РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 03

дисциплина:Сетевые технологии (09.03.03)

Анализ трафика в Wireshark

Студент: Стелина Петрити

Группа: НПИбд-02-21

МОСКВА

2023 г.

Цели работы

Изучение посредством Wireshark кадров Ethernet, анализ PDU протоколов транспортного и прикладного уровней стека TCP/IP.

3.3. Задания для выполнения
3.3.1. MAC-адресация
3.3.1.1. Постановка задачи

1. Изучение возможностей команды ipconfig для ОС типа Windows (ifconfig для систем типа Linux).
2. Определение MAC-адреса устройства и его типа.

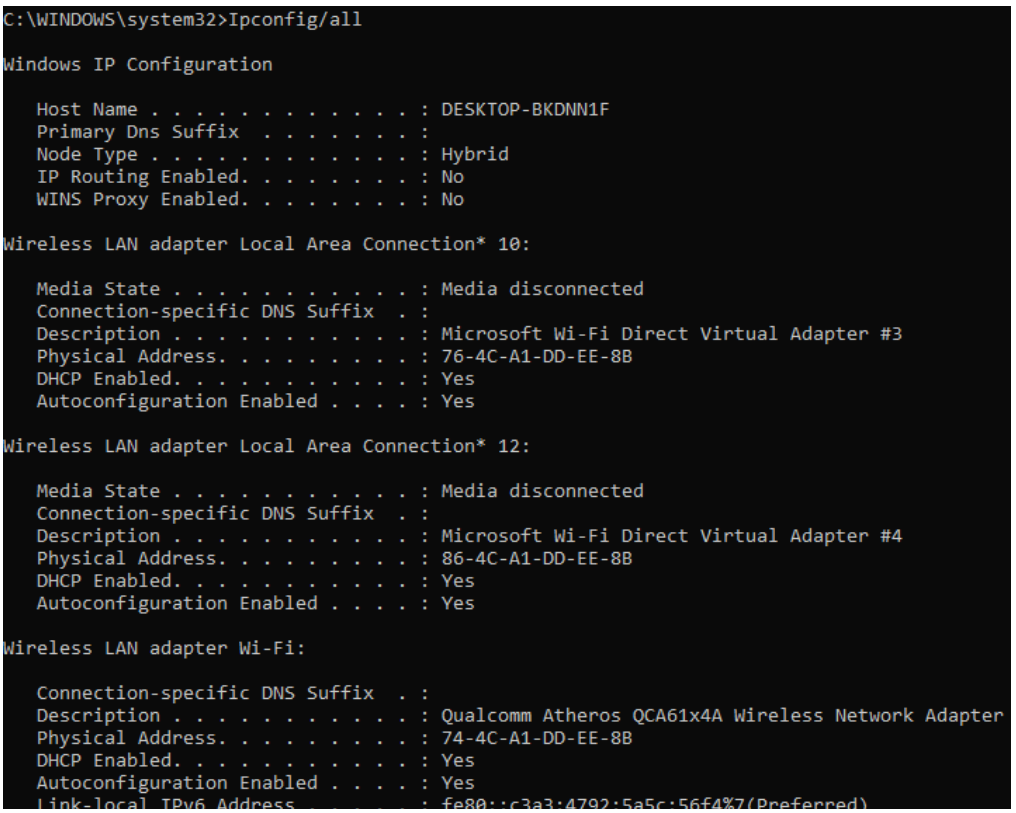
3.3.1.2. Порядок выполнения работы

1. С помощью команды ipconfig для ОС типа Windows или ifconfig для систем типа Linux выведите информацию о текущем сетевом соединении. Используйте разные опции команды. В отчёте поясните детально полученную в каждом случае при выводе информацию. Подтвердите свой ответ скриншотами.

Ipconfig/all

Команда "ipconfig /all" в командной строке (Windows) используется для отображения подробной информации о сетевых интерфейсах и конфигурации вашего компьютера. Эта команда предоставляет обширную информацию о ваших сетевых адаптерах, IP-адресах, настройках DNS и многом другом. Когда вы запустите эту команду, вы увидите список сведений, связанных с сетью, для каждого сетевого адаптера в вашей системе.

рис.1.информация о сетевых подключениях



2. Определите MAC-адреса сетевых интерфейсов на вашем компьютере. Подтвердите свой ответ скриншотом.
"Физический адрес" под каждым сетевым интерфейсом - это MAC-адрес.
Например:

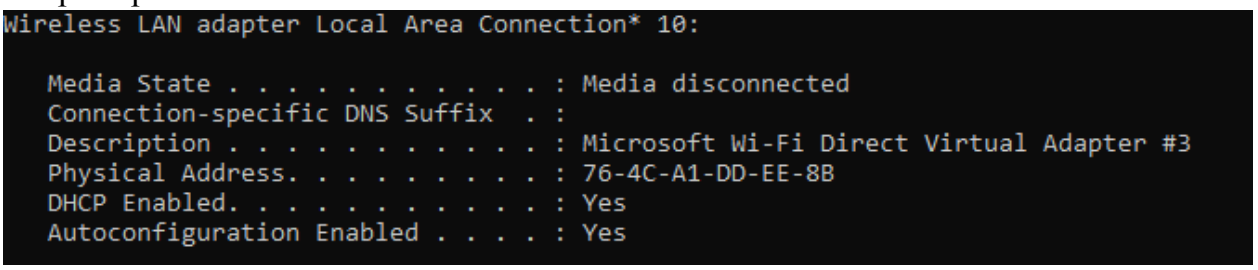


рис.2.Physical Address- MAC-адрес

3. Опишите структуру MAC-адресов вашего устройства. Какая часть адреса идентифицирует производителя? Какая часть адреса идентифицирует сетевой интерфейс? Определите, каким является адрес — индивидуальным или групповым, глобально администрируемым или локально администрируемым.
Поясните свой ответ. Используйте шестнадцатеричную запись MAC-адреса для пояснения.

Префикс производителя (OUI - Organizationally Unique Identifier): Первые 6 символов MAC-адреса идентифицируют производителя оборудования. Этот префикс уникален для каждого производителя и называется OUI.

Идентификатор устройства (NIC - Network Interface Controller): После OUI идет последовательность символов, которая уникально идентифицирует сетевой интерфейс устройства, произведенного этим производителем.

Мой MAC-адрес: 76-4C-A1-DD-EE-8B

76-4C-A1 идентифицирует производителя (OUI)

-DD-EE-8B идентифицирует конкретный сетевой интерфейс(NIC-specific identifier)

3.3.2. Анализ кадров канального уровня в Wireshark

3.3.2.1. Постановка задачи

1. Установить на домашнем устройстве Wireshark.
2. С помощью Wireshark захватить и проанализировать пакеты ARP и ICMP в части кадров канального уровня.

3.3.2. Анализ кадров канального уровня в Wireshark
3.3.2.1. Постановка задачи

1. Установить на домашнем устройстве Wireshark.

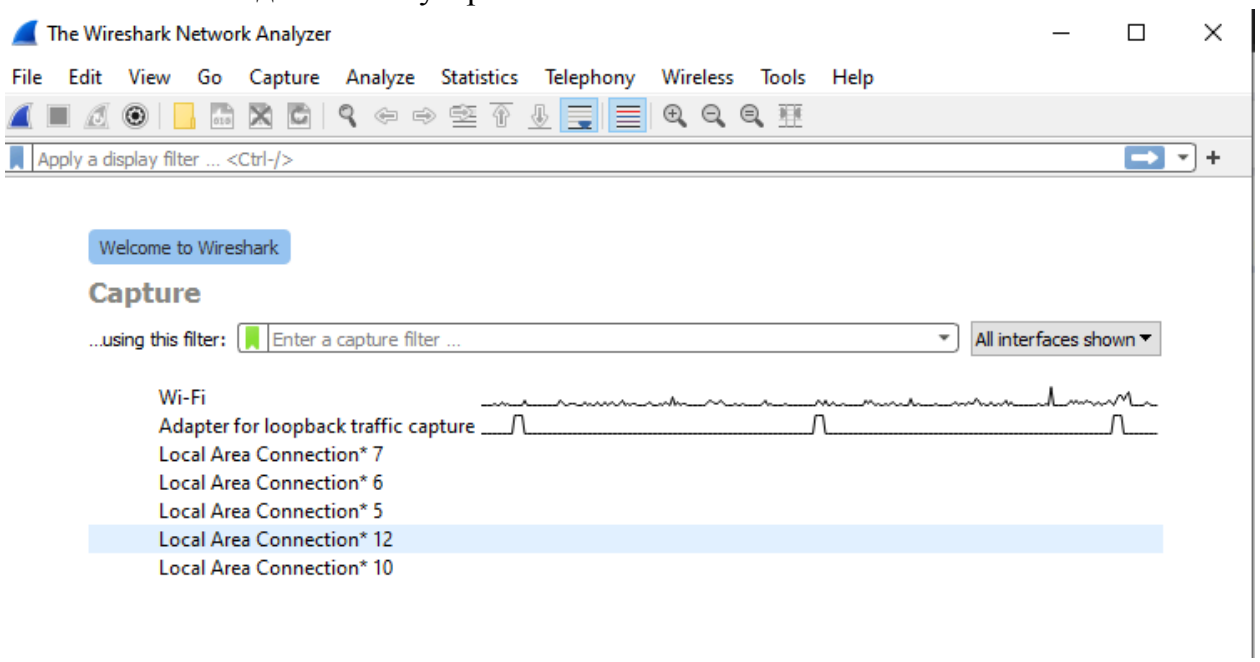


рис.3.Установите Wireshark на свое домашнее устройство.

2. С помощью Wireshark захватить и проанализировать пакеты ARP и ICMP в части кадров канального уровня.Убедитесь, что начался процесс захвата трафика.

