ГУАП

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | Д.Д. Савельева |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 |
| АНАЛИЗ ТРАФИКА КОМПЬЮТЕРНЫХ СЕТЕЙ УТИЛИТОЙ WIRESHARK |
| по курсу: Инфокоммуникационные системы и сети |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4329 |  |  |  | Д.С. Шаповалова |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

# **1. Цель работы:**

Изучить структуру протокольных блоков данных, анализируя реальный трафик с помощью утилиты Wireshark.

# **2. Ход работы:**

*1. Анализ трафика утилиты ping*

Был отслежен и проанализирован трафик утилитой Wireshark, создаваемый утилитой ping, при запуске из командной строки. В качестве “размера\_пакета” были поочерёдно использованы значения от 100 до 10000.

Результат отслеживания представлен на рисунках 1.1-1.3:

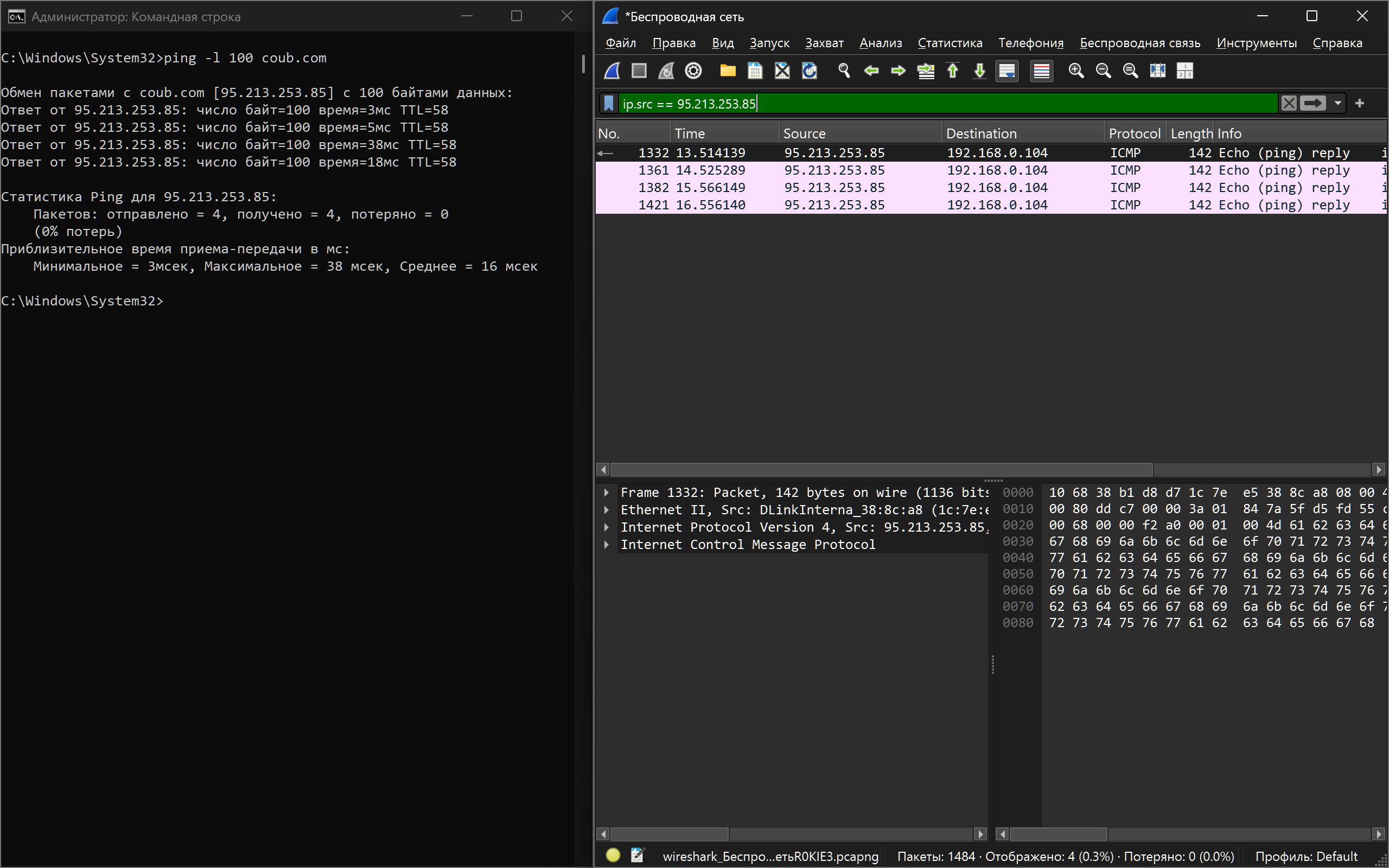


Рисунок 1.1 – ping -l 100 coub.com

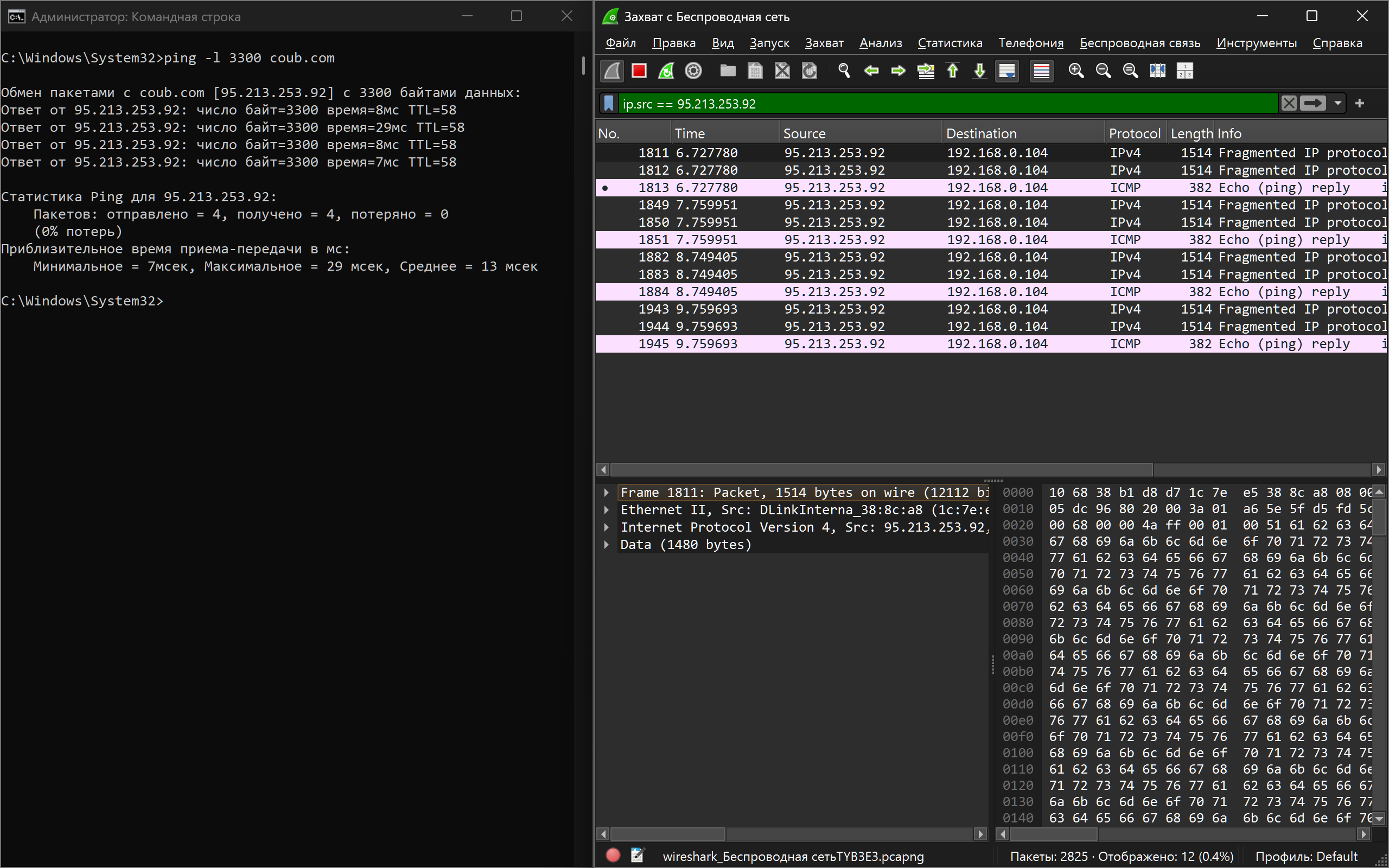


Рисунок 1.2.1 – ping -l 3300 coub.com

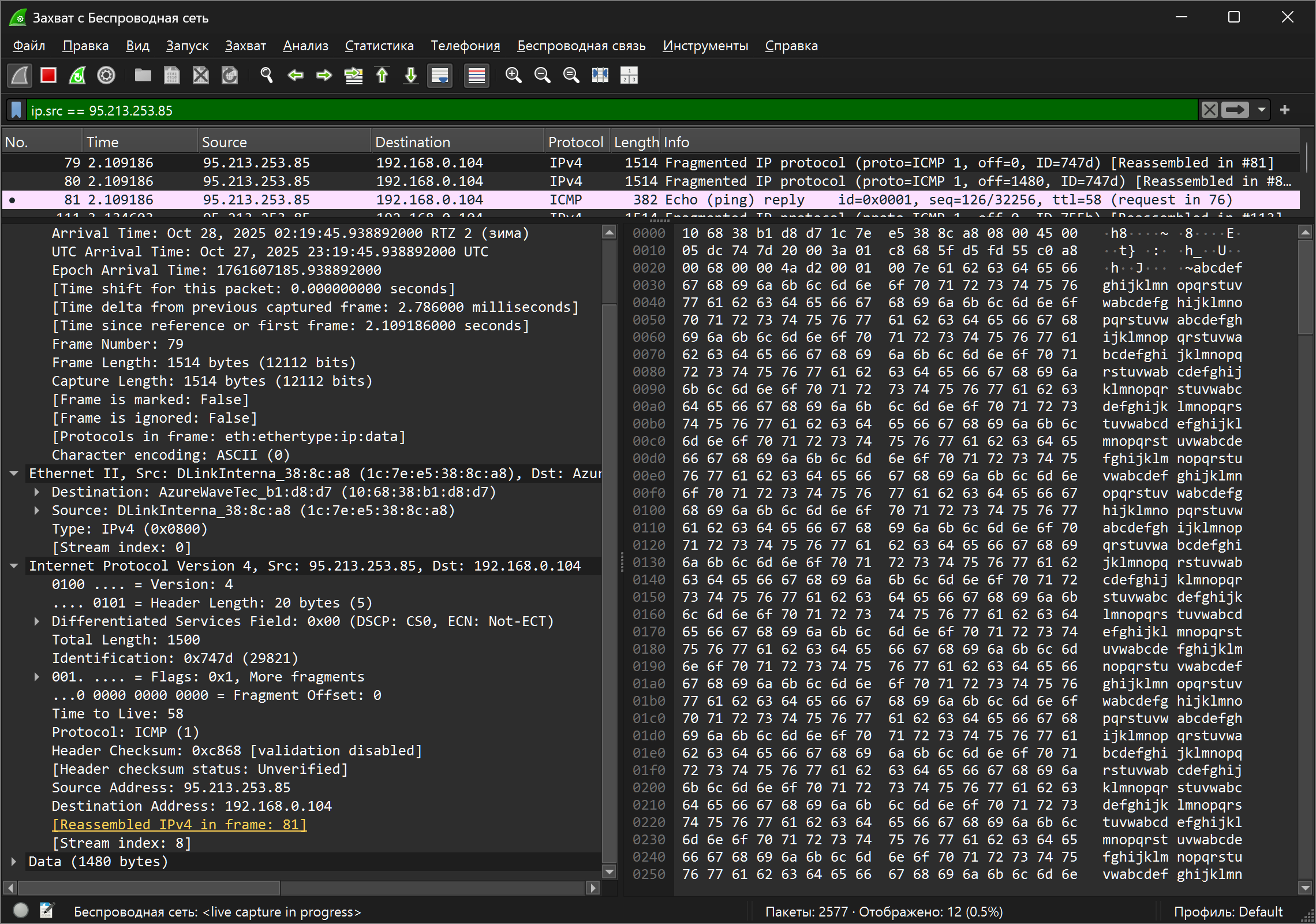


Рисунок 1.2.2 – ping -1 3300 coub.com, подробнее

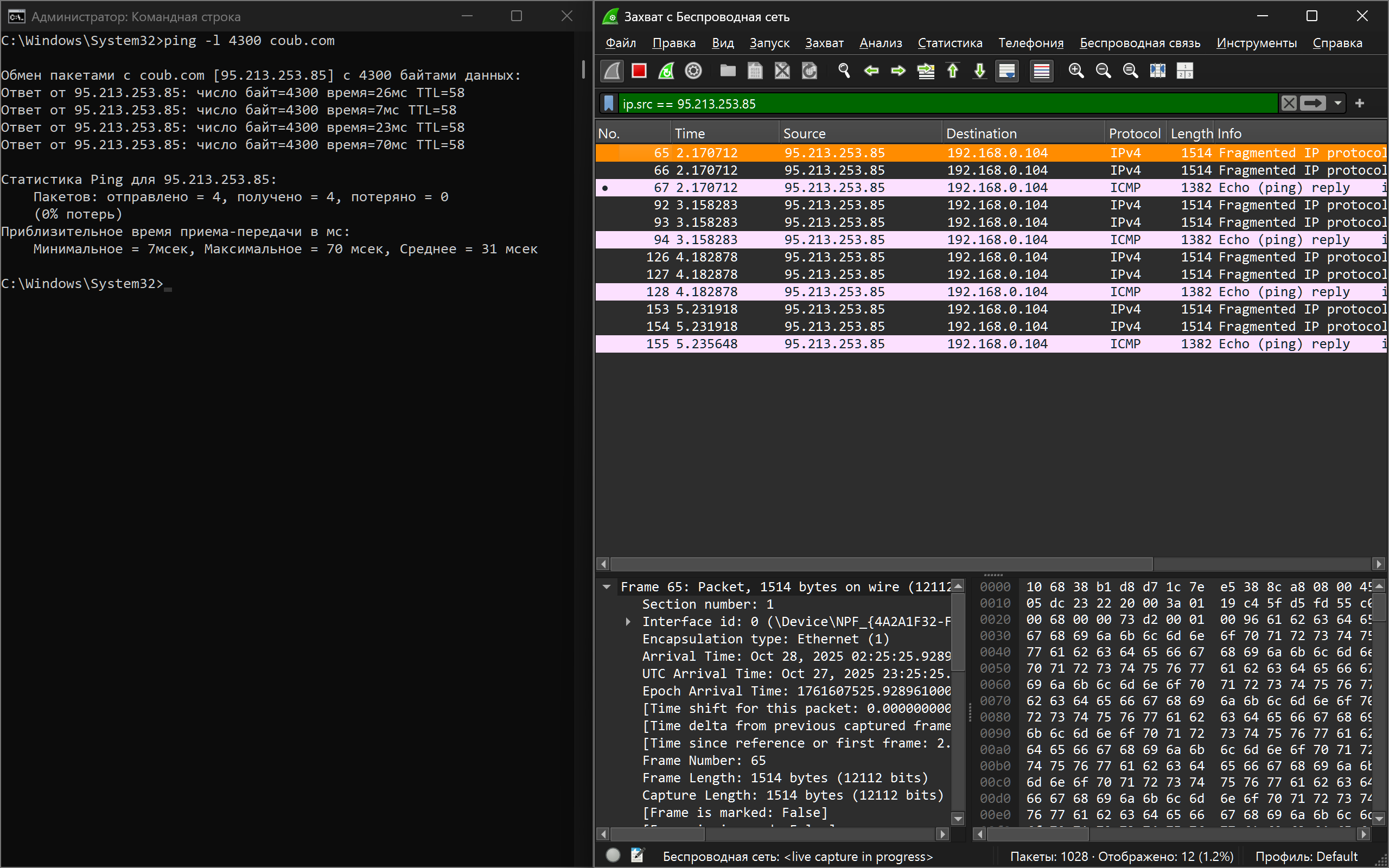


Рисунок 1.3.1 – ping -1 4000 coub.com

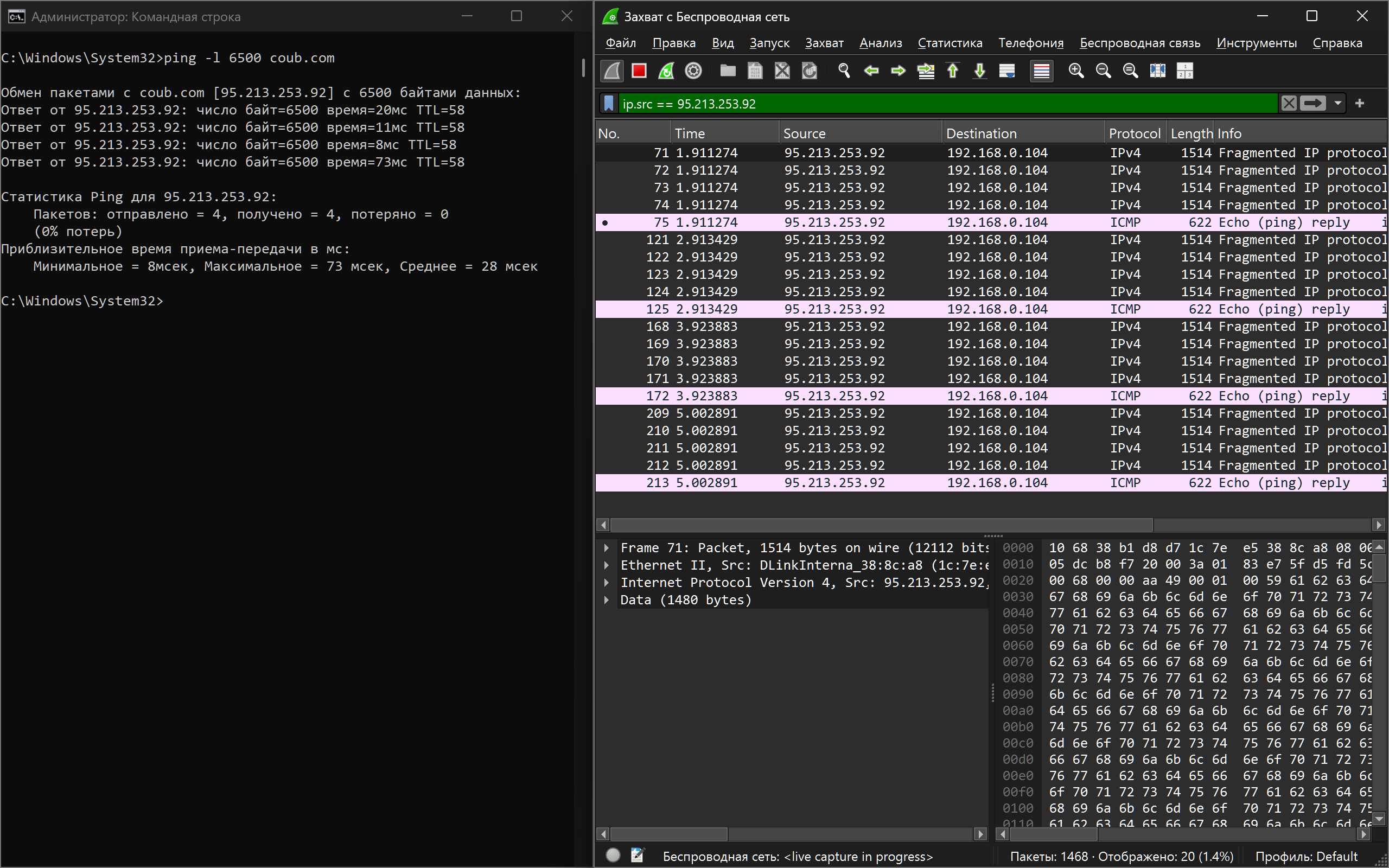


Рисунок 1.3.2 – ping -l 6500 coub.com

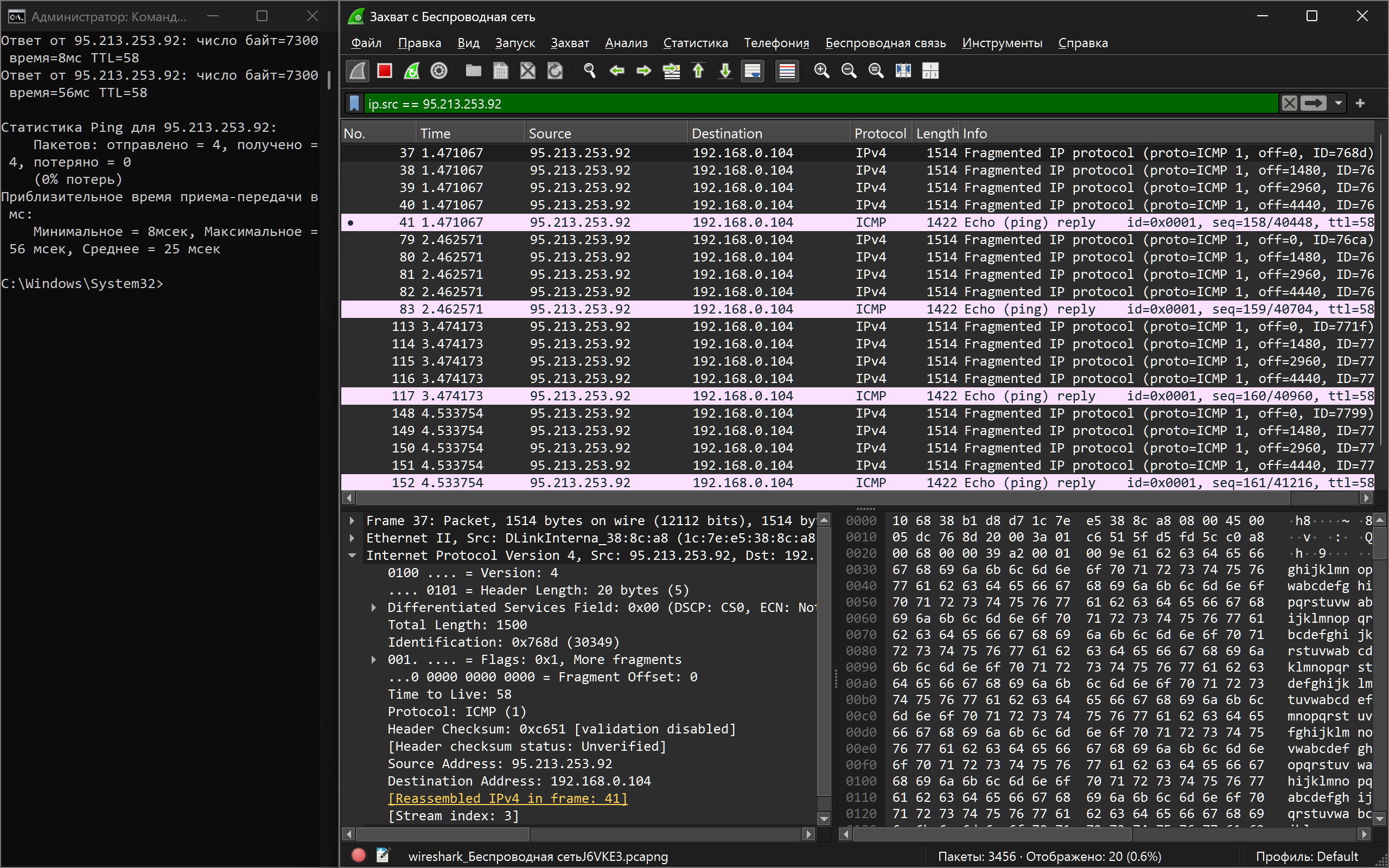


Рисунок 1.3.3 – ping -1 7300 coub.com

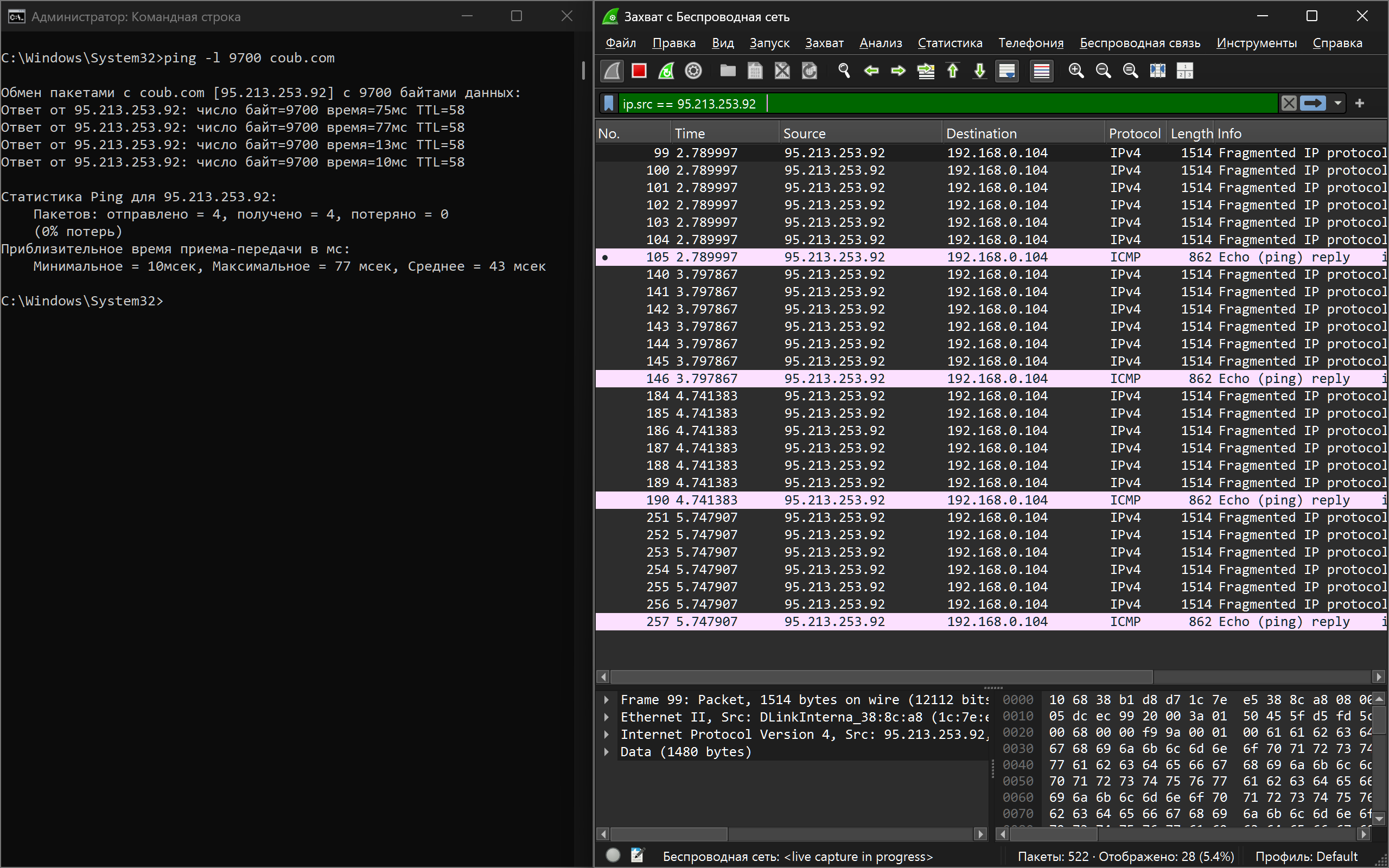


Рисунок 1.4 - ping -l 9700 coub.com

1. Имеет ли место фрагментация исходного пакета, какое поле на это указывает?

Фрагментация возникает, если размер ICMP-пакета (Echo Request/Reply) превышает максимально допустимое значение MTU для данного канала (обычно 1500 байт для Ethernet).

На фрагментацию указывает поле "Flags" (Флаги) в заголовке IP-пакета.

