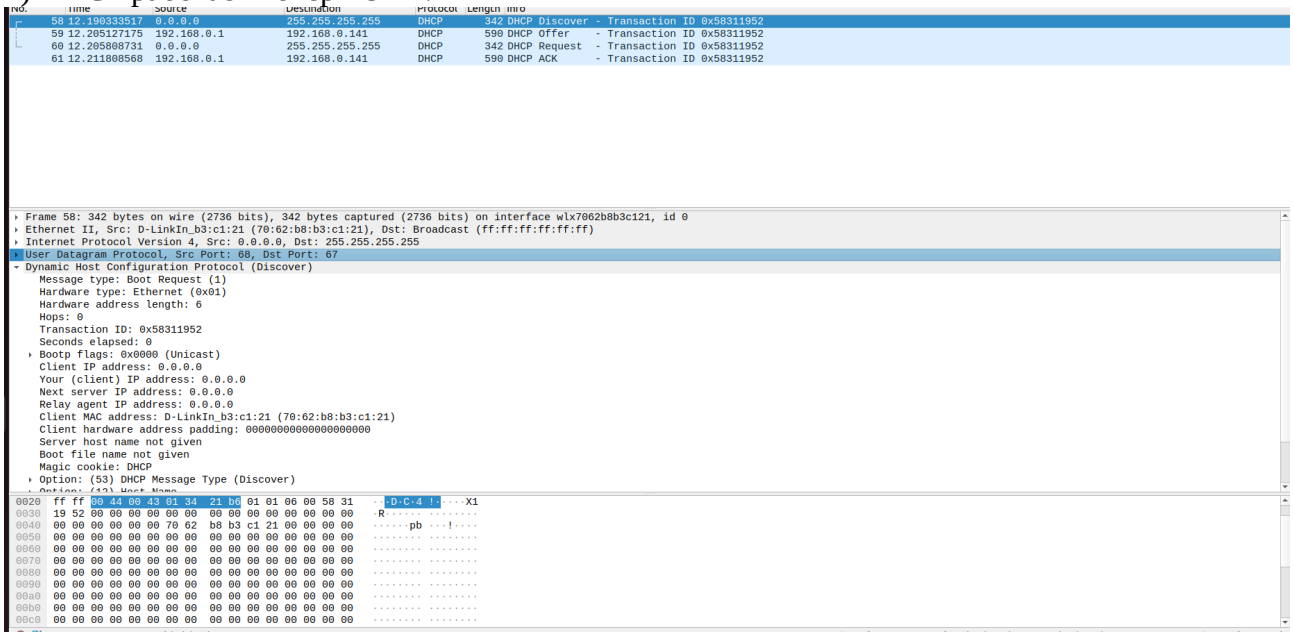
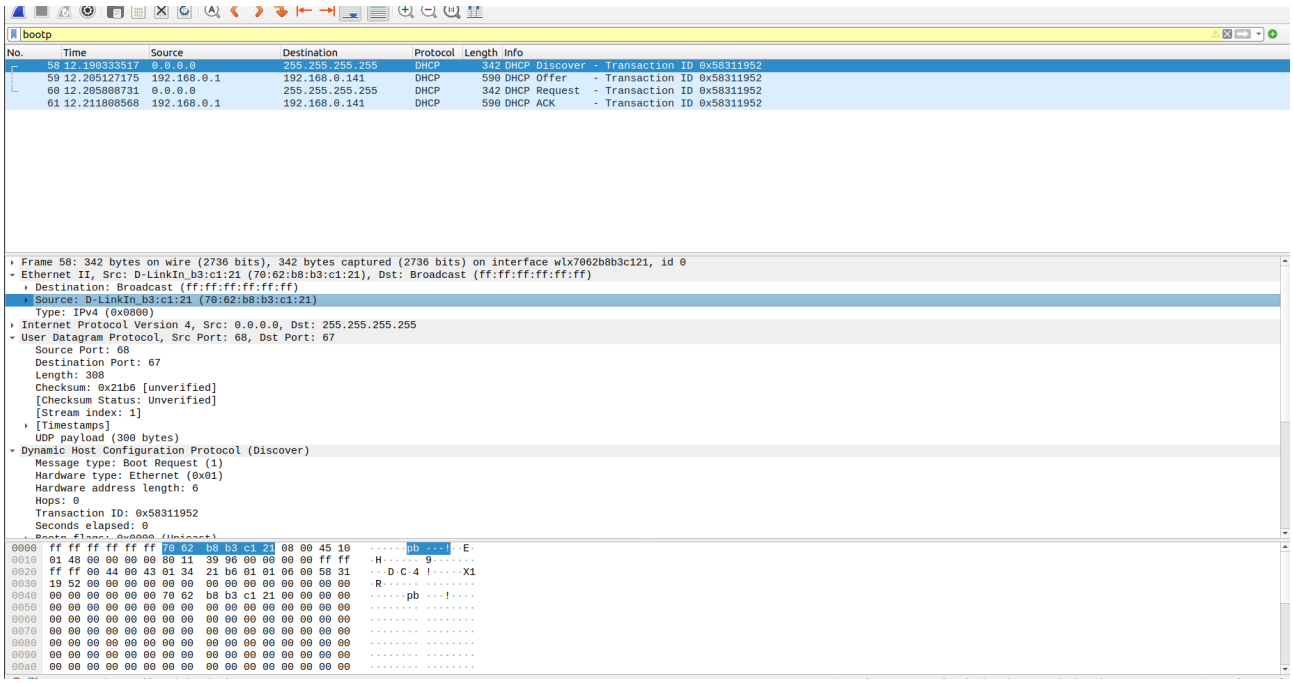
Курбатов Ярослав, РПО МКН

1. Wireshark: DHCP

1) DHCP работает поверх UDP:

