

Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
«Университет ИТМО»

## **Инфокоммуникационные системы и технологии**

Курсовая работа №1

**Выполнил:**  
Студент гр. К3120  
А.П. Алексеевич  
**Преподаватель:**  
Г.А. Карапетян

Санкт-Петербург  
2020 г.

# Содержание

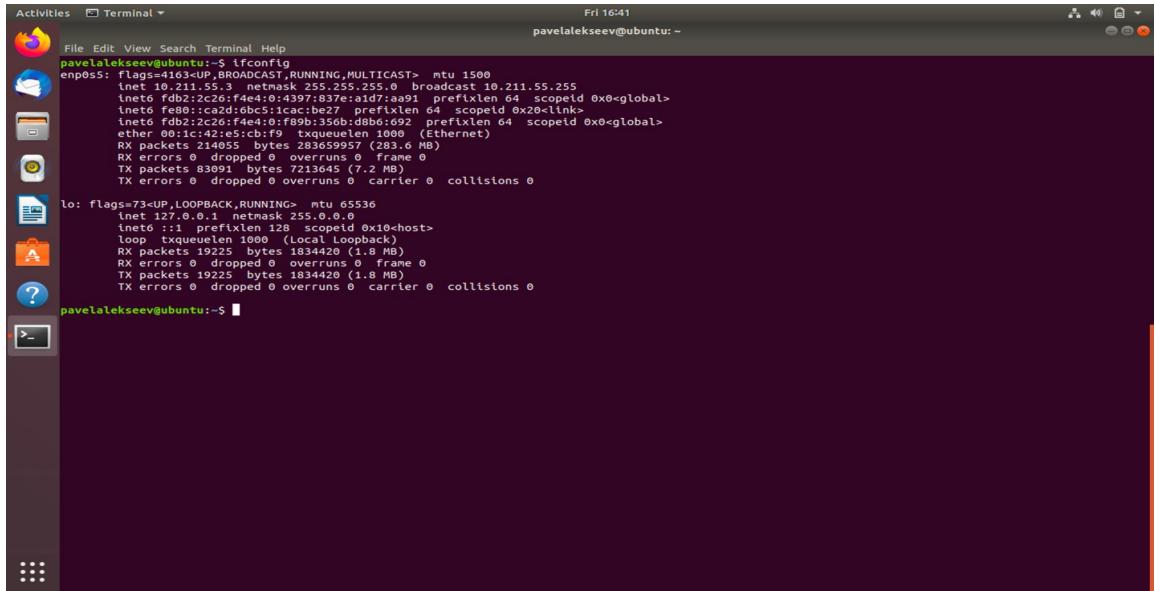
<b>Лабораторная работа №1 .....</b>	<b>3</b>
Провести анализ сетевой конфигурации ОС .....	3
Изучение и практическая работа с утилитой ping .....	4
Установка, настройка и запуск СУБД (SQL -ориентированной: MySQL).....	5
Установка,настройка и запуск WEB-сервера Apache.....	6
Настройка установки и запуска apache2 .....	6
Установка, запуск, настройка FTP-сервера .....	7
Создал пользователя для удаленного SSH-подключения с возможностями администратора ....	9
<b>Лабораторная работа №2 .....</b>	<b>9</b>
<b>Ethernet .....</b>	<b>9</b>
Обзор:.....	10
Структура Ethernet-фрейма. Стандарты. ....	10
<b>Wireshark установка и знакомство. .....</b>	<b>12</b>
Характеристики трафика.....	13
Генерация трафика.....	14
<b>Лабораторная работа №3. ....</b>	<b>15</b>
<b>Стандарт IPv4 .....</b>	<b>16</b>
<b>Классовая адресация .....</b>	<b>16</b>
<b>Заголовки ip-пакета .....</b>	<b>17</b>
<b>Стандарт TCP .....</b>	<b>19</b>
<b>Заголовки TCP-пакета .....</b>	<b>20</b>
Генерирую трафик и анализирую .....	23
<b>Лабораторная работа № 4 .....</b>	<b>24</b>
<b>Альтернатива .....</b>	<b>25</b>
<b>Безопасность .....</b>	<b>25</b>
<b>Практическая работа с HTTP .....</b>	<b>26</b>
<b>Итоги .....</b>	<b>27</b>

# Лабораторная работа №1

## «Подготовка серверной инфраструктуры»

### Провести анализ сетевой конфигурации ОС

Используя команду ifconfig вывел на экран все свои сетевые интерфейсы.



```
pavelalekseev@ubuntu:~$ ifconfig
enp0s5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.10 brd 192.168.1.255 scope global
          link layer ...
          brd 192.168.1.255
          netmask 255.255.255.0 broadcast 192.168.1.255
          ...
          ether 00:0c:2e:6f:40:01 brd ...
          txqueuelen 1000 (Ethernet)
          RX packets 214955 bytes 283659957 (283.6 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 83091 bytes 7213644 (7.2 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 brd 127.0.0.0 ...
          link layer ...
          netmask 255.0.0.0 broadcast 127.0.0.0
          ...
          loop txqueuelen 0 (Local Loopback)
          RX packets 19225 bytes 1834420 (1.8 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 19225 bytes 1834420 (1.8 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
pavelalekseev@ubuntu:~$
```

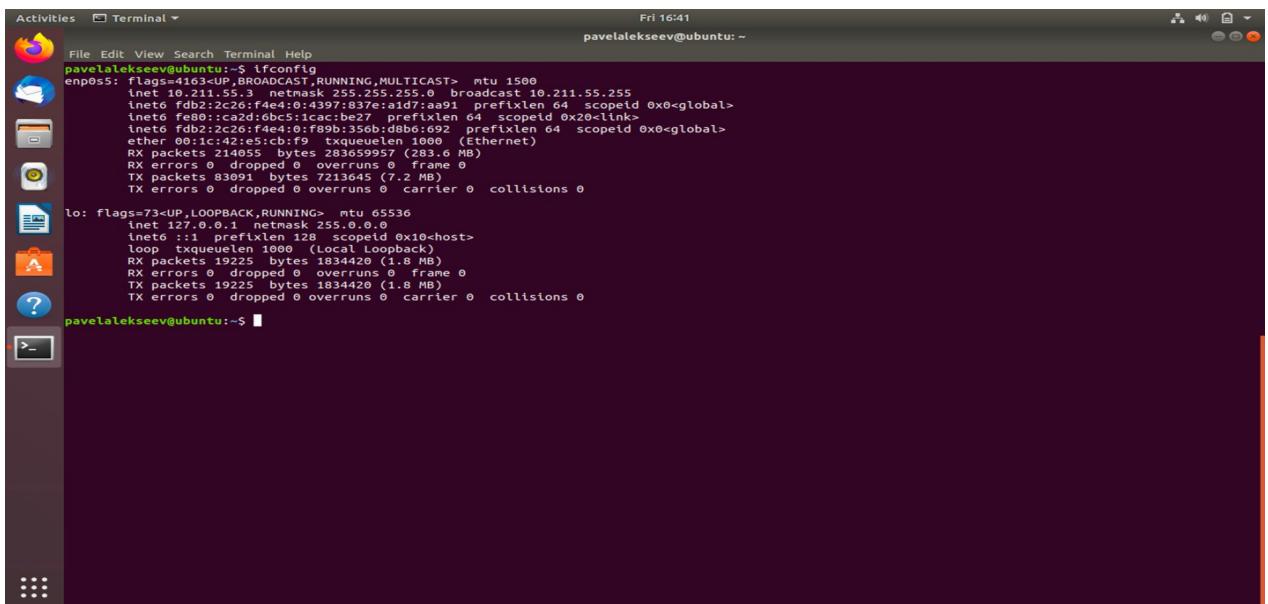
- lo- интерфейс обратной петли.
- enp0s5- интерфейс подключенный к карте Ethernet.

### Характеристики интерфейсов

- flags- список установленных флагов интерфейса.
- UP, BROADCAST, MULTICAST – включен, принимает такой пакет, групповой пакет
- LOOPBACK-канал коммуникаций с 1 конечной точкой .
- Inet – ip адрес интерфейса.
- Mtu – размер максимального блока , который можно передать.
- Netmask- маска подсети.
- В данной строке (3) указан доп. ID.
- Loop txqueuelen- устанавливает длину очереди передачи для устройства.
- RX и TX packets –число пакетов полученных и отправленных.
- RX и TX bytes– размер этих пакетов .
- Errors–ошибки,Dropped–отброшенные пакеты,Overruns–переполненные пакеты,carrier–потерянные пакеты ,collisions–стыки пакетов.

# Изучение и практическая работа с утилитой ping

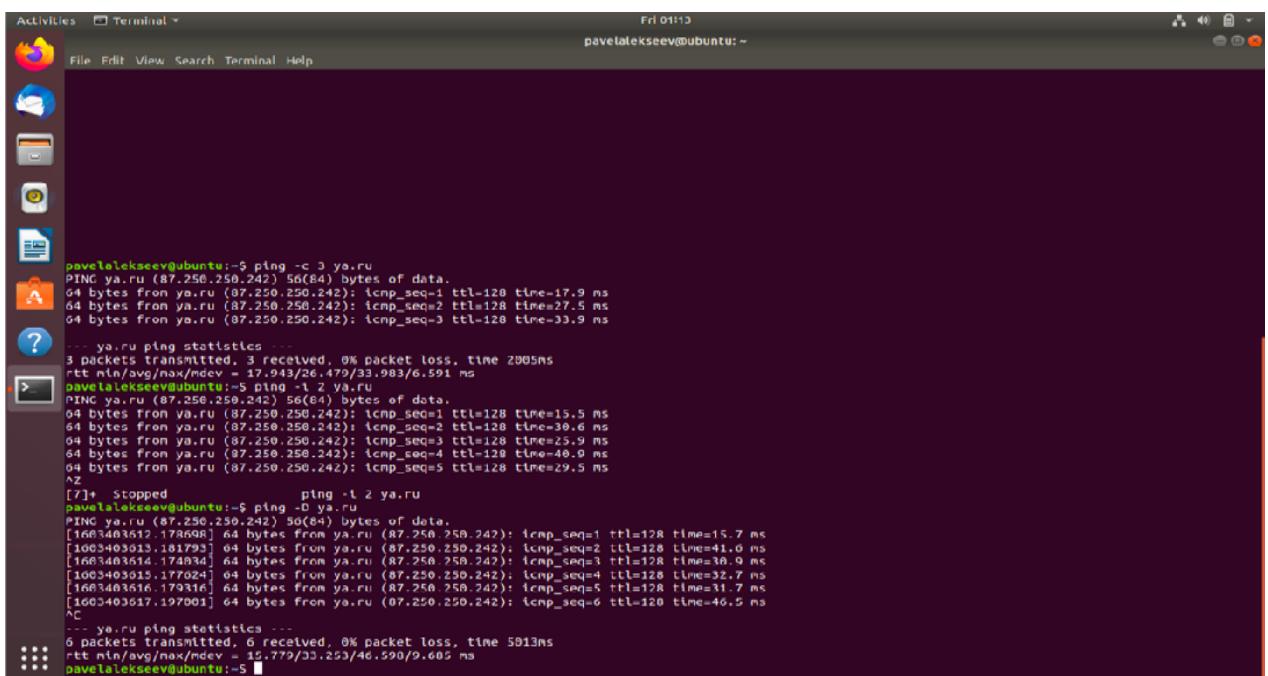
Ping – команда , которая помогает проверить качество соединения , доступность удаленного хоста . После окончания данной команды , мы можем наблюдать . Результат работы ( сколько ICMP пакетов было передано , сколько получено , сколько потеряно время передачи).



```
Fri 16:41 pavalekseev@ubuntu:~$ ifconfig
enp0s5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.211.55.3 netmask 255.255.255.0 broadcast 10.211.55.255
                inet6 fdb2:2c26:fd4e:0:4397:837e:a1d7:a9a prefixlen 64 scoprid 0x0<global>
                inet6 fe80::ca2d:6bc5:fcac:be2%enp0s5 prefixlen 64 scoprid 0x20<link>
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scoprid 0x10<host>
loop: txqueuelen 1000  (Local Loopback)
        RX packets 19225 bytes 1834420 (1.8 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 19225 bytes 1834420 (1.8 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
pavalekseev@ubuntu:~$
```

- icmpq\_seq-ID для потока .
- ttl- кол-во узлов до целого узла.