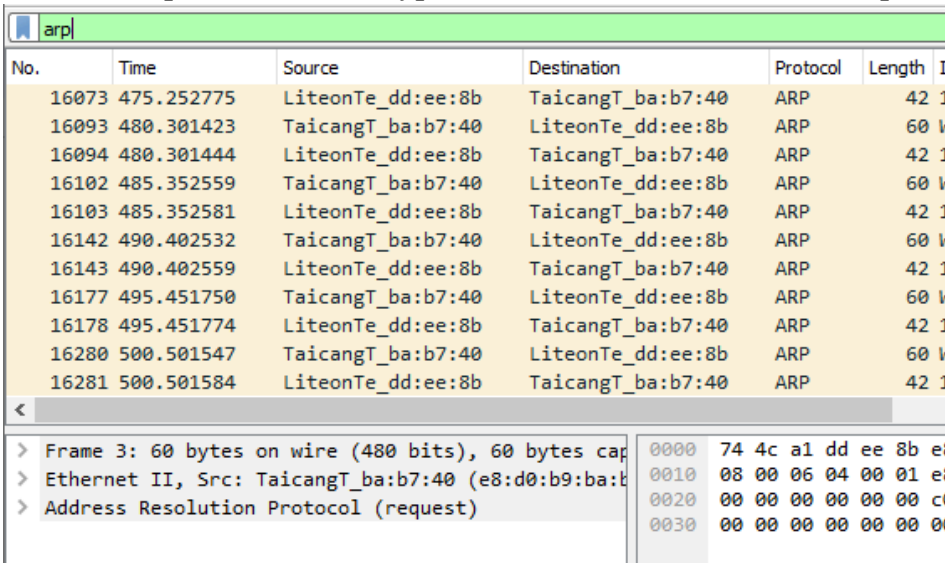


рис.4.пакет ARP

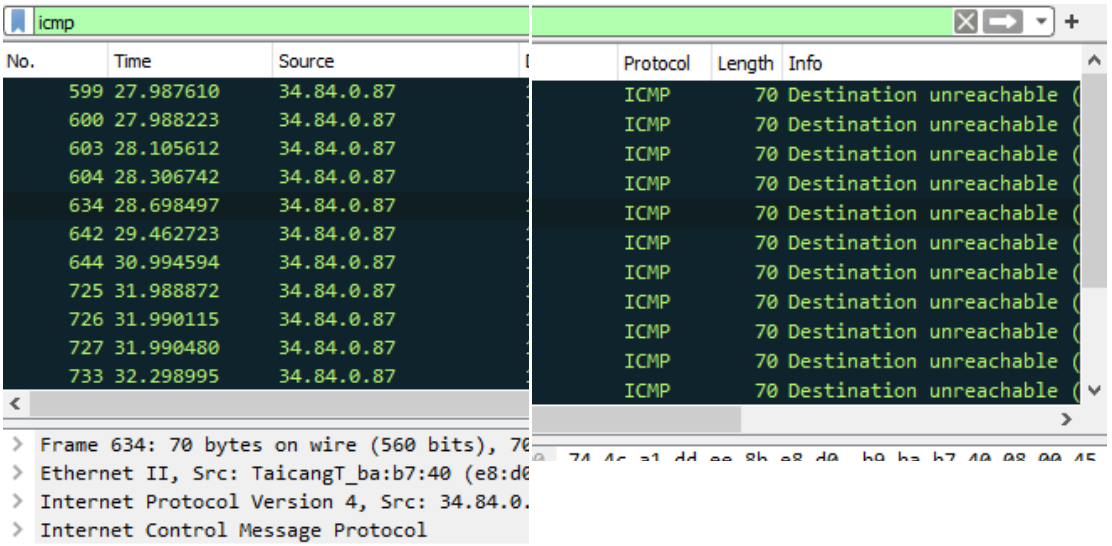
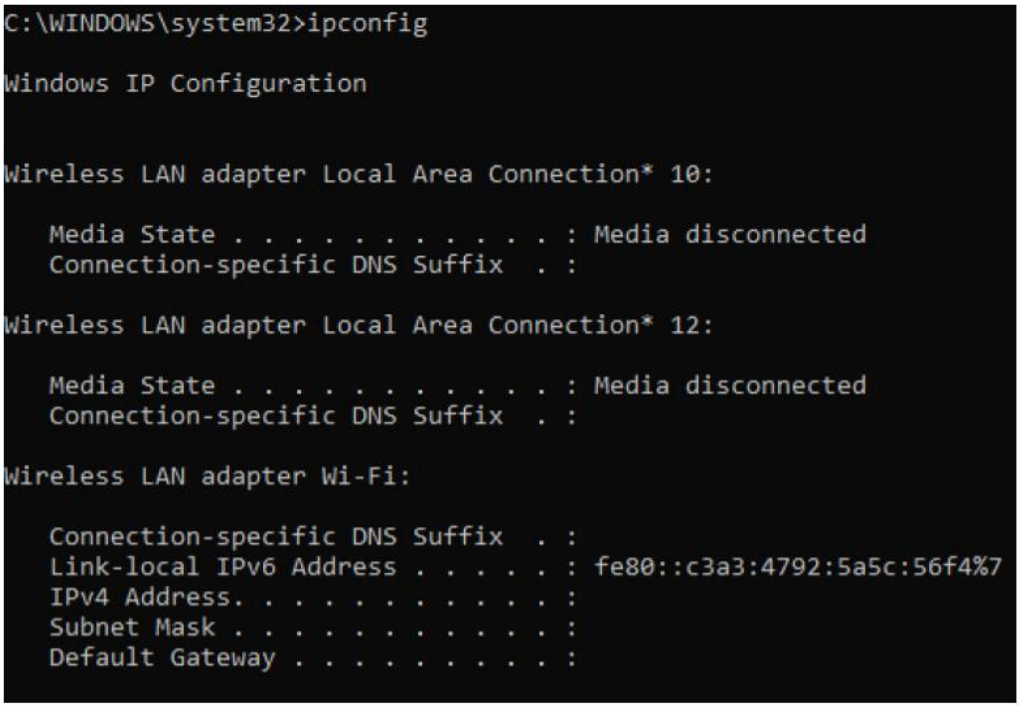


рис.5.пакет ICMP

Чтобы увидеть, начался ли захват трафика, нужно просто понаблюдать, как новые строки информации о пакетах добавляются в ARP и ICMP окна Wireshark. Когда вы видите какое-либо действие, это означает, что процесс захвата пакетов работает.

3. На вашем устройстве в консоли определите с помощью команды ipconfig для ОС типа Windows или ifconfig для систем типа Linux IP-адрес вашего устройства и шлюз по умолчанию (default gateway).

рис.5. IP-адрес и шлюз по умолчанию (default gateway)



4. На вашем устройстве в консоли с помощью команды ping адрес_шлюза пропингуйте шлюз по умолчанию. Для остановки процесса используйте комбинацию клавиш Ctrl + c или изначально при помощи параметров команды ping задайте число сообщений эхо-запроса.

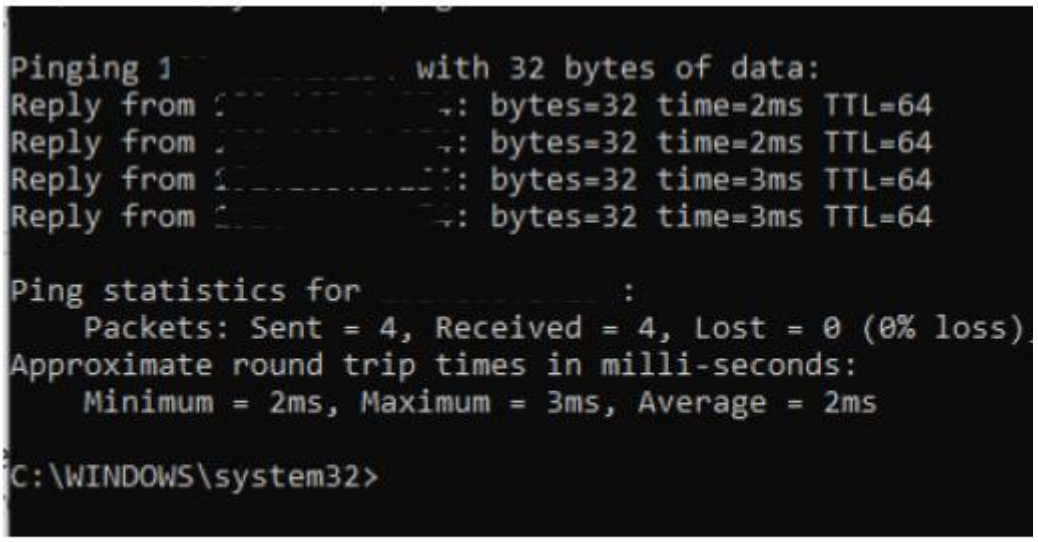


рис.6. ping адрес_шлюза пропингуйте шлюз по умолчанию

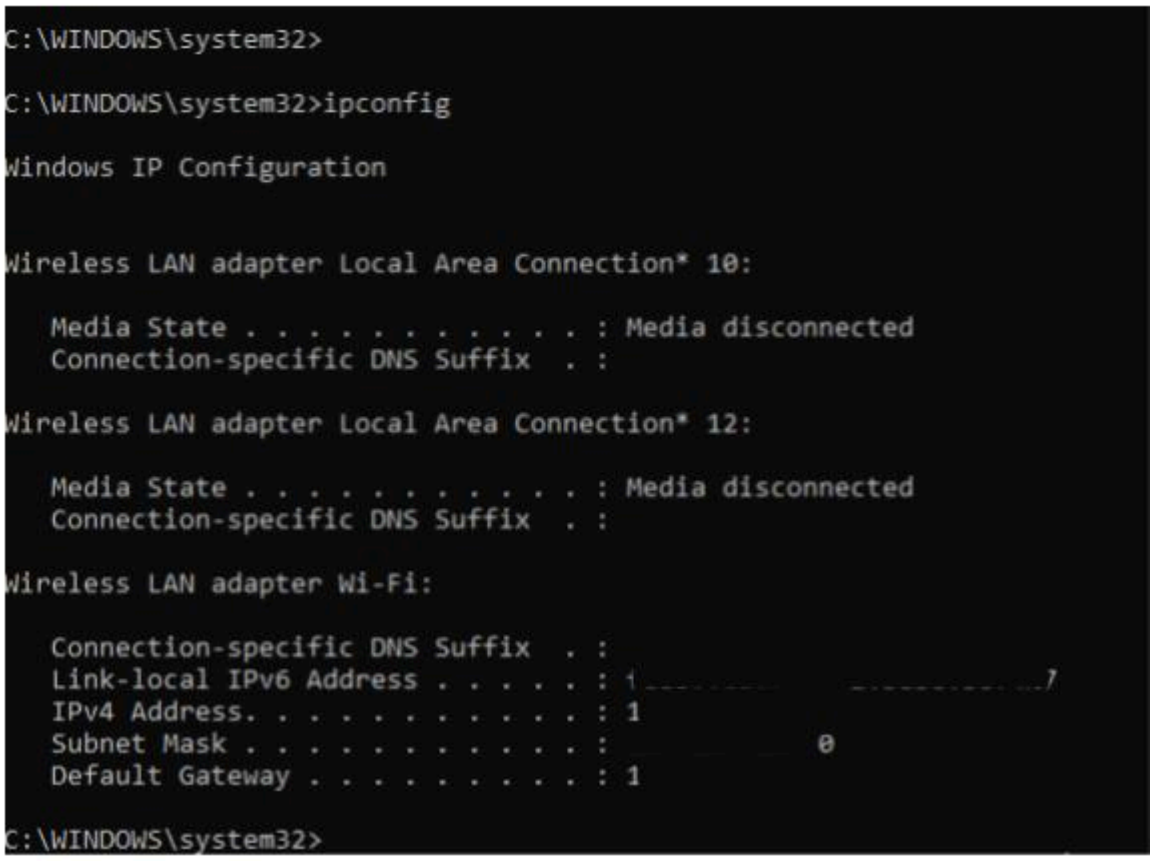


рис.7. ping отображает соответствующее имя хоста, поэтому ctrl + c сработало

5. В Wireshark остановите захват трафика. В строке фильтра пропишите фильтр arp or icmp. Убедитесь, что в списке пакетов отобразятся только пакеты ARP или ICMP, в частности пакеты, которые были сгенерированы с помощью команды ping, отправленной с вашего устройства на шлюз по умолчанию.

