МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

‘Компьютерные сети’

*Выполнил:*

Студент группы P33312

Соболев Иван Александрович

*Преподаватель:*

Алиев Тауфик Измайлович



Санкт-Петербург, 2024

**Цель работы**

Изучить структуру протокольных блоков данных, анализируя реальный трафик на

компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

**Задание**

1. Запустить Wireshark. В появившемся окне выбрать интерфейс, для которого

необходимо осуществлять анализ проходящих через него пакетов. В качестве

интерфейса, используемого для захвата трафика, выбрать физический адаптер,

через который компьютер подключён к Интернету (обычно этот адаптер

называется Local или “Подключение по локальной сети”). Если меню для выбора

адаптера не появляется при запуске Wireshark, нужно запустить из “Меню” команду

“Capture->Options”. После выбора адаптеру, нужно запустить процесс захвата

трафика (кнопка Start).

2. Инициировать процесс передачи трафика по сети (например, в браузере открыть

сайт, заданный по варианту, или запустить соответствующую сетевую утилиту).

3. Установить значение “Фильтра”, чтобы из всего множества перехватываемых

пакетов Wireshark отобразил только те, которые имеют отношение к

выполняемому заданию. Для корректного создания фильтра следует пользоваться

всплывающими подсказками Wireshark, которые активизируются при наборе

фильтра. В качестве альтернативного способа можно использовать интерактивный

конструктор фильтра, нажав на кнопку “Expression” в правой части элемента

“Фильтр”

4. Дождаться появления данных в списке захваченных пакетов и убедиться, что

количество пакетов достаточно для выполнения задания.

5. Сохранить захваченный трафик в файл-трассу (pcap).

6. Описать в отчёте структуру наблюдаемых PDU (т.е. протокольных блоков данных:

кадров, пакетов, сегментов) как для запросов, так и ответов. Указать название и

назначение всех заголовков всех уровней OSI-модели в пакетах с учётом порядка

инкапсуляции (для этого нужно раскрывать соответствующие значки «+» в поле с

детальной информацией о выбранном пакете).

7. Написать в отчёте ответы на вопросы задания (для этого может потребоваться

самостоятельно изучить назначение соответствующей заданию сетевой утилиты,

использованной для создания трафика).

8. Поместить в отчёт скриншоты окна Wireshark, иллюстрирующие ответы из

вышеуказанных п.6 и п.7.

URL - Адрес сайта, в названии которого лексически входит фамилия студента:

<https://math-sobolev.ru/>

**Выполнение**

Этап 1. Анализ трафика утилиты ping

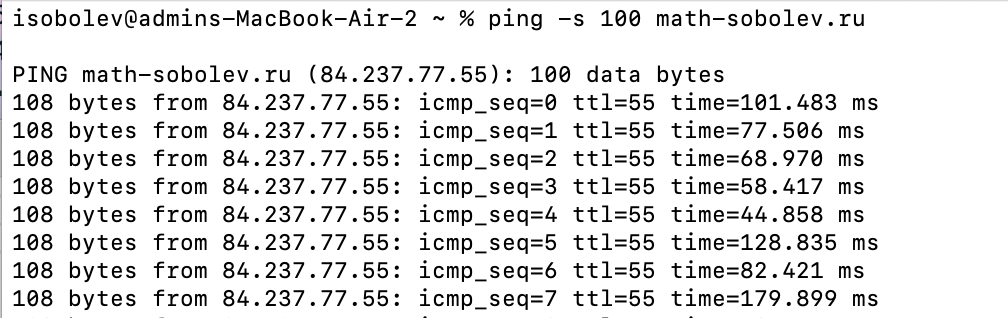
Необходимо отследить и проанализировать трафик, создаваемый утилитой ping, запустив её следующим образом из командной строки:

**“ping -l размер\_пакета адрес\_сайта\_по\_варианту”**.

В качестве “размера\_пакета” необходимо поочерёдно использовать различные значения

от 100 до 10000, самостоятельно выбрав шаг изменения.