## Пример использования доп. ключей



```
Fri 01:13 pavalekseev@ubuntu:~$ ping -c 3 ya.ru
PING ya.ru (87.250.250.242) 56(84) bytes of data.
64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=128 time=17.9 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=128 time=27.5 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=128 time=33.9 ms
--- ya.ru ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 17.943/26.479/33.983/6.591 ms
pavalekseev@ubuntu:~$ ping -l 2 ya.ru
PING ya.ru (87.250.250.242) 56(84) bytes of data.
64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=128 time=15.5 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=128 time=30.6 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=128 time=25.9 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=4 ttl=128 time=40.9 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=5 ttl=128 time=29.5 ms
^Z
[7]+  Stopped                  ping -l 2 ya.ru
pavalekseev@ubuntu:~$ ping -D ya.ru
PING ya.ru (87.250.250.242) 50(84) bytes of data.
[1603403512.178608] 64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=128 time=15.7 ms
[1603403513.181793] 64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=128 time=41.0 ms
[1603403514.174934] 64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=128 time=30.9 ms
[1603403515.177024] 64 bytes from ya.ru (87.250.250.242): icmp_seq=4 ttl=128 time=32.7 ms
[1603403516.179116] 64 bytes from ya.ru (87.250.250.242): icmp_seq=5 ttl=128 time=31.7 ms
[1603403517.197001] 64 bytes from ya.ru (87.250.250.242): icmp_seq=6 ttl=120 time=40.5 ms
^C
--- ya.ru ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5013ms
rtt min/avg/max/mdev = 15.779/33.253/46.590/9.605 ms
pavalekseev@ubuntu:~$
```

- ping -c 3 ya.ru- выведет 3 пакета и покажет результат .

- ping -i 2 ya.ru - будет выводить пакеты с определенным интервалом.
- ping-D ya.ru - интенсивность отправляемых пакетов.

P.S: При обычном использовании данной команды придется останавливать вручную при помощи **ctrl+c**

## Установка, настройка и запуск СУБД (SQL-ориентированной: MySQL)

- СУБД- система управления базами данных.(создает базы данных, позволяет редактировать их, безопасность)
- База данных(БД)- определенный набор данных ,созданный по своим правилам имеющий определенную структуру.

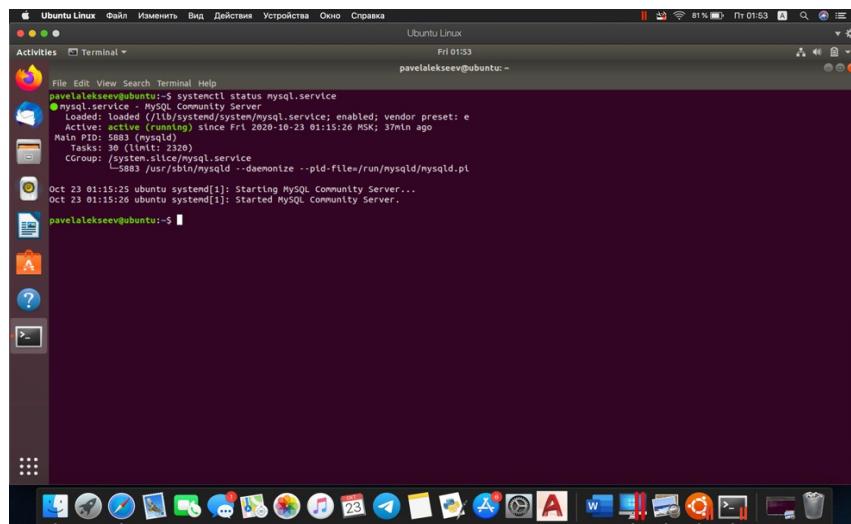
В нашей базе данных используется некий язык SQL , благодаря которому и происходит обмен данными.

Принцип работы : USER отправляет SQL запрос -> через СУБД -> доходит до базы данных -> затем обратно через СУБД -> возвращает данные пользователю.

### Установил MySQL , настроил , запустил

```
pavelalekseev@ubuntu:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
libl ballo1 libevent-core-2.1-6 libhtml-template-perl mysql-client-5.7
mysql-client-core-5.7 mysql-common mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
libl ballo1 libevent-core-2.1-6 libhtml-template-perl mysql-client-5.7
mysql-client-core-5.7 mysql-common mysql-server mysql-server-5.7
mysql-server-core-5.7
0 upgraded, 9 newly installed, 0 to remove and 125 not upgraded.
Need to get 19.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get: http://us.archive.ubuntu.com/ubuntu bionic/main amd64 mysql-common all 5.8+1.0.4 [7,308 B]
Get: http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libl ballo1 amd64 0.3.110-Subuntu0.1 [6,476 B]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-core-5.7 amd64 5.7.31-Subuntu0.18.04.1 [6,653 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-core-5.7 amd64 5.7.31-Subuntu0.18.04.1 [1,948 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-core-5.7 amd64 5.7.31-Subuntu0.18.04.1 [7,452 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libevent-core-2.1-6 amd64 2.1.6-4build1 [85.9 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-server-5.7 amd64 5.7.31-Subuntu0.18.04.1 [2,931 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libhtml-template-perl all 2.97-1 [59.0 kB]
Fetched 19.2 MB in 9s (2,244 kB/s)
Preconfiguring packages...
Selecting previously unselected package mysql-common.
(Reading database ... 135322 files and directories currently installed.)
Preparing to unpack .../mysql-common_5.8+1.0.4_all.deb ...
Unpacking mysql-common (5.8+1.0.4) ...
Selecting previously unselected package libl ballo1:amd64.
Preparing to unpack .../libl ballo1_0.3.110-Subuntu0.1_amd64.deb ...
Unpacking libl ballo1:amd64 (0.3.110-Subuntu0.1_amd64) ...
Selecting previously unselected package mysql-client-core-5.7.
Preparing to unpack .../mysql-client-core-5.7_5.7.31-Subuntu0.18.04.1_amd64.deb ...
Unpacking mysql-client-core-5.7 (5.7.31-Subuntu0.18.04.1)
```

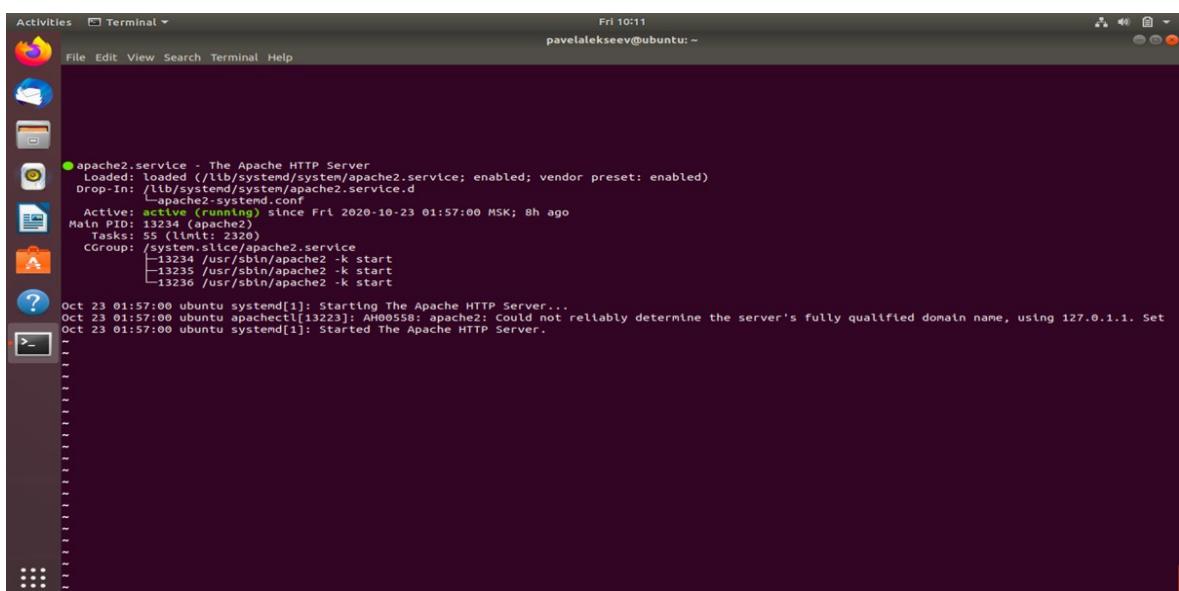
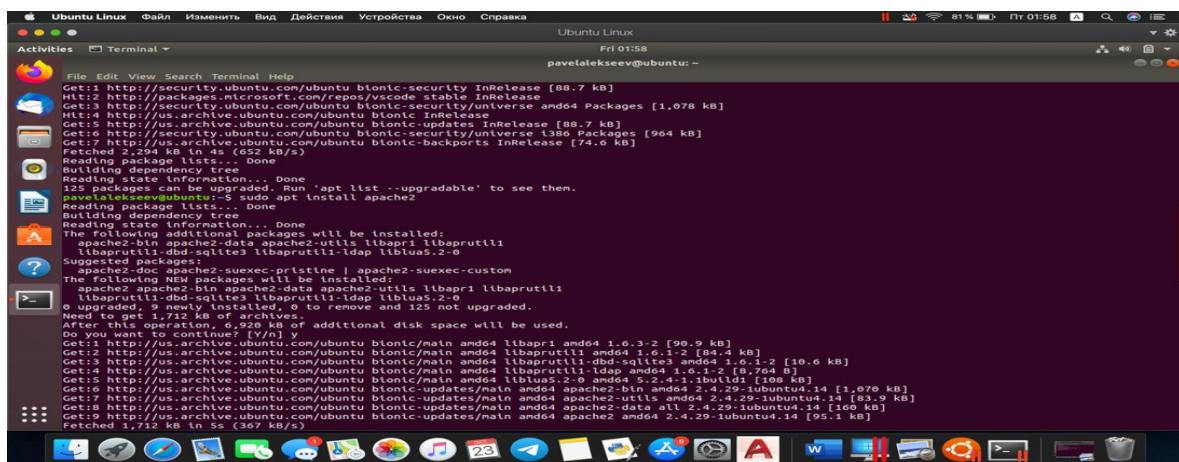
```
pavelalekseev@ubuntu:~$ ping -c 3 ya.ru
PING ya.ru (87.250.250.242) 56(84) bytes of data.
64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=128 time=17.9 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=128 time=27.5 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=128 time=33.9 ms
...
-- ya.ru ping statistics --
6 packets transmitted, 0 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 17.943/26.479/33.983/6.591 ms
pavelalekseev@ubuntu:~$ ping -D ya.ru
PING ya.ru (87.250.250.242) 56(84) bytes of data.
64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=128 time=15.5 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=128 time=30.6 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=128 time=25.5 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=4 ttl=128 time=15.9 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=5 ttl=128 time=29.5 ms
...
[?] Stopped ping -c 3 ya.ru
pavelalekseev@ubuntu:~$ ping -D ya.ru
PING ya.ru (87.250.250.242) 56(84) bytes of data.
[1603403013.181793] 64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=128 time=15.7 ms
[1603403013.181793] 64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=128 time=41.6 ms
[1603403014.174034] 64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=128 time=30.9 ms
[1603403014.174034] 64 bytes from ya.ru (87.250.250.242): icmp_seq=4 ttl=128 time=31.7 ms
[1603403016.179316] 64 bytes from ya.ru (87.250.250.242): icmp_seq=5 ttl=128 time=31.7 ms
[1603403017.197001] 64 bytes from ya.ru (87.250.250.242): icmp_seq=6 ttl=128 time=46.5 ms
...
-- ya.ru ping statistics --
6 packets transmitted, 6 received, 0% packet loss, time 5013ms
rtt min/avg/max/mdev = 15.779/33.253/46.598/9.685 ms
pavelalekseev@ubuntu:~$
```