Если установлен флаг "More fragments", это означает, что данный пакет является промежуточным фрагментом, а не последним.

На скриншотах (Рисунок 1.1) для маленького пакета (100 байт) флаг "More fragments" всего не установлен, что говорит об отсутствии фрагментации.

На скриншотах для больших пакетов (Рисунок 1.2, 1.3, 1.4) можно увидеть несколько ICMP-пакетов с установленным флагом "More fragments" для всех, кроме последнего.

2. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?

На наличие фрагментации указывают поля заголовка IP-пакета:

• Flags (бит MF — More Fragments)

• Fragment Offset (смещение фрагмента в общем пакете).

Если пакет не фрагментирован, MF=0 и Offset=0. Если наблюдается фрагментация, то хотя бы один пакет имеет MF=1 или ненулевой Offset.

3. Чему равно количество фрагментов при передаче ping-пакетов?

Количество фрагментов зависит от размера полезной нагрузки (-l), который вы указали, и от MTU (Maximum Transmission Unit) вашей сети. MTU — это максимальный размер кадра/cегмента, который может быть передан без фрагментации (обычно 1500 байт для Ethernet).

* Для 100: Размер пакета меньше MTU. Фрагментации нет, количество фрагментов = 1.
* Для 3300-4300: (Размер IP-заголовка + ICMP-заголовок + 3300 байт данных) > MTU. Требуется фрагментация. Количество фрагментов = 3.
* Для 6500-7300: Количество фрагментов больше – 5.
* Для 9700**:** Количество фрагментов ещё больше – 7.

4. Построить график, в котором на оси абсцисс находится размер\_пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый ping-пакет.

Рисунок 1.5 – График зависимости числа фрагментов от размера пакета

5. Как изменить поле TTL с помощью утилиты ping?

Поле TTL (Time to Live) в заголовке IP-пакета можно изменить с помощью ключа -i в ОС Windows.