**ping -s 100 math-sobolev.ru**

****

**Вопросы**

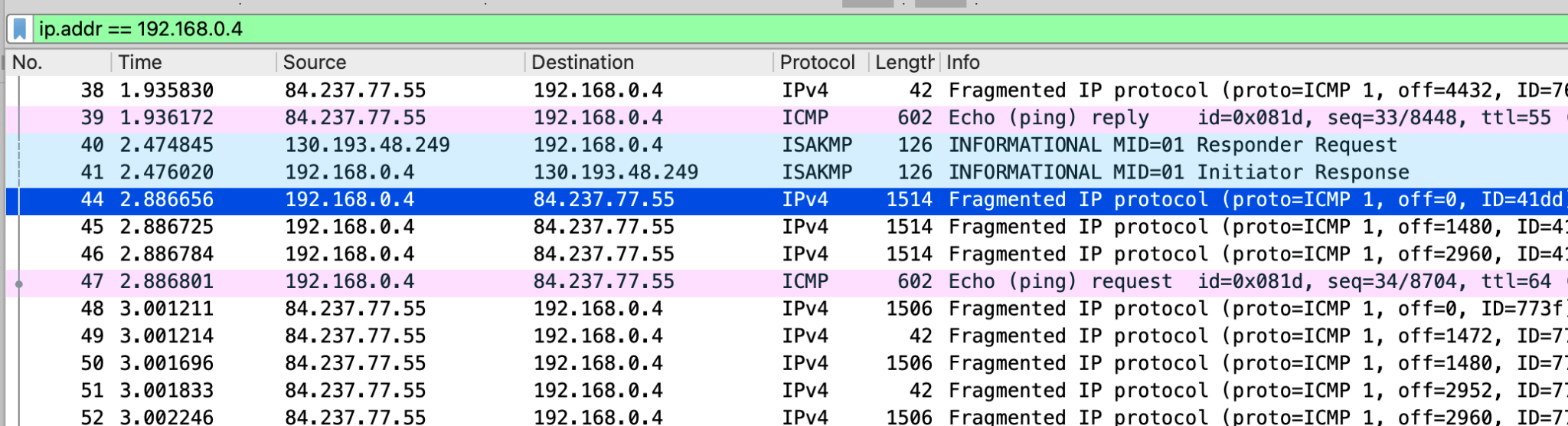
1. **Имеет ли место фрагментация исходного пакета, какое поле на это**