## Установка,настройка и запуск WEB-сервера Apache

**Apache**- сервер который контролирует доступ веб-пользователя к различным серверным файлам.

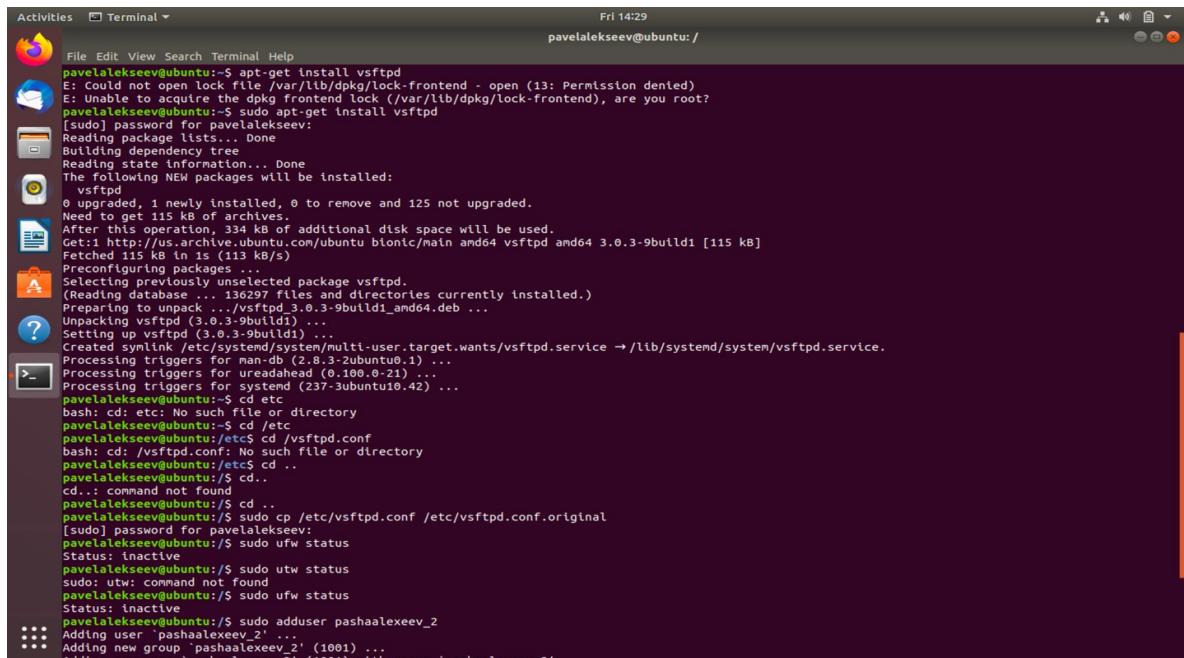
### Настройка установка и запуска apache2



# Установка, запуск, настройка FTP-сервера

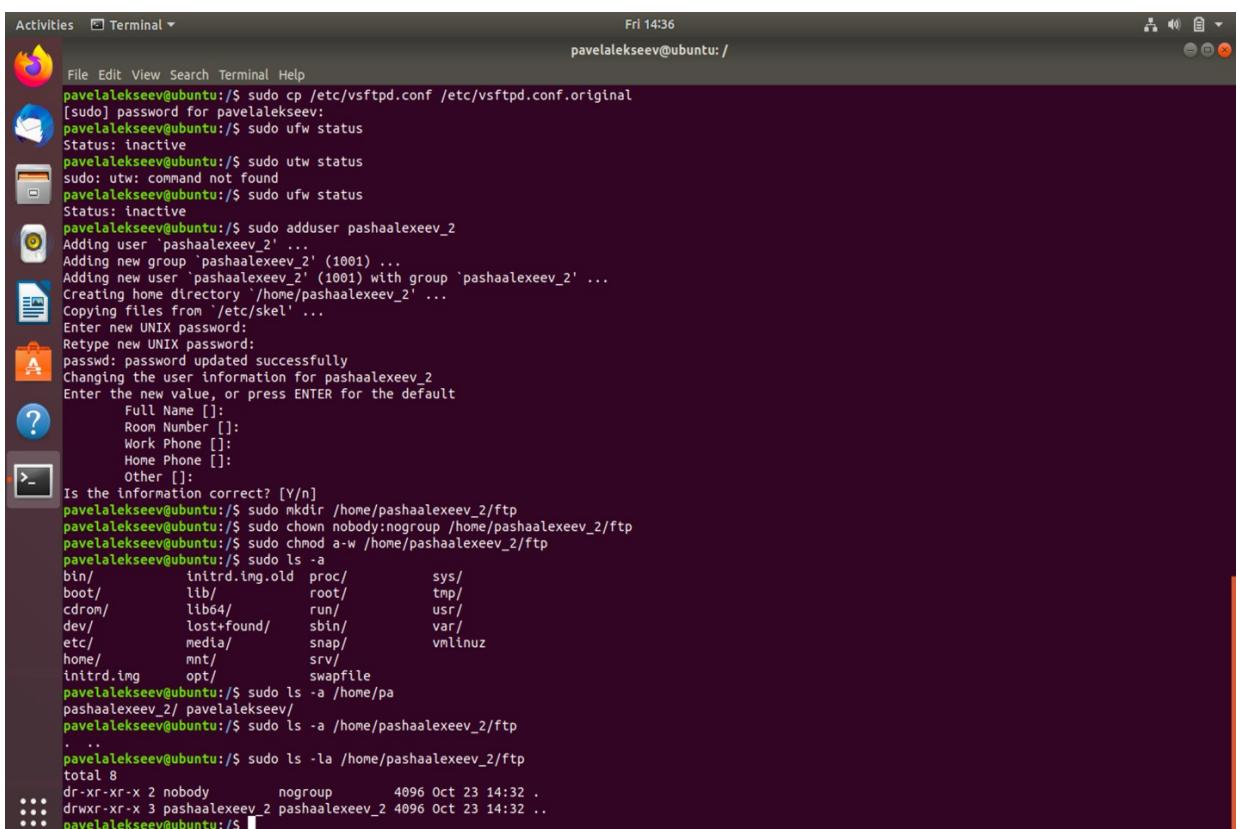
**FTP-сервер**-оптимальное решение для тех случаев когда нужно загрузить файлы на сервер, дать доступ коллегам.(протокол приема и передачи данных)

- Установил



```
pavelalekseev@ubuntu:~$ apt-get install vsftpd
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
[pavelalekseev@ubuntu:~]$ sudo apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 125 not upgraded.
Need to get 115 kB of archives.
After this operation, 334 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 vsftpd amd64 3.0.3-9build1 [115 kB]
Fetched 115 kB in 0s (113 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 136297 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.3-9build1_amd64.deb ...
Unpacking vsftpd (3.0.3-9build1) ...
Setting up vsftpd (3.0.3-9build1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /lib/systemd/system/vsftpd.service.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.42) ...
[pavelalekseev@ubuntu:~]$ cd etc
bash: cd: etc: No such file or directory
[pavelalekseev@ubuntu:~]$ cd /etc
[pavelalekseev@ubuntu:~/etc]$ cp vsftpd.conf /etc/vsftpd.conf
bash: cd: /vsftpd: No such file or directory
[pavelalekseev@ubuntu:~/etc]$ cd ..
[pavelalekseev@ubuntu:~]$ cd ...
[pavelalekseev@ubuntu:~]$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original
[sudo] password for pavelalekseev:
[pavelalekseev@ubuntu:~]$ sudo ufw status
Status: inactive
[pavelalekseev@ubuntu:~]$ sudo utw status
sudo: utw: command not found
[pavelalekseev@ubuntu:~]$ sudo ufw status
Status: inactive
[pavelalekseev@ubuntu:~]$ sudo adduser pashaalexeev_2
Adding user `pashaalexeev_2' ...
Adding new group `pashaalexeev_2' (1001) ...
[pavelalekseev@ubuntu:~]$
```

- Создаю еще одного пользователя для работы с сервером , даю ему права , а также копирую сервер



```
pavelalekseev@ubuntu:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original
[sudo] password for pavelalekseev:
[pavelalekseev@ubuntu:~]$ sudo ufw status
Status: inactive
[pavelalekseev@ubuntu:~]$ sudo utw status
sudo: utw: command not found
[pavelalekseev@ubuntu:~]$ sudo ufw status
Status: inactive
[pavelalekseev@ubuntu:~]$ sudo adduser pashaalexeev_2
Adding user `pashaalexeev_2' ...
Adding new group `pashaalexeev_2' (1001) ...
Adding new user `pashaalexeev_2' (1001) with group `pashaalexeev_2' ...
Creating home directory `/home/pashaalexeev_2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for pashaalexeev_2
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n]
[pavelalekseev@ubuntu:~]$ sudo mkdir /home/pashaalexeev_2/ftp
[pavelalekseev@ubuntu:~]$ sudo chown nobody:nogroup /home/pashaalexeev_2/ftp
[pavelalekseev@ubuntu:~]$ sudo chmod a-w /home/pashaalexeev_2/ftp
[pavelalekseev@ubuntu:~]$ sudo ls -a
bin/          initrd.img.old  proc/        sys/
boot/         lib/           root/       tmp/
cdrom/        lib64/         run/        usr/
dev/          lost+found/   sbin/       var/
etc/          media/        snap/      vmlinuz
home/         mnt/          srv/       swapfile
initrd.img    opt/          swapfile
[pavelalekseev@ubuntu:~]$ sudo ls -a /home/pashaalexeev_2
pashaalexeev_2/  pavelalekseev/
[pavelalekseev@ubuntu:~]$ sudo ls -a /home/pashaalexeev_2/ftp
...
[pavelalekseev@ubuntu:~]$ sudo ls -la /home/pashaalexeev_2/ftp
total 8
dr-xr-xr-x 2 nobody     nogroup        4096 Oct 23 14:32 .
drwxr-xr-x 3 pashaalexeev_2 pashaalexeev_2 4096 Oct 23 14:32 ..
[pavelalekseev@ubuntu:~]$
```

- Далее открываю содержимое данного файла меню и добавлю функции

```

Activities Terminal Fri 14:36
pavelalekseev@ubuntu:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original
[psudo] password for pavelalekseev:
pavelalekseev@ubuntu:~$ sudo ufw status
Status: inactive
pavelalekseev@ubuntu:~$ sudo ufw status
sudo: ufw: command not found
pavelalekseev@ubuntu:~$ sudo ufw status
Status: inactive
pavelalekseev@ubuntu:~$ sudo adduser pashaalexeev_2
Adding user `pashaalexeev_2' ...
Adding new group `pashaalexeev_2' (1001) ...
Adding new user `pashaalexeev_2' (1001) with group `pashaalexeev_2' ...
Creating home directory `/home/pashaalexeev_2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for pashaalexeev_2
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
pavelalekseev@ubuntu:~$ sudo mkdir /home/pashaalexeev_2/ftp
pavelalekseev@ubuntu:~$ sudo chown nobody:nogroup /home/pashaalexeev_2/ftp
pavelalekseev@ubuntu:~$ sudo chmod a-w /home/pashaalexeev_2/ftp
pavelalekseev@ubuntu:~$ sudo ls -a
bin/      intrd.img.old  proc/      sys/
boot/     lib/          root/     tmp/
cdrom/    lib64/        run/      usr/
dev/      lost+found/   sbin/     var/
etc/      media/       snap/    vmlinuz
home/    mnt/          srv/      swapfile
initrd.img  opt/        swapfile
pavelalekseev@ubuntu:~$ sudo ls -a /home/pashaalexeev_2
pashaalexeev_2/  pavelalekseev/
pavelalekseev@ubuntu:~$ sudo ls -a /home/pashaalexeev_2/ftp
.
.
.
pavelalekseev@ubuntu:~$ sudo ls -la /home/pashaalexeev_2/ftp
total 8
dr-xr-xr-x 2 nobody    nogroup        4096 Oct 23 14:32 .
drwxr-xr-x 3 pashaalexeev_2 pashaalexeev_2 4096 Oct 23 14:32 ..
pavelalekseev@ubuntu:~$ 

```

Write `enable=yes`, `chroot_local_user = yes`, добавил юзера в `lockal root`, определил доступность соединения `pasv_min_port` и `pasv_max_port`, при помощи `userlist` дал права только нашему пользователю.

