

# Установка

Установка програми відбувалась як зазвичай - через пакетний менеджер. Але додатково додав свого користувача у групу wireshark.

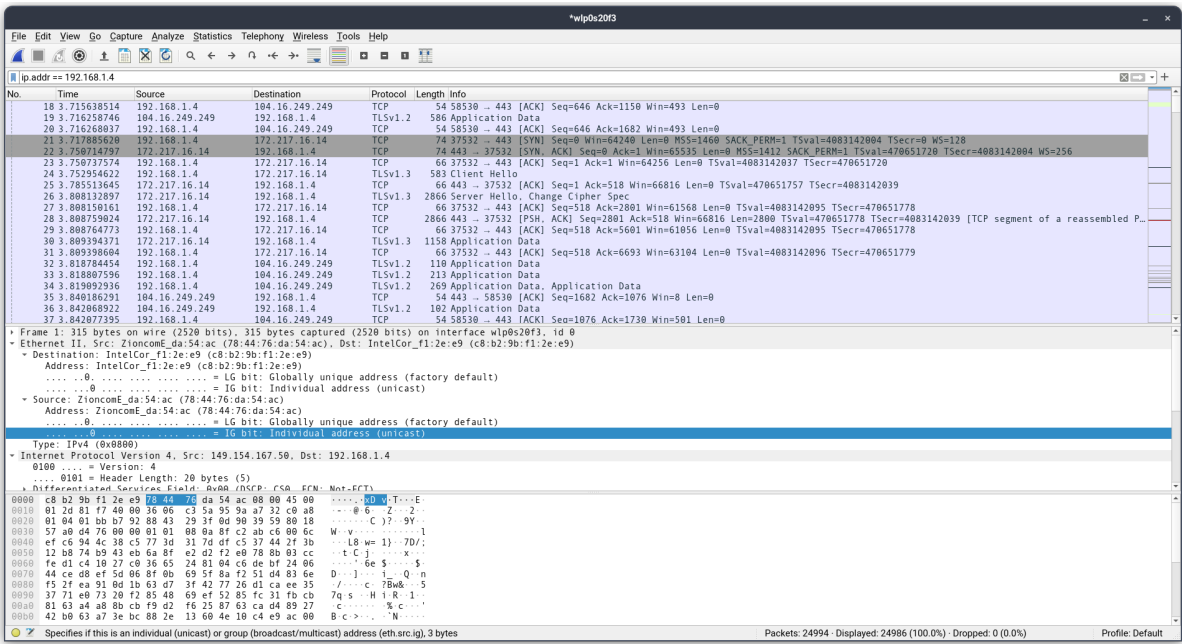
```
sudo usermod -aG wireshark lolpie
```

У групі wireshark містяться користувачі, які можуть виконувати перехоплення пакетів. Це альтернатива, до запуску програми від імені адміністратора (sudo)

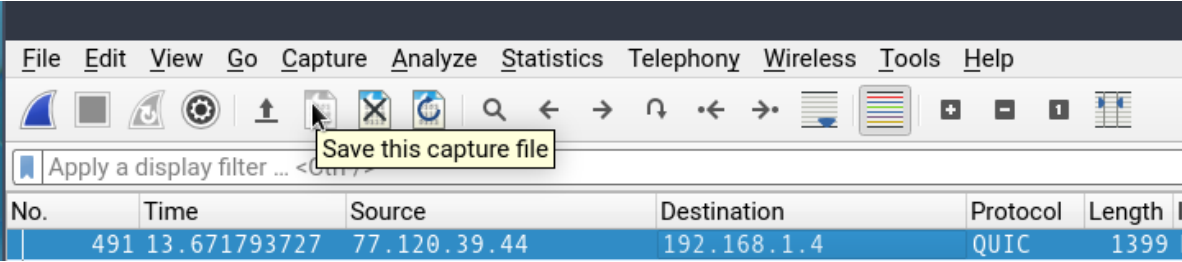
# Інформація про систему

- ip<sub>4</sub> комп'ютера == **192.168.1.4**
- ip<sub>4</sub> телефону == **192.168.1.11**
- ip<sub>6</sub> == **fe80::e99f:48da:f29f:51ed**
- mac == **c8:b2:9b:f1:2e:e9**