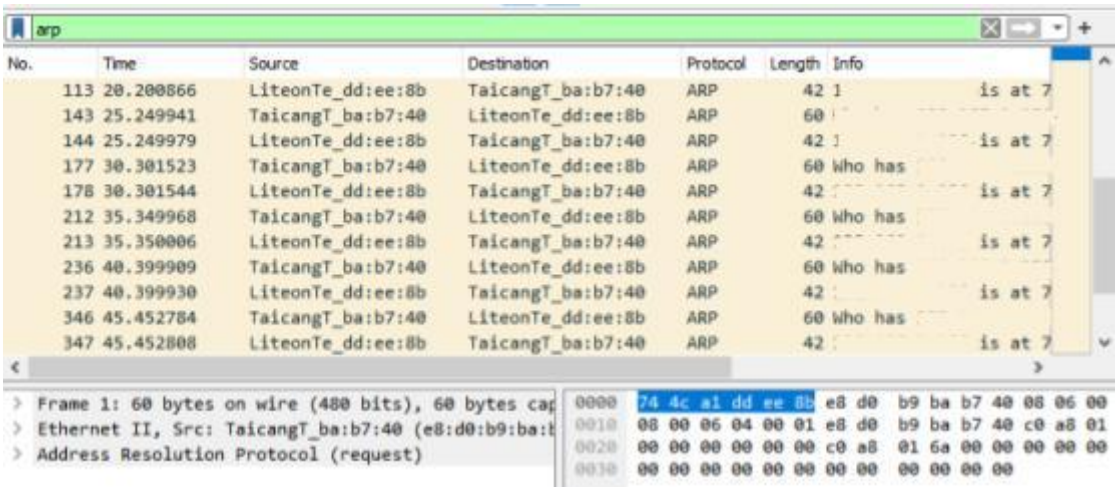


рис.7.остановка перехвата трафика ARP

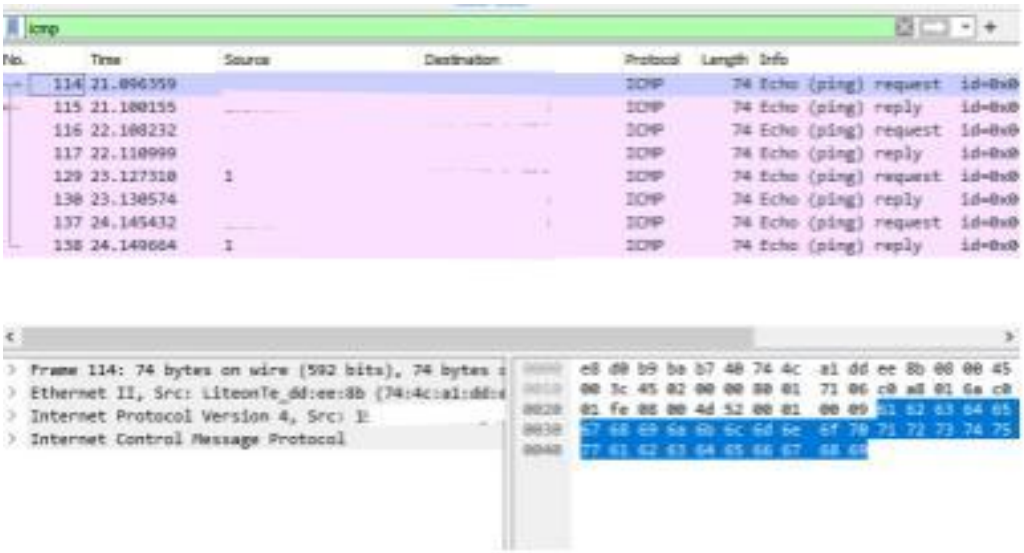


рис.8.остановка перехвата трафика ICMP

6. Изучите эхо-запрос и эхо-ответ ICMP в программе Wireshark:
– На панели списка пакетов (верхний раздел) выберите первый указанный кадр ICMP — эхо-запрос. Изучите информацию на панели сведений о пакете в средней части экрана. В отчёте укажите длину кадра, к какому типу Ethernet относится кадр, определите MAC-адреса источника и шлюза, определите тип MAC-адресов.

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :74:4c:a1:dd:ee:8b
- MAC-адрес шлюза: e8:d0:b9:ba:b7:40
- тип MAC-адресов:IPv4 (0.0800) *рис.10*

рис.9.первый указанный кадр ICMP — эхо-запрос Frame:397

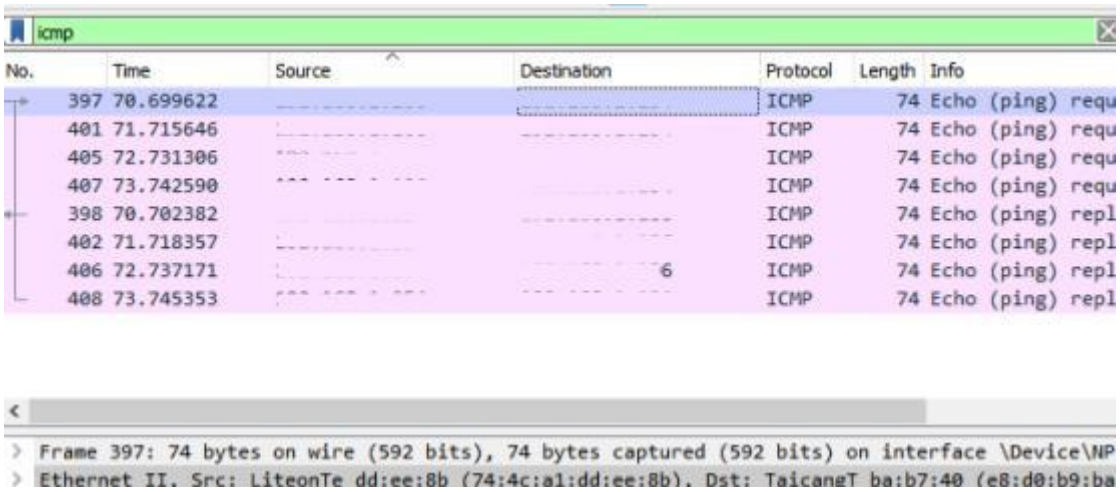
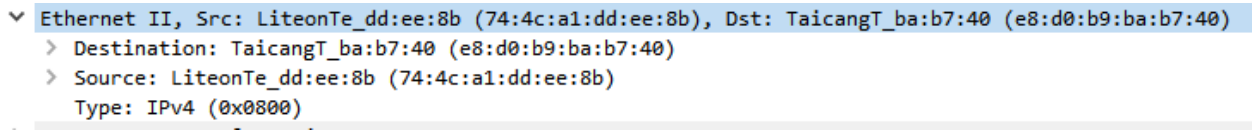


рис.10.тип MAC-адресов



– На панели списка пакетов (верхний раздел) выберите второй указанный кадр ICMP — эхо-ответ. Изучите информацию на панели сведений о пакете в средней части экрана. В отчёте укажите длину кадра, к какому типу Ethernet относится кадр, определите MAC-адреса источника и шлюза, определите тип MAC-адресов.

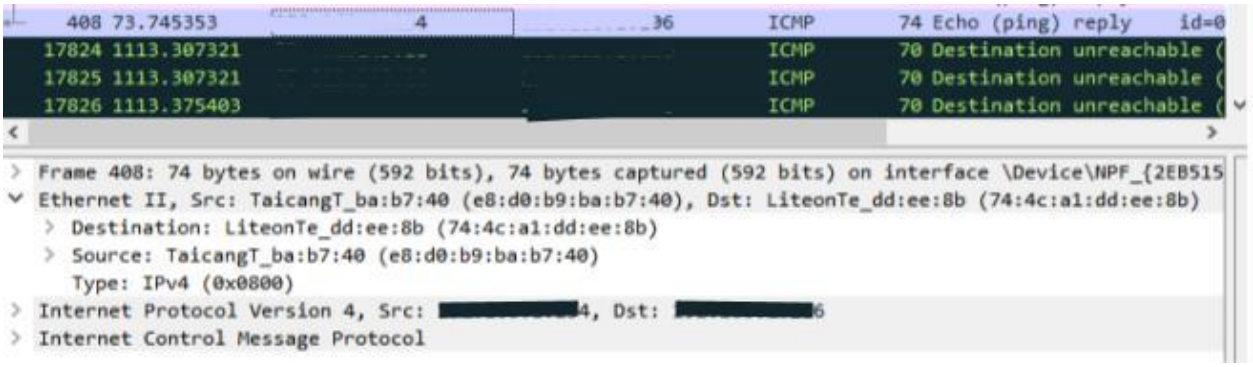
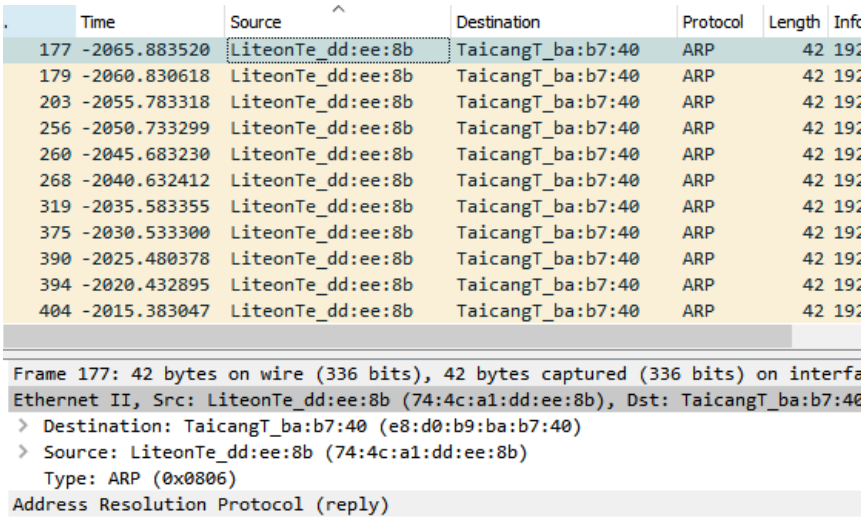


рис.11.второй указанный кадр ICMP — эхо-ответ Frame 408

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- тип MAC-адресов:IPv4 (0.0800)

7. Изучите кадры данных протокола ARP. Изучите данные в полях заголовка Ethernet II.