- Запуск

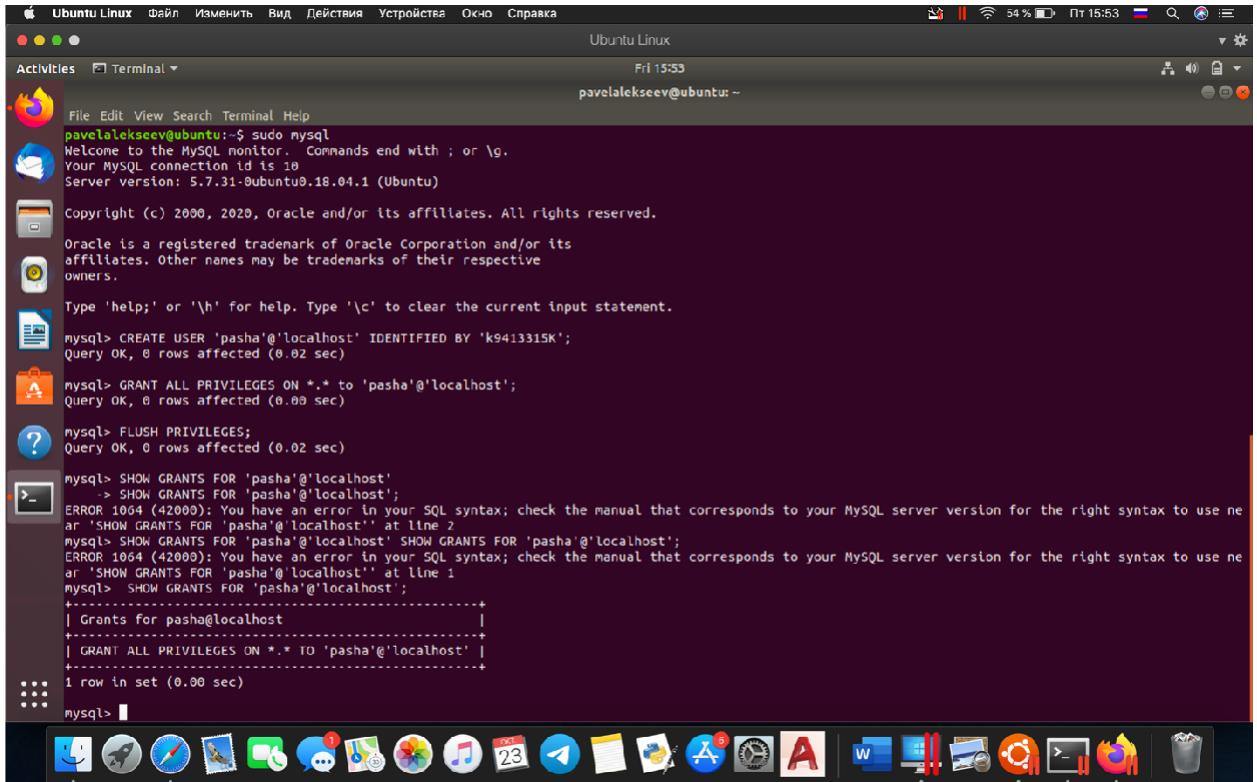
```

Activities Terminal Fri 15:33
pavelalekseev@ubuntu:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original
[psudo] password for pavelalekseev:
pavelalekseev@ubuntu:~$ sudo chown pashaalexeev_2:pashaalexeev_2 /home/pashaalexeev_2/ftp/files
pavelalekseev@ubuntu:~$ echo "vsftpd sample file" | sudo tee /home/pashaalexeev_2/ftp/files/sample.txt
vsftpd sample file
pavelalekseev@ubuntu:~$ sudo nano /etc/vsftpd.conf
pavelalekseev@ubuntu:~$ echo 'pashaalexeev_2' | sudo tee -a /etc/vsftpd.userlist
[sudo] password for pavelalekseev:
pashaalexeev_2
pavelalekseev@ubuntu:~$ cat /etc/
Display all 223 possibilities? (y or n)
pavelalekseev@ubuntu:~$ cat /etc/vsftpd.userlist
pashaalexeev_2
pavelalekseev@ubuntu:~$ sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-10-23 14:00:35 MSK; 1h 32min ago
     Main PID: 24091 (vsftpd)
        Tasks: 1 (limit: 2320)
       CGroup: /system.slice/vsftpd.service
               └─24091 /usr/sbin/vsftpd /etc/vsftpd.conf

Oct 23 14:00:35 ubuntu systemd[1]: Starting vsftpd FTP server...
Oct 23 14:00:35 ubuntu systemd[1]: Started vsftpd FTP server.
pavelalekseev@ubuntu:~$ 

```

# Создал пользователя для удаленного SSH-подключения с возможностями администратора



```
pavelalekseev@ubuntu:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.31-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'pasha'@'localhost' IDENTIFIED BY 'k9413315K';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'pasha'@'localhost';
Query OK, 0 rows affected (0.09 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW GRANTS FOR 'pasha'@'localhost';
+-----+
| Grants for pasha@localhost          |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'pasha'@'localhost' |
+-----+
1 row in set (0.00 sec)

mysql>
```

Сделал пользователя pasha. И при помощи команды grant all privileges дал ему права как у администратора.

## Лабораторная работа №2

### “Изучение технологии Ethernet”

### Ethernet

Ethernet-это семейство технологий пакетной передачи данных между устройствами для компьютерных и промышленных сетей. Даже название данной технологии (Ethernet (“эфирная сеть”)) отражает первоначальный принцип работы данной разработки: всё, передаваемое одним узлом, одновременно принимается всеми остальными. В настоящее время практически всегда подключение происходит через коммутаторы , так что кадры, отправляемые одним узлом, доходят лишь до адресата - это повышает скорость работы и безопасность сети.

**IEEE 802** — группа стандартов семейства IEEE, касающихся локальных вычислительных сетей (LAN) и сетей мегаполисов (MAN).

В частности, стандарты IEEE 802 ограничены сетями с пакетами переменной длины.

Число 802 являлось следующим свободным номером для стандарта, хотя часто ассоциируется с датой принятия стандарта — февраль 1980 года.

Службы и протоколы, указанные в IEEE 802, находятся на двух нижних уровнях (канальный и физический) семиуровневой сетевой модели OSI.

## Обзор:

- **1BASE5** — также известный, как StarLAN, стал первой модификацией Ethernet технологии, использующей витую пару. Работал на скорости 1 Мбит/с, но не нашёл коммерческого применения.
- **1000BASE-T**, IEEE 802.3ab — основной гигабитный стандарт, опубликованный в 1999 году, использует витую пару категории 5e. В передаче данных участвуют 4 пары, каждая пара используется одновременно для передачи по обоим направлениям со скоростью — 250 Мбит/с.
- **10GBASE-CX4** — технология 10-гигабитного Ethernet для коротких расстояний (до 15 метров), используется медный кабель CX4 и коннекторы (устройство для соединения электрических цепей между собой) InfiniB .

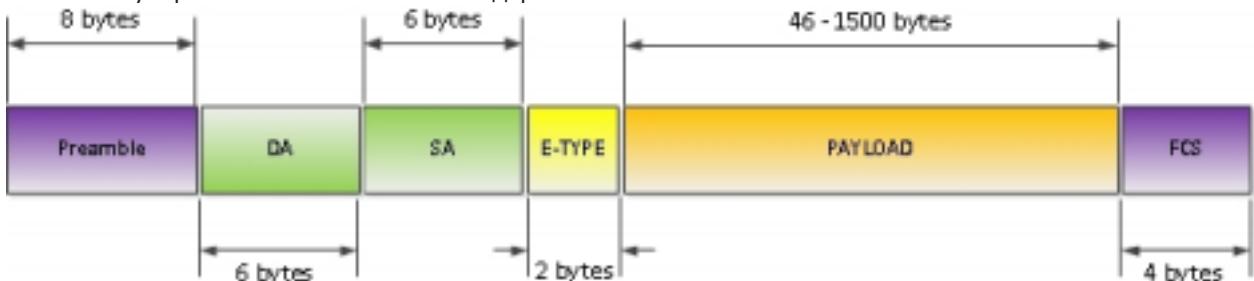
Стандарт 10-гигабитного Ethernet включает в себя семь стандартов физической среды для LAN, MAN и WAN(глобальная вычислительная сеть). В настоящее время он описывается поправкой IEEE 802.3ae и должен войти в следующую ревизию стандарта IEEE 802.3.

## Структура Ethernet-фрейма. Стандарты.

**Ethernet-фрейм** - фрагмент данных протокола канального уровня модели OSI, передаваемый по линии связи.

### 1) Ethernet II

Данный формат был создан в сотрудничестве 3-х компаний – DEC, Intel и Xerox. В связи с этим, стандарт также носит название DIX Ethernet standard. Данная версия стандарта была опубликована в 1982г (первая версия, Ethernet I – в 1980г. Различия в версиях небольшие, формат в целом остался неизменным). В 1997г. году данный стандарт был добавлен IEEE к стандарту 802.3, и на данный момент, подавляющее большинство пакетов в Ethernet сетях инкапсулированы согласно этого стандарта.



**Preamble** – последовательность бит, по сути, не являющаяся частью ETH заголовка определяющая начало Ethernet фрейма.

**DA** (Destination Address) – MAC адрес назначения, может быть юникастом, мультикастом, бродкастом.

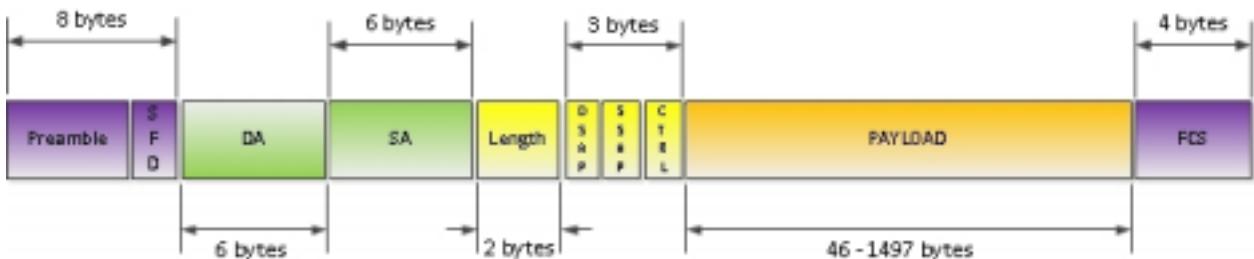
**SA** (Source Address) – MAC адрес отправителя. Всегда юникаст.

**E-TYPE** (EtherType) – Идентифицирует L3 протокол (к примеру 0x0800 –Ipv4, 0x86DD – IPv6, 0x8100- указывает что фрейм тегирован заголовком 802.1q, и т.д.