# Інтерфейс



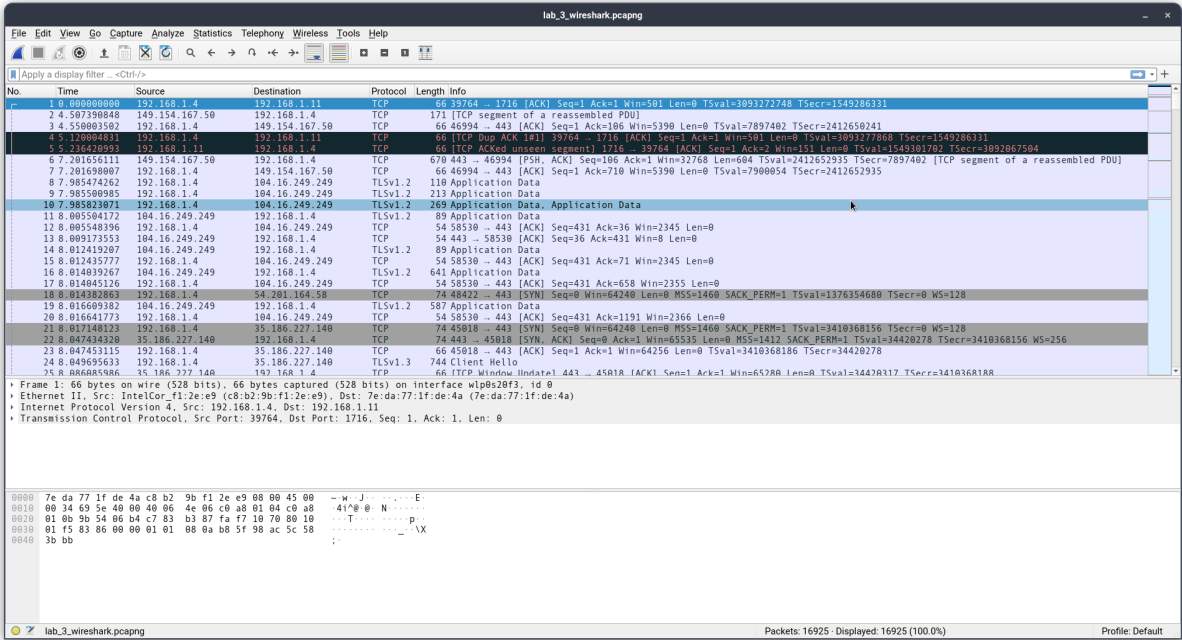
у нас є 3 основних поля - список перехоплених пакетів, оброблений вміст пакета та сирий вміст пакета відповідно. Також, для виконання цієї лабораторної роботи, пригодиться поле фільтрації, яке знаходиться над списком пакетів. Також, при потребі цей список можна зберегти у файл



# Фільтрація

# Перехоплення

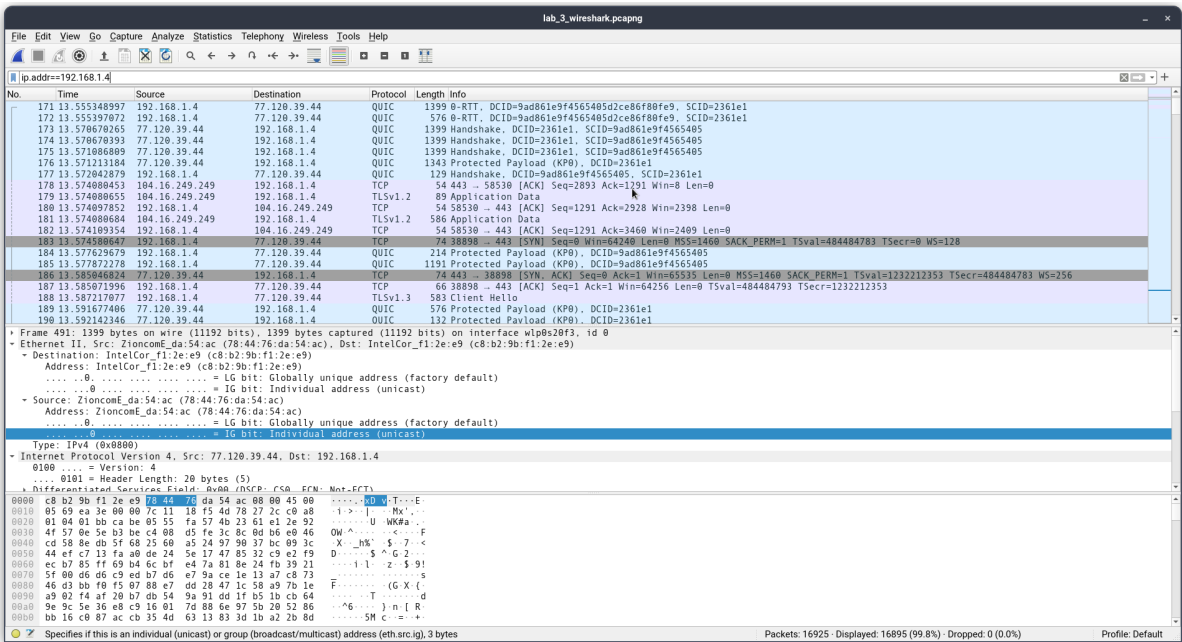
Спочатку нам потрібно дістати певний набір пакетів, по якому пізніше будемо фільтрувати. Для того запускаємо нове перехоплення і робимо будь які дії в мережі. Також можна завантажити файл з перехопленими пакетами, що був збережений раніше.