**указывает?**

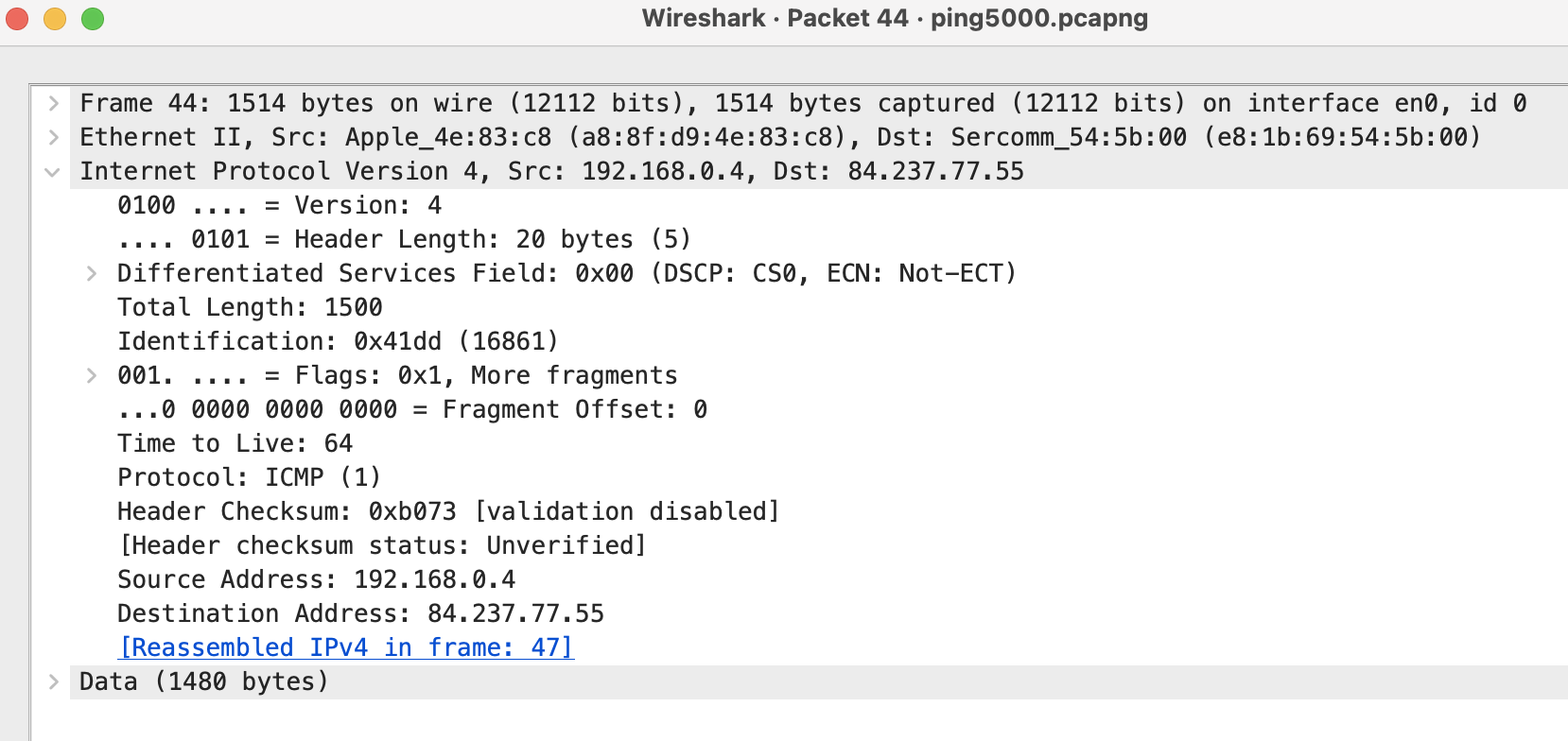
Да, так как с увеличением размера пакета, мы можем видеть, что на уровне IP-

протокола появляются фрагменты “Fragmented IP protocol”. Флаг More Fragments

как раз указывает на наличие фрагментации исходного пакета.



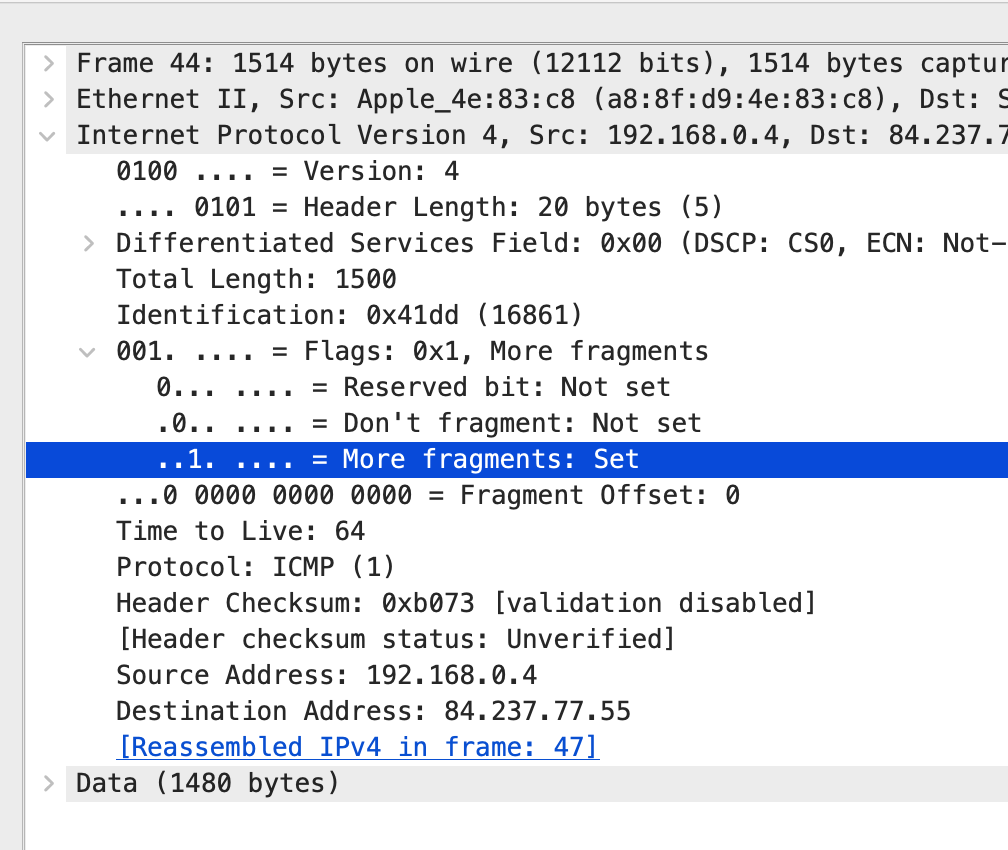
Ниже представлено содержимое кадра 44. Мы видим, что пакет IPv4, который в нем лежит, имеет размер 1500 байт (с учетом заголовка), а размер кадра Ethernet равен 1514 байтам.



