
160 или 192+	Данные																															

```
Internet Protocol Version 4, Src: 192.168.0.126, Dst: 20.54.37.73
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
   Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
   Total Length: 40
   Identification: 0xa431 (42033)
   Flags: 0x40, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
   Fragment Offset: 0
   Time to Live: 128
   Protocol: TCP (6)
   Header Checksum: 0x0000 [validation disabled]
   [Header checksum status: Unverified]
   Source Address: 192.168.0.126
   Destination Address: 20.54.37.73
```

Структуры IP-пакета совпадает.

- На канальном уровне – протокол Etherhet



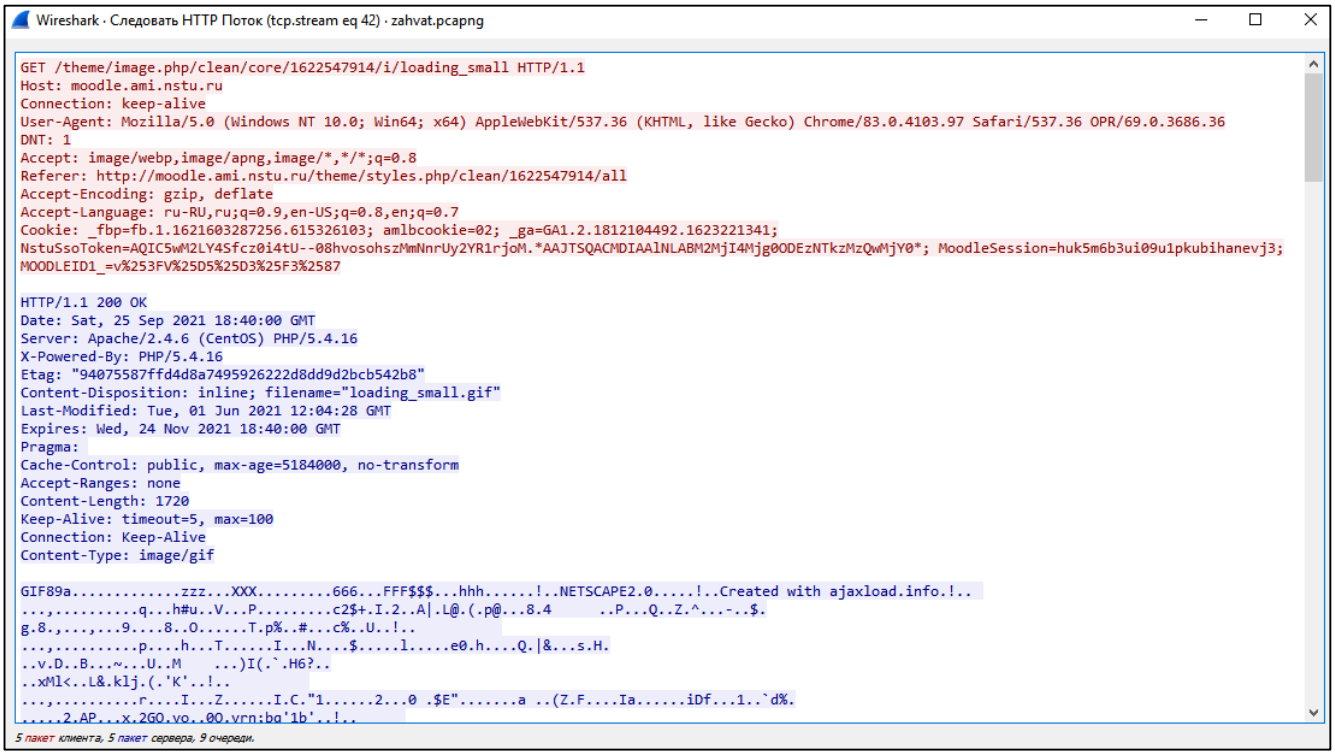

```
Ethernet II, Src: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a), Dst: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
   Destination: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
      Address: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
   Source: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
      Address: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
   Type: IPv4 (0x0800)
```

Структуры Ethernet-кадра совпадает.