**Payload** – L3 пакет размером от 46 до 1500 байт

**FCS** (Frame Check Sequences) – 4 байтное значение CRC используемое для выявления ошибок передачи. Вычисляется отправляющей стороной, и помещается в поле FCS. Принимающая сторона вычисляет данное значение самостоятельно и сравнивает с полученным.

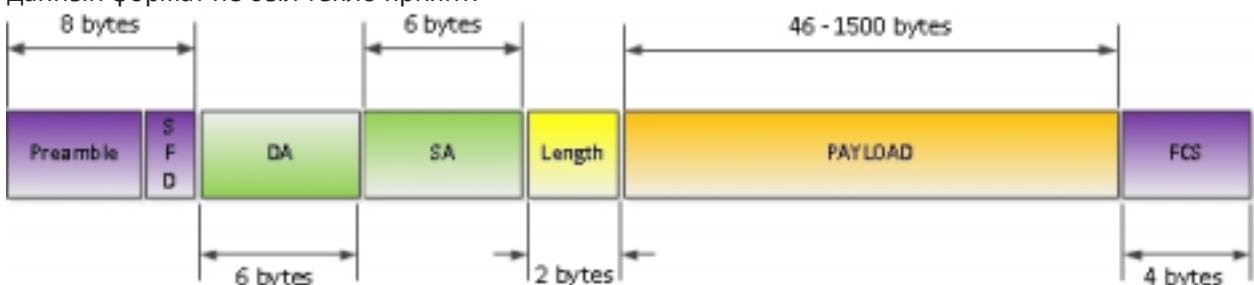
## 2) Ethernet\_802.3/802.2 (802.3 with LLC header)



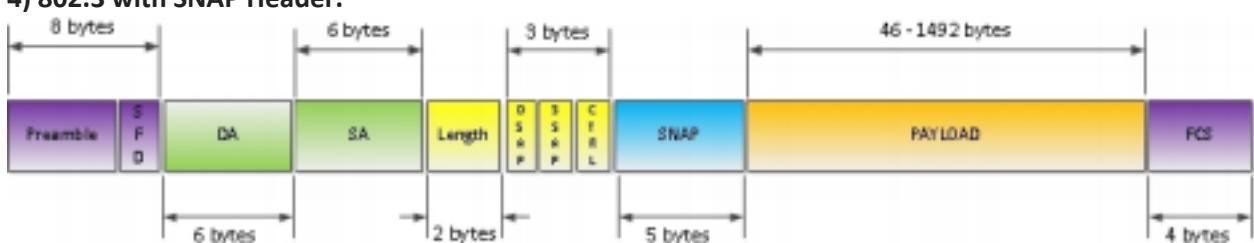
После Ethernet II был создан данный формат. Сразу видно, что поле E-TYPE преобразовано в поле Length, которое указывало на количество байт следующее за этим полем и до поля FCS. Теперь, понять у кого длиннее можно было уже на втором уровне системы OSI. Но, указатель на тип протокола этого уровня был нужен, и IEEE дало миру следующую инновацию — два поля по 1 байту — Source Service Access Point (SSAP) и Destination Service Access Point (DSAP). Цель, также самая, — идентифицировать вышестоящий протокол, но какова реализация! Теперь, благодаря наличию двух полей в рамках одной сессии пакет мог передаваться между разными протоколами, либо же один и тот же протокол мог по-разному называться на двух концах одной сессии. В IEEE фрейм формате появляется 1 байтное поле Control. Отвечающее за Connection-less или же Connection-oriented соединение.

## 3) «Raw» 802.3

Данный «недостандарт» явил в мир Novell. Заполучив ещё в процессе разработки спецификации стандарта 802.3/802.2, и выкинув LLC заголовок, в Novell получили вполне себе неплохой фрейм формат, но одним существенным недостатком – отсутствием возможности указания вышестоящего протокола. Поэтому они ограничили этот фрейм-формат исключительно IPX протоколом, который сами же и поддерживали. Но публикой данный формат не был тепло принят.



## 4) 802.3 with SNAP Header.



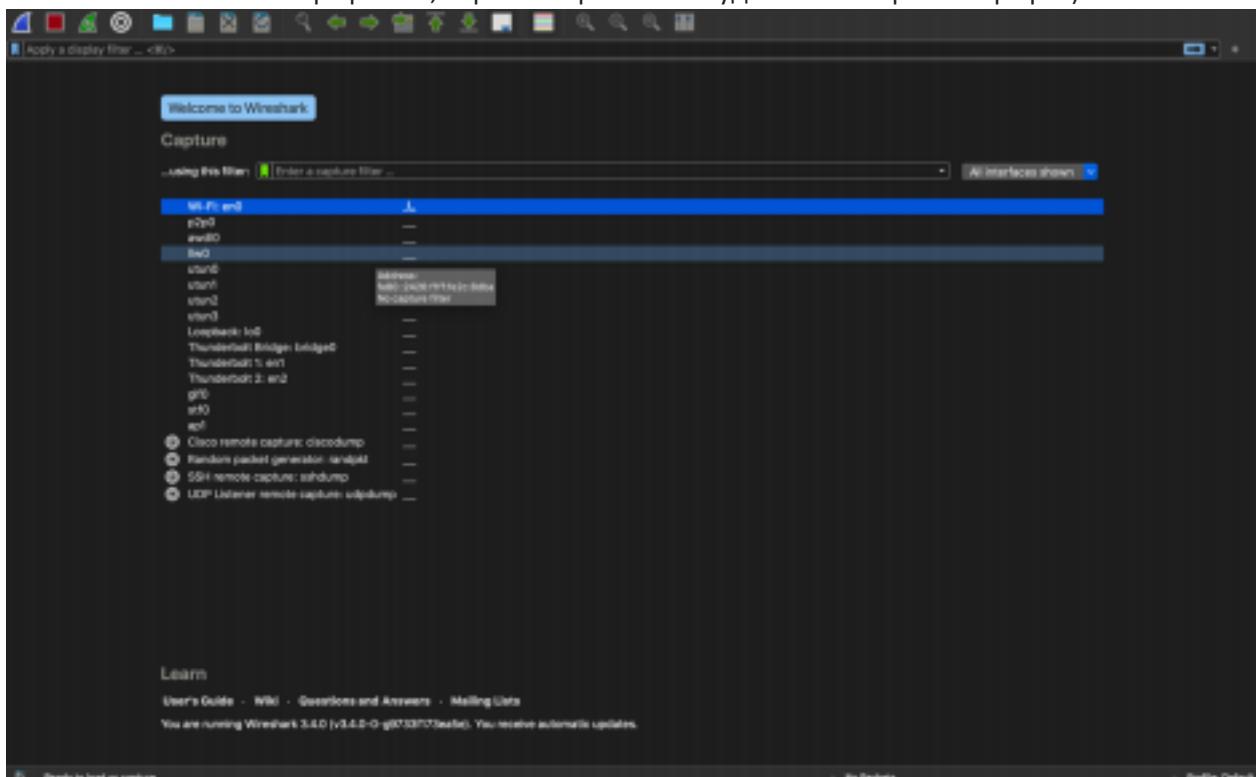
В комитет IEEE приходило осознание того, что номера протоколов и деньги кончаются. Пользователи требовали 3-х байтный LLC заголовок. И из-за нехватки номеров протоколов (их всего могло быть 128), IEEE вводит новый стандарт фрейма Ethernet SNAP . Основное нововведение — добавление 5-ти байтного поля Subnetwork Access Protocol (SNAP), которое в свою очередь состоит из двух частей – 3х байтного поля Organizationally Unique Identifier (OUI) и 2x байтного Protocol ID (PID).

**OUI** (vendor code) – позволяет идентифицировать проприетарные протоколы указанием вендора.

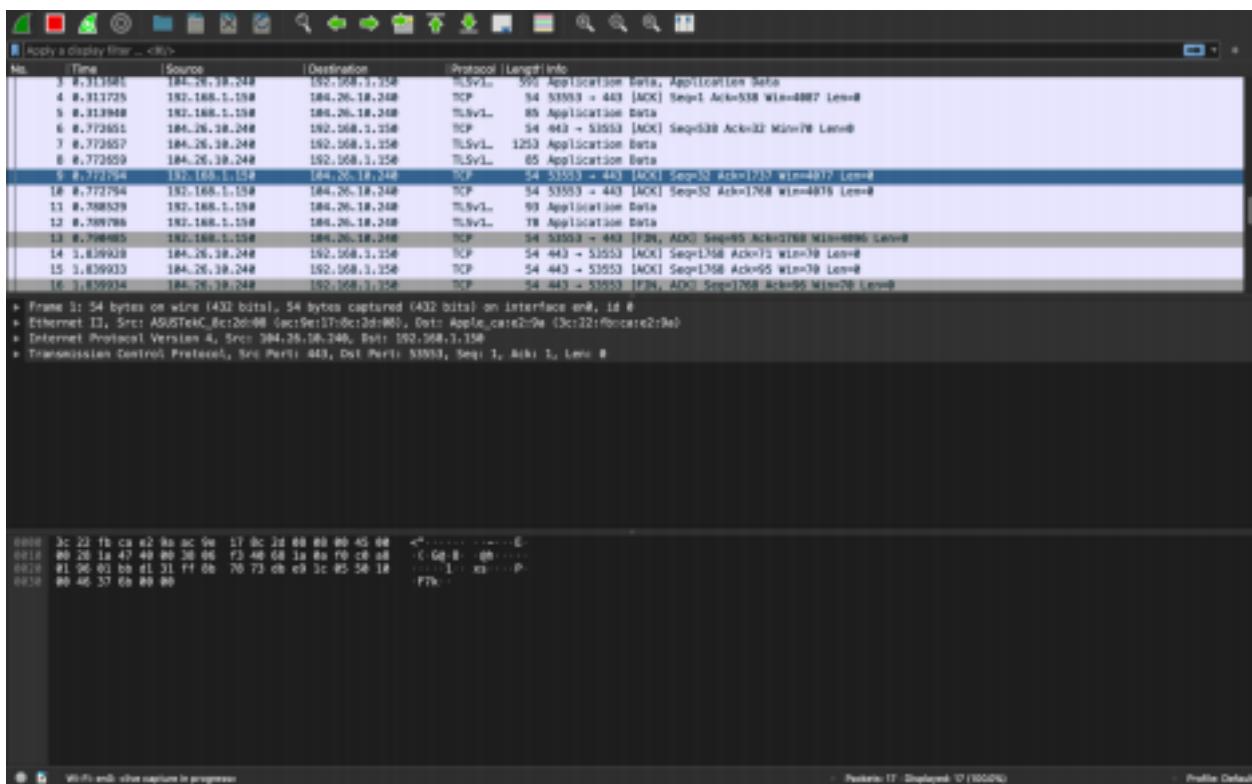
**PID** это, по сути, то же поле EtherType из DIX Ethernet II — 2 байта под указание протокола вышестоящего уровня.