Сам заголовок Ethernet II содержит MAC адреса — 12 байт и 2 байта, отвечающие за тип протокола выше (IPv4).

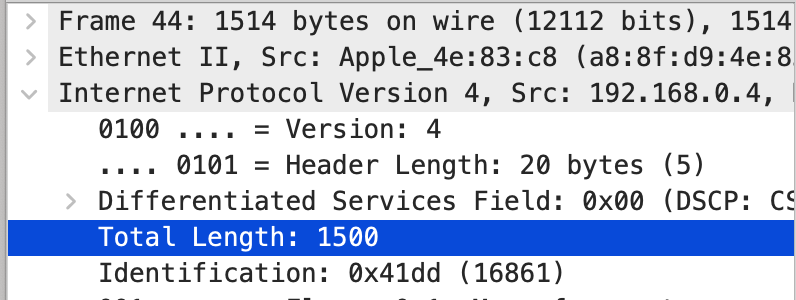
**2. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?**

Флаг «More fragments» в пакете IPv4 отвечает за то, является ли фрагмент не последним.



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

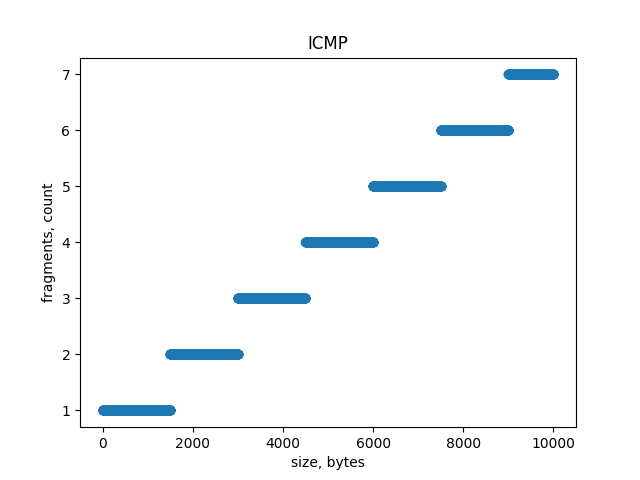
Количество фрагментов зависит от размера пакета и соответствующего соединению MTU (Maximum Transmission Unit – максимальный размер передаваемого блока), который обычно равен 1500 байт (1480 байт без учета заголовка).



Количество фрагментов = ceil(Длина сообщения (байт) / 1500 байт (фрагмент))

ceil() - округление вверх.