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

	Time	Source	Destination	Protocol	Length	Info
1425	258.205569	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=126/32256, ttl=1 (no response found!)
1426	258.205743	192.168.0.1	192.168.0.126	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1427	258.206200	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=127/32512, ttl=1 (no response found!)
1428	258.206320	192.168.0.1	192.168.0.126	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1429	258.206724	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=128/32768, ttl=1 (no response found!)
1430	258.206848	192.168.0.1	192.168.0.126	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
1434	258.209113	192.168.0.1	192.168.0.126	ICMP	120	Destination unreachable (Port unreachable)
1441	259.719091	192.168.0.1	192.168.0.126	ICMP	120	Destination unreachable (Port unreachable)
1448	261.233569	192.168.0.1	192.168.0.126	ICMP	120	Destination unreachable (Port unreachable)
1453	263.755975	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=129/33024, ttl=2 (no response found!)
1454	263.757598	109.174.12.1	192.168.0.126	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1455	263.760508	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=130/33280, ttl=2 (no response found!)
1456	263.761890	109.174.12.1	192.168.0.126	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1457	263.764360	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=131/33536, ttl=2 (no response found!)
1458	263.765756	109.174.12.1	192.168.0.126	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1470	264.779684	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=132/33792, ttl=3 (no response found!)
1471	264.781715	78.40.80.50	192.168.0.126	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1472	264.784497	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=133/34048, ttl=3 (no response found!)
1473	264.785588	78.40.80.50	192.168.0.126	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1474	264.788225	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=134/34304, ttl=3 (no response found!)
1475	264.789271	78.40.80.50	192.168.0.126	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1490	265.804352	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=135/34560, ttl=4 (no response found!)
1491	265.805994	195.34.36.57	192.168.0.126	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1492	265.808844	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=136/34816, ttl=4 (no response found!)
1493	265.810418	195.34.36.57	192.168.0.126	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1494	265.812955	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=137/35072, ttl=4 (no response found!)
1495	265.814521	195.34.36.57	192.168.0.126	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1499	266.827829	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=138/35328, ttl=5 (no response found!)
1500	266.829938	195.34.50.188	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1501	266.832824	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=139/35584, ttl=5 (no response found!)
1502	266.834774	195.34.50.188	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1503	266.837733	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=140/35840, ttl=5 (no response found!)
1504	266.839885	195.34.50.188	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1505	267.851574	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=141/36096, ttl=6 (no response found!)
1506	267.874744	212.188.42.149	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1507	267.875835	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=142/36352, ttl=6 (no response found!)
1508	267.899012	212.188.42.149	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1509	267.900079	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=143/36608, ttl=6 (no response found!)
1510	267.922996	212.188.42.149	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1515	268.916988	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=144/36864, ttl=7 (no response found!)
1516	268.956500	212.188.29.85	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1517	268.959603	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=145/37120, ttl=7 (no response found!)
1518	268.999070	212.188.29.85	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1519	269.002112	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=146/37376, ttl=7 (no response found!)
1520	269.041676	212.188.29.85	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
1524	270.019180	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=147/37632, ttl=8 (no response found!)
1553	273.964170	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=148/37888, ttl=8 (no response found!)
1568	277.963753	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=149/38144, ttl=8 (no response found!)
1587	281.965130	192.168.0.126	89.208.236.251	ICMP	106	Echo (ping) request id=0x0001, seq=150/38400, ttl=9 (no response found!)
1589	282.016061	212.188.54.213	192.168.0.126	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)

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



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

Перехват и восстановление логина с паролем:

6895	376.546900	192.168.0.126	217.71.130.131	TCP	54 57799 → 21 [ACK] Seq=1 Ack=1 Win=262144 Len=0
6896	376.563052	217.71.130.131	192.168.0.126	FTP	74 Response: 220 (vsFTPd 3.0.2)
6897	376.564363	192.168.0.126	217.71.130.131	FTP	70 Request: USER pmi-b6603
6898	376.566465	217.71.130.131	192.168.0.126	TCP	60 21 → 57799 [ACK] Seq=21 Ack=17 Win=29696 Len=0
6899	376.566717	217.71.130.131	192.168.0.126	FTP	88 Response: 331 Please specify the password.
6900	376.567055	192.168.0.126	217.71.130.131	FTP	69 Request: PASS BeSwulj5
6908	376.608794	217.71.130.131	192.168.0.126	TCP	60 21 → 57799 [ACK] Seq=55 Ack=32 Win=29696 Len=0