Синтаксис: ping -i <TTL> -l <размер> <адрес>

Пример: ping -i **50** -l 100 coub.com

6. Что содержится в поле данных ping-пакета?

Поле данных ICMP-пакета (Echo Request/Reply), генерируемого утилитой ping, по умолчанию содержит набор повторяющихся буквенно-цифровых символов. Обычно это последовательность из символов латинского алфавита (abcdefghijklmnopqrstuvwabc...).

Назначение этого заполнения — достичь запрошенного размера пакета (-l). Если установить -l 100, в данных будет 100 байт этого служебного заполнения. По шестнадцатеричному представлению пакета в Wireshark, можно увидеть данную повторяющуюся последовательность.

*Структура наблюдаемых PDU для ICMP Echo Request и Echo Reply*

Структура PDU ICMP Echo Reply:

Уровень 2 (Канальный): Ethernet-кадр с MAC-адресами источника (устройства на стороне сервера) и назначения ( компьютера).

Уровень 3 (Сетевой): IP-пакет, содержащий IP-адреса отправителя (сервер) и получателя (ваш компьютер). Поле TTL указывает оставшееся время жизни пакета.

Уровень 4 (Транспортный/Служебный): В IP-пакете размещено ICMP-сообщение типа Echo Reply. Его заголовок содержит значения: Type = 0 (Echo Reply), Code = 0, контрольную сумму. Идентификатор и порядковый номер совпадают с теми, что были в соответствующем Echo Request, что подтверждает принадлежность ответа именно этому запросу. Полезная нагрузка возвращается в неизменном виде.

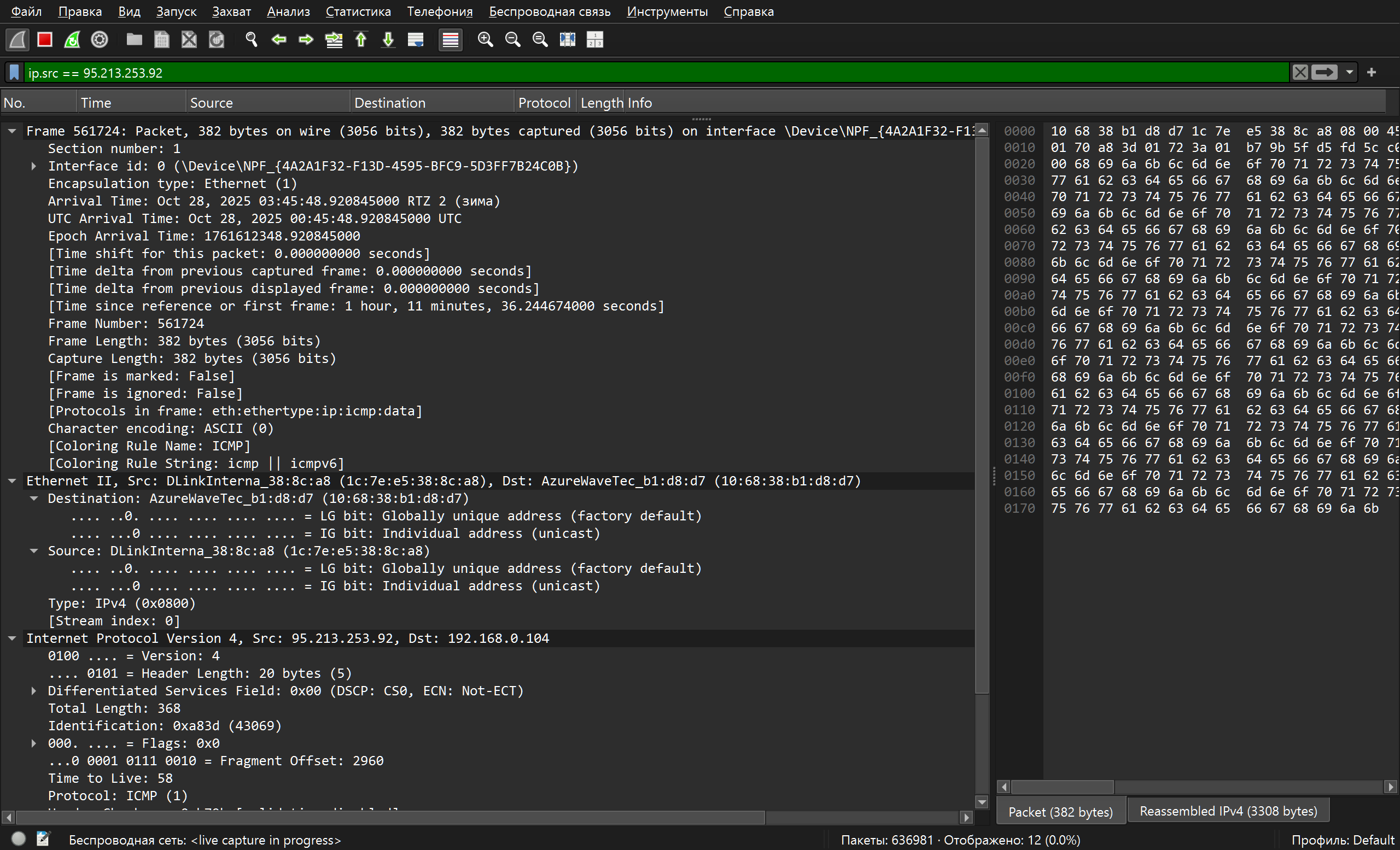


Рисунок 2.1 – ICMP для Echo Reply

Структура PDU при фрагментации IP-пакета

Уровень 2 (Канальный): Каждый фрагмент большого пакета передаётся отдельным Ethernet-кадром. Все они содержат MAC-адреса источника и назначения, а также поле EtherType = IPv4.