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

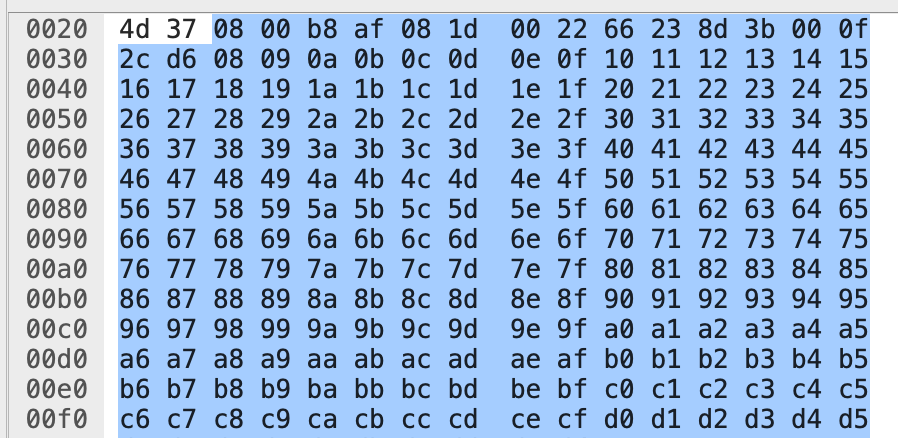


**5. Как изменить поле TTL с помощью утилиты ping?**



**6. Что содержится в поле данных ping-пакета?**

ASCII-символы



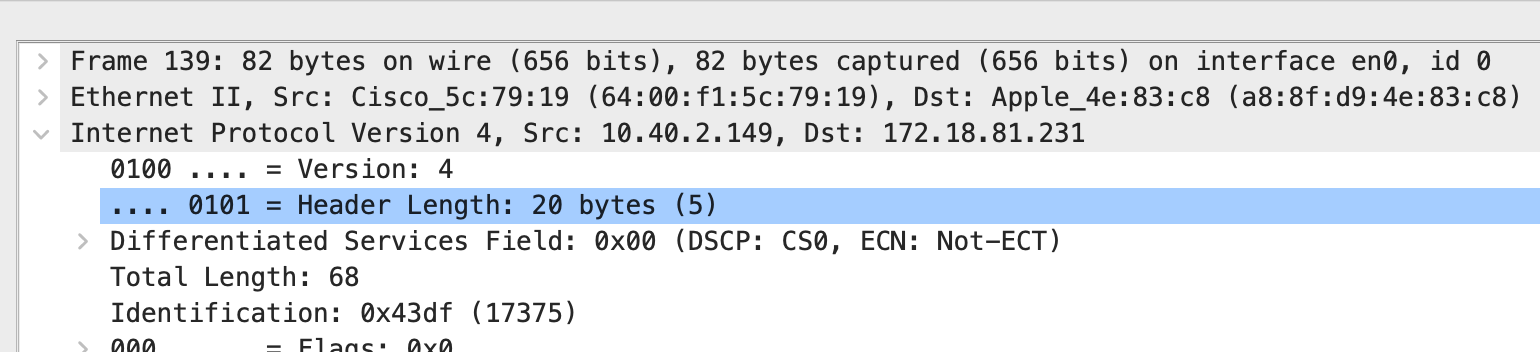
**2.2. Анализ трафика утилиты tracert (traceroute)**