Wireshark · Следовать TCP Поток (tcp.stream eq 50) · zahvat.pcapng

```
220 (vsFTPd 3.0.2)
USER pmi-b6603
331 Please specify the password.
PASS BeSwulj5
230 Login successful.
SYST
215 UNIX Type: L8
FEAT
211-Features:
 EPRT
 EPSV
 MDTM
 PASV
 REST STREAM
 SIZE
 TVFS
 UTF8
211 End
OPTS UTF8 ON
200 Always in UTF8 mode.
PWD
257 "/home/NSTU/pmi-b6603"
CWD /home/NSTU/pmi-b6603
250 Directory successfully changed.
PWD
257 "/home/NSTU/pmi-b6603"
TYPE A
200 Switching to ASCII mode.
PASV
227 Entering Passive Mode (217,71,130,131,149,27).
LIST -a
150 Here comes the directory listing.
226 Directory send OK.
TYPE A
200 Switching to ASCII mode.
PASV
227 Entering Passive Mode (217,71,130,131,22,151).
```

Пакет 6896. 21 пакет клиента, 35 пакет сервера, 42 очереди. Щёлкните, чтобы выбрать.

Перехват и восстановление содержимого файла:

ftp-data

	Time	Source	Destination	Protocol	Length	Info
523	110.215068	217.71.130.131	192.168.0.126	FTP-DATA	965	FTP Data: 911 bytes (PASV) (LIST -a)
1299	230.222009	217.71.130.131	192.168.0.126	FTP-DATA	965	FTP Data: 911 bytes (PASV) (LIST -a)
6618	350.258050	217.71.130.131	192.168.0.126	FTP-DATA	965	FTP Data: 911 bytes (PASV) (LIST -a)
6948	376.923684	217.71.130.131	192.168.0.126	FTP-DATA	965	FTP Data: 911 bytes (PASV) (LIST -a)
7074	401.906886	217.71.130.131	192.168.0.126	FTP-DATA	965	FTP Data: 911 bytes (PASV) (LIST -a)
7091	401.959706	192.168.0.126	217.71.130.131	FTP-DATA	2146	FTP Data: 2092 bytes (PASV) (STOR test2.txt)
7112	402.013387	217.71.130.131	192.168.0.126	FTP-DATA	1032	FTP Data: 978 bytes (PASV) (LIST -a)

Wireshark · Следовать TCP Поток (tcp.stream eq 53) · zahvat.pcapng

```
Часть 3. Вкратце об истории Олимпийских игр

Древние греки придумали Олимпийские игры, пока вели одну из своих нескончаемых войн. Основных причин было две: во-первых, во время баталий солдатам и
офицерам некогда было заниматься спортом, а ведь эллины (так называли себя древние греки) стремились тренироваться всё время, не занятое упражнениями в
философии; во-вторых, воинам хотелось поскорее вернуться домой, а отпуск на войне не предоставлялся. Было ясно, что войска нуждались в перемирии и что
единственной возможностью его объявить могли стать Олимпийские игры: ведь непременное условие Олимпиады – прекращение войны.

Сначала эллины хотели проводить Олимпийские игры ежегодно, но впоследствии поняли, что частые перерывы в боевых действиях бесконечно удлиняют войны,
поэтому Олимпийские игры стали объявлять только раз в четыре года. Зимних игр в те времена, конечно же, не было, потому что в Элладе не было ни ледовых
арен, ни горнолыжных трасс.

В Олимпийских играх мог участвовать любой гражданин, но богатые могли позволить себе дорогостоящее спортивное снаряжение, а бедные – нет. Чтобы богатые не
побеждали бедных только оттого, что их спортинвентарь лучше, все атлеты мерились силой и ловкостью обнажёнными.

– А почему игры назывались Олимпийскими? – спросите вы. – Боги с Олимпа тоже принимали в них участие?

Нет, боги, кроме ссор между собой, никаким другим спортом не занимались, но любили с нескрываемым от смертных азартом следить за спортивными состязаниями
из поднебесья. А чтобы богам сподручнее было наблюдать за перипетиями соревнований, первый стадион построили в святилище, которое называлось Олимпия, –
так игры получили свое название.

Боги и те на время игр заключали между собой перемирие и клялись не помогать своим избранникам. Более того, они даже разрешали эллинам считать победителей
богами – правда, временными, всего на один день. Чемпионы-олимпийцы удостаивались оливковых и лавровых венков: медалей тогда еще не придумали, а лавр в
Древней Греции ценился на вес золота, так что лавровый венок тогда был всё равно что золотая медаль сегодня.
```

1 пакет клиента, 0 пакет сервера, 0 очереди.