тепер в нас є набір пакетів, і ми можемо перейти до фільтрації

# ip.addr

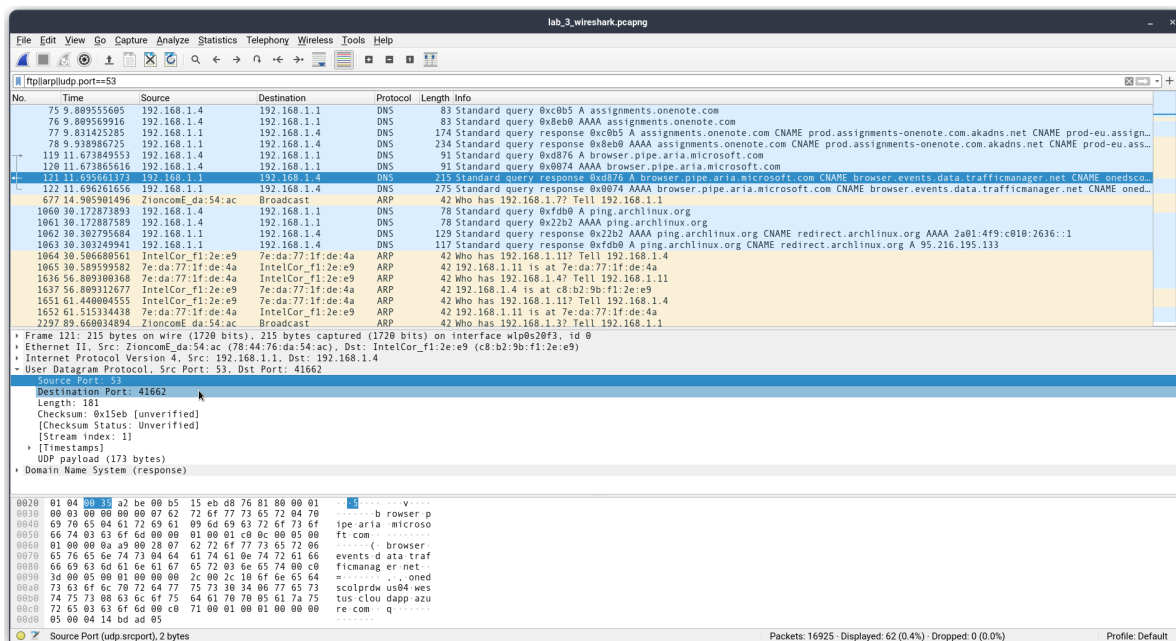
ip.addr дозволяє знайти всі пакети, в яких ipv4 відправника(Source), або ipv4 отримувача(Destination) рівне шуканому. Спробуємо цей фільтр із ipv4 комп'ютера - 192.168.1.4