Уровень 3 (Сетевой): Каждый фрагмент формируется как отдельный IP-пакет, имеющий одинаковое значение поля Identification. В поле Fragment offset задаётся смещение данных текущего фрагмента относительно начала исходного пакета. Флаг MF (More Fragments) указывает, является ли фрагмент промежуточным (MF=1) или последним (MF=0).

Уровень 4 (Транспортный/Служебный): Полный заголовок ICMP (Echo Request или Reply) содержится только в первом фрагменте. Последующие фрагменты включают лишь часть данных (payload) из исходного ICMP-пакета.

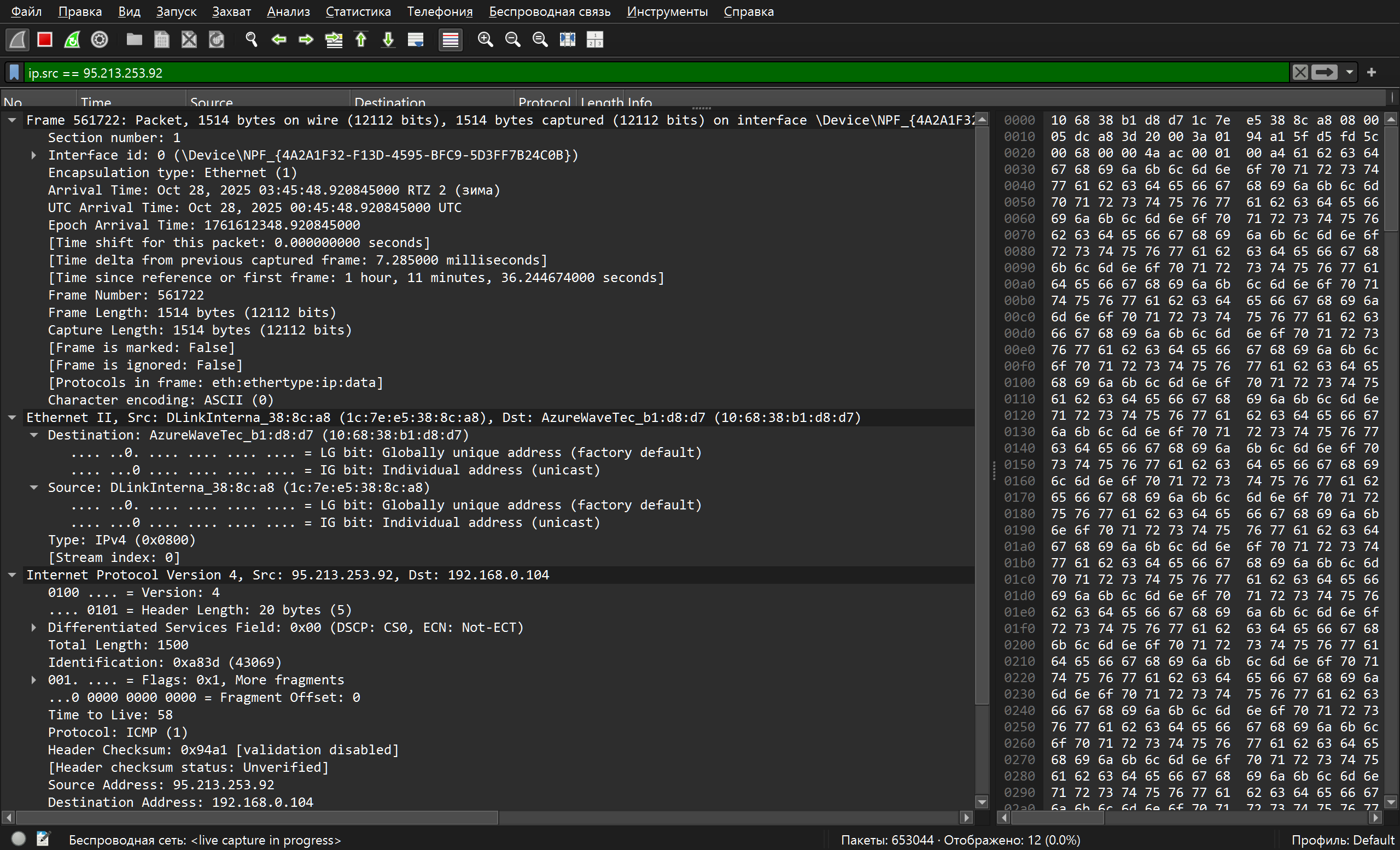


Рисунок 2.2 – Fragmented IP protocol

*2. Анализ трафика утилиты tracert (traceroute)*

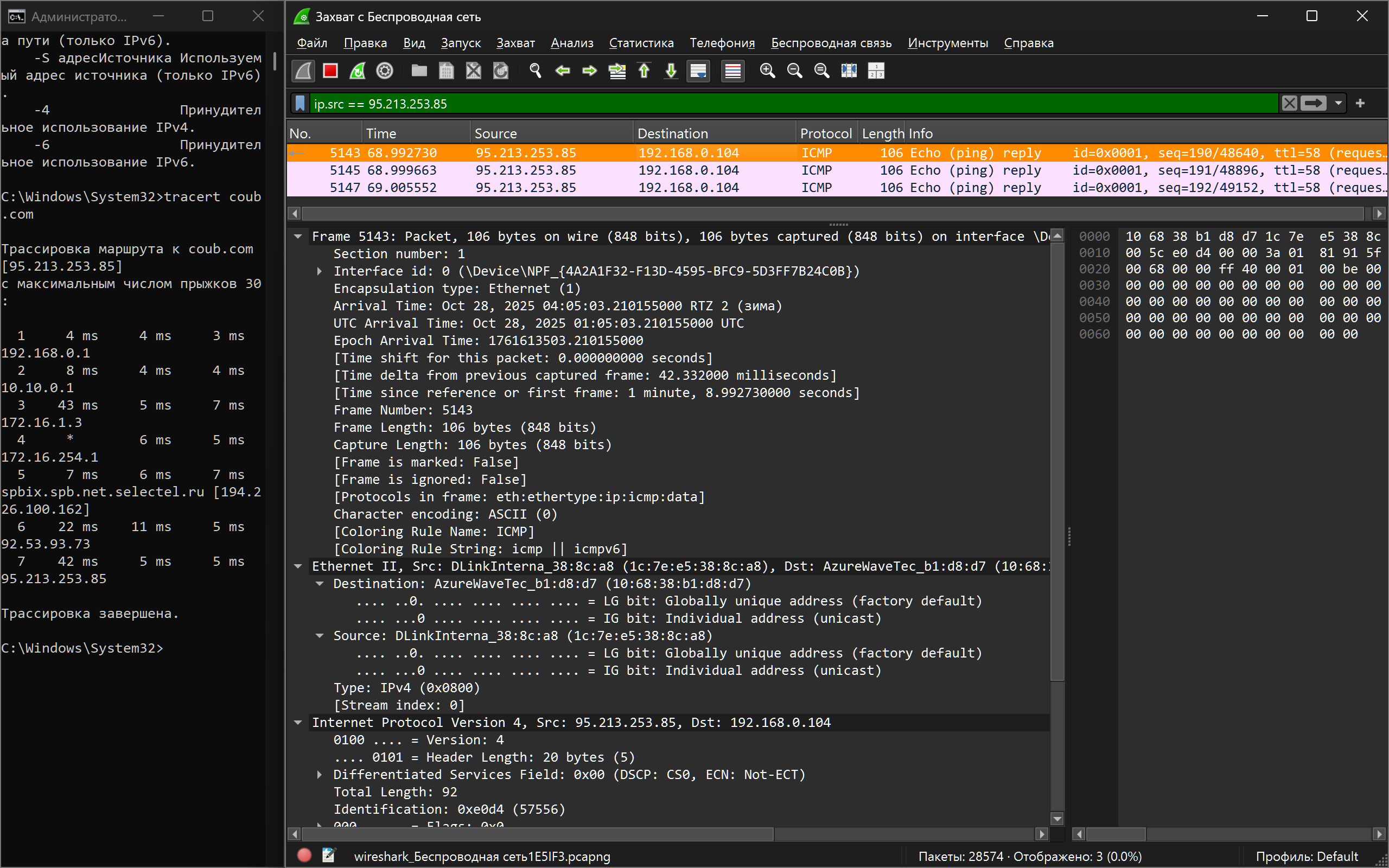
Был отслежен и проанализирован трафик утилитой Wireshark, создаваемый утилитой traceroute, при запуске из командной строки. 

Рисунок 3.1 – tracert coub.com

1. Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?

* Заголовок IP: Стандартный заголовок IPv4 без опций имеет фиксированный размер 20 байт. Это видно в строке "Header Length: 20 bytes (5)" в Wireshark.
* Поле данных (Payload): Размер поля данных зависит от типа пакета.
  + В ICMP-пакетах, генерируемых tracert (как UDP-зонды, так и ответы), поле данных включает в себя:
    1. ICMP-заголовок (8 байт для сообщений об ошибках, таких как "Time Exceeded").
    2. Вложенный IP-заголовок и первые 8 байт исходного пакета, который вызвал ошибку (ещё 28 байт: 20 байт IP + 8 байт данных).
  + Таким образом, общий размер IP-пакета (заголовок + данные) для ICMP Time Exceeded обычно составляет 20 + 36 = 56 байт, что можно увидеть в поле "Total Length" заголовка IP.

2. Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert? Для ответа на этот вопрос нужно проследить изменение TTL при передаче по маршруту, состоящему из более чем двух хопов.

Как изменяется: Утилита tracert отправляет серию пакетов, в которых значение поля TTL последовательно увеличивается на 1, начиная с 1. Первый пакет имеет TTL=1, второй — TTL=2, третий — TTL=3 и так далее.

Почему это происходит: Это является основным алгоритмом работы утилиты.