рис.12.кадры данных протокола ARP



Тип Ethernet II

- MAC-адрес источника:(74:4c:a1:dd:ee:8b) Поле MAC-адреса источника представляет адрес управления доступом к мультимедиа (MAC) устройства, отправившего пакет ARP. Он идентифицирует аппаратный адрес отправителя (MAC-адрес) в локальной сети.

- MAC-адрес шлюза:(e8:d0:b9:ba:b7:40)Поле MAC-адрес назначения представляет собой MAC-адрес устройства-получателя или широковещательный адрес.

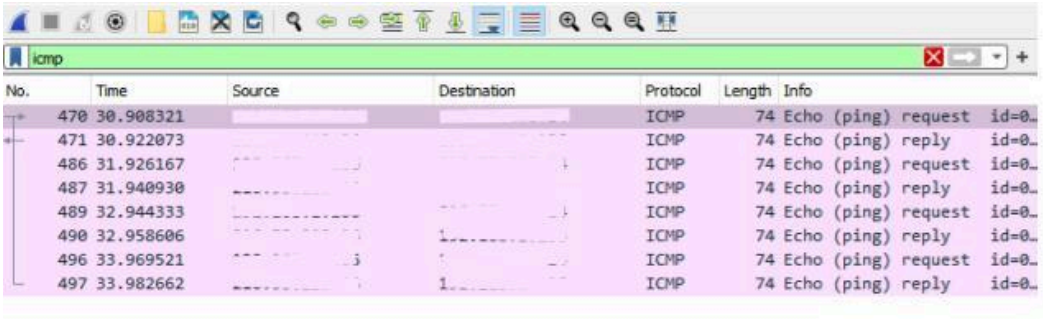
- Тип:(ARP (0x0806))Поле Type указывает тип фрейма Ethernet. В ARP-пакетах значение типа равно 0x0806, что является шестнадцатеричным представлением. Это значение указывает, что кадр содержит данные протокола ARP, позволяя принимающему устройству понять, что пакет является кадром ARP.

8. Начните новый процесс захвата трафика в Wireshark. На вашем устройстве в консоли пропингуйте по имени какой-нибудь известный вам адрес, например ping rudn.ru.

```
C:\WINDOWS\system32>ping google.com

Pinging google.com [                    ] with 32 bytes of data:
Reply from                 : bytes=32 time=12ms TTL=118
Reply from                 : bytes=32 time=12ms TTL=118
Reply from                 : bytes=32 time=15ms TTL=118
Reply from                 : bytes=32 time=12ms TTL=118

Ping statistics for                :
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 15ms, Average = 12ms

C:\WINDOWS\system32>
```

рис.13.ping google.com

Сначала ICMP:



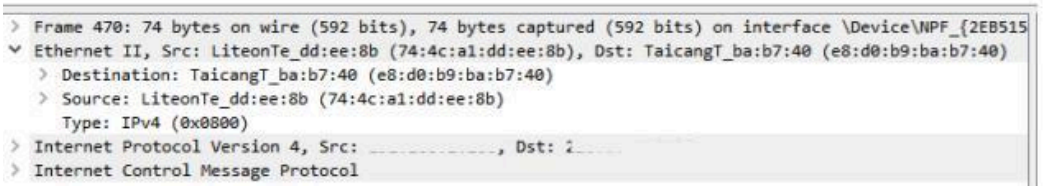


рис.14. ICMP — эхо-запрос

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :74:4c:a1:dd:ee:8b
- MAC-адрес шлюза: e8:d0:b9:ba:b7:40
- тип MAC-адресов:IPv4 (0.0800)

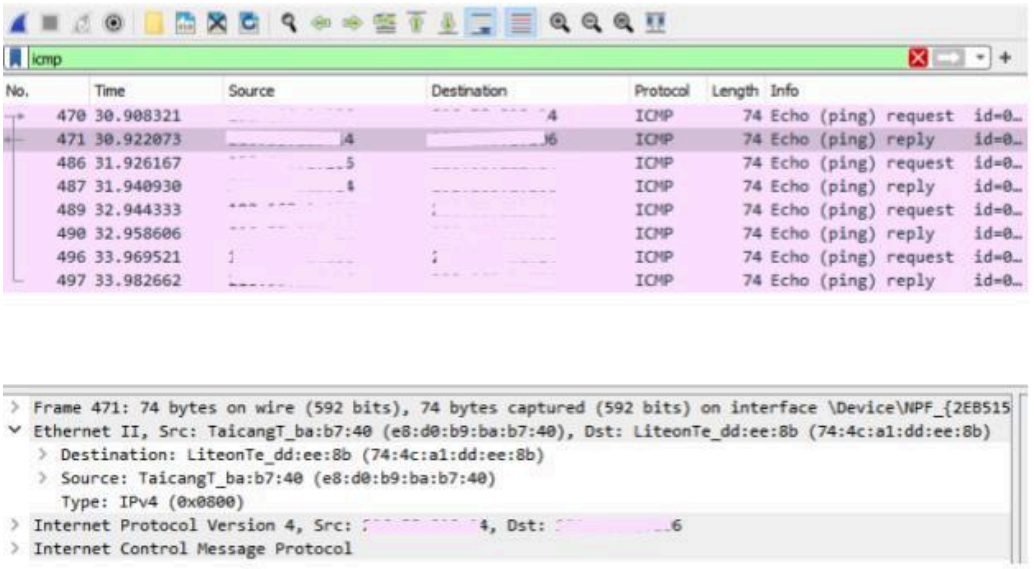


рис.15. ICMP — эхо-ответ

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- тип MAC-адресов:IPv4 (0.0800)

ARP:

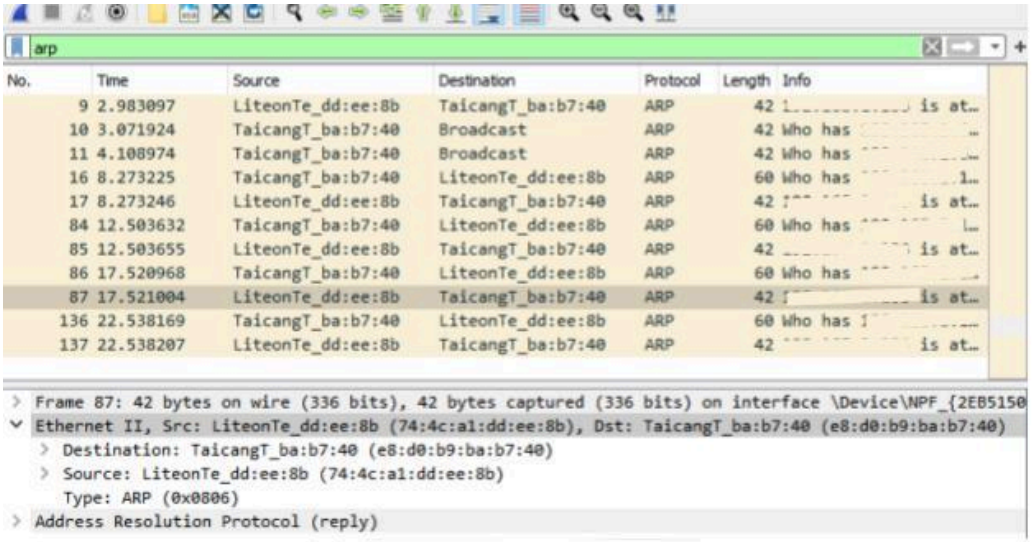


рис.16. ARP — эхо-запрос

Тип Ethernet II

- MAC-адрес источника:74:4c:a1:dd:ee:8b
- MAC-адрес шлюза:e8:d0:b9:ba:b7:40
- Тип:ARP (0x0806)

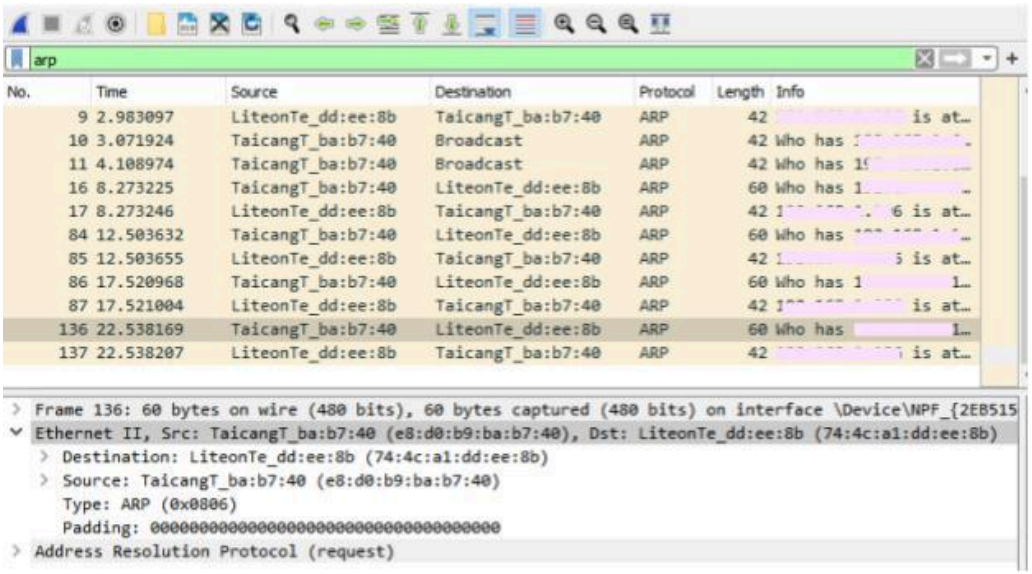


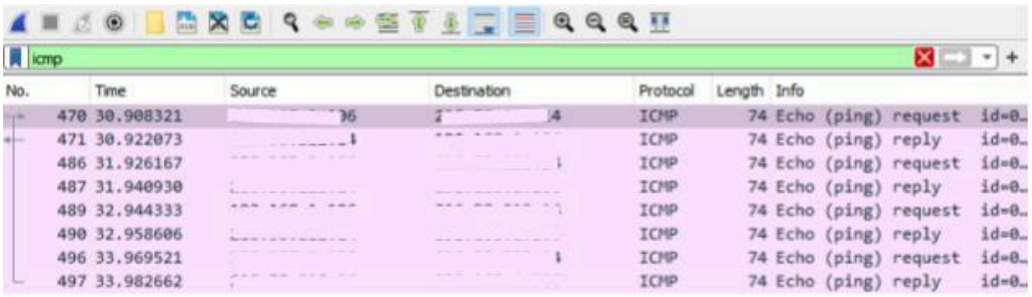
рис.17. ICMP — эхо-ответ

Тип Ethernet II

- MAC-адрес источника:e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- Тип:ARP (0x0806)

9. В Wireshark остановите захваттрафика. Изучите запросы и ответы протоколов ARP и ICMP. Определите MAC-адреса источника и получателя, определите тип MAC-адресов.