## Wireshark установка и знакомство.

- Скачиваем новейшую версию для нашей ОС с официального сайта.
- Запускаем и видим главную страницу приложения.(здесь отображены все наши сетевые интерфейсы , через которые мы и будем анализировать трафик)



Например возьмем интерфейс en0

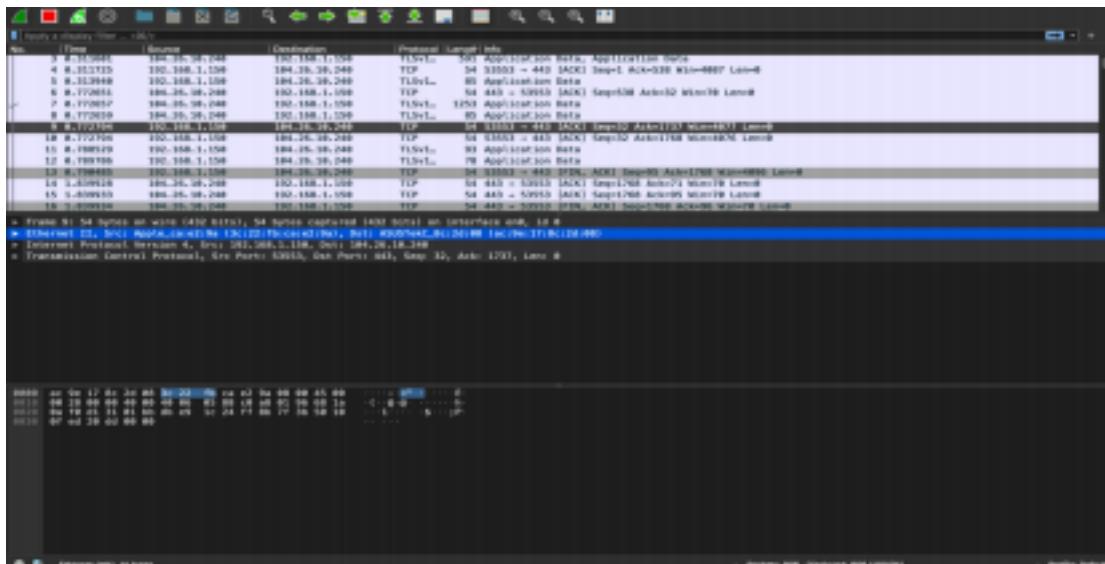


После запуска мы можем наблюдать генерацию нашего трафика.

## Характеристики трафика

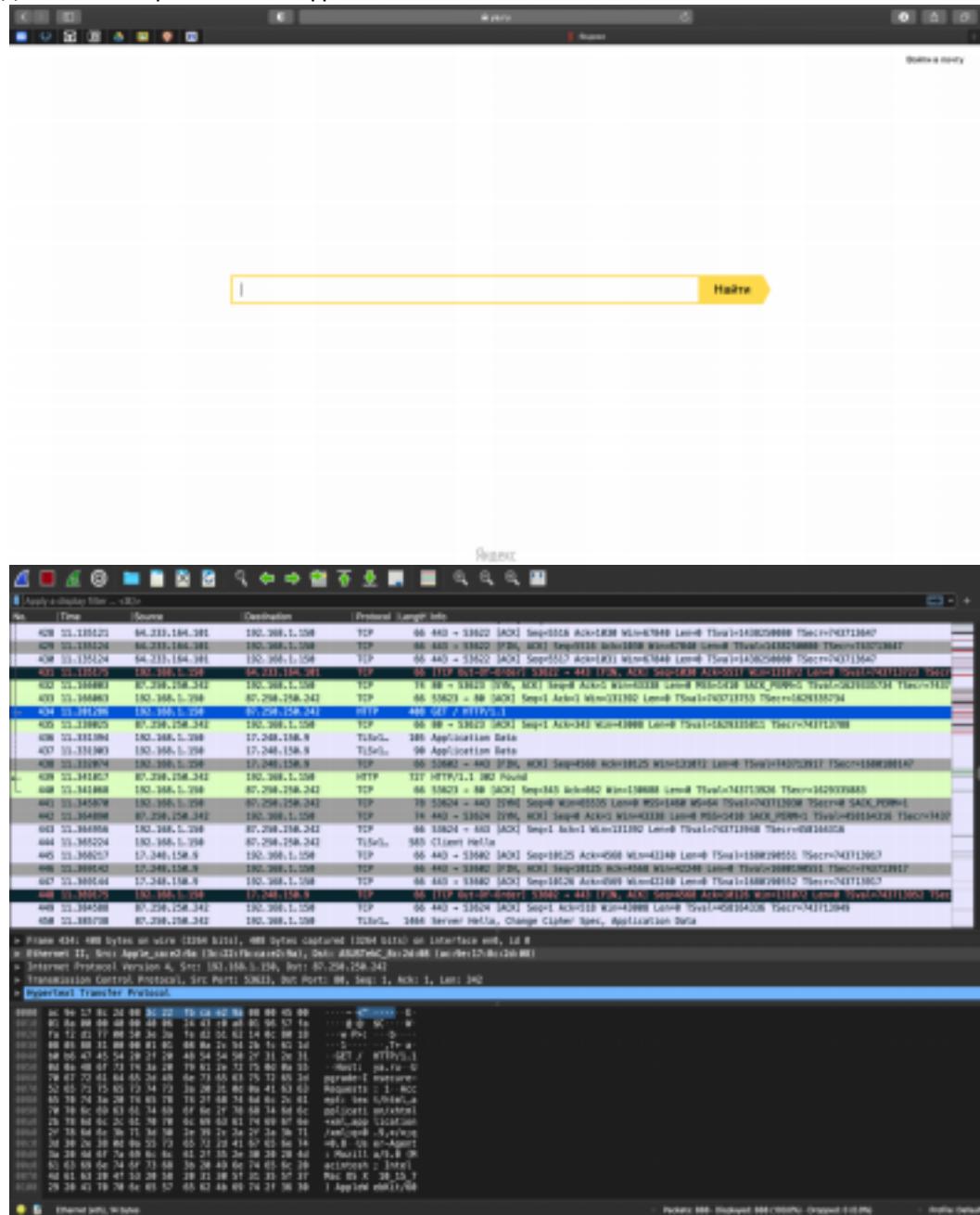
- No** – номер под которым идет фрейм данного интерфейса.
- Time** - анализирование сетевых данных, поступающих на сеть компьютера(в миллисекундах, с разбивкой на наносекунды.).
- Source** – начальный адрес.
- Destination** - адрес назначения.
- Protocol** - протокол, используемый в кадре Ethernet, IP - пакете или сегменте TCP(ARP, DNS, TCP, HTTP и др.)
- Length** - длина кадра в байтах

Если мы выберем один из фреймов, то можем посмотреть его подробное описание



## Генерация трафика.

1. Для начала запустим Wireshark и запустим анализ трафика как было сделано ранее 2. В поисковую строку вбиваю любой запрос
  3. Находим свой запрос в списке фреймов.



Idx	Time	Source	Destination	Protocol	Length	Info
437	31-331983	192.168.1.198	17.248.198.9	TLSv1..	98	Application Data
438	31-330774	192.168.1.198	17.248.198.9	TCP	68	53623 - > [ACK] Seq=1568 Win=38129 Len=8 TSeq=143713812 TSec=r14381188147
439	31-341811	87.218.3.238	182.158.3.238	HTTP	272	HTTP/1.1 202 Found
440	31-341868	192.168.1.198	87.158.3.242	TCP	68	53623 - > [ACK] Seq=343 Acks=682 Win=138688 Len=8 TSeq=143713826 TSec=r14381188148
441	31-348576	192.168.1.198	87.158.3.242	TCP	78	53624 - 440 [FIN, ACK] Seq=65515 Len=8 MSS=1418 SACK_PERMITTED TSeq=14381188149
442	31-364986	87.258.256.342	182.168.1.198	TCP	74	53624 - 440 [FIN, ACK] Seq=10338 Len=8 MSS=1418 SACK_PERMITTED TSeq=14381188150
443	31-364956	192.168.1.198	87.258.256.242	TCP	68	53624 - 440 [ACK] Seq=1 Win=131862 Len=8 TSeq=143713946 TSec=r45184316
444	31-365224	192.168.1.198	87.258.256.242	TLSv1..	583	Client Hello
445	31-366177	17.248.198.9	182.158.3.238	TCP	68	443 - 53682 [ACK] Seq=11254 Acks=40586 Win=42248 Len=8 TSeq=1438119851 TSec=r143713817
446	31-389383	17.248.198.9	182.158.3.238	TCP	68	443 - 53682 [FIN, ACK] Seq=18129 Acks=40586 Win=42248 Len=8 TSeq=1438119853 TSec=r143713817
447	31-389384	17.248.198.9	182.158.3.238	TCP	68	443 - 53682 [ACK] Seq=18130 Acks=40587 Win=42248 Len=8 TSeq=1438119852 TSec=r143713817
448	31-389385	192.168.1.198	17.248.198.9	TCP	68	L127_01-07-07-07-07-07
449	31-389458	87.258.256.342	182.168.1.198	TCP	68	443 - 53682 [ACK] Seq=1518 Win=42248 Len=8 TSeq=1438119849
450	31-389326	87.258.3.238	182.158.3.238	TLSv1..	1464	Server Hello, Change Cipher Spec, Application Data

## Описание

- 1) Фрейм 434 – это и есть наш ya.ru.
  - 2) На этой же строке видно с какого ip-адреса был отправлен запрос(87.250.250.242). 3) Ip-адрес получателя(87.250.250.242)

- 4) Протокол (HTTP).
  - 5) Длина кадра (408 байт).

Так же в строке, которая ниже, мы можем наблюдать подробную информацию про наш фрейм.

- Frame 434: 488 bytes on wire (3884 bits), 488 bytes captured (3884 bits) in interface en0, id 8
- Ethernet II, Src: Application2 [00:0C:29:00:00:0A], Dst: ApplicationC [00:0C:29:00:00:B0] (GeForce370/8c:29:00:B0)
- Internet Protocol Version 4, Src: 192.168.1.198, Dst: 87.236.29.242
- Transmission Control Protocol, Src Port: 5843, Dst Port: 80, Seq: 1, Ack: 1, Len: 342
- Hypertext Transfer Protocol

## Описание

- 1) На первой строке видно сколько байт(бит) было передано (408 байт (3264 бит)), сколько захватил мой интерфейс (408 байт(3264 бит)). Следовательно все работает исправно. 2) На второй строчке мы видим стандарт Ethernet (в моем случае это Ethernet II), src – источник ( с MAC-адресом) (Apple\_ca:e2:9a (3c:22:fb:ca:e2:9a)), dst – получатель (имя (MAC-адреса))(ASUSTekC\_8c:2d:08(ac:9e:17:8c:2f:08)).
  - 3) Третья строка показывает какой протокол используется во фрейме (IPv4) и как и в (2) источника и получателя .
  - 4) В четвертой строке расписана информация про протокол TCP, который задействовал данный фрейм.
  - 5) В пятой строке указан основной протокол – HTTP.

Ещё ниже есть блок в котором отображается содержимое нашего пакета (в реальном виде и в виде обозначения в 16 системе счисления)

## **Лабораторная работа №3.**

### **Изучение стандартов TCP/IP**

**Internet Protocol** (IP, досл. «межсетевой протокол») — маршрутизуемый протокол сетевого уровня стека TCP/IP. Именно IP стал тем протоколом, который объединил отдельные компьютерные сети во всемирную сеть Интернет. Неотъемлемой частью протокола является адресация сети.

IP объединяет сегменты сети в единую сеть, обеспечивая доставку пакетов данных между любыми узлами сети через произвольное число промежуточных узлов (маршрутизаторов). Он классифицируется как протокол сетевого уровня по сетевой модели OSI. IP не гарантирует надёжной доставки пакета до адресата — в частности, пакеты могут прийти не в том порядке, в котором были отправлены, продублироваться казаться повреждёнными или не прийти вовсе. Гарантию безошибочной доставки

пакетов дают некоторые протоколы более высокого уровня — транспортного уровня сетевой модели OSI, — например, TCP, которые используют IP в качестве транспорта.

При доставке IP пакета он проходит через разные каналы доставки. Возможно возникновение ситуации, когда размер пакета превысит возможности узла системы связи. В этом случае протокол предусматривает возможность дробления пакета на уровне IP в процессе доставки. Соответственно, к конечному получателю пакет придет в виде нескольких пакетов, которые необходимо собрать в один перед дальнейшим анализом. Возможность дробления пакета с последующей сборкой называется IP фрагментацией.

## Стандарт IPv4

**IPv4**- четвёртая версия интернет протокола (IP). Первая широко используемая версия. IPv4 использует 32-битные (четырёхбайтные) адреса, ограничивающие адресное пространство  $4\ 294\ 967\ 296$  ( $2^{32}$ ) возможными уникальными адресами.