# ftp | | arp | | udp.port

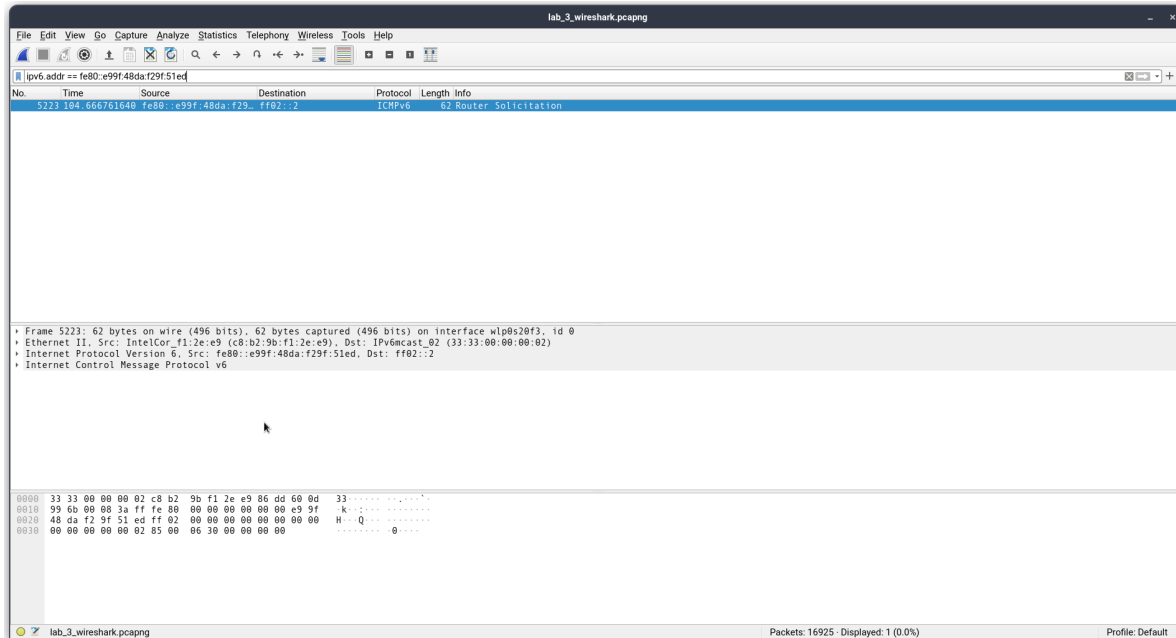
Даним фільтром ми перевіряємо, чи протокол = arp, або протокол = frp, або `udp.port == 53`. На скріні, у секції з обробленим змістом можна переконавшись в тому, що `udp.post` дійсно == 53.