Пакет с TTL=1 достигает первого маршрутизатора. Тот уменьшает TTL на 1, получает 0, отбрасывает пакет и отправляет обратно ICMP "Time-to-live exceeded". Так мы узнаём первый хоп.

Пакет с TTL=2 проходит первый маршрутизатор (TTL уменьшается до 1), но достигает второго маршрутизатора. Тот уменьшает TTL до 0, отбрасывает пакет и отправляет ошибку. Так мы узнаём второй хоп.

Этот процесс повторяется, пока пакет с достаточно большим TTL не достигнет целевого хоста. Ответ целевого хоста (например, "Port Unreachable") сигнализирует об окончании построения маршрута.

3. Чем отличаются ICMP-пакеты, генерируемые утилитой tracert, от ICMP-пакетов, генерируемых утилитой ping (см. предыдущее задание).

ICMP-пакеты, генерируемые утилитами tracert и ping, имеют фундаментальные различия в своей сути, несмотря на то что обе используют протокол ICMP для диагностики.

Утилита ping работает по принципу эхо-запроса. Она генерирует ICMP Echo Request (Тип 8) и ожидает в ответ ICMP Echo Reply (Тип 0) от целевого узла. Пакеты ping отправляются с большим значением TTL (обычно 64 или 128), чтобы гарантировать, что они достигнут конечной цели. Главная задача ping — подтвердить доступность узла и измерить время двусторонней задержки (RTT).

Утилита tracert (в ее реализации для Windows) работает по иному принципу. Она сама не генерирует ICMP Echo Request. Вместо этого она отправляет серию UDP-пакетов на заведомо неиспользуемый порт целевого узла. Ключевое отличие заключается в манипуляции полем TTL. tracert намеренно отправляет первые пакеты с TTL=1, затем с TTL=2 и так далее, чтобы каждый из них был отброшен на определенном промежуточном маршрутизаторе.

В ответ на эти отброшенные UDP-пакеты tracert получает не эхо-ответы, а сообщения об ошибках — ICMP Time-to-live exceeded (Тип 11) от каждого маршрутизатора на пути. Когда же UDP-пакет наконец достигает цели, утилита получает финальное сообщение об ошибке — ICMP Destination Unreachable (Port Unreachable) (Тип 3, Код 3). Таким образом, tracert использует ICMP-пакеты с ошибками как инструмент для построения карты маршрута, в то время как ping использует штатные ICMP-запросы и ответы для проверки связи.

4. Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?

ICMP Reply (Ответ):

Пример: ICMP Echo Reply (Тип 0).

Назначение: Это штатный, ожидаемый ответ на успешно доставленный запрос. Он означает: "Я получил твой пакет и корректно его обработал".

В tracert: Такой пакет приходит только от целевого хоста в конце маршрута (в виде ICMP Destination Unreachable, что для tracert является аналогом "порт недоступен, но хост достигнут").

ICMP Error (Ошибка):

Примеры: ICMP Time Exceeded (Тип 11), ICMP Destination Unreachable (Тип 3).

Назначение: Это сообщение об аномальной ситуации. Оно означает: "Я не смог доставить твой пакет по назначению, и вот причина".

В tracert: Эти пакеты приходят от промежуточных маршрутизаторов (Time Exceeded) и являются основным источником информации для построения маршрута.

Оба типа сообщений необходимы для диагностики сети. Reply подтверждает конечную доступность, а Error указывает на проблему на пути (например, обрыв на определённом участке, что без tracert и ICMP Errors было бы сложно локализовать).

5. Что изменится в работе tracert, если убрать ключ “-d”? Какой дополнительный трафик при этом будет генерироваться?

Ключ -d в Windows tracert отключает reverse DNS-запросы, то есть преобразование IP-адресов маршрутизаторов в их доменные имена.

Что изменится без -d: В выводе tracert вместо голых IP-адресов (например, 192.168.1.1) появятся попытки отобразить доменные имена маршрутизаторов (например, router1.isp.my-city.net).

Какой трафик сгенерируется: Для этого tracert будет инициировать обратные DNS-запросы (PTR-запросы) для каждого уникального IP-адреса, полученного в ходе работы. Это создаст дополнительный DNS-трафик (UDP-пакеты на порт 53), что может заметно увеличить время выполнения всей операции tracert, особенно если DNS-серверы отвечают медленно.

*3. Анализ HTTP-трафика*

Изначально для анализа был выбран сайт coub.com. Однако при анализе трафика в Wireshark пакеты с протоколом HTTP обнаружены не были. Вместо этого весь обмен данными между браузером и сервером происходил по протоколу TLS (Transport Layer Security). Это указывает на то, что сайт coub.com использует защищённое соединение HTTPS, при котором HTTP-сообщения шифруются и инкапсулируются в TLS-сессии, что делает их недоступными для прямого анализа содержимого средствами Wireshark без использования закрытых ключей сервера.

Для выполнения задачи анализа структуры HTTP-запросов и ответов был использован специальный сайт, поддерживающий незашифрованное HTTP-соединение: <http://httpforever.com/>.

Был отслежен и проанализирован HTTP-трафик, создаваемый браузером при посещении данного сайта. В списке захваченных пакетов были проанализированы пары HTTP-сообщений (запрос-ответ). Для анализа использовался фильтр http.

В списке захваченных пакетов были проанализированы следующие пары HTTP-сообщений (запрос-ответ):

● GET-сообщение от клиента (браузера);

● ответ сервера.

При выполнении данного пункта возникли проблемы с условными GET-запросами. Они не приходили от браузера к веб-сайту. Это могло произойти по причине, того что сайт не поддерживает условные GET-запросы.



Рисунок 4.1 – get запрос

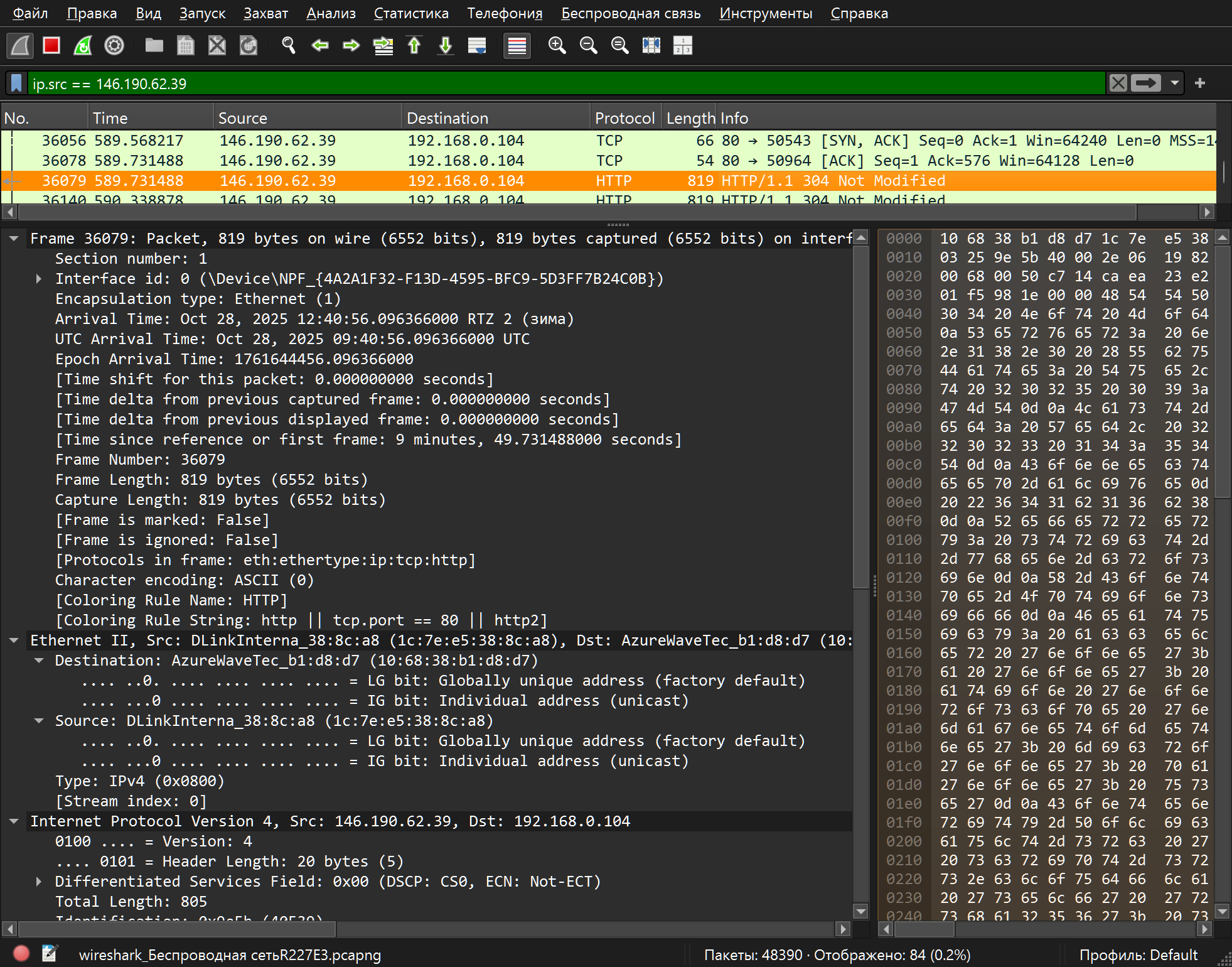


Рисунок 4.2 – Условный Get-запрос (304 not modified)

Структура PDU HTTP GET-запроса (первичное обращение)

* Уровень 2 (Канальный): Ethernet II кадр, содержащий MAC-адреса отправителя (клиента) и получателя (шлюза).
* Уровень 3 (Сетевой): Internet Protocol Version 4 пакет с IP-адресами источника (клиент) и назначения (сервер httpforever.com).
* Уровень 4 (Транспортный): Transmission Control Protocol сегмент. Устанавливается соединение между клиентом и сервером (порт клиента – случайный высокий, порт сервера – 80). Заголовок TCP содержит номера последовательностей и подтверждений, флаги (PSH, ACK), обеспечивающие надежную доставку.
* Уровень 7 (Прикладной): Hypertext Transfer Protocol – непосредственно HTTP-запрос.
  + Request Method: GET
  + Request URI: /
  + HTTP-Version: HTTP/1.1
  + Заголовки (Headers): Содержат метаинформацию: Host (адрес сервера), User-Agent (информация о браузере и ОС), Accept (типы контента, которые понимает клиент), Accept-Encoding (поддерживаемые методы сжатия).

Таким образом:

Первичный get-запрос всегда приводит к получению полного содержимого (ответ 200 + данные страницы).

Вторичный запрос может привести либо к ответу 304 (без тела), либо к новому 200 (если ресурс изменился).