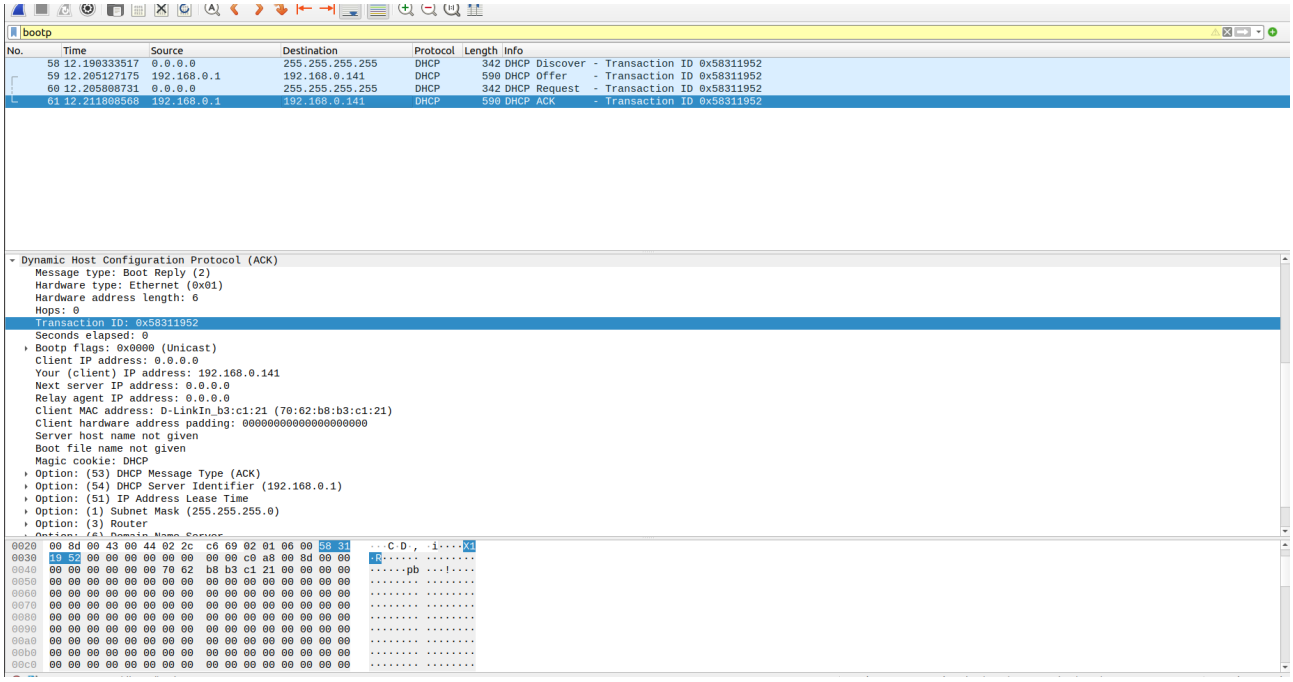


2) чтобы узнать мой mac address, посмотрим вкладку thernet:



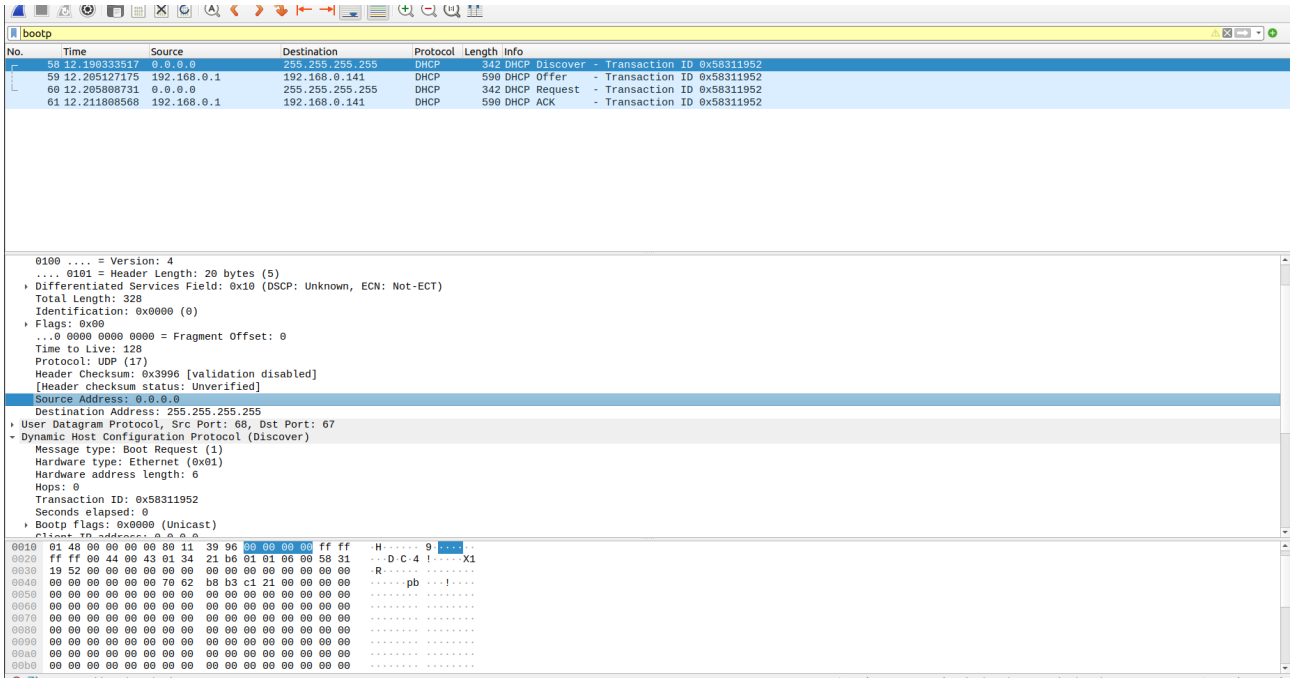
адрес 70:62:b8:b3:c1:21.