Весь диалог (2092 bytes) | Показать данные как windows-1251 | Поток 53

10. Определить последовательность прохождения запросов, реализующих один из протоколов в соответствии с вариантом из таблицы (ARP). Построить схему работы протокола и формат пакетов.

arp

No.	Time	Source	Destination	Protocol	Length	Info
1041	186.967717	Tp-LinkT_94:36:04	ASUSTekC_7c:c1:2a	ARP	60	Who has 192.168.0.126? Tell 192.168.0.1
1042	186.967732	ASUSTekC_7c:c1:2a	Tp-LinkT_94:36:04	ARP	42	192.168.0.126 is at 38:d5:47:7c:c1:2a

```
Frame 1041: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{C2979791-FD18-46A7-94F4-B077AE467D76}, id 0
Ethernet II, Src: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04), Dst: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
    Sender IP address: 192.168.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.126
```

```
Frame 1042: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{C2979791-FD18-46A7-94F4-B077AE467D76}, id 0
Ethernet II, Src: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a), Dst: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
    Sender IP address: 192.168.0.126
    Target MAC address: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
    Target IP address: 192.168.0.1
```

Пояснение:

ARP (Address Resolution Protocol) – протокол разрешения адресов необходим для определения MAC-адреса по IP-адресу. ARP-протокол относится к канальному уровню.

Маршрутизатор Tp-Link послал широковещательный ARP-запрос на все подключенные к нему устройства, используя соответствующий MAC-адрес. Компьютер Asus, получив запрос, сообщает свой Mac-адрес, посылая ARP-ответ маршрутизатору. Форматы ARP-запросов и ARP-ответов одинаковы.

Заголовок запроса ARP протокола:

Заголовки ARP-пакетов совпадают.

11.Найти в перехваченном трафике пакеты, передаваемые по протоколу в соответствии с вариантом задания (STP), определить назначение данного протокола.

stp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
12	2.000062	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
18	4.000057	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
20	6.000137	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
21	8.000178	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
37	10.000222	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
50	12.000254	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
61	14.000301	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
65	16.000345	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
70	18.000376	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
78	20.000397	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
95	22.000456	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
110	24.000499	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
114	26.000529	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
119	28.000594	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
121	30.000638	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
124	32.000668	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
150	34.000694	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
155	36.000741	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
166	38.000770	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
172	40.000839	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
183	42.000881	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
184	44.000926	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
197	46.000967	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
199	48.001005	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
209	50.001048	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001
218	52.001045	Tp-LinkT_94:36:02	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/0/10:fe:ed:94:36:04 Cost = 0 Port = 0x8001

Spanning Tree Protocol (STP, протокол остовного дерева) — канальный протокол. Основной задачей STP является устранение петель в топологии произвольной сети Ethernet, в которой есть один или более сетевых мостов, связанных избыточными соединениями. STP решает эту задачу, автоматически блокируя соединения, которые в данный момент для полной связности коммутаторов являются избыточными.

Необходимость устранения топологических петель в сети Ethernet следует из того, что их наличие в реальной сети Ethernet с коммутатором с высокой вероятностью приводит к бесконечным повторам передачи одних и тех же кадров Ethernet одним и более коммутатором, отчего пропускная способность сети оказывается почти полностью занятой этими бесполезными повторами; в этих условиях, хотя формально сеть может продолжать работать, на практике её производительность становится настолько низкой, что может выглядеть как полный отказ сети.

Принцип действия:

1. Выбирается один корневой мост (*Root Bridge*).

2. Далее каждый коммутатор просчитывает кратчайший путь к корневому. Соответствующий порт называется корневым портом (*Root Port*). У любого некорневого коммутатора может быть только один корневой порт.

3. После этого для каждого сегмента сети, к которому присоединён более чем один мост (или несколько портов одного моста), просчитывается кратчайший путь к корневому мосту(порту). Мост, через который проходит этот путь, становится назначенным для этой сети (*Designated Bridge*), а соответствующий порт — назначенным портом (*Designated port*).

4. Далее во всех сегментах, с которыми соединено более одного порта моста, все мосты блокируют все порты, не являющиеся корневыми и назначенными. В итоге получается древовидная структура (математический граф) с вершиной в виде корневого коммутатора.