*4. Анализ DNS-трафика*

Был отслежен и проанализирован трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

Настроен Wireshark-фильтр: dns

Очищен кэш DNS с помощью команды в командной строке: ipconfig /flushdns

Очищен кэш браузера

Выполнен переход на сайт coub.com

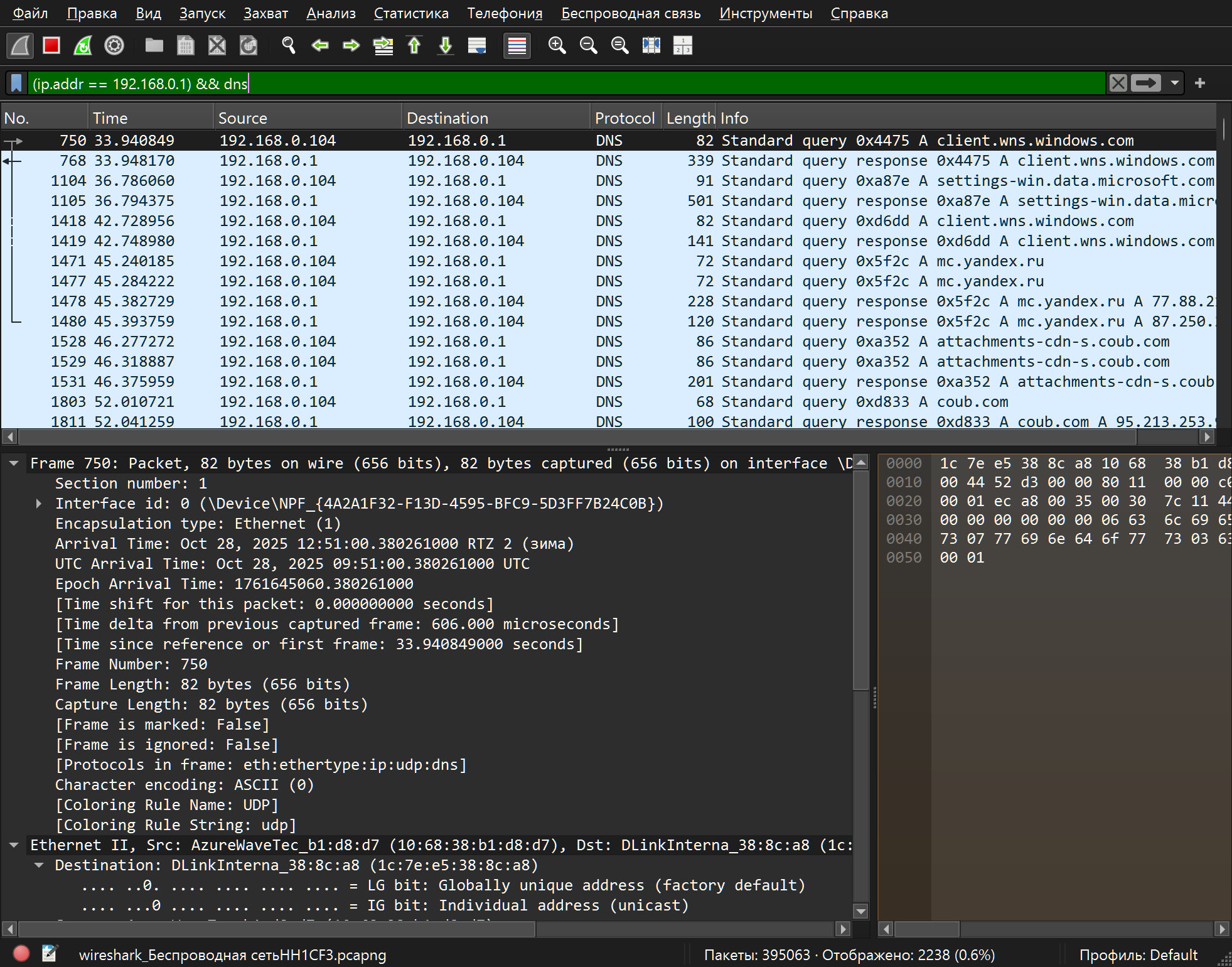


Рисунок 5.1 – Трафик протокола DNS

**Структура наблюдаемых PDU DNS**

**Структура PDU DNS Standard Query (Запрос)**

**Уровень 2 (Канальный):** Ethernet II кадр, содержащий MAC-адреса отправителя (клиентского устройства) и получателя (шлюза или DNS-сервера).

**Уровень 3 (Сетевой):** Internet Protocol Version 4 пакет с IP-адресом источника (клиента) и IP-адресом назначения (DNS-сервера, например, 8.8.8.8 или сервера провайдера).

**Уровень 4 (Транспортный):** User Datagram Protocol сегмент.

**Source Port:** Случайный высокий порт, выбранный клиентом.

**Destination Port:** 53 (стандартный порт DNS).

**Уровень 7 (Прикладной):** Domain Name System (query) – непосредственно DNS-запрос.

**Transaction ID:** Уникальный идентификатор для сопоставления запроса и ответа.

**Flags:** Флаги запроса, например, Recursion desired (показывает, что клиент запрашивает рекурсивный поиск).

**Questions:** Количество запросов (обычно 1).

**Queries:** Содержит доменное имя (coub.com) и тип записи (например, A – IPv4-адрес, AAAA – IPv6-адрес, PTR - обратный).

**1. Почему адрес, на который отправлен DNS-запрос, не совпадает с адресом посещаемого сайта?**

DNS-запрос направляется на DNS-сервер (например, на IP-адрес провайдера или публичный DNS вроде 8.8.8.8), а не на IP-адрес самого сайта coub.com. Это происходит потому, что приложению (браузеру) для установки соединения с веб-сервером необходим его IP-адрес, а не доменное имя. Функция DNS как раз и заключается в преобразовании понятных человеку доменных имен (coub.com) в машиночитаемые IP-адреса. Только после успешного разрешения имени браузер может отправить HTTP/HTTPS запрос по полученному IP-адресу.

**2. Какие бывают типы DNS-запросов?**

* **По механизму разрешения:**
  + **Рекурсивные (Recursive):** Клиент требует от DNS-сервера полный ответ. Сервер самостоятельно обходит цепочку других серверов, если не знает ответа, и возвращает клиенту конечный результат или ошибку.
  + **Итеративные (Iterative):** Сервер, не имея полного ответа, возвращает клиенту адрес следующего, более авторитетного DNS-сервера, к которому нужно обратиться. Клиент повторяет запрос уже к этому серверу.
* **По типу запрашиваемой записи (в поле**Type**запроса):**
  + **A (Address):** Запрос IPv4-адреса.
  + **AAAA (Quad-A):** Запрос IPv6-адреса.
  + **NS (Name Server):** Запрос авторитетных серверов имен для домена.
  + **CNAME (Canonical Name):** Запрос канонического имени (алиаса).
  + **MX (Mail Exchange):** Запрос серверов электронной почты для домена.
  + **PTR (Pointer):** Обратный запрос — определение доменного имени по IP-адресу.

**3. В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?**

Независимые DNS-запросы для ресурсов (изображений, стилей, скриптов) потребуются в том случае, если эти ресурсы размещены на других доменных именах (поддоменах или сторонних сервисах), отличных от основного домена сайта.

Пример для coub.com:

* Основная страница: coub.com => требуется DNS-запрос типа A для получения IP.
* Изображение размещено на: cdn.coub.com или images.thirdparty-cdn.com => для загрузки каждого такого изображения браузеру потребуется выполнить новый, независимый DNS-запрос к имени cdn.coub.com или images.thirdparty-cdn.com, чтобы получить их IP-адреса.

Этот механизм позволяет распределять нагрузку и использовать специализированные сервисы для доставки контента (CDN).

*5. Анализ ARP-трафика (Address Resolution Protocol)*

Был отслежен и проанализирован трафик протокола ARP, сгенерированный в результате выполнения следующих действий:

* Очищена ARP-таблица с помощью команды: netsh interface ip delete arpcache
* Проверка очистки выполнена командой: arp -a
* Очищен кэш браузера
* Выполнен переход на сайт coub.com

**Структура наблюдаемых PDU протокола ARP**

**Структура PDU ARP-запроса (ARP Request)**

* **Уровень 2 (Канальный):** Address Resolution Protocol (request) - протокол работает непосредственно на канальном уровне.
  + **Target MAC address:** 00:00:00:00:00:00 (широковещательный адрес) - запрос адресован всем устройствам в локальной сети.
  + **Source MAC address:** MAC-адрес сетевой карты компьютера.
  + **Frame Type:** ARP (0x0806) - указывает, что кадр содержит ARP-пакет.
* **Сетевой уровень (логическая информация):** Хотя ARP работает на канальном уровне, он оперирует IP-адресами:
  + **Sender MAC address:** MAC-адрес компьютера.
  + **Sender IP address:** IP-адрес компьютера.
  + **Target MAC address:** 00:00:00:00:00:00 (неизвестен, поэтому нули).
  + **Target IP address:** IP-адрес шлюза (маршрутизатора), для которого требуется найти MAC-адрес.
  + **Opcode:** 1 (request) - указывает, что это ARP-запрос.