3) значение Transaction-ID – 0x58311952:

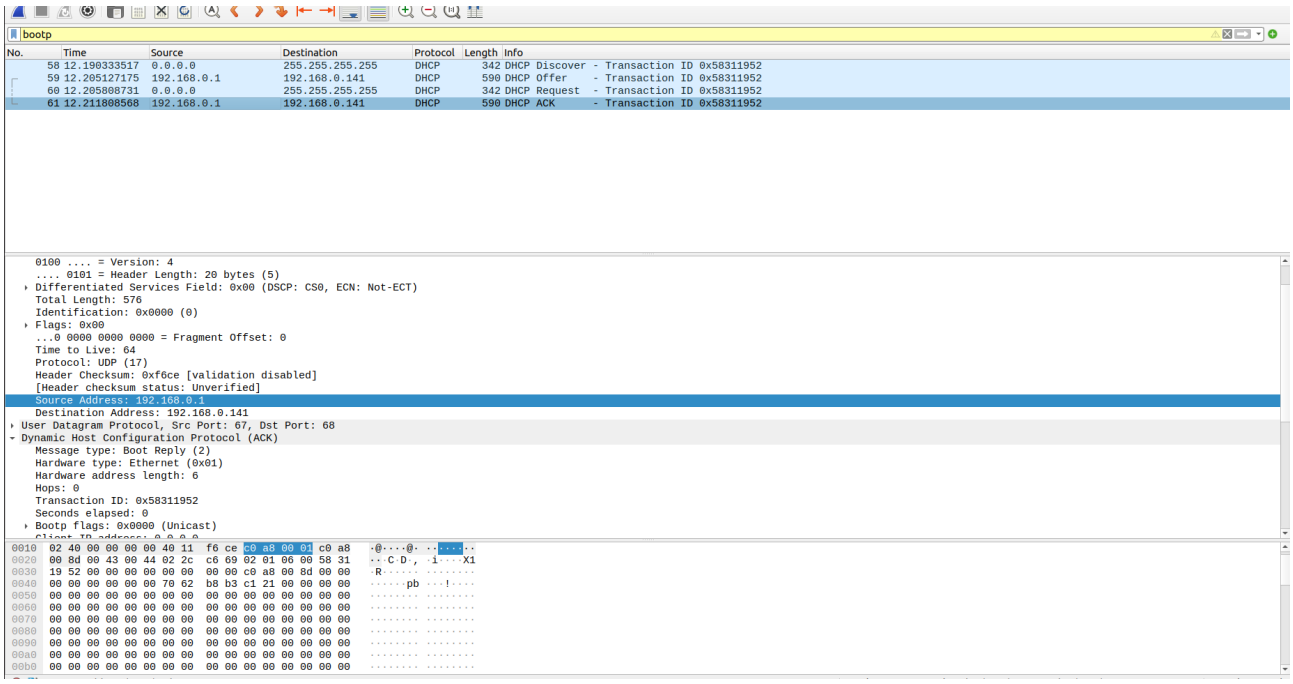


Поле используется чтобы клиенты могли отличать свои запросы от чужих, так как в локальной сети одновременно может протекать несколько DHCP запросов/ответов.

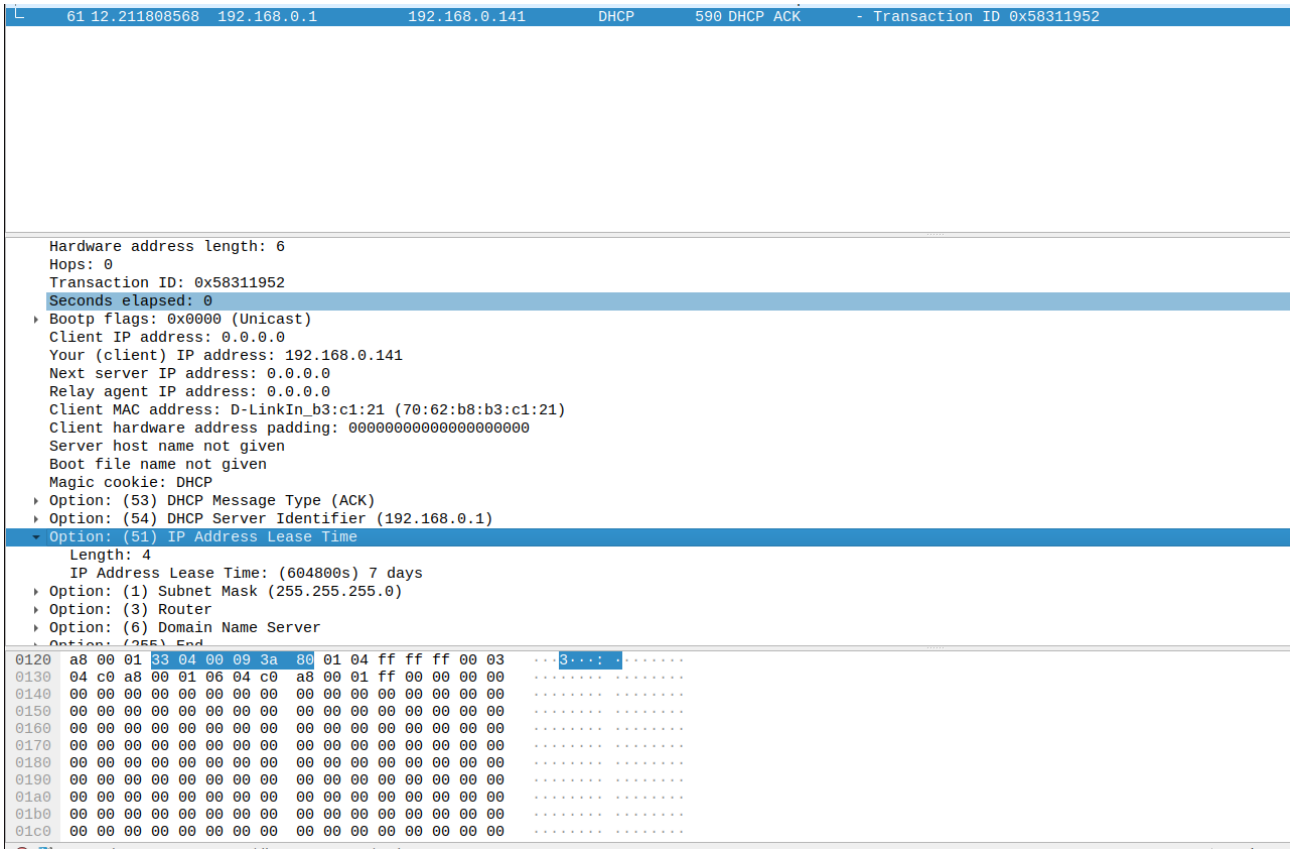
4) Хост использует адрес отправки 0.0.0.0 и адрес назначения 255.255.255.255 (бродкаст):