Сначала ICMP:



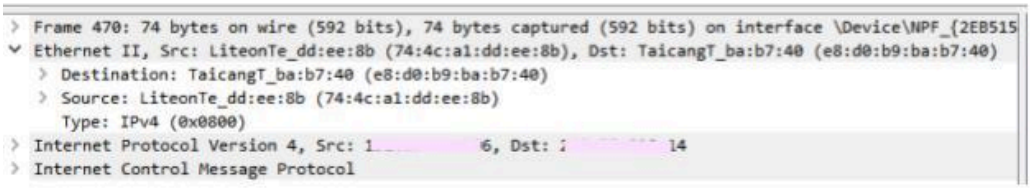


рис.14. ICMP — эхо-запрос

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II

- MAC-адрес источника :74:4c:a1:dd:ee:8b
- MAC-адрес шлюза: e8:d0:b9:ba:b7:40
- тип MAC-адресов:IPv4 (0.0800)

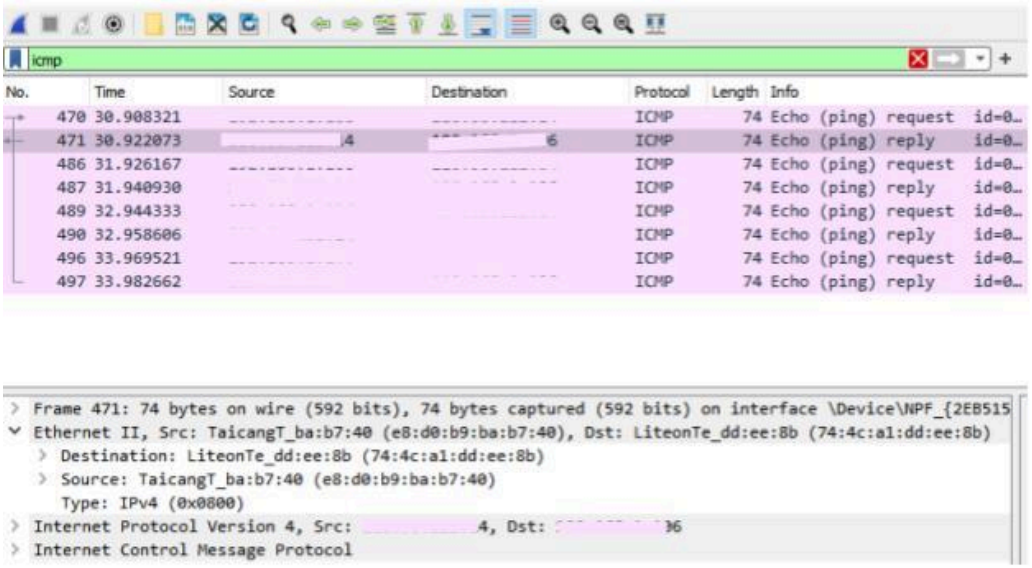


рис.15. ICMP — эхо-ответ

- длина кадра: 74 bytes (592 bits)
- тип Ethernet : Ethernet II
- MAC-адрес источника :e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- тип MAC-адресов:IPv4 (0.0800)

ARP:

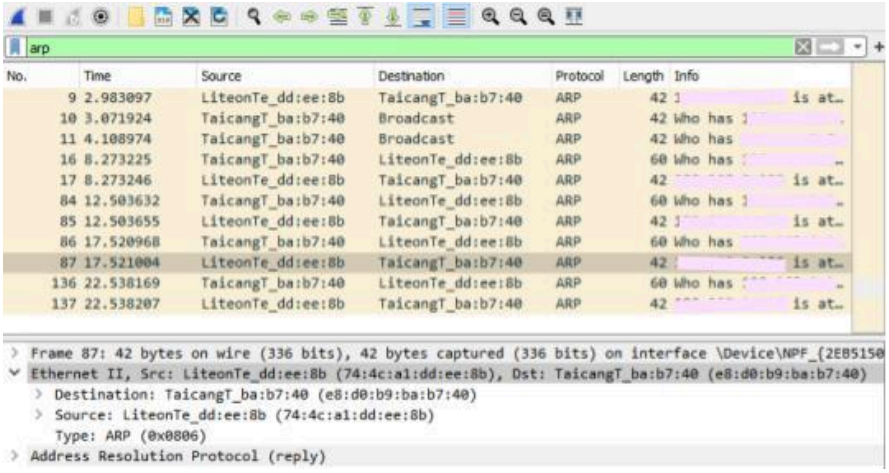


рис.16. ARP — эхо-запрос

Тип Ethernet II

- MAC-адрес источника:74:4c:a1:dd:ee:8b
- MAC-адрес шлюза:e8:d0:b9:ba:b7:40
- Тип:ARP (0x0806)

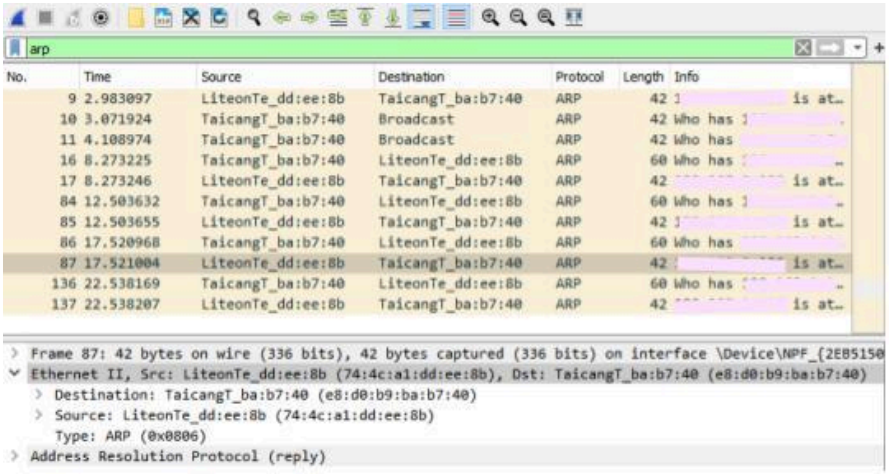


рис.17. ICMP — эхо-ответ

Тип Ethernet II

- MAC-адрес источника:e8:d0:b9:ba:b7:40
- MAC-адрес шлюза: 74:4c:a1:dd:ee:8b
- Тип:ARP (0x0806)

3.3.3. Анализ протоколов транспортного уровня в Wireshark

3.3.3.1. Постановка задачи

С помощью Wireshark захватить и проанализировать пакеты HTTP,DNS в части заголовков и информации протоколов TCP, UDP, QUIC.

3.3.3.2. Порядок выполнения работы

1. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата трафика.

рис.18.активный устройстве сетевой интерфейс wi-fi

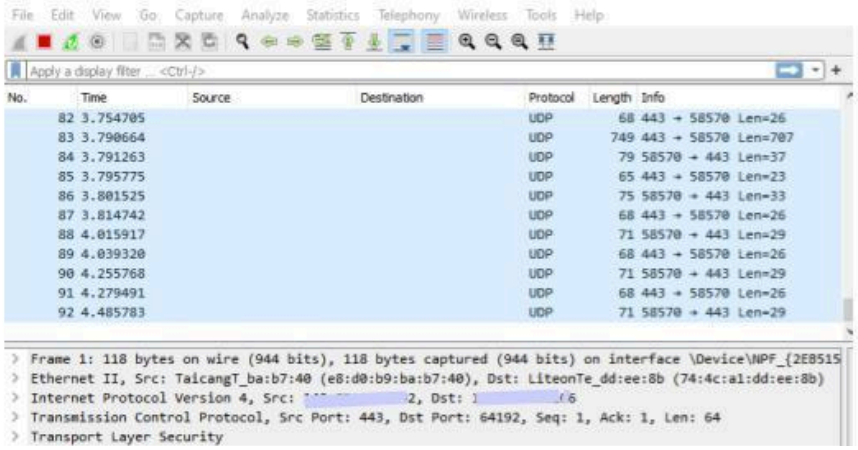


рис.19.процесс начал захватывать трафик.

2. На вашем устройстве в браузере перейдите на сайт,работающий по протоколу HTTP (например, на сайт CERN http://info.cern.ch/). При необходимости получения большей информации для Wireshark поперемещайтесь по ссылкам или разделам сайта в браузере.

рис.20.сайт CERN http://info.cern.ch/

3. В Wireshark в строке фильтра укажите http и проанализируйте информацию по протоколу TCP в случае запросов и ответов. В отчёте приведите пояснение по информации, захваченной в Wireshark.

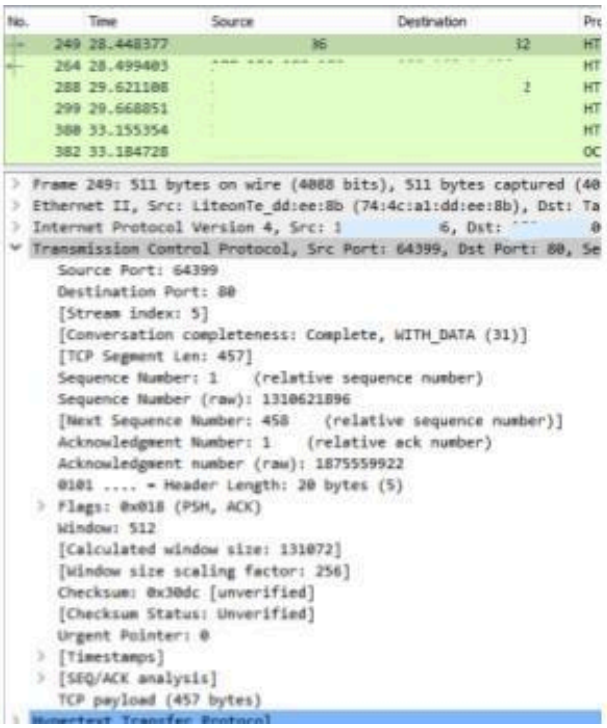


рис.21. HTTP— эхо-запрос

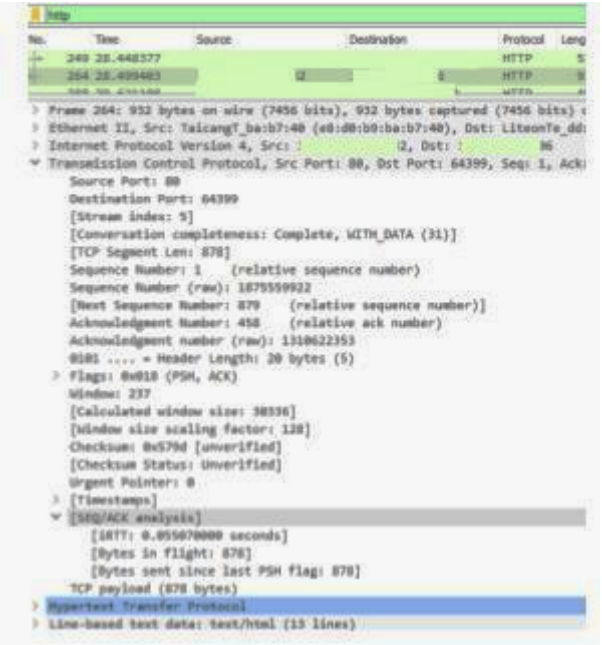


рис.22. HTTP— эхо-ответ

- Source and Destination Ports:
  - TCP использует номера портов для идентификации исходного и конечного приложений или служб.
  - В HTTP-запросе исходным портом обычно является случайно присвоенный эфемерный номер порта на вашем компьютере.
  - Порт назначения для HTTP обычно является портом 80 для обычного HTTP или портом 443 для HTTPS (безопасный HTTP).