Також, варто зазначити, що `||` це або, але ми також можемо використовувати `or`



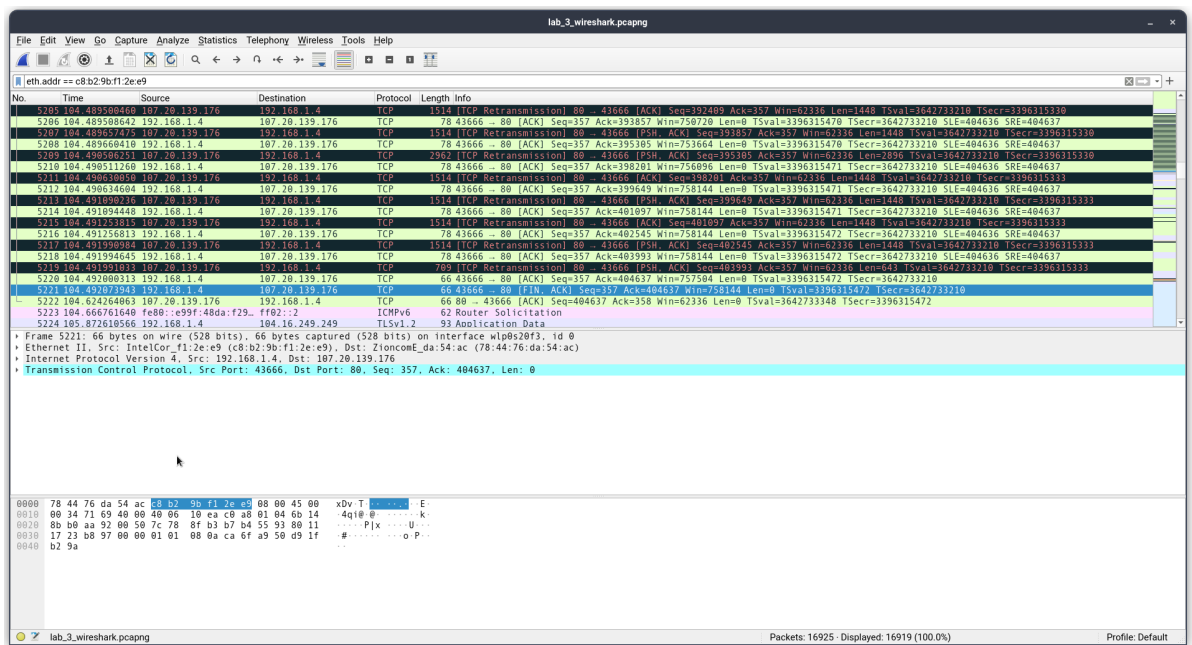
## ipv6.addr

цей фільтр подібний на перший, тільки тут ми замість `ipv4` використовуємо `ipv6`(який в мене `fe80::e99f:48da:f29f:51ed`)



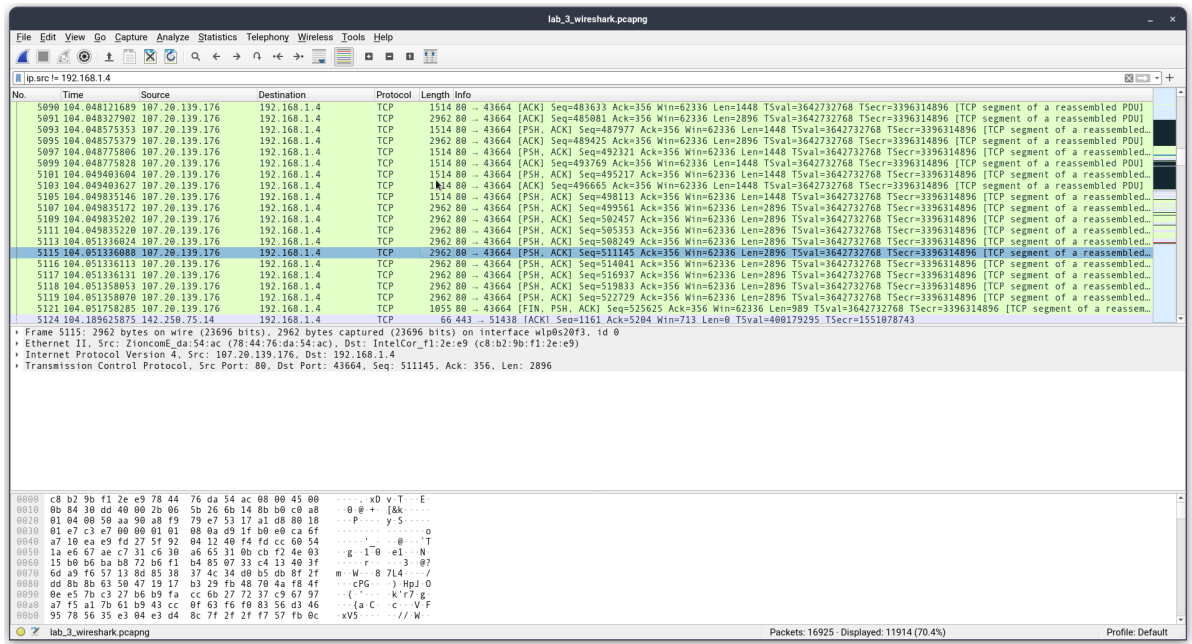
## eth.addr

даним фільтром ми перевіряємо чи MAC адреса відправника/отримувача рівна шуканій. В мене `MAC == c8:b2:9b:f1:2e:e9`