5) IP-адрес DHCP-севера – 192.168.0.1 (роутер):



6) Lease time – это способ не допустить ситуации, когда один хост навсегда забирает из пула IP-адрес. Хост должен либо запросить продление своего адреса до окончания lease time, либо запросить новый после окончания lease. В моем случае этот период составляет 7 дней:



2. Программирование

1. Проверка целостности пакетов

Ruby MRI ~>3.0.0

Вычисление CRC8: 01-crc/crc8.rb
Запуск: ruby 01-crc/main.rb (ввод текстам, ctrl-d для сабмита)

Пример:

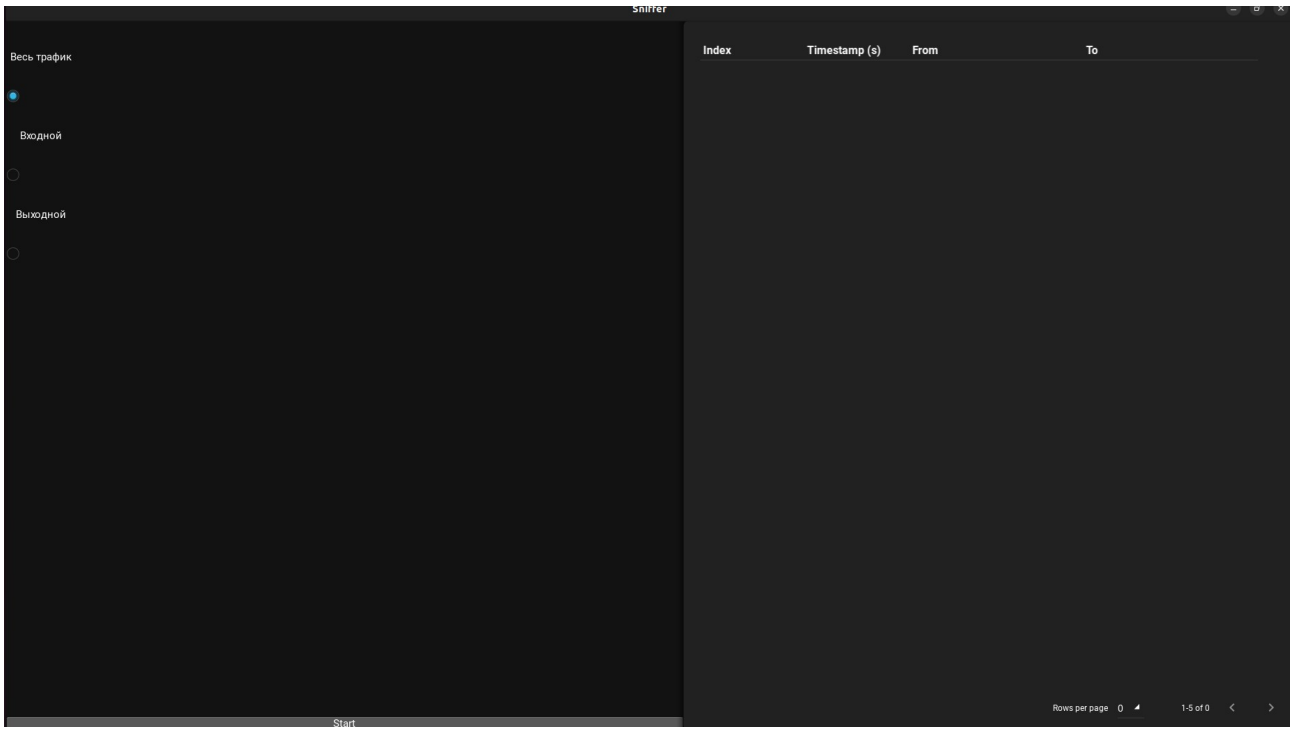
```
viralpraxis@primary:~/Documents/spbu/2023-spring/networks/spbu-masters-compnet/homeworks/13$ ruby 01-crc/main.rb
aiosdmjaqwrwiejmrfiowesjrfiowesjwerf
dpqwejkfrmwolkfrwejwfnwerjfg
9q34ier9230rfpsmfslk dfoiwmejiporjfmweiopjfewfs
wesfkdowsefjweoifj
[info]: changed byte at chunk 10
[info]: changed byte at chunk 11
[info]: changed byte at chunk 19
[info]: changed byte at chunk 22
[info]: changed byte at chunk 26
chunk: 1 payload: [97, 105, 111, 115, 100] crc8: 25 correct: true
chunk: 2 payload: [109, 106, 97, 113, 119] crc8: 202 correct: true
chunk: 3 payload: [114, 119, 105, 101, 106] crc8: 80 correct: true
chunk: 4 payload: [109, 114, 102, 105, 111] crc8: 188 correct: true
chunk: 5 payload: [119, 101, 115, 106, 114] crc8: 88 correct: true
chunk: 6 payload: [102, 105, 111, 119, 101] crc8: 99 correct: true
chunk: 7 payload: [115, 106, 119, 101, 114] crc8: 109 correct: true
chunk: 8 payload: [102, 10, 100, 112, 113] crc8: 231 correct: true
chunk: 9 payload: [119, 101, 106, 107, 102] crc8: 185 correct: true
chunk: 10 payload: [114, 109, 119, 111, 109] crc8: 181 correct: false ( actual: 178 )
chunk: 11 payload: [107, 102, 114, 120, 101] crc8: 118 correct: false ( actual: 181 )
chunk: 12 payload: [106, 119, 102, 110, 119] crc8: 248 correct: true
chunk: 13 payload: [101, 114, 106, 102, 103] crc8: 164 correct: true
chunk: 14 payload: [10, 57, 113, 51, 52] crc8: 148 correct: true
chunk: 15 payload: [105, 101, 114, 57, 50] crc8: 167 correct: true
chunk: 16 payload: [51, 48, 114, 102, 112] crc8: 52 correct: true
chunk: 17 payload: [115, 109, 102, 115, 108] crc8: 181 correct: true
chunk: 18 payload: [107, 32, 100, 102, 111] crc8: 50 correct: true
chunk: 19 payload: [105, 120, 109, 101, 106] crc8: 118 correct: false ( actual: 164 )
chunk: 20 payload: [105, 112, 111, 114, 106] crc8: 254 correct: true
chunk: 21 payload: [102, 109, 119, 101, 105] crc8: 145 correct: true
chunk: 22 payload: [111, 112, 107, 102, 101] crc8: 91 correct: false ( actual: 48 )
chunk: 23 payload: [119, 102, 115, 10, 119] crc8: 140 correct: true
chunk: 24 payload: [101, 115, 102, 107, 100] crc8: 168 correct: true
chunk: 25 payload: [111, 119, 115, 101, 102] crc8: 151 correct: true
chunk: 26 payload: [106, 119, 102, 111, 105] crc8: 10 correct: false ( actual: 183 )
chunk: 27 payload: [102, 106, 10] crc8: 249 correct: true
viralpraxis@primary:~/Documents/spbu/2023-spring/networks/spbu-masters-compnet/homeworks/13$
```