- Sequence and Acknowledgment Numbers:

- TCP использует порядковые номера для отслеживания порядка следования пакетов данных.
- Порядковый номер указывает позицию первого байта в сегменте данных.
- Номер подтверждения подтверждает получение данных вплоть до этого порядкового номера.
- Эти цифры помогают обеспечить надежную доставку данных.

- Flags:

TCP-пакеты включают в себя различные флаги для управления передачей данных. Наши флаги - это:

- ACK (подтверждение): Указывает, что пакет подтверждает полученные данные.
- PSH (Push): подает сигнал получателю о немедленной доставке данных в приложение.

- Window:
  - Размер окна - это важное значение, которое указывает на размер принимающего буфера.
  - Это помогает регулировать поток данных и обеспечивает эффективную передачу данных.
  - Больший размер окна может обеспечить более быструю передачу данных.

- Acknowledgment of Data:
  - Поскольку данные отправляются от клиента к серверу в HTTP-запросе, вы увидите подтверждение данных в соответствующих пакетах подтверждения TCP (ACK).
  - Номер подтверждения в пакете ACK указывает следующий ожидаемый порядковый номер от сервера.

Изучая эти сведения о протоколе TCP в Wireshark, вы можете получить представление об установлении TCP-соединения, потоке данных и подтверждении данных между клиентом и веб-сервером во время HTTP-запроса. Эта информация необходима для понимания надежности и эффективности передачи данных по сети.

4. Wireshark в строке фильтра укажите dns и проанализируйте информацию по протоколу UDP в случае запросов и ответов. В отчёте приведите пояснение по информации, захваченной в Wireshark.

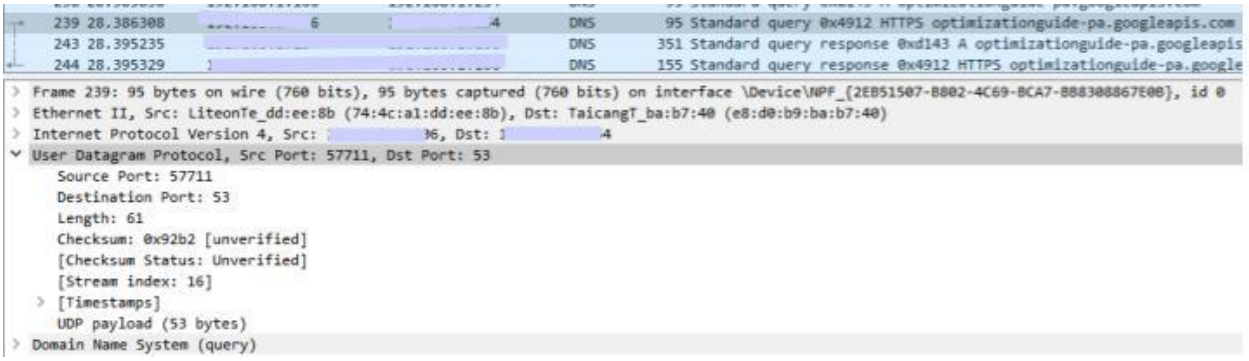


рис.23. dns— эхо-запрос

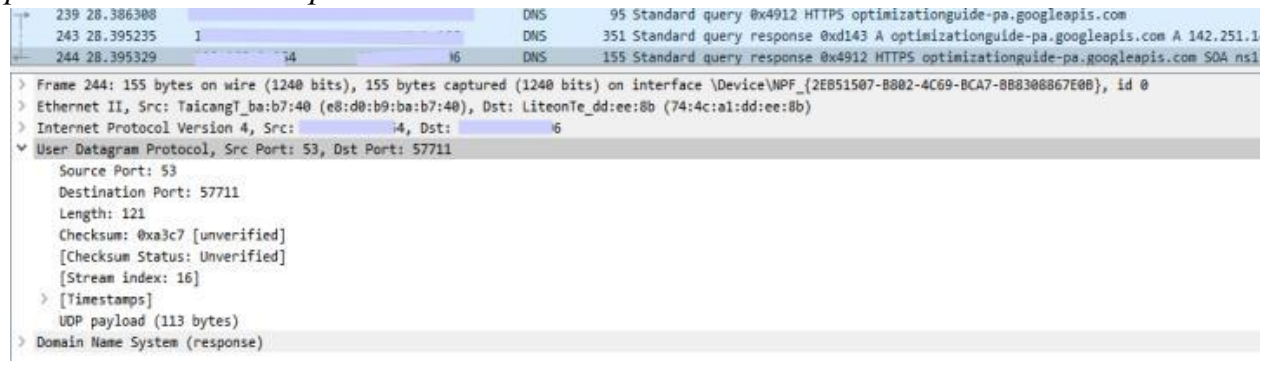


рис.24. dns— эхо-ответ

Запрос DNS:

- User Datagram Protocol (UDP): Это указывает на то, что пакет использует транспортный протокол UDP.
- Src Port (Source Port): номер порта источника, с которого отправляется DNS-запрос. В данном случае это порт 57711.
- Dst Port (Destination Port): Номер порта назначения, на который отправляется DNS-запрос. Порт 53 является стандартным портом для DNS.
- Length: общая длина UDP-пакета, включая UDP-заголовок и полезную нагрузку. В данном случае это 61 байт.
- Checksum: Поле контрольной суммы используется для проверки ошибок. Он в шестнадцатеричном формате (0x92b2) и помечен как "непроверенный". Контрольные суммы UDP необязательны, и в данном случае они не проверяются.
- Stream index: Этот индекс помогает Wireshark организовывать и отображать связанные пакеты. В данном случае он помечен как stream index 16.
- Timestamps: Это указывает на то, что в пакет включены временные метки, которые могут использоваться для различных целей, включая измерение времени в пути туда и обратно.
- UDP payload (53 bytes): Фактические данные, переносимые пакетом UDP. В DNS-запросах эта полезная нагрузка обычно включает в себя DNS-запрос, включая запрашиваемое доменное имя.

Ответ DNS:

User Datagram Protocol (UDP):: Как и в случае с запросом, это указывает на использование транспортного протокола UDP.
- Src Port (Source Port): номер порта источника для ответа DNS. В данном случае

это стандартный DNS-порт, 53.
Dst Port (Destination Port): Номер порта назначения, на который отправляется ответ DNS. Порт 57711 совпадает с исходным портом исходного запроса.
- Length: общая длина UDP-пакета, включая UDP-заголовок и полезную нагрузку. В данном случае это 121 байт.
- Checksum: Поле контрольной суммы, снова помеченное как "непроверенное". Контрольные суммы UDP являются необязательными.

- Stream index: тот же индекс потока, что и у запроса DNS (индекс потока 16), указывающий, что этот ответ связан с запросом.
- Timestamps: Временные метки также включены в пакет.
- UDP payload (53 bytes): Фактические данные, переносимые пакетом UDP. В ответах DNS это обычно включает разрешенный IP-адрес (или другие записи DNS), соответствующий доменному имени из исходного запроса.

Таким образом, эти сведения о захвате Wireshark показывают обмен пакетами запросов DNS и ответов. Запрос DNS инициируется с исходного порта (57711) на порт DNS-сервера (53). Ответ DNS возвращается с сервера на исходный порт (57711) с разрешенной информацией. Для этого обмена используется протокол UDP, и хотя контрольные суммы присутствуют, в данном случае они непроверены. Временные метки предоставляют дополнительный контекст для измерения информации, связанной со временем, такой как время в пути туда и обратно.

5. Wireshark в строке фильтра укажите quic и проанализируйте информацию по протоколу quic в случае запросов и ответов. В отчёте приведите пояснение по информации, захваченной в Wireshark.