Традиционной формой записи IPv4 является запись в виде четырёх десятичных чисел (от 0 до 255), разделённых точками. Через дробь указывается длина маски подсети.

После определенного времени люди начали понимать, что адресов просто не хватит и была придумана классовая адресация.

## Классовая адресация

**Класс А:** Первый бит адреса равен нулю, таким образом, класс А занимает половину всего адресного пространства. Адрес сети занимает 7 бит, адрес узла — 24 бита, следовательно класс А содержит 128 подсетей по 16 777 216 адресов в каждой. ( 0.XXX.XXX.XXX — 127.XXX.XXX.XXX )

**Класс В:** Адрес начинается с битов 1,0, таким образом, класс В занимает четверть всего адресного пространства. Адрес сети занимает 14 бит, адрес узла — 16, следовательно класс В содержит 16 384 подсетей по 65 536 адресов в каждой (128.0.XXX.XXX — 191.255.XXX.XXX)

**Класс С:** Адрес начинается с битов 1,1,0, таким образом, класс С занимает 1/8 адресного пространства. Адрес сети занимает 21 бит, адрес узла — 8 бит, следовательно класс С содержит 2 097 152 сетей по 256 адресов в каждой. (192.0.0.XXX — 223.255.255.XXX)

**Класс D:** Адрес начинается с битов 1,1,1,0. Класс D занимает 1/16 адресного пространства. Используется для многоадресной рассылки.  
(224.XXX.XXX.XXX — 239.XXX.XXX.XXX)

**Класс E:** Адрес начинается с битов 1,1,1,1. Такие адреса запрещены. Зарезервировано для использования в будущем. (240.XXX.XXX.XXX — 255.XXX.XXX.XXX)

Для сравнения прикрепляю шкалу распределения:

A	B	C	D	E
---	---	---	---	---

## Заголовки ip-пакета

- **Версия**

Первым полем заголовка пакета является версия протокола размером в четыре бита. Для IPv4 это 4.

- **Размер заголовка (Internet Header Length)**

Следующие четыре бита содержат размер заголовка пакета в 32-битных словах. Поскольку число опций не постоянно, указание размера важно для отделения заголовка от данных. Минимальное значение равно 5 , максимальное — 15 .

- **Differentiated Services Code Point (DSCP)**

Изначально называлось «тип обслуживания», в настоящее время определяется, как «Differentiated Services». Используется для разделения трафика на классы обслуживания, например, для установки чувствительному к задержкам трафику, такому как VoIP, большего приоритета.

- **Указатель перегрузки (Explicit Congestion Notification, ECN)**

Предупреждение о перегрузке сети без потери пакетов. Является необязательной функцией и используется только если оба хоста её поддерживают.

- **Размер пакета**

16-битный полный размер пакета в байтах, включая заголовок и данные. Минимальный размер равен 20 байтам (заголовок без данных), максимальный — 65535 байт. Хосты должны поддерживать передачу пакетов размером до 576 байт, но современные реализации обычно поддерживают гораздо больший размер. Пакеты большего размера, чем поддерживает канал связи, фрагментируются.

- **Идентификатор**

Преимущественно используется для идентификации фрагментов пакета, если он был фрагментирован. Существуют эксперименты по его

использованию для других целей, таких как добавление информации о трассировке пакета для упрощения отслеживания пути пакета с подделанным адресом источника.

- **Флаги**

Поле размером три бита содержащее флаги контроля над фрагментацией. Биты, от старшего к младшему, означают:

- 0: Зарезервирован, должен быть равен 0.
- 1: Не фрагментировать
- 2: У пакета ещё есть фрагменты

Если установлен флаг «не фрагментировать», то в случае необходимости фрагментации такой пакет будет уничтожен. Может использоваться для передачи данных хостам, не имеющим достаточных ресурсов для обработки фрагментированных пакетов. Флаг «есть фрагменты» должен быть установлен в 1 у всех фрагментов пакета, кроме последнего. У нефрагментированных устанавливается в 0 — такой пакет считается собственным последним фрагментом.

- **Смещение фрагмента**

Поле размером в 13 бит, указывает смещение поля данных текущего фрагмента относительно начала поля данных первого фрагментированного пакета в блоках по 8 байт.

### **«Время жизни» (TTL) пакета**

Определяет максимальное количество маршрутизаторов на пути следования пакета. Наличие этого параметра не позволяет пакету бесконечно ходить по сети. Каждый маршрутизатор при обработке пакета должен уменьшить значение TTL на единицу. Пакеты, время жизни которых стало равно нулю, уничтожаются, а отправителю посыпается сообщение ICMP *Time Exceeded*. На отправке пакетов с разным временем жизни основана трассировка их пути прохождения. Максимальное значение TTL=255. Обычное начальное значение TTL=64 (зависит от ОС).

- **Протокол**

Указывает, данные какого протокола IP содержит пакет (например, TCP или ICMP). Присвоенные номера протоколов можно найти на сайте IANA.

- **Контрольная сумма заголовка**

16-битная контрольная сумма, используемая для проверки целостности заголовка. Каждый хост или маршрутизатор сравнивает контрольную сумму заголовка со значением этого поля и отбрасывает пакет, если они не совпадают. Целостность данных IP не проверяет —

она проверяется протоколами более высоких уровней (такими, как TCP или UDP), которые тоже используют контрольные суммы.

Поскольку TTL уменьшается на каждом шаге прохождения пакета, сумма тоже должна вычисляться на каждом шаге. Метод пересчёта контрольной суммы определён в RFC 1071.

- **Адрес источника**

32-битный адрес отправителя пакета. Может не совпадать с настоящим адресом отправителя из-за трансляции адресов .

- **Адрес назначения**

32-битный адрес получателя пакета. Также может меняться при трансляции адресов.

IPv4 Header Format																																							
Отс туп туп	ок тет	0								1								2								3													
Окт ет	Би т	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	0	Версия	Размер заголовка		Differentiated Services Code Point				<b>Explicit Congestion Notification</b>		Размер пакета (полный)																												
4	32	Идентификатор								Флаги								Смещение фрагмента																					
8	64	Время жизни				Протокол				Контрольная сумма заголовка																													
12	96	IP-адрес источника																																					
16	128	IP-адрес назначения																																					
20	160	Опции																																					
20 или 24+ или 19 или 2+	160 или 19 или 2+	Данные																																					

## Стандарт TCP

**TCP — сетевая модель** передачи данных, представленных в цифровом виде. Модель описывает способ передачи данных от источника информации к получателю. В модели предполагается прохождение информации через четыре уровня, каждый из которых описывается правилом (протоколом передачи)

Стек протоколов подразделяется на 4 уровня:

1. Прикладной
2. Транспортный
3. Межсетевой
4. Канальный

## Заголовки TCP-пакета

Структура заголовка						
Бит	0 — 3	4 — 9	10 — 15	16 — 31		
0	Порт источника, <b>Source Port</b>			Порт назначения, <b>Destination Port</b>		
32	Порядковый номер, <b>Sequence Number (SN)</b>					
64	Номер подтверждения, <b>Acknowledgment Number (ACK SN)</b>					
96	Длина заголовка, <b>(Data offset)</b>	Зарезервировано	Флаги	Размер Окна, <b>Window size</b>		
128	Контрольная сумма, <b>Checksum</b>		Указатель важности, <b>Urgent Point</b>			
160	Опции (необязательное, но используется практически всегда)					
160/192+	Данные					

- **Порт источника, Порт назначения**

Эти 16-битные поля содержат номера портов — числа, которые определяются по специальному списку. Порт источника идентифицирует приложение клиента, с которого отправлены пакеты. Ответные данные передаются клиенту на основании этого номера. Порт назначения идентифицирует порт, на который отправлен пакет.

- **Порядковый номер** (32 бита) — измеряется в байтах, и каждый переданный байт полезных данных увеличивает это значение на 1.
- **Номер подтверждения** (32 бита) — если установлен флаг ACK, то это поле содержит порядковый номер октета, который отправитель данного сегмента желает получить. Это означает, что все предыдущие

октеты были успешно получены. Каждая сторона подсчитывает свой Sequence number для переданных данных и отдельно Acknowledgement number для полученных данных. Sequence number каждой из сторон соответствует Acknowledgement number другой стороны.

- **Длина заголовка**

Длина заголовка занимает 4 бита и указывает значение длины заголовка, измеренное в 32-битовых словах. Минимальный размер составляет 20 байт , а максимальный — 60 байт . Длина заголовка определяет смещение полезных данных относительно начала сегмента.

- **Зарезервировано**

Зарезервировано (6 бит) для будущего использования и должно устанавливаться в ноль. Из них два (5-й и 6-й) уже определены:

- **CWR** — флаг установлен отправителем, чтобы указать, что получен пакет с установленным флагом ECE
- **ECE** — указывает, что данный узел способен на ECN и для указания отправителю о перегрузках в сети
- **Флаги (управляющие биты)**

Это поле содержит 6 битовых флагов:

- **URG** —Когда узел отправляет сегмент с URG флагом, то узел-получатель принимает его на отдельном канале.
- **ACK** — поле задействовано
- **PSH** —инструктирует получателя протолкнуть данные, накопившиеся в приёмном буфере, в приложение пользователя.
- **RST** — оборвать соединения, сбросить буфер
- **SYN** — синхронизация номеров последовательности **FIN** — флаг, будучи установлен, указывает на завершение соединения
- **Размер окна**

определяет количество байт данных после передачи которых отправитель ожидает подтверждения от получателя, что данные получены. Иначе говоря, получатель пакета располагает для приёма данных буфером длиной "размер окна" байт.

По умолчанию размер окна измеряется в байтах, поэтому ограничен  $2^{16}$  (65535) байтами. Однако благодаря TCP опции Window scale option этот размер может быть увеличен до 1 Гбайта. Чтобы задействовать эту опцию, обе стороны должны согласовать это в своих SYN сегментах.

- **Контрольная сумма**

Поле контрольной суммы — это 16-битное дополнение к сумме всех 16-битных слов заголовка (включая псевдозаголовок) и данных. Если сегмент, по которому вычисляется контрольная сумма, имеет длину не кратную 16-битам, то длина сегмента увеличивается до кратной 16-ти, за счёт дополнения к нему справа нулевых битов заполнения. Биты заполнения (0) не передаются в сообщении и служат только для расчёта контрольной суммы. При расчёте контрольной суммы значение самого поля контрольной суммы принимается равным 0.

- **Указатель важности**

16-битовое значение положительного смещения от порядкового номера в данном сегменте. Это поле указывает порядковый номер октета, которым заканчиваются важные данные. Поле принимается во внимание только для пакетов с установленным флагом URG.