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

Широковещательный ARP-запрос:

```
> Frame 1041: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{C2979791-FD18-46A7-94F4-B077AE467D76}, id 0
> Ethernet II, Src: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04), Dst: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
    Sender IP address: 192.168.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.126
```

DHCP-пакетов не было перехвачено, так как в локальной сети данному компьютеру выделен статический IP-адрес 192.168.0.126, а остальным устройствам маршрутизатор назначает IP-адреса динамически.

Структура заголовка ARP-запроса:

Структура заголовка DHCP-запроса:

Dynamic Host Configuration Protocol				
Bit Offset	0–15		16–31	
0	OpCode	Hardware Type	Hardware Length	Hops
32	Transaction ID			
64	Seconds Elapsed		Flags	
96	Client IP Address			
128	Your IP Address			
160	Server IP Address			
196	Gateway IP Address			
228+	Client Hardware Address (16 bytes)			
	Server Host Name (64 bytes)			
	Boot File (128 bytes)			
	Options			

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

Передача пакетов ARP:

```
Ethernet II, Src: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a), Dst: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
  Destination: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
  Source: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
  Type: ARP (0x0806)
```

Передача пакетов ICMP:

```
Ethernet II, Src: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a), Dst: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
  Destination: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
  Source: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
  Type: IPv4 (0x0800)
```

Передача пакетов DNS:

```
Ethernet II, Src: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04), Dst: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
  Destination: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
  Source: Tp-LinkT_94:36:04 (10:fe:ed:94:36:04)
  Type: IPv4 (0x0800)
```

Передача пакетов MDNS:

```
Ethernet II, Src: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a), Dst: IPv6mcast_fb (33:33:00:00:00:fb)
  Destination: IPv6mcast_fb (33:33:00:00:00:fb)
  Source: ASUSTekC_7c:c1:2a (38:d5:47:7c:c1:2a)
```

В поле тип данных указывается название протокола верхнего уровня (уровня 3 по модели OSI).

14. Построить статистику по используемым за время сеанса протоколам.

Wireshark · Статистика Иерархии Протоколов · zahvat.pcapng

Протокол	Процент Пакетов	Пакеты	Процент Байтов	Байты	Бит/с	Конечные Пакеты	Конечные Байты	Конечные Бит/с
Frame	100.0	7152	100.0	4701045	91k	0	0	0
Ethernet	100.0	7152	2.1	100128	1953	0	0	0
Logical-Link Control	3.0	213	0.2	8521	166	0	0	0
Spanning Tree Protocol	2.9	206	0.2	7210	140	206	7210	140
Internetwork Packet eXchange	0.1	7	0.0	210	4	0	0	0
Service Advertisement Protocol	0.1	7	0.0	462	9	7	462	9
Link Layer Discovery Protocol	0.0	1	0.0	44	0	1	44	0
Internet Protocol Version 6	0.1	9	0.0	360	7	0	0	0
User Datagram Protocol	0.1	9	0.0	72	1	0	0	0
Multicast Domain Name System	0.1	5	0.0	686	13	5	686	13
Link-local Multicast Name Resolution	0.1	4	0.0	168	3	4	168	3
Internet Protocol Version 4	96.9	6927	2.9	138588	2704	0	0	0
User Datagram Protocol	21.3	1525	0.3	12200	238	0	0	0
Simple Service Discovery Protocol	3.5	247	1.6	76247	1487	247	76247	1487
NetBIOS Name Service	0.1	6	0.0	300	5	6	300	5
NetBIOS Datagram Service	0.3	21	0.1	4221	82	0	0	0
SMB (Server Message Block Protocol)	0.3	21	0.1	2499	48	0	0	0
SMB MailSlot Protocol	0.3	21	0.0	525	10	0	0	0
Microsoft Windows Browser Protocol	0.3	21	0.0	693	13	0	0	0
VSS Monitoring Ethernet trailer	0.3	21	0.0	21	0	21	21	0
Multicast Domain Name System	0.1	5	0.0	686	13	5	686	13
Link-local Multicast Name Resolution	0.1	4	0.0	168	3	4	168	3
GQUIC (Google Quick UDP Internet Connections)	3.7	264	2.0	95601	1865	264	95601	1865
Domain Name System	1.9	136	0.2	10780	210	136	10780	210
Data	11.8	842	6.1	284573	5552	842	284573	5552
Transmission Control Protocol	71.3	5097	83.9	3941987	76k	4010	3179957	62k
Transport Layer Security	12.1	868	73.0	3431984	66k	860	3391970	66k
Hypertext Transfer Protocol	1.4	101	8.4	395809	7722	48	37535	732
Portable Network Graphics	0.1	8	0.3	15088	294	8	18606	363
Media Type	0.1	10	3.8	178663	3486	10	96612	1885
Line-based text data	0.1	10	2.8	132892	2592	10	135094	2635
HTML Form URL Encoded	0.0	3	0.0	189	3	3	2912	56
eXtensible Markup Language	0.3	20	1.9	89155	1739	20	96710	1886
Compuserve GIF	0.0	2	0.0	1763	34	2	2270	44
FTP Data	0.1	7	0.2	7625	148	1	0	0
Line-based text data	0.1	6	0.1	5533	107	6	5533	107
File Transfer Protocol (FTP)	1.4	101	0.0	2091	40	101	0	0
Data	0.3	18	0.0	533	10	18	533	10
Internet Group Management Protocol	0.2	12	0.0	96	1	12	96	1
Internet Control Message Protocol	4.1	293	0.5	22312	435	287	21796	425
NetBIOS Name Service	0.1	6	0.0	300	5	6	300	5
Address Resolution Protocol	0.0	2	0.0	74	1	2	74	1