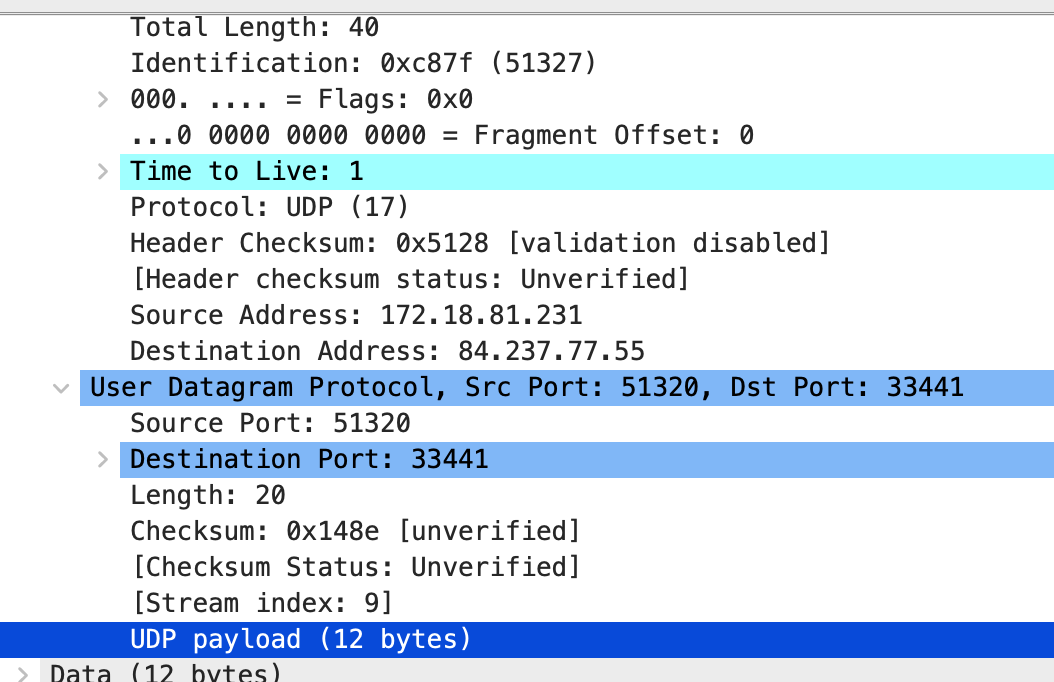
Необходимо отследить и проанализировать трафик, создаваемый утилитой tracert (или traceroute в Linux), запустив её следующим образом из командной строки: “tracert -d адрес\_сайта\_по\_варианту”

**traceroute -n math-sobolev.ru**

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

Ниже представлен ответ от маршрутизатора с типом time-limit-exceeds. В заголовке IP содержится 20 байт. В поле данных 12 байтов.