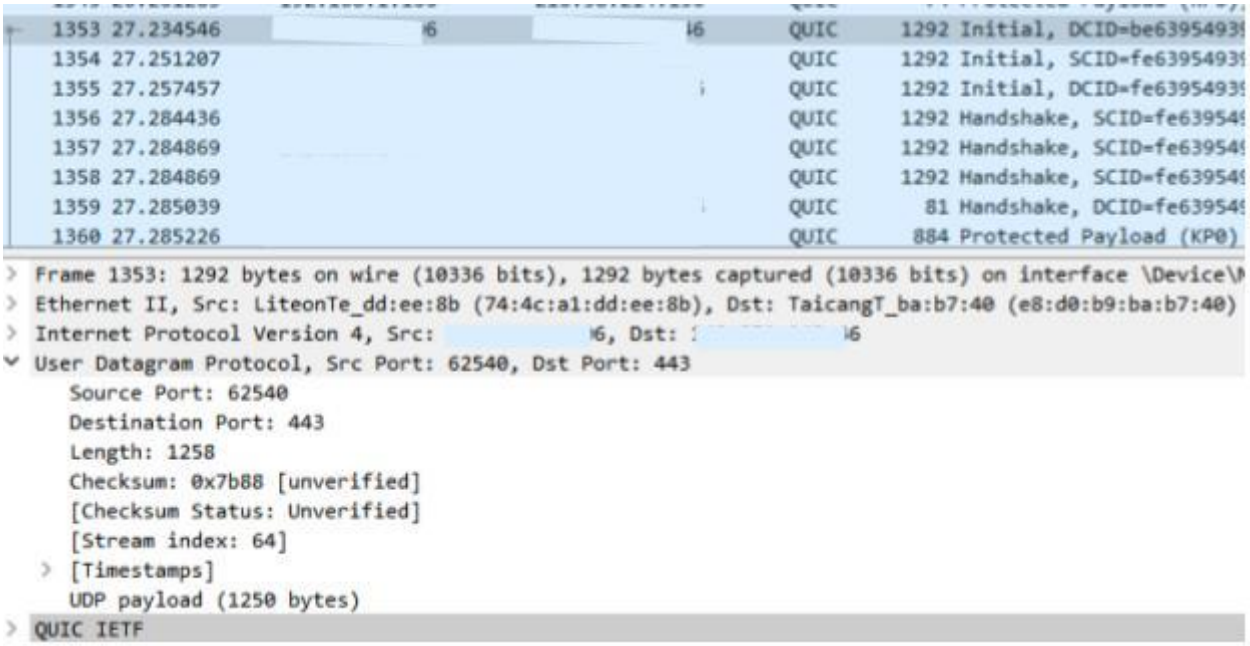


рис.25. quic— эхо-запрос

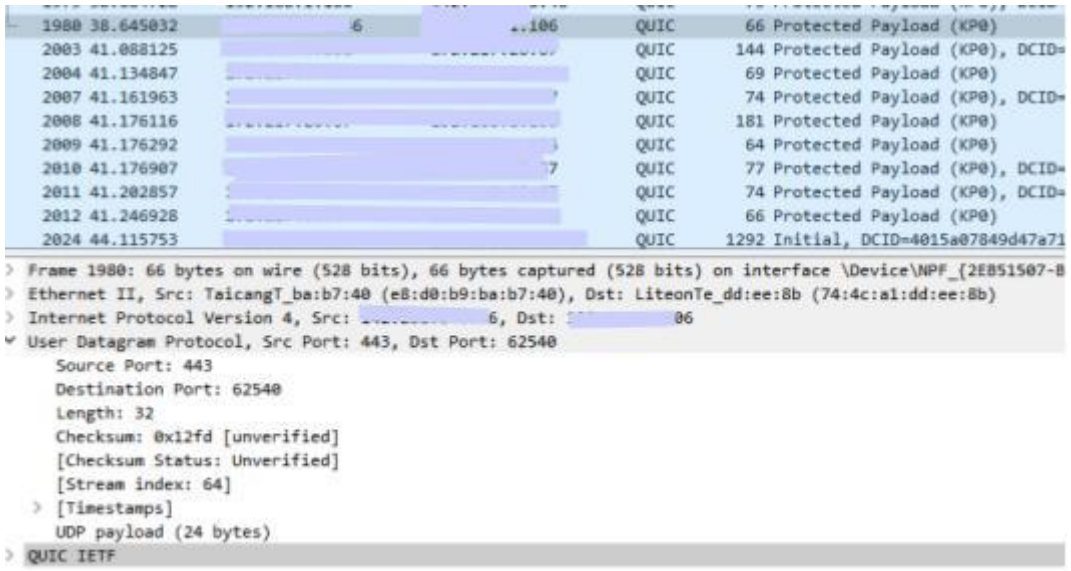


рис.26. quic— эхо-ответ

Давайте разберем их подробно:

Запрос (Request):

- Протокол: User Datagram Protocol (UDP)
- Исходный порт (Src Port): 62540
- Порт назначения (Dst Port): 443
- Длина (Length): 1258 байт
- Контрольная сумма (Checksum): 0x7b88 [непроверенная]
- Статус контрольной суммы: Непроверенная
- Индекс потока: 64
- Метка времени (Timestamps)
- Полезная нагрузка UDP (UDP payload): 1250 байт

Ответ (Response):

- Протокол: User Datagram Protocol (UDP)
- Исходный порт (Src Port): 443
- Порт назначения (Dst Port): 62540
- Длина (Length): 32 байта
- Контрольная сумма (Checksum): 0x12fd [непроверенная]
- Статус контрольной суммы: Непроверенная
- Индекс потока: 64
- Метка времени (Timestamps)
- Полезная нагрузка UDP (UDP payload): 24 байта

Теперь давайте объясним, что означает каждое из этих полей:

- Протокол (Protocol): Указывает на то, что это UDP-пакет.

- Исходный порт (Source Port): Порт отправителя (вашего компьютера) для идентификации отправителя.

- Порт назначения (Destination Port): Порт получателя (какой-либо удаленный сервер или служба), куда направлен запрос или ответ.

- Длина (Length): Общая длина UDP-пакета в байтах, включая заголовок и данные.

- Контрольная сумма (Checksum): Значение контрольной суммы, которое помогает обнаруживать ошибки в данных (но в данном случае она отмечена как непроверенная).

- Статус контрольной суммы (Checksum Status): Показывает, что контрольная сумма пакета не была проверена.

- Индекс потока (Stream Index): Индекс потока, связанный с этим пакетом.

- Метка времени (Timestamps): Метка времени может использоваться для отслеживания времени отправки и приема пакетов.

- Полезная нагрузка UDP (UDP payload): Это собственно данные, передаваемые внутри UDP-пакета. Длина полезной нагрузки указана в байтах.

Запрос был отправлен с порта 62540 на порт 443, а ответ был получен с порта 443 на порт 62540. Ответ имеет гораздо меньшую длину по сравнению с запросом.

6. Остановите захват трафика в Wireshark

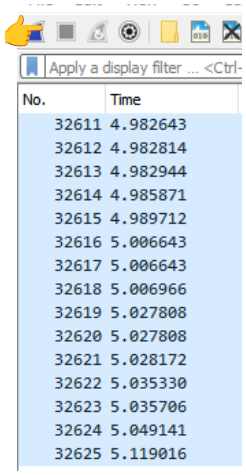


рис.26. Остановка захват трафика в Wireshark(нет красный свет (emoji знак)

3.3.4. Анализ handshake протокола TCP в Wireshark

3.3.4.1. Постановка задачи

С помощью Wireshark проанализировать handshake протокола TCP.

3.3.4.2. Порядок выполнения работы

1. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата трафика.

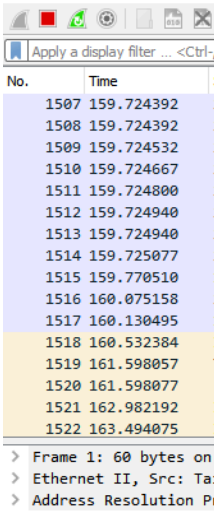
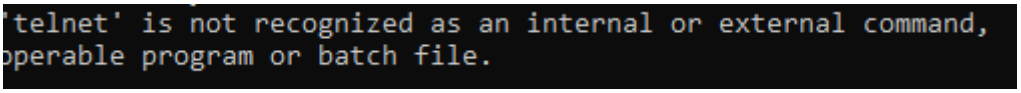


рис.27.Начался процесс перехвата трафика

2. На вашем устройстве или используйте подсоединение по telnet или ssh к вашему маршрутизатору (например с помощью PUTTY или соответствующих команд в консоли), или соединение по HTTP с каким-то сайтом для захвата в Wireshark пакетов TCP.



3. В Wireshark проанализируйте handshake протокола TCP, в отчёте приведите пример с пояснениями изменения значений соответствующих сообщений при установлении соединения по TCP.

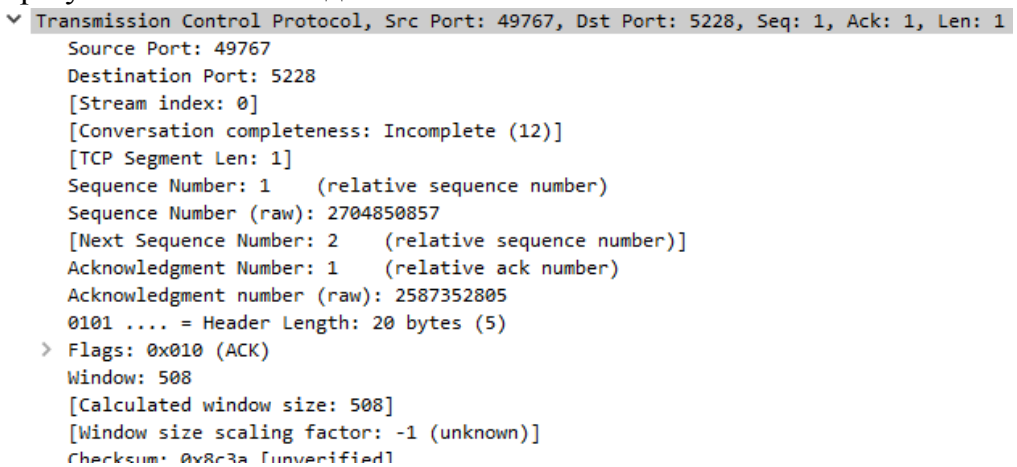


рис.28. tcp-эхо-запрос

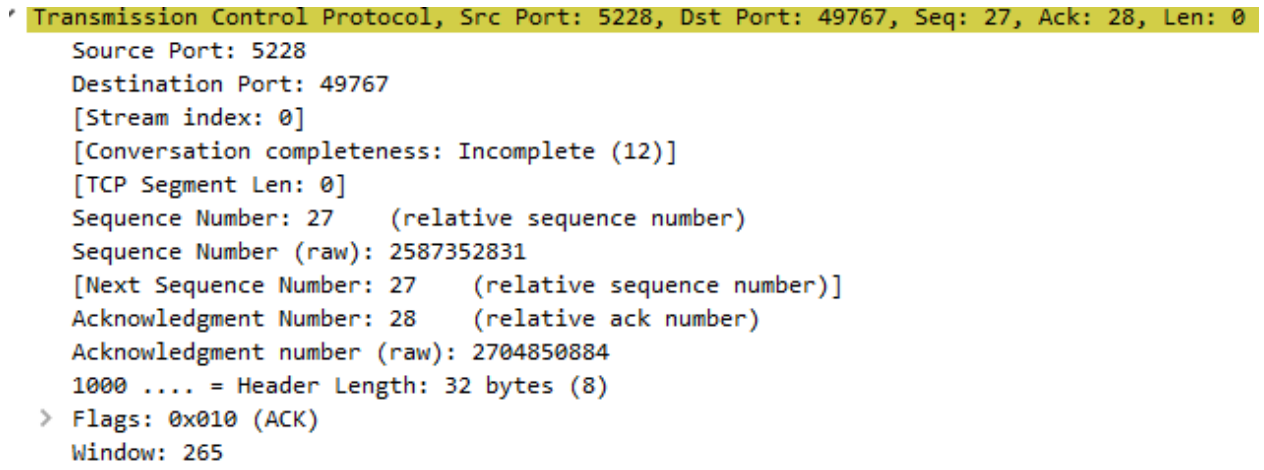


рис.29. tcp-эхо-ответ

При анализе подтверждения по протоколу TCP в Wireshark вы обычно обращаете внимание на процесс трехстороннего подтверждения, который происходит при установлении TCP-соединения. Тремя этапами рукопожатия являются:

SYN (синхронизировать) - Инициирующее устройство отправляет TCP-пакет с установленным флагом SYN на сервер для запроса соединения.

SYN-ACK (Синхронизировать-подтверждение) - Сервер отвечает TCP-пакетом, содержащим флаги SYN и ACK, установленные для подтверждения запроса и инициирования соединения.

ACK (подтверждение) - Инициирующее устройство отправляет окончательный TCP-пакет с установленным флагом ACK для подтверждения ответа сервера, завершая настройку соединения.

Подтверждение (от клиента к серверу):
Порт источника: тот же порт источника, что и раньше.
Порт назначения: Тот же порт назначения, что и раньше.
Порядковый номер: увеличивается на 1 по сравнению с предыдущим порядковым номером клиента.
Номер подтверждения: увеличивается на 1 по сравнению с предыдущим номером подтверждения сервера.
Флаги: Флаг SYN не установлен, флаг подтверждения установлен.