2. Подсчет сетевого трафика

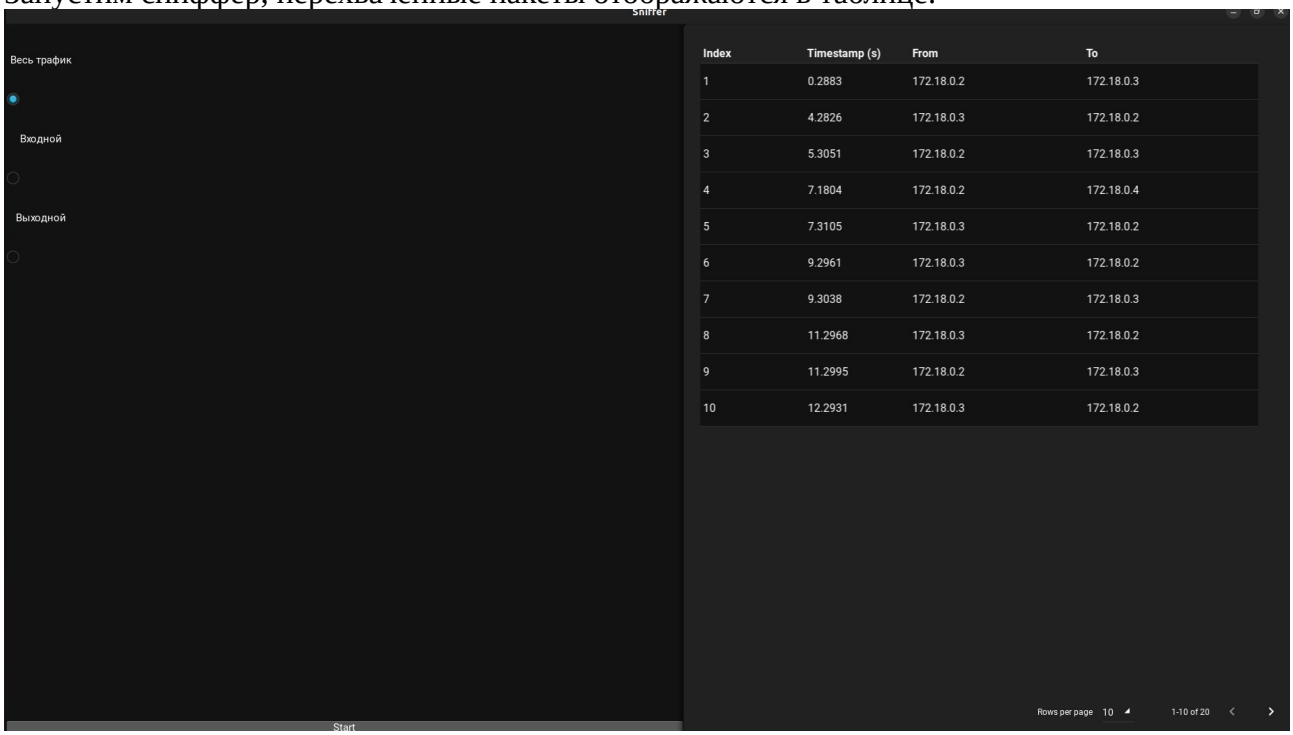
Python ~> 3.10
Зависимости указаны в 02-sniffer/requirements.txt

Запуск: [sudo] python main.py (нужны права su)