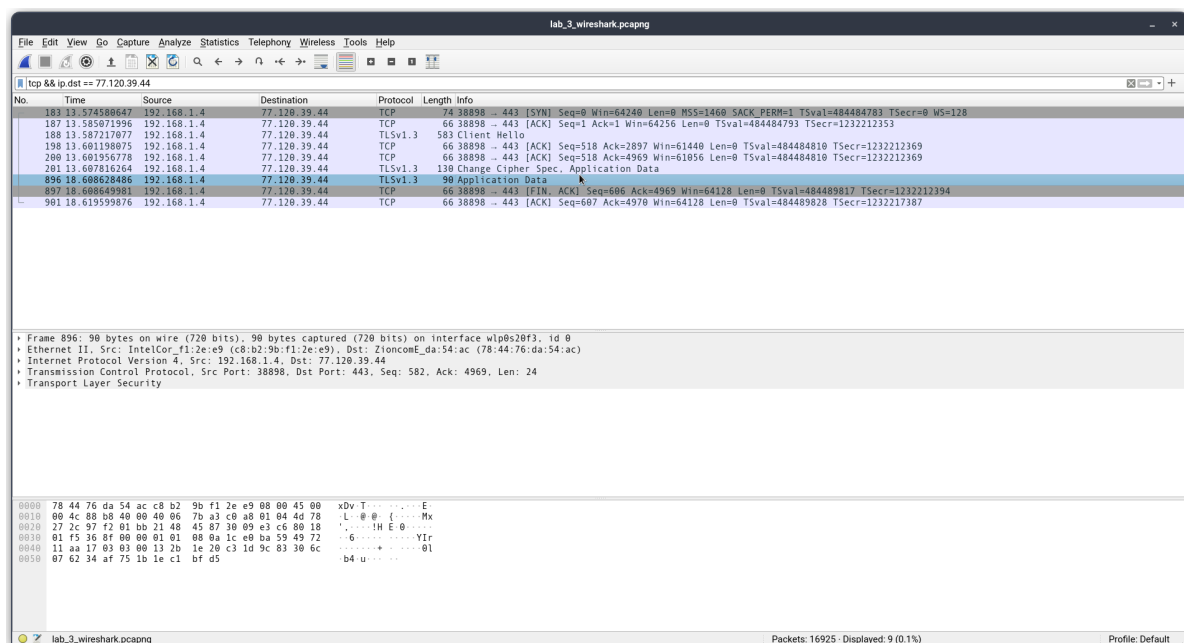
## ip.src

src у фільтрі, перевіряє на рівність тільки відправника (source)









## tcp && ip.dst

перевіряємо чи протокол == TCP, і чи ip отримувача == зазначеному. Тут, як шукане ip, вибралось одне з пакетів. Також, && це "і", яке можна замінити and



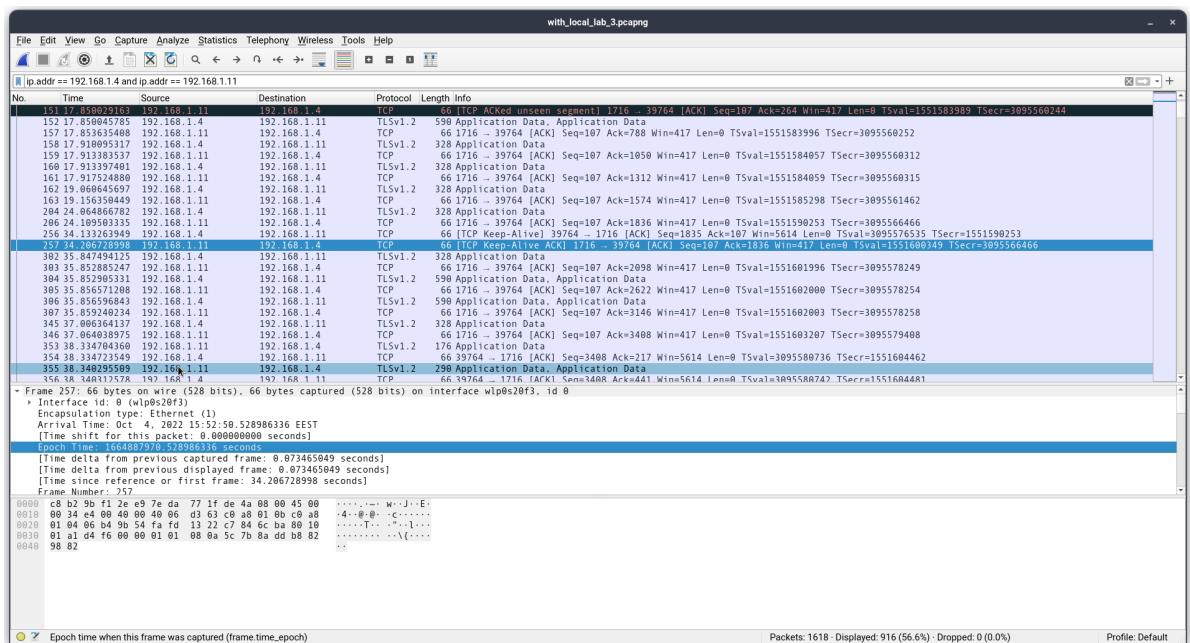
## свій фільтр

На лінукс системах є програма, що дозволяє взаємодіяти з андроїд телефонами бездротово - kde connect. Її функціонал такий:

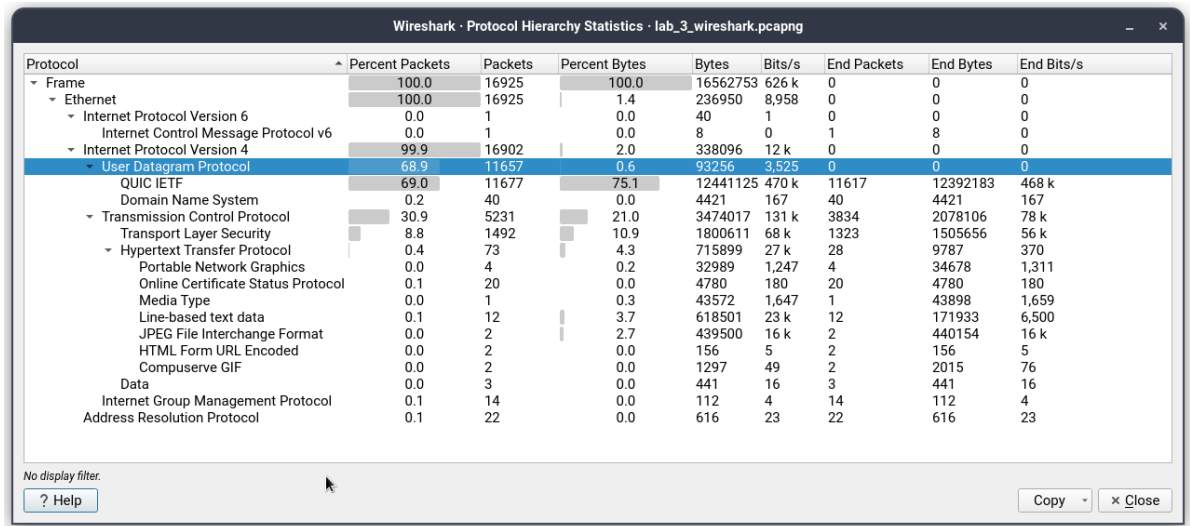
-  Надіслати файли
-  Надіслати вміст буфера
-  Віддалений показ слайдів
-  Керування відтворенням
-  Виконати команду
-  Дистанційне введення

Так як програма працює бездротово, пристрої мають бути в одній локальній мережі, і пакети, що надсилаються між ними можна перехопити. Що ми і зробимо. ір комп'ютера == **192.168.1.4**, телефона == **192.168.1.11**

нагадаю, що addr перевіряє чи ір відправника/отримувача == шуканому, а так як в нас в обох випадках addr, то нам поверне всі пакети що містять шукані ір, не важливо від того хто відправник а хто отримувач



# Опис сторінки



**Protocol Hierarchy** - як зрозуміло з назви, це ієрархія протоколів, зображена у вигляді дерева.

На цій сторінці можна побачити інформацію про протоколи у даному списку пакетів(захопленні)

Стовпці:

- **Protocol** - ім'я протоколу
- **Percent Packets** - відсоток кількості пакетів протоколу
- **Packets** - загальна кількість пакетів протоколу(тут є число яке ми можемо перевірити - 1 пакет для ipv6)
- **Percent Bytes** - відсоток сумарної кількості байтів пакетів протоколу
- **Bytes** - сумарна кількість байтів пакетів протоколу
- **Bits/s** - пропускна здатність протоколу
- **End Packets** - кількість пакетів цього протоколу, де це був найвищий протокол у стеку (тобто останній оброблений).
- **End Bytes** - сумарна кількість байтів пакетів протоколу, де це був найвищий протокол у стеку (тобто останній оброблений).
- **End Bits/s** - пропускна здатність протоколу, коли він був найвищий протокол у стеку (тобто останній оброблений).