- **Опции**

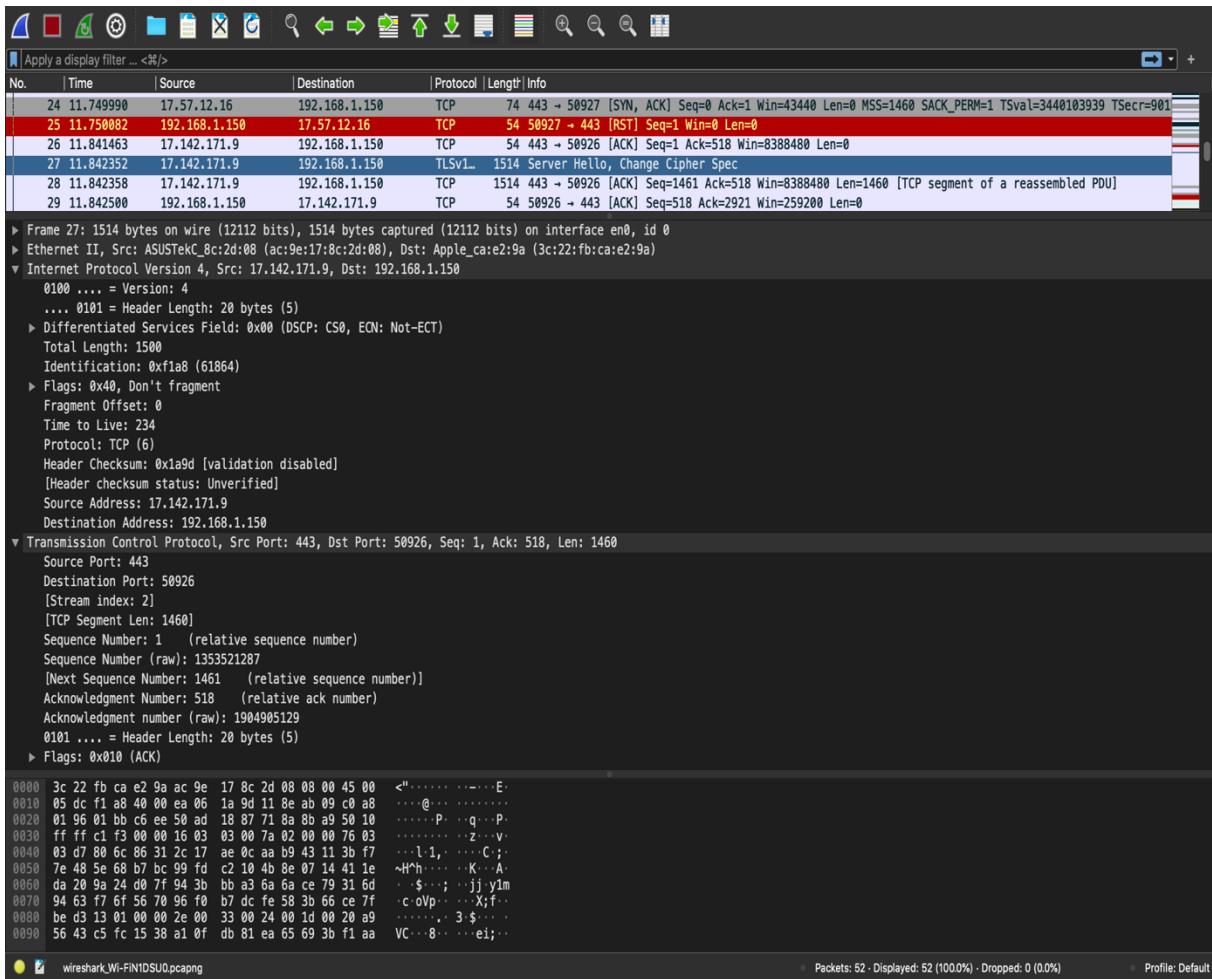
Могут применяться в некоторых случаях для расширения протокола. Иногда используются для тестирования.

### **Внешние и внутренние адреса**

IP адреса бывают внутренними и внешними. В чем тут дело. Как мы уже знаем, в IPv4 у нас ограниченное количество адресов - на всех не хватает. А так еще в интернете нельзя иметь два одинаковых адреса, ведь в таком случае нельзя будет однозначно понять кому передавать данные. Но как дать возможность выходить в интернет всем, кто захочет? Переходить на 6 версию? Можно, но это дорого и долго. Тут нам на помощь приходит технология NAT - Network Address Translation, а точнее ее надстройка PAT - Port address translation, суть которой в том, что много устройств могут выходить в интернет с одним и тем же адресом. Но как такое возможно, если мы сказали что нельзя иметь два одинаковых адреса?

Суть в том, что у вас есть ваш внутренний адрес, который выдает провайдер, с которым вы находитесь внутри локальной сети, а есть внешний адрес, который провайдер вам дает для выхода в интернет. И основная идея заключается в том, что несмотря на то, что у вас и у других пользователей одинаковые адреса, их можно отличить, благодаря тому, что к адресу добавляется порт, это уникальное значение после двоеточия, которое присваивает провайдер и которое является дополнительным идентификатором, позволяющим различать адреса. Это позволяет решать проблему с нехваткой адресов, и является дополнительным слоем безопасности.

# Генерирую трафик и анализирую



Анализируем TCP. Выбрал первый который увидел.

**Source port** и **Destination port** — это соответственно номера портов получателя и отправителя, идентифицирующие приложений на отправляющем и принимающем узлах.

**Sequence number** и **Acknowledgment number** — это порядковый номер сегмента и номер подтверждения, которые используются для надёжной доставки.

**Header length** — Это четырёхбитное поле, содержащее в себе длину заголовка TCP сегмента.

**Flags** — поле с флагами, которые используются в процессе обмена информацией и описывают дополнительное назначение сегмента. Могут принимать следующие значения:

- **URG** — поле «Указатель важности» задействовано.
- **ACK** — поле «Номер подтверждения» задействовано.

- • **PSH** — инструктирует получателя протолкнуть данные, накопившиеся в приёмном буфере, в приложение пользователя.
- • **RST** — оборвать соединения, сбросить буфер (очистка буфера).
- • **SYN** — синхронизация номеров последовательности.
- • **FIN** — флаг, будучи установлен, указывает на завершение соединения.

**Window** — содержит размер окна.

**Checksum** — контрольная сумма заголовка и данных.

**Urgent pointer** - признак важности (срочности) данного сегмента.

8

**Options** — дополнительное необязательное поле, которое может использоваться, например, для тестирования протокола.

Все ровно по тем же критериям что и выше.

## Лабораторная работа № 4 “HTTP Технология”

HTTP — широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам). Протокол задействован на 7 уровне (прикладной) в соответствие с моделью OSI. На данный момент актуальной является версия HTTP 1.1.

Протокол HTTP предполагает использование клиент-серверной структуры передачи данных. Клиентское приложение формирует запрос и отправляет его на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту. После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны аналогичным образом.

Простыми словами протокол HTTP позволяет нам использовать наши веб-браузеры. Протокол осуществляет обмен между пользовательскими приложениями, осуществляющими доступ к веб-ресурсам и веб-сервером. Но и это не единственная особенность HTTP. Так же данный протокол служит как “проводник” для других протоколов транспортного уровня (OSI), он передаёт информацию другим протоколам. API многих программных

продуктов также подразумевает использование HTTP для передачи данных — сами данные при этом могут иметь любой формат, например, XML или JSON.

Обычно передача данных осуществляется через соединение TCP/IP и по стандарту используется 80 порт (но может использоваться и любой другой)



## **Альтернатива**

Сегодня для передачи данных существует не только HTTP, но и другие протоколы. Одним из таких является SPDY. SPDY -это улучшенная версия HTTP, которая была создана для пользовательского удобства, а именно позволяет уменьшить скорость передачу данных и обеспечивает более надежную безопасность.

Увеличение скорости обеспечивается посредством сжатия, приоритизации и мультиплексирования дополнительных ресурсов, необходимых для веб-страницы, чтобы все данные можно было передать в рамках одного соединения. На данный момент поддержка протокола SPDY есть в браузерах Firefox, Chromium/Chrome, Opera, Internet Explorer и Amazon Silk.

## **Безопасность**

Протокол HTTP не использует шифрование, а обеспечивает свою безопасность при помощи расширений, которые делают упаковку в криптографические протоколы (SSL, TLS). HTTPS – это и есть данное расширение.

HTTPS широко используется для защиты информации от перехвата, а также, как правило, обеспечивает защиту в том случае, если сертификат проверяется на клиенте, и при этом приватный ключ сертификата не был скомпрометирован, пользователь не подтверждал использование неподписанного сертификата, и на компьютере пользователя не были внедрены сертификаты центра сертификации злоумышленника.

## Практическая работа с HTTP

Генерируем трафик и анализируем его, и посмотрим при помощи Wireshark, где применяется данный протокол (для этого в нашем браузере запустим браузер)

При помощи фильтра нашел нужный нам протокол

No.	Time	Source	Destination	Protocol	Length	Info
1855	154.838959	192.168.1.150	151.139.128.14	HTTP	358	GET /MFcvWaADAgEAME4wTDBKMAkGBss0AwIaB0AEFLzeKSaCVhNd%2F1Xvw5L5GJNFZp2SB80sa%2BAyYe0rjTht0dMk4WZQ0m
1858	154.854091	192.168.1.150	151.139.128.14	HTTP	358	GET /MFcvWaADAgEAME4wTDBKMAkGBss0AwIaB0AEFLzeKSaCVhNd%2F1Xvw5L5GJNFZp2SB80sa%2BAyYe0rjTht0dMk4WZQ0m
1869	154.861668	151.139.128.14	192.168.1.150	OCSP	538	Response
1879	154.889585	151.139.128.14	192.168.1.150	OCSP	793	Response

Далее рассмотрим этот пакет более подробно:

```
Frame 1855: 358 bytes on wire (2864 bits), 358 bytes captured (2864 bits) on interface en0, id 0
Ethernet II, Src: Apple_c4e2:9a (3c:22:fb:c4:e2:9a), Dst: ASUSTek_C_8c:2d:08 (ac:9e:17:8c:2d:08)
Internet Protocol Version 4, Src: 192.168.1.150, Dst: 151.139.128.14
Transmission Control Protocol, Src Port: 53605, Dst Port: 80, Seq: 1, Ack: 1, Len: 292
Hypertext Transfer Protocol (HTTP)
GET /MFcvWaADAgEAME4wTDBKMAkGBss0AwIaB0AEFLzeKSaCVhNd%2F1Xvw5L5GJNFZp2SB80sa%2BAyYe0rjTht0dMk4WZQ0mmsgIRAOw7ZfhYq0Cd9hTftRg6U0%3D HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /MFcvWaADAgEAME4wTDBKMAkGBss0AwIaB0AEFLzeKSaCVhNd%2F1Xvw5L5GJNFZp2SB80sa%2BAyYe0rjTht0dMk4WZQ0mmsgIRAOw7ZfhYq0Cd9hTftRg6U0%3D HTTP/1.1\r\n]
[Severity level: Chat]
[Content Sequence]
Request Method: GET
Request URL: /MFcvWaADAgEAME4wTDBKMAkGBss0AwIaB0AEFLzeKSaCVhNd%2F1Xvw5L5GJNFZp2SB80sa%2BAyYe0rjTht0dMk4WZQ0mmsgIRAOw7ZfhYq0Cd9hTftRg6U0%3D
Request Version: HTTP/1.1
Return-Path: <[REDACTED]>
Host: ocsp.sectigo.com\r\n
Accept: */*\r\n
Accept-Language: ru_RU\r\n
Connection: keep-alive\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: com.apple.trustd/2.0\r\n
[Full request URL: http://ocsp.sectigo.com/MFcvWaADAgEAME4wTDBKMAkGBss0AwIaB0AEFLzeKSaCVhNd%2F1Xvw5L5GJNFZp2SB80sa%2BAyYe0rjTht0dMk4WZQ0mmsgIRAOw7ZfhYq0Cd9hTftRg6U0%3D]
[HTTP request 1/1]
[Response in frame: 1869]
```

Рассмотрим все детали подробнее:

- 1) **HTTP/1.1** – версия протокола / 200 – код состояния, сообщающий об успешности запроса или причине неудачи / OK\r\n – сообщение состояния – краткое описание кода состояния.
- 2) **Content-Type** – Указывает тип носителя ресурса.
- 3) **Connection** – определяет будет ли сетевое соединение открытым.
- 4) **Server** - Содержит информацию о программном обеспечении, используемом исходным сервером для обработки запроса.
- 5) **Ext**- указывает ожидание, которые ожидались от сервера.

## **Итоги**

В данной курсовой работе я собрал все лабораторный работы, которые я сделал за весь первый семестр. Я изучил технологию Ethernet, такие протоколы как IP, TCP, UDP, HTTP. На- учился работать в программе для анализа трафика Wireshark, а также генерировать и анализировать Ethernet-кадры, IP-пакеты, TCP-сегменты, а также HTTP-сообщения.