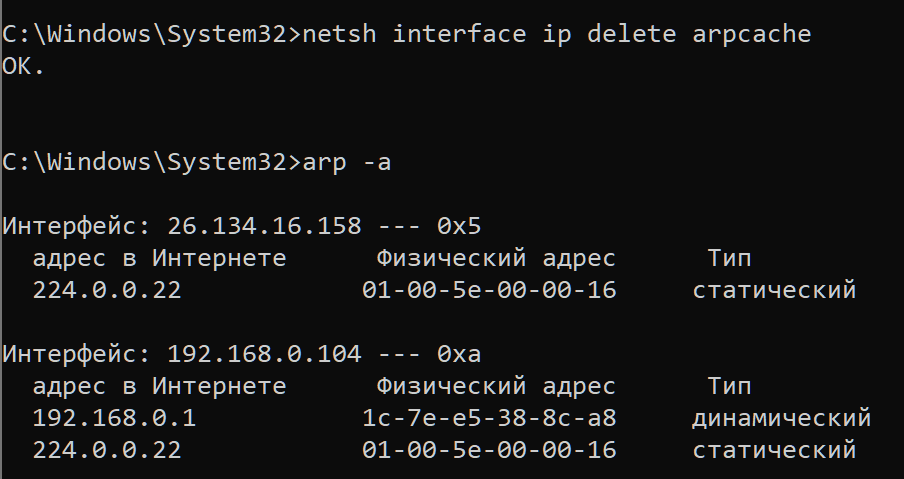


Рисунок 6.1 – Очищенная arp таблица

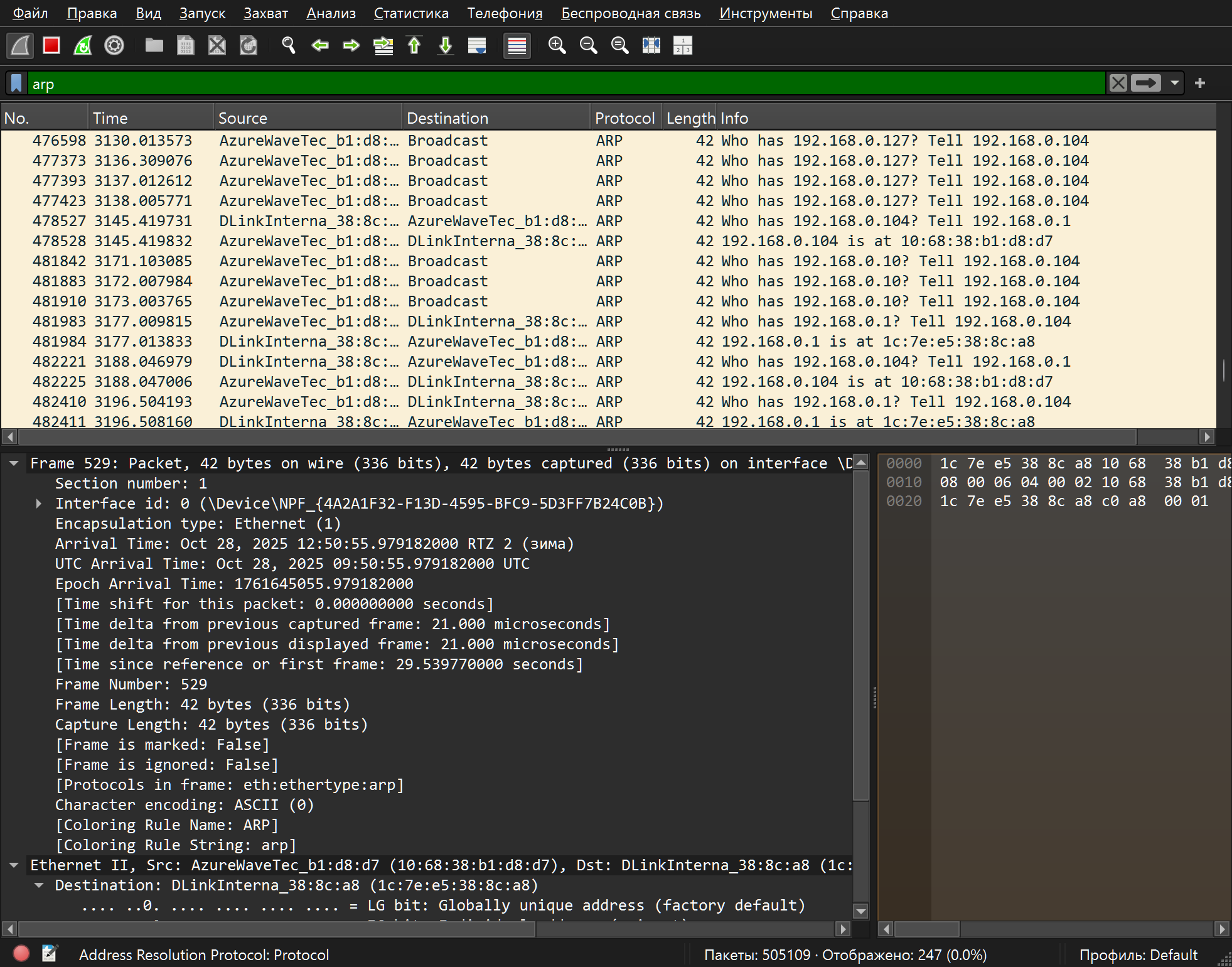


Рисунок 6.2 - Захваченные пакеты ARP

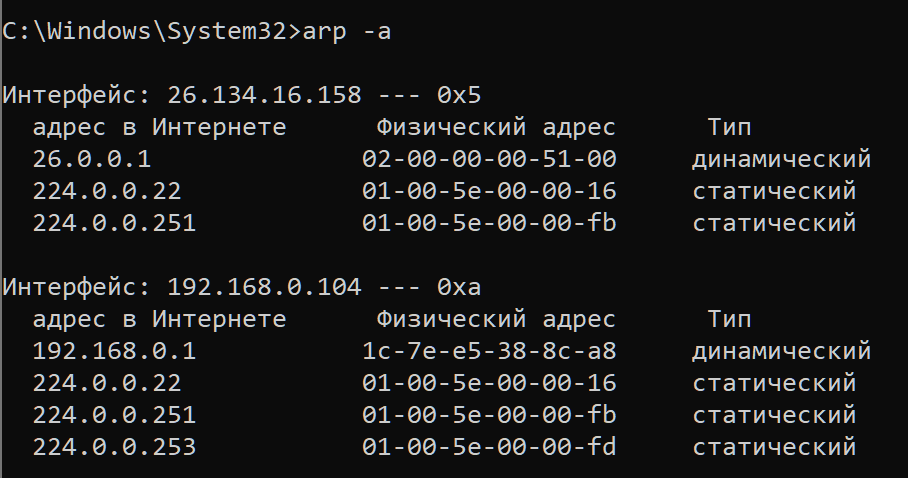


Рисунок 6.3 – новая ARP таблица

**Структура PDU ARP-ответа (ARP Reply)**

* **Уровень 2 (Канальный):** Address Resolution Protocol (reply)
  + **Target MAC address:** MAC-адрес компьютера (уже известен отправителю).
  + **Source MAC address:** MAC-адрес маршрутизатора (шлюза).
  + **Frame Type:** ARP (0x0806)
* **Сетевой уровень (логическая информация):**
  + **Sender MAC address:** MAC-адрес маршрутизатора (шлюза) – это искомый адрес.
  + **Sender IP address:** IP-адрес маршрутизатора.
  + **Target MAC address:** MAC-адрес компьютера.
  + **Target IP address:** IP-адрес компьютера.
  + **Opcode:** 2 (reply) - указывает, что это ARP-ответ.

**1. Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют?**

* **В ARP-запросе (Request):**
  + **Source MAC address:** Физический адрес вашего компьютера. Идентифицирует сетевое устройство, которое ищет MAC-адрес.
  + **Destination MAC address:** ff:ff:ff:ff:ff:ff (широковещательный адрес). Означает, что пакет адресован всем устройствам в локальной сети.
* **В ARP-ответе (Reply):**
  + **Source MAC address:** Физический адрес маршрутизатора (шлюза по умолчанию). Идентифицирует устройство, которое обладает запрашиваемым IP-адресом.
  + **Destination MAC address:** MAC-адрес вашего компьютера. Означает, что ответ адресован конкретно вашему устройству.

**2. Какие MAC-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Какие устройства они идентифицируют?**

* **Source MAC address:** MAC-адрес вашего компьютера (в исходящих пакетах) или MAC-адрес маршрутизатора (во входящих пакетах).
* **Destination MAC address:** MAC-адрес маршрутизатора (в исходящих пакетах) или MAC-адрес вашего компьютера (во входящих пакетах).

**Важное отличие от ARP:** В HTTP-пакетах (работающих через IP) MAC-адреса **не идентифицируют конечные устройства** (ваш компьютер и сервер coub.com), а идентифицируют только **следующий узел на пути следования пакета** в пределах локальной сети. Поскольку сервер coub.com находится в интернете, все пакеты к нему сначала отправляются на шлюз (маршрутизатор), который уже дальше маршрутизирует их по интернету.

**3. Для чего ARP-запрос содержит IP-адрес источника?**

IP-адрес источника в ARP-запросе необходим для двух целей:

1. **Чтобы получатель ответа знал, кому его отправить.** Устройство, которое узнает свой MAC-адрес в запросе, использует IP-адрес источника как адрес назначения для ARP-ответа.
2. **Для обновления ARP-таблиц других узлов в сети.** Все устройства в локальной сети, получившие широковещательный ARP-запрос, могут обновить свои ARP-таблицы, сопоставив IP-адрес источника с его MAC-адресом, даже если они не являются целевыми для этого запроса. Это повышает эффективность работы сети.

*6. Анализ трафика утилиты nslookup*

Был отслежен и проанализирован трафик протокола DNS, сгенерированный утилитой nslookup в результате выполнения следующих действий:

1. Настроен Wireshark-фильтр: dns
2. Запущена в командной строке команда: nslookup coub.com
3. Запущена в командной строке команда: nslookup -type=NS coub.com

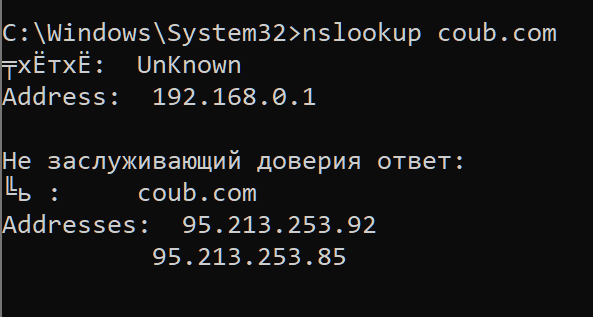


Рисунок 7.1 - nslookup coub.com

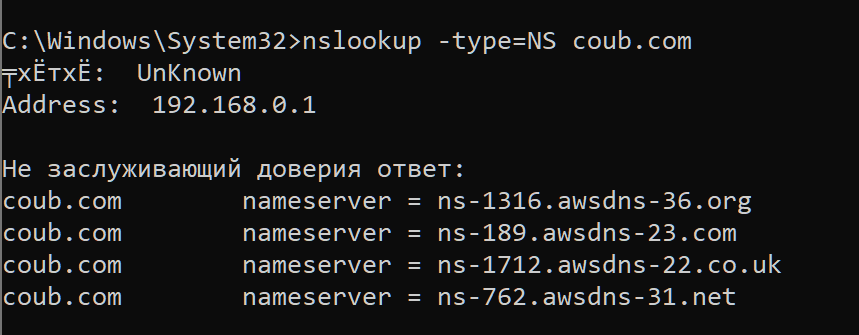


Рисунок 7.2 - nslookup -type=NS coub.com

Структура наблюдаемых PDU DNS

Структура PDU DNS-запроса (Standard Query)

* Уровень 2 (Канальный): Ethernet II кадр содержит MAC-адрес источника (клиент) и MAC-адрес назначения (шлюз/маршрутизатор или локальный DNS-сервер). Канальный уровень обеспечивает доставку UDP-пакета до ближайшего узла в локальной сети.
* Уровень 3 (Сетевой): Internet Protocol Version 4 пакет с адресом источника (клиент) и адресом назначения (DNS-сервер). Заголовок IP содержит TTL, идентификатор пакета, контрольную сумму.
* Уровень 4 (Транспортный): User Datagram Protocol сегмент с исходным портом клиента и целевым портом 53 (DNS). Этот уровень обеспечивает передачу данных между приложением клиента и DNS-сервером без установки соединения.
* Уровень 7 (Прикладной): Domain Name System (query) содержит:
  + Transaction ID - уникальный идентификатор для сопоставления запроса и ответа
  + Flags - флаги запроса (запрос/рекурсия)
  + Questions - количество запросов
  + Queries - имя запрашиваемого домена (coub.com) и тип записи (например, A - IP-адрес, NS - имя сервера)