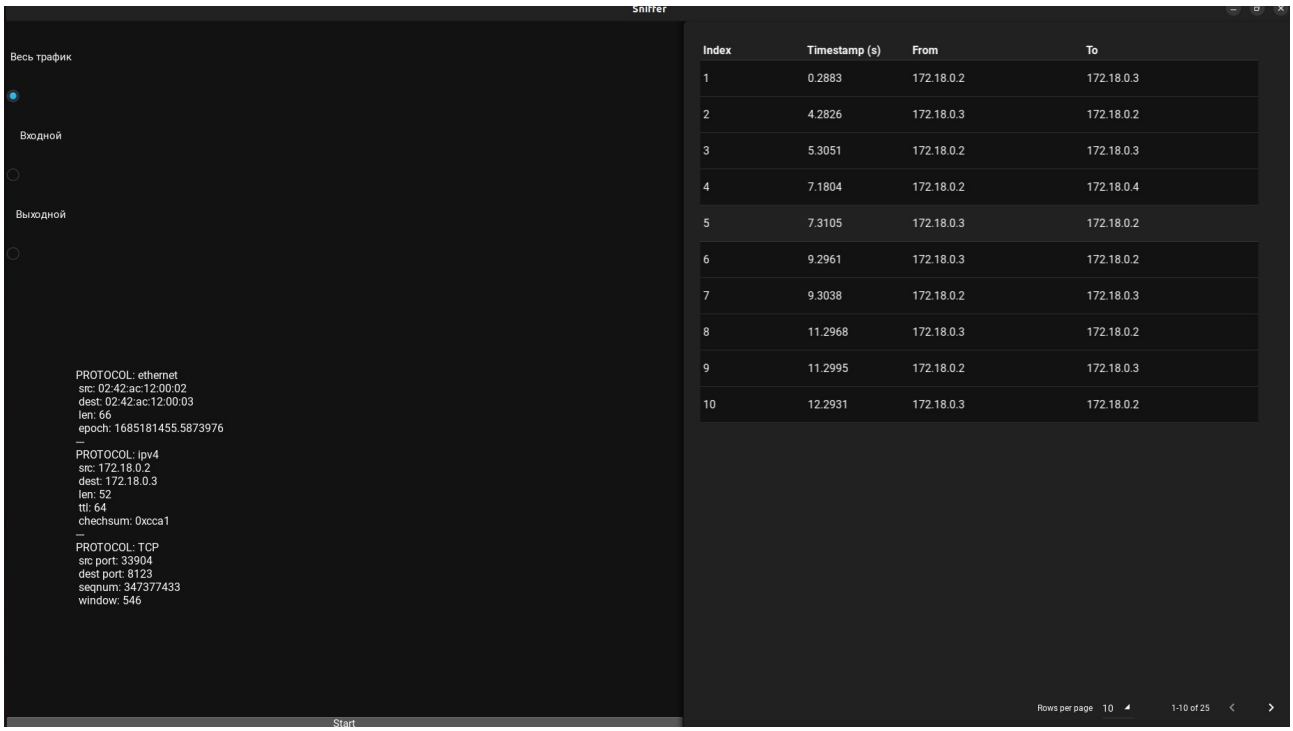
Начальное состояние UI:



Запустим сниффер, перехваченные пакеты отображаются в таблице:

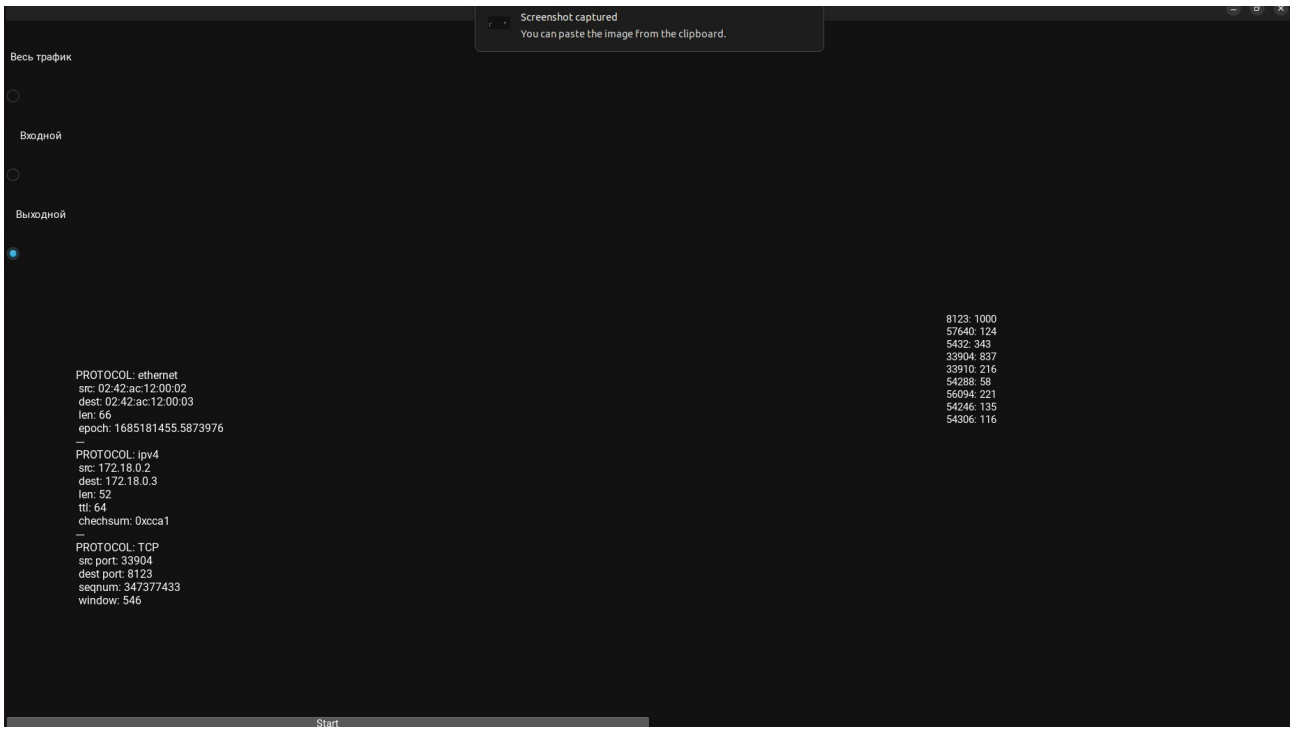


При клике на пакет в таблице отображается информация о пакетах протоколов, которые инкапсулированы в этот пакет:



Переключимся на режим “Входной трафик”:

И на режим “выходной трафик”:



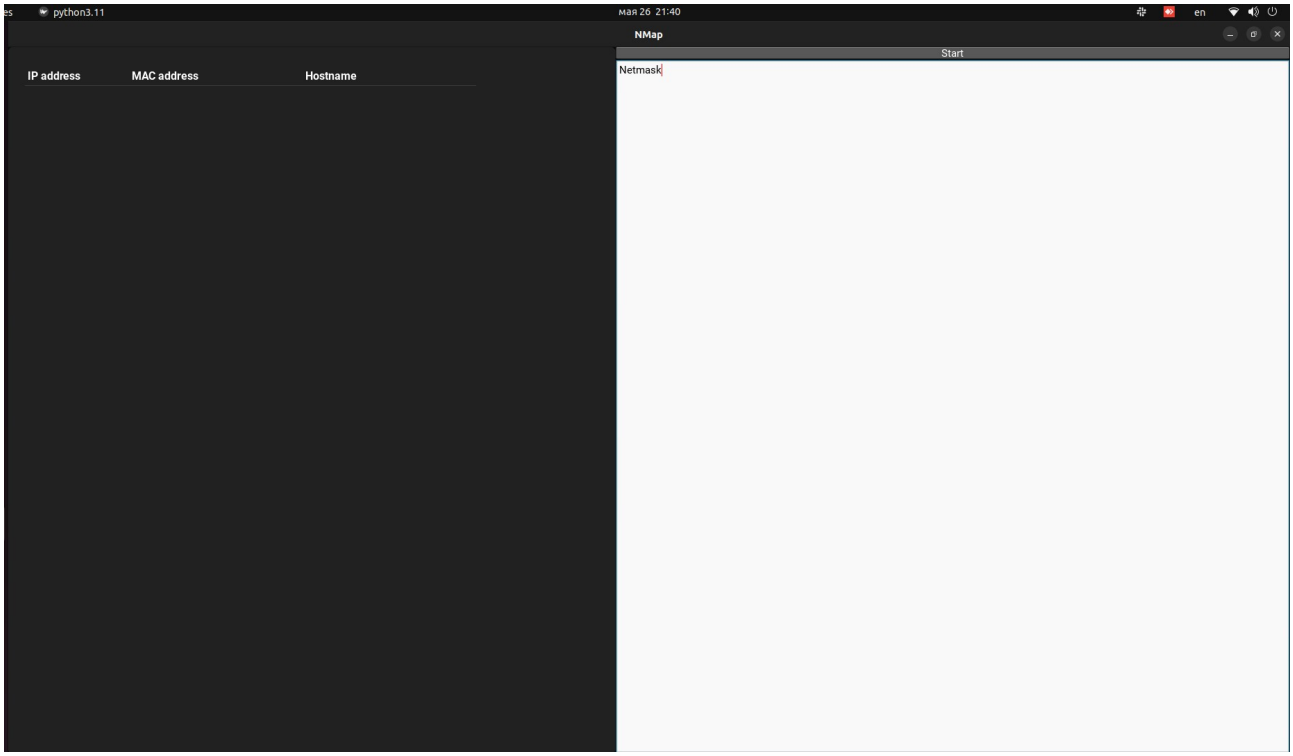
NB: для удобства тестирования в таблице отображается только каждый 200-ый пакет (чтобы за ними можно уследить)

3. Определение всех компьютеров в сети

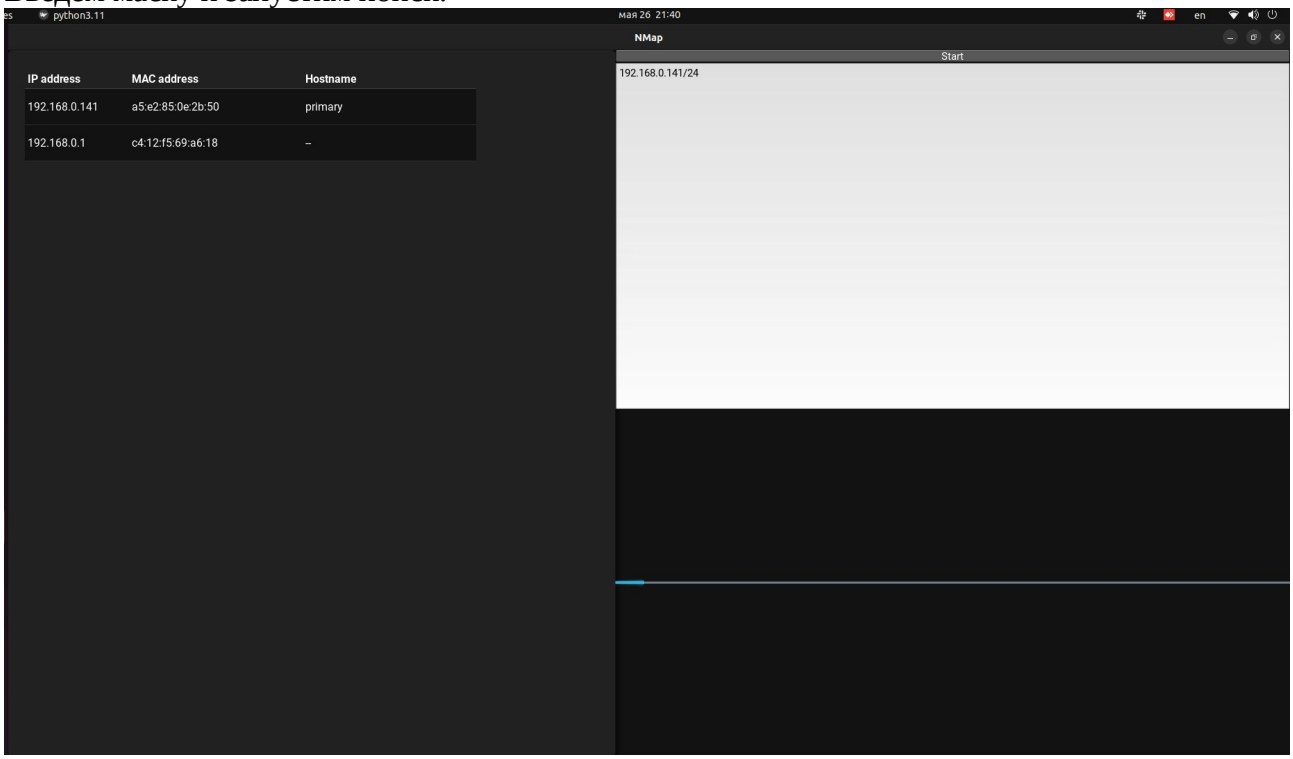
python ~> 3.11
Зависимости указаны в 03-nmap/requirements.txt