15. Изучить процесс установления соединения по протоколу TCP.

Рассмотрим процесс соединение по протоколу TCP на примере:

Клиент						Сервер					
Шаг / Действие	Порядковый номер (ISN)	Номер подтверждения (ACK)	Порт отправ.	Порт получ.	Флаги	Шаг / Действие	Порядковый номер (ISN)	Номер подтверждения (ACK)	Порт отправ.	Порт получ.	Флаги
1 отправил	1111		7070	8080	**SYN**	1 получил	1111		7070	8080	SYN
3 получил	2222	111**2**	8080	7070	SYN **ACK**	2 отправил	2222	111**2**	8080	7070	**SYN ACK**
4 отправил	1112	222**3**	7070	8080	**ACK**	5 получил	111**2**	222**3**	7070	8080	**ACK**
< Передача данных >											
6 отправил	1112		7070	8080	**FIN**	6 получил	1112		7070	8080	**FIN**
7 получил		1113	8080	7070	**ACK**	7 отправил		111**3**	8080	7070	**ACK**

Соединение устанавливается в три этапа (процесс «трёхкратного рукопожатия» TCP).

Первое рукопожатие: **1)** клиент вызывает connect() для запуска запроса на соединение, устанавливает бит флага SYN в 1, случайным образом генерирует ISN и отправляет сегмент синхронизации.

Второе рукопожатие: **2)** после того, как Сервер по очереди вызовет socket(), bind() и listen(), он будет отслеживать указанный адрес сокета. После того, как сервер получает сегмент сообщения синхронизации от клиента, запрос подключения клиента отслеживается битом флага SYN = 1, поэтому сервер вызывает функцию accept(), чтобы принять запрос подключения, и устанавливает биты флага SYN и ACK в true и номер подтверждения ACK = ISN клиента +1, случайным образом генерируют свой порядковый номер ISN и отправляет сегмент (сегмент синхронизации + подтверждения) клиенту для подтверждения запроса на соединение.

Третье рукопожатие: **3)** после того, как клиент получает сегмент сообщения сервера, он проверяет, установлен ли флаг ACK и соответствует ли номер подтверждения ISN клиента + 1.

4) Если это правильно, бит флага ACK устанавливается в 1, ACK = ISN сервера + 1, и пакет отправляется на сервер, клиент переходит в состояние ESTABLISHED.

5) Сервер проверяет, установлен ли флаг ACK 1 и является ли ACK = ISN сервера + 1 после получения клиентского сегмента. Если это правильно, соединение установлено успешно, и сервер также переходит в состояние ESTABLISHED, завершая трехстороннее рукопожатие. Затем данные могут передаваться между клиентом и сервером.

6) После передачи данных для прекращения соединения клиент устанавливает флаг FIN, номер подтверждения клиента + 1, ACK = ISN сервера + 1.

7) Сервер, получив сегмент с флагом FIN формирует сегмент с установленным флагом ACK = ISN клиента + 1. Клиент получает подтверждение, и соединение считается закрытым.