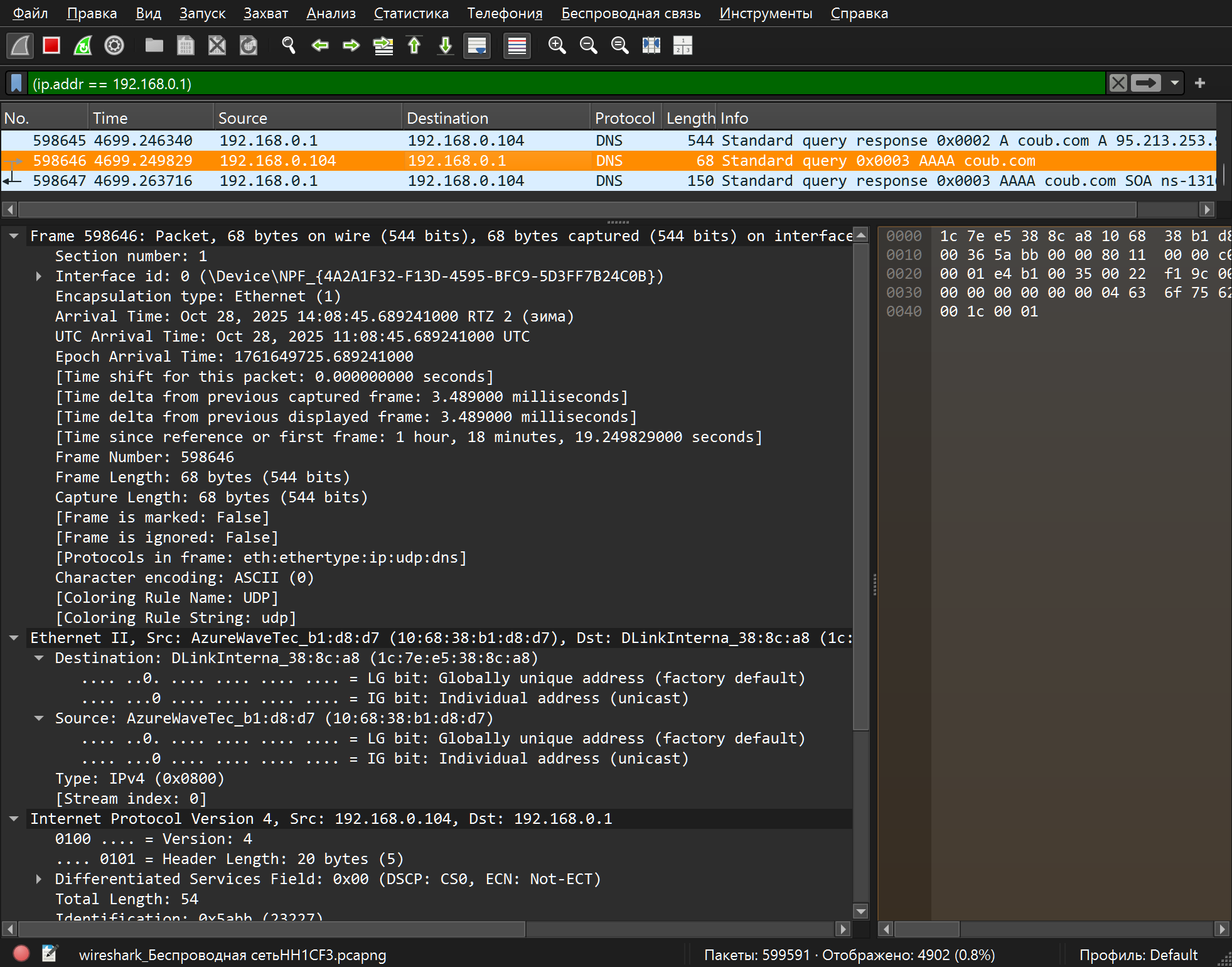


Рисунок 7.3 – DNS-запрос

Структура PDU DNS-ответа (Standard Query Response)

* Уровень 2 (Канальный): Ethernet II кадр содержит MAC-адрес источника (DNS-сервер/шлюз) и MAC-адрес назначения (клиент).
* Уровень 3 (Сетевой): Internet Protocol Version 4 пакет с адресом источника (DNS-сервер) и адресом назначения (клиент). Заголовок IP аналогичен запросу (TTL, идентификатор, контрольная сумма).
* Уровень 4 (Транспортный): User Datagram Protocol сегмент с исходным портом 53 (DNS-сервер) и портом клиента.
* Уровень 7 (Прикладной): Domain Name System (response) содержит:
  + Transaction ID - совпадает с идентификатором из запроса
  + Flags - флаги ответа
  + Questions - повторяет исходный запрос
  + Answers - содержит найденные записи (IP-адреса для A-записей, NS для серверов имен и др.)
  + Authority - секция с авторитетными серверами
  + Additional - дополнительные записи

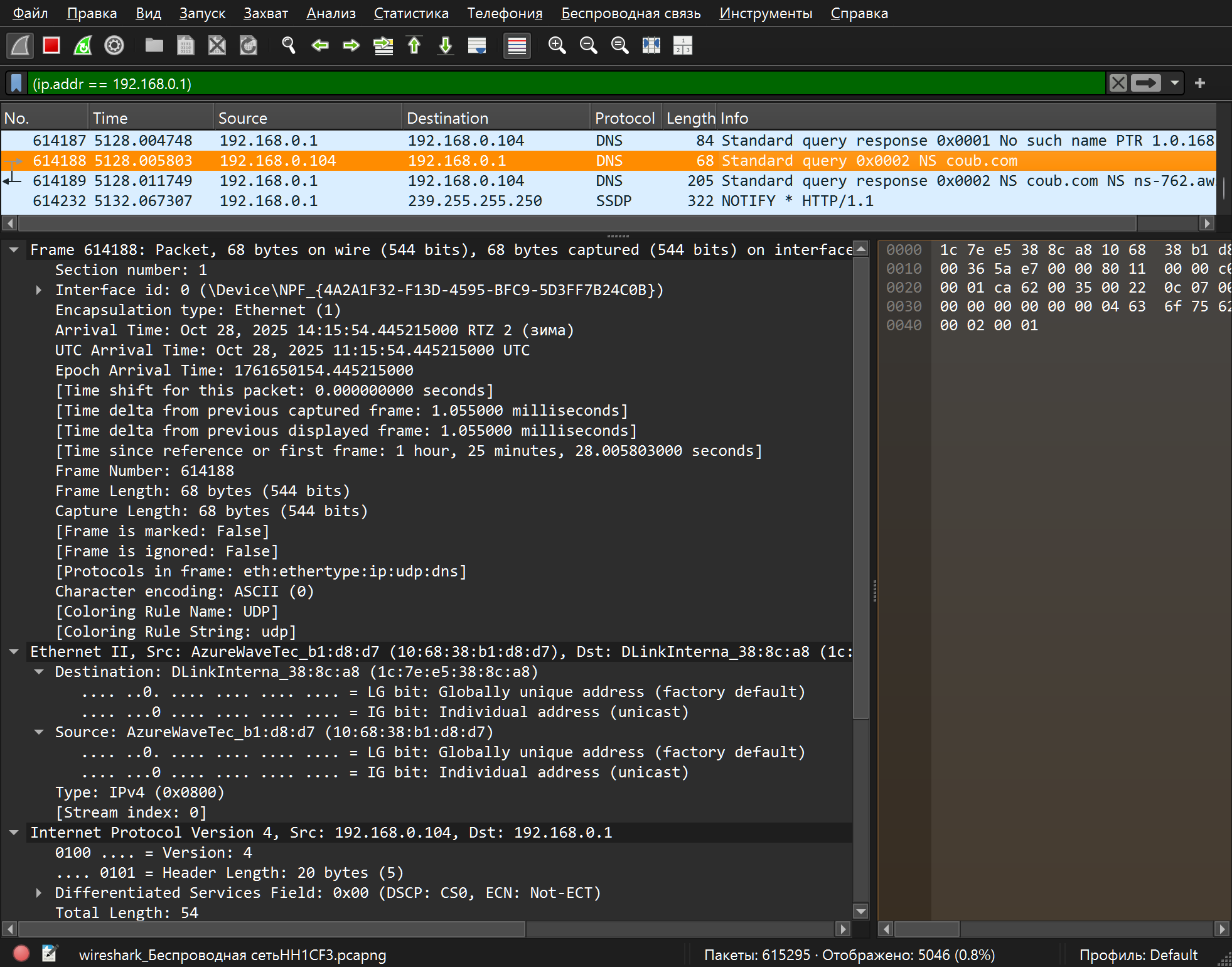


Рисунок 7.4 – DNS-запрос для NS типа

**Сравнение трафика для разных типов запросов**

**Для команды**nslookup coub.com**(тип A):**

* В секции **Answers** содержатся A-записи с IPv4-адресами серверов coub.com
* Ответ включает непосредственно IP-адреса для подключения к веб-серверу

**Для команды**nslookup -type=NS coub.com**(тип NS):**

* В секции **Answers** содержатся NS-записи с именами авторитетных серверов домена
* В секции **Additional** обычно содержатся A-записи с IP-адресами этих серверов имен

**Ответы на контрольные вопросы**

**1. Чем различается трасса трафика в п.2 и п.4, указанных выше?**

Основное различие заключается в **типе DNS-запроса и содержании ответа**:

* nslookup coub.com генерирует запросы типа **A**, направленные на получение IPv4-адресов для подключения к веб-серверу. В ответе содержатся непосредственно IP-адреса.
* nslookup -type=NS coub.com генерирует запросы типа **NS**, направленные на получение информации об авторитетных серверах имен для домена. В ответе содержатся доменные имена этих серверов.

**2. Что содержится в поле «Answers» DNS-ответа?**

Поле **Answers** содержит одну или несколько **Resource Records** (ресурсных записей), непосредственно отвечающих на запрос. Каждая запись включает:

* **Name** - доменное имя, к которому относится запись
* **Type** - тип записи (A, NS, AAAA, CNAME и др.)
* **Class** - обычно IN (Internet)
* **Time to Live** - время жизни записи в кэше
* **Data** - непосредственно данные (IP-адрес для A-записи, имя сервера для NS-записи)

**3. Каковы имена серверов, возвращающих авторитативный (authoritative) отклик?**

Авторитетными являются серверы, указанные в секции **Authority** DNS-ответа. Для домена coub.com это могут быть серверы вида:

* ns1.somednsserver.com
* ns2.somednsserver.com
* ns3.somednsserver.com

Флаг Authoritative в заголовке DNS-ответа указывает, пришел ли ответ непосредственно с авторитетного сервера.

7. FTP

# **3. Вывод:**

В ходе выполнения лабораторной работы был проведён комплексный анализ сетевого трафика с использованием программного обеспечения Wireshark. Практическим путём исследованы принципы функционирования ключевых сетевых протоколов стека TCP/IP и их соответствие эталонной модели OSI.

Экспериментально подтверждён механизм инкапсуляции данных, при котором блоки PDU каждого вышележащего уровня последовательно инкапсулируются в служебные заголовки нижележащих уровней. На примере протокола ICMP продемонстрирован процесс фрагментации IP-пакетов при превышении размера MTU, а также идентифицированы ключевые поля заголовка IP (Flags, Fragment Offset), ответственные за управление этим процессом.

Анализ работы утилиты traceroute выявил алгоритм построения маршрута следования пакетов, основанный на целенаправленной манипуляции полем TTL для получения ICMP-сообщений типа Time Exceeded от промежуточных маршрутизаторов. Установлено, что в реализации Windows данная утилита использует UDP-пакеты, а не ICMP Echo Request.

Исследование протокола HTTP позволило выявить механизмы оптимизации передачи данных, в частности, работу с кэшированием через условные GET-запросы (If-Modified-Since) и соответствующие ответы 304 Not Modified. Отмечена повсеместная распространённость HTTPS, обеспечивающего шифрование трафика на транспортном уровне.

Изучение протокола DNS продемонстрировало распределённую архитектуру системы доменных имён, включая различия между рекурсивными и итеративными запросами, а также между запросами типов A (разрешение имён) и NS (поиск авторитетных серверов). Эксперимент с утилитой nslookup наглядно показал многоэтапный процесс разрешения доменных имён.

Анализ протокола ARP подтвердил его роль в установлении соответствия между логическими (IP) и физическими (MAC) адресами в пределах локальной сети. Зафиксирован широковещательный характер ARP-запросов и точечный характер ARP-ответов.

Таким образом, работа позволила экспериментально верифицировать теоретические положения о многоуровневой организации сетевого взаимодействия, изучить структуру и назначение служебных полей заголовков основных сетевых протоколов, а также приобрести практические навыки использования специализированного программного обеспечения для диагностики и анализа сетевого трафика. Полученные результаты подтверждают принцип модульности сетевых протоколов и их слабую связанность в рамках эталонных моделей.