4. В Wireshark в меню «Статистика» выберете «График Потока». В отчёте приведите пояснения по изменениям значений соответствующих сообщений при установлении соединения по TCP.

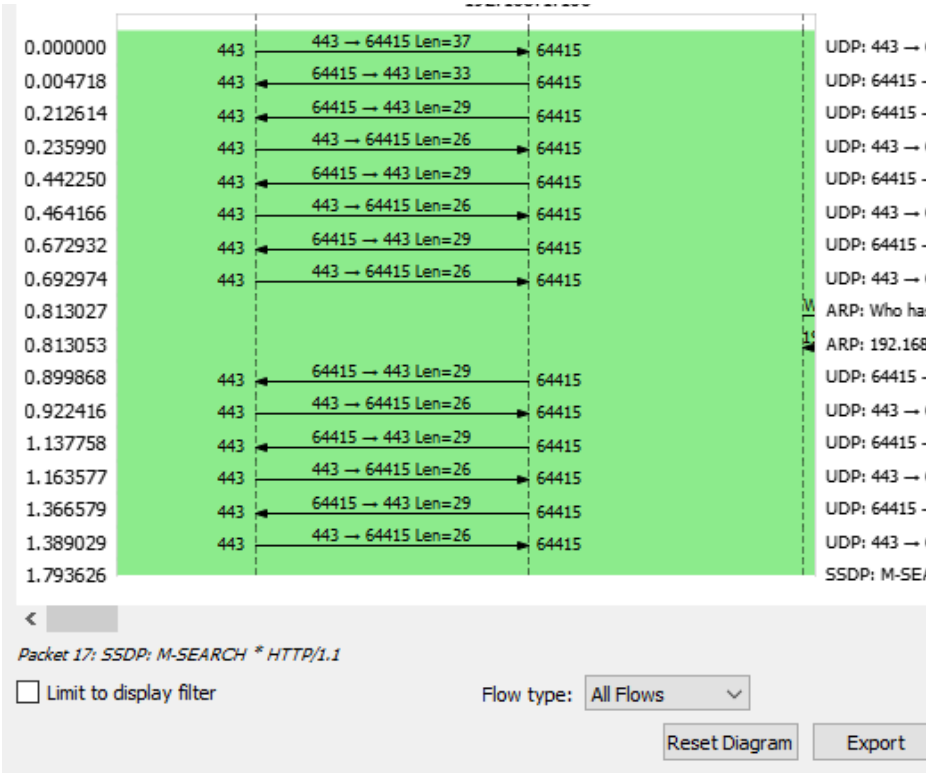


рис.30.График Потока

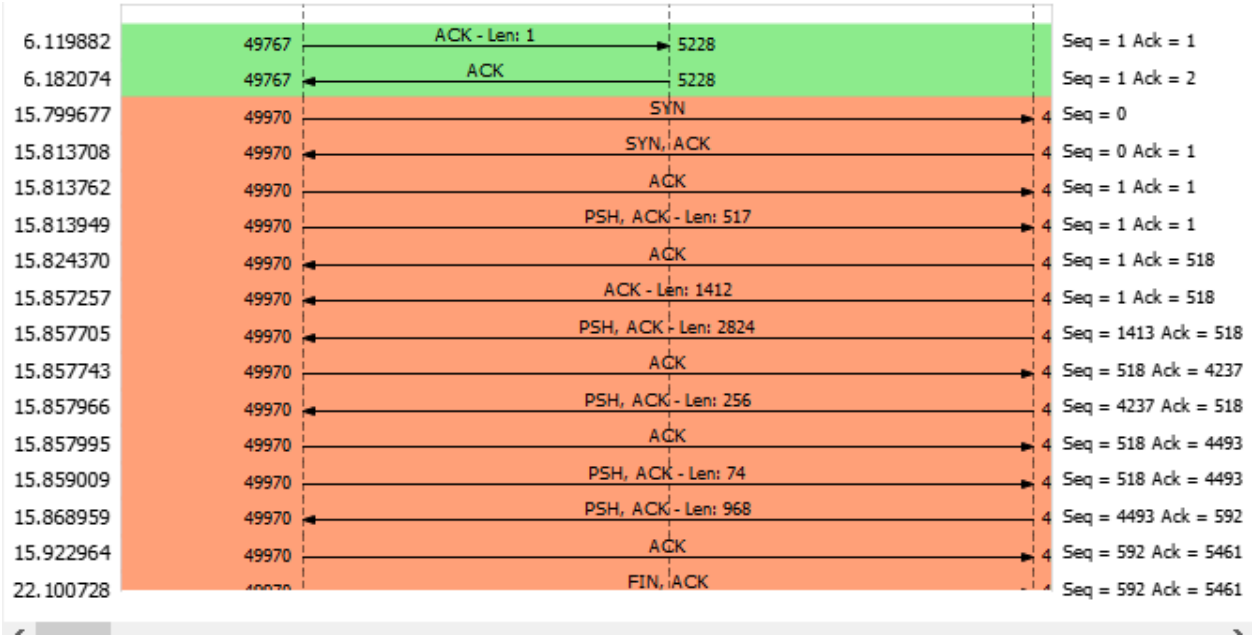


рис.31.TCP flows

Трехстороннее рукопожатие TCP включает в себя три основных пакета: SYN, SYN-ACK и ACK-подтверждение. Давайте разберем изменения в их значениях:

Пакет SYN (синхронизировать) (клиент-сервер):

Порт источника: Случайно выбранный временный порт на стороне клиента.
Порт назначения: Хорошо известный порт, связанный со службой на сервере (например, 80 для HTTP, 22 для SSH).
Порядковый номер: Случайное число, выбранное клиентом, часто называемое

начальным порядковым номером (ISN).
Номер подтверждения: 0 (поскольку это начальный пакет и подтверждать пока нечего).
Флаги: установлен флаг SYN (указывающий на начало соединения), флаг подтверждения не установлен (поскольку данные еще не получены).

SYN-ACK (Синхронизация-подтверждение) Пакет (от сервера к клиенту):
Порт источника: Хорошо известный порт, связанный со службой на сервере.
Порт назначения: исходный порт из пакета SYN (временный порт на стороне клиента).
Порядковый номер: случайное число, выбранное сервером (ISN сервера).
Номер подтверждения: Номеру подтверждения присваивается порядковый номер клиента + 1. Это подтверждает получение SYN-пакета от клиента.
Флаги: установлен флаг SYN (указывает на продолжение инициализации соединения), установлен флаг ACK (подтверждает SYN от клиента).

ACK (Подтверждение) Пакет (от клиента к серверу):
Порт источника: тот же самый эфемерный порт на стороне клиента.
Порт назначения: Тот же самый хорошо известный порт на стороне сервера.
Порядковый номер: увеличивается на 1 по сравнению с предыдущим порядковым номером клиента. Это подтверждает получение пакета SYN-ACK от сервера.
Номер подтверждения: увеличивается на 1 по сравнению с предыдущим номером подтверждения сервера. Это подтверждает получение SYN-ACK от сервера.
Флаги: Флаг SYN не установлен (поскольку настройка соединения завершена), установлен флаг ACK (подтверждение SYN-ACK с сервера).
Таким образом, ключевые изменения во время трехстороннего подтверждения связи TCP включают в себя последовательность пакетов и установку флагов:

Пакет SYN инициирует соединение и содержит начальный порядковый номер клиента.
Пакет SYN-ACK подтверждает SYN клиента, содержит начальный порядковый номер сервера и подтверждает порядковый номер клиента.
Пакет ACK подтверждает SYN-ACK от сервера, увеличивая порядковые номера как клиента, так и сервера на 1.
Эти шаги гарантируют, что обе стороны соединения синхронизированы и знают начальные порядковые номера друг друга, что обеспечивает надежную передачу данных.

5. Остановите захват трафика в Wireshark.

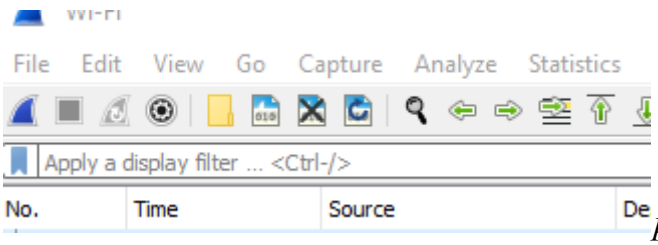


рис.32.захват трафика в Wireshark

3.3.4. Анализ handshake протокола TCP в Wireshark

3.3.4.1. Постановка задачи

С помощью Wireshark проанализировать handshake протокола TCP.
Анализ "handshake" протокола TCP с использованием Wireshark - это отличный способ изучить как работает установление соединения между двумя устройствами. Протокол TCP используется для надежной передачи данных в сети. Handshake TCP состоит из трех этапов: установление соединения, передача данных и завершение соединения. Давайте разберем каждый этап на примере с помощью Wireshark:

1. Установление соединения (TCP Three-Way Handshake):
- Когда клиент хочет установить соединение с сервером, он отправляет TCP пакет с флагом SYN (Synchronize) и начальным номером последовательности (ISN - Initial Sequence Number).
- Когда сервер получает этот пакет, он отправляет ответный пакет с флагами SYN и ACK (Acknowledgment). В этом пакете сервер устанавливает свой ISN.
- Клиент, получив ответ от сервера, отправляет третий пакет с флагом ACK. Этот пакет подтверждает установление соединения.

В Wireshark вы можете найти эти пакеты, используя фильтр `tcp.flags.syn == 1 && tcp.flags.ack == 0` для поиска пакета SYN от клиента и `tcp.flags.syn == 1 && tcp.flags.ack == 1` для поиска пакета SYN-ACK от сервера.

2. Передача данных:
- После успешного установления соединения клиент и сервер могут обмениваться данными, отправляя TCP пакеты с флагом ACK.

Для отображения данных, передаваемых между клиентом и сервером, вы можете просто просматривать TCP пакеты в Wireshark и анализировать их содержимое.

3. Завершение соединения (TCP Four-Way Handshake):
- Когда клиент или сервер хочет завершить соединение, они отправляют TCP пакет с флагом FIN (Finish).
- Завершение соединения состоит из двух этапов: сначала одна сторона отправляет FIN, а затем другая подтверждает завершение соединения.

Для анализа завершения соединения, вы можете использовать фильтры `tcp.flags.fin == 1 && tcp.flags.ack == 0` и `tcp.flags.fin == 1 && tcp.flags.ack == 1` для поиска соответствующих пакетов.

Обратите внимание, что в Wireshark вы можете также просматривать дополнительные сведения о каждом пакете, такие как порты, IP-адреса, длина данных и т. д., что помогает в более подробном анализе handshake и обмена данными в TCP соединении.

Выводы

В этой лабораторной работе я изучил фреймы Ethernet с помощью Wireshark и научился анализировать протоколы PDU транспортного и прикладного уровней стека TCP/IP.