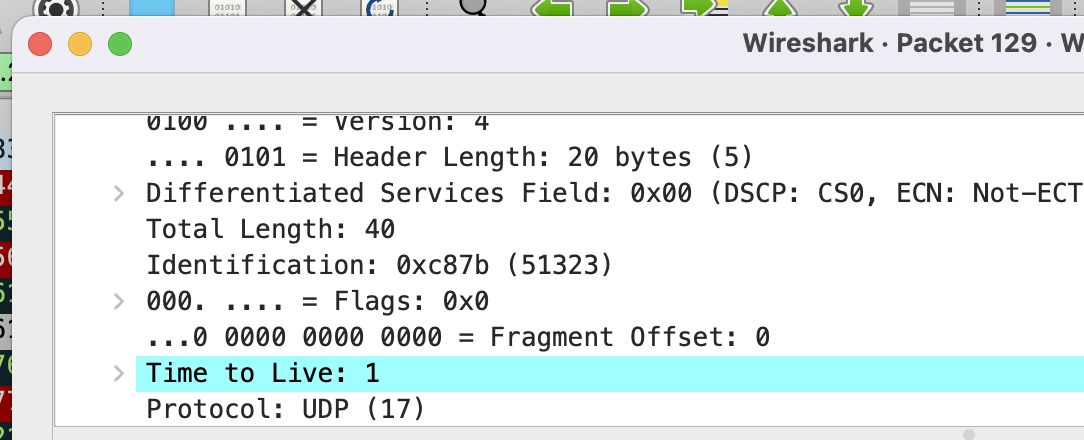
****

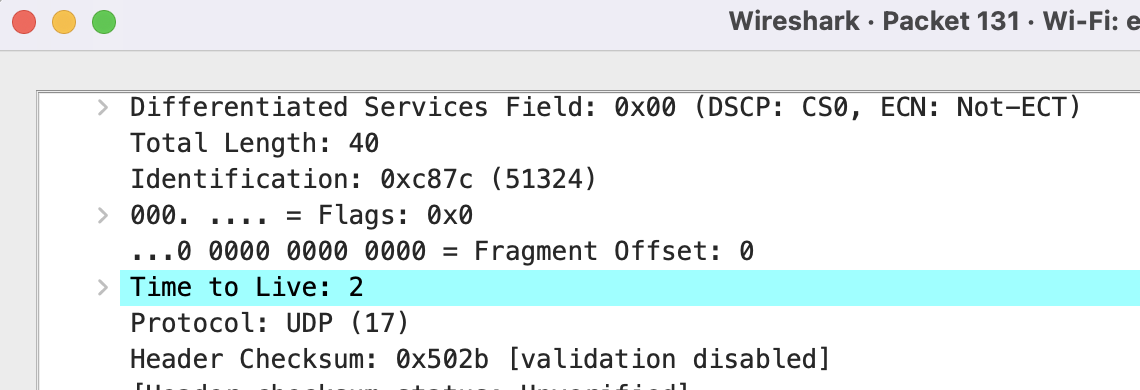
**2. Как и почему изменяется поле TTL в следующих друг за другом ICMP- пакетах tracert? Для ответа на этот вопрос нужно проследить изменение TTL при передаче по маршруту, состоящему из более чем двух хопов.**

Поле TTL в заголовке ICMP пакета используется для предотвращения зацикливания пакетов в сети. При отправке пакета его TTL устанавливается на определенное начальное значение (обычно 64), и каждый маршрутизатор, через который проходит пакет, уменьшает значение TTL на 1. Если значение TTL достигает нуля, пакет отбрасывается, и маршрутизатор отправляет обратно ICMP сообщение о превышении TTL (Time Exceeded).

При выполнении команды traceroute каждый отправляемый пакет имеет начальное значение TTL, которое постепенно уменьшается по мере прохождения через маршрутизаторы по пути к целевому узлу. Таким образом, каждый маршрутизатор, через который проходит пакет, уменьшает значение TTL перед его передачей дальше.

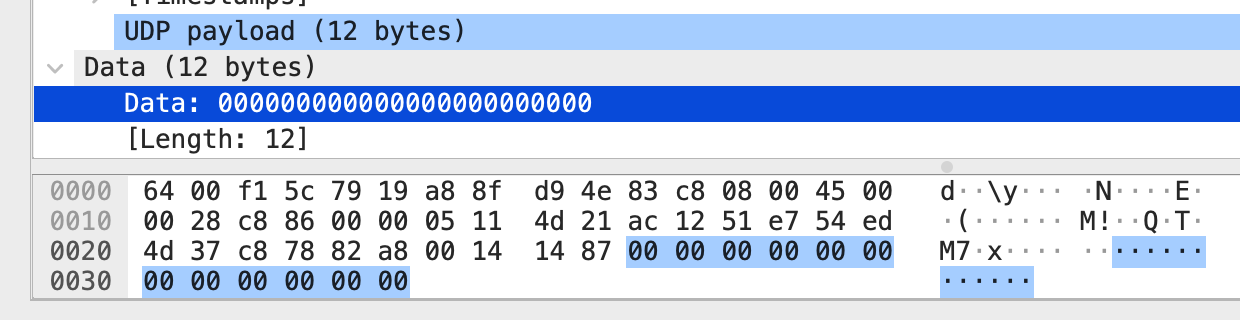
Для определения пути tracert отправляет пакеты с TTL, который увеличивается. Таким образом, когда TTL становится в маршрутизаторе равным 0, то тот посылает ICMP сообщение с информацией о том, что TTL exceeds. Таким образом, по умолчанию сначала посылается пакет с TTL равным 1, затем с TTL равным 2 и т. д. до тех пор пока не достигнем конечного узла.





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

В отличие от пакетов, генерируемых утилитой ping, пакеты, генерируемые утилитой tracert, в поле данных содержат нули. Кроме этого, ICMP-пакеты, генерируемые утилитой ping, содержат TTL = 64, а пакеты, генерируемые traceroute последовательно увеличивают TTL, пока не достигнут пункта назначения.



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