Запуск: python 03-nmap/nmap/main.py

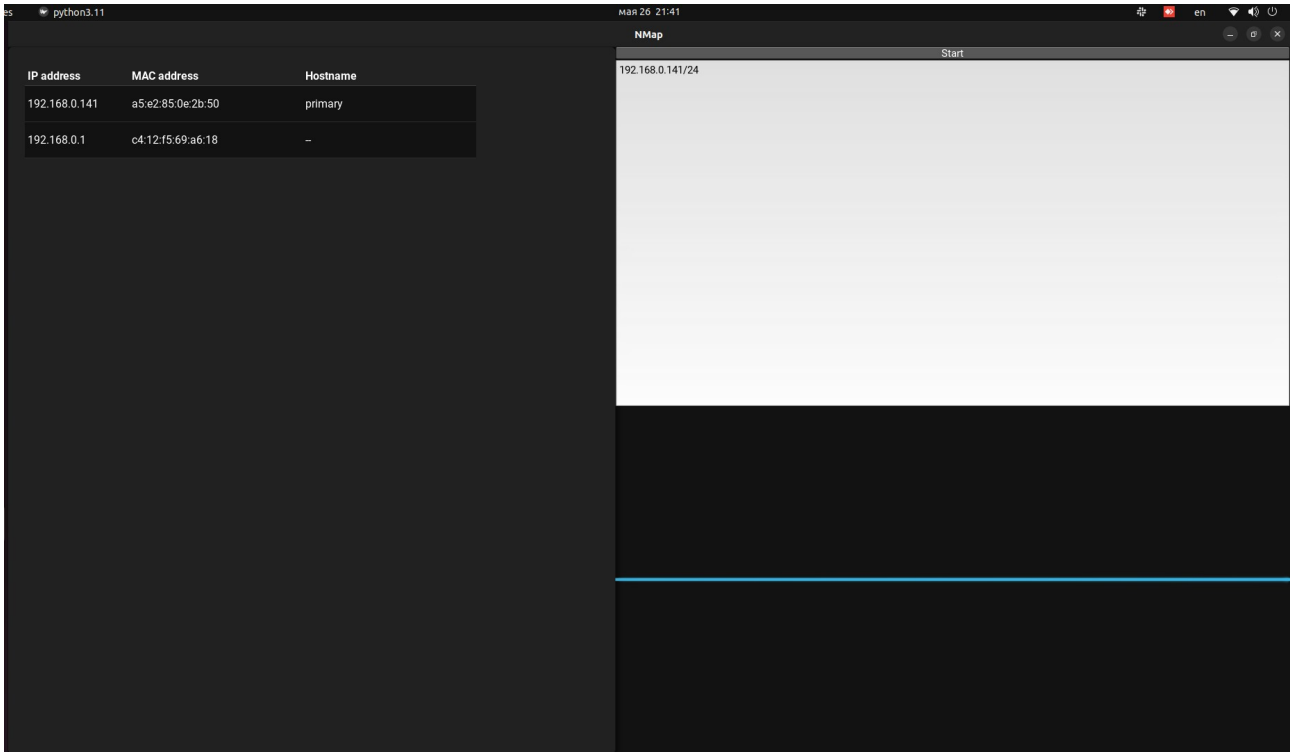
Иначальное состояние UI:



Введем маску и запустим поиск:



В результате программа обнаружили в локальной сети только мой роутер (в данный момент только он в сети и есть):



NB: отмечу, что первый IP адрес (адрес хоста) находится следующим образом: берется список всех адресов всех сетевых интерфейсов, из них находится тот, что удовлетворяет маске (см. self_resolver.py).

Задачи

Задача 1
а)
f(p) = Np(1 – p)^(N – 1)
найдем производную f(p), считая N константой:

z(p) = dp/df f(p) = n (1 – np) (1 – p) ^ (n – 2)

т.к. p лежит в (0, 1), очевидно, z(p) = 0 при p = 1 / N

Ответ: N^(-1).

б)
Найдем lim Np(1 – p)^(N – 1) при N → inf

подставим p = 1 / N

lim (1 – 1 / n)^(n – 1) при n → inf

пусть y = lim{n→inf} (1 – 1 / n) ^ (n – 1)

ln y = lim{n → inf} ln(1 – 1 / n) ^ (n – 1) = lim{n → inf} ln(1 – 1 / n) / (n – 1) ^ (-1)
очевидно, можем применить правило Лопиталя:

ln y = lim{n → inf} ln(1 – 1/n) / (n-1)^(-1) = lim{n → inf} -1 * (n – 1)^2 / (n – 1) n = -1 * (n – 1) n
ln y = -1
y = e^-1 ~ 0.367

Ответ: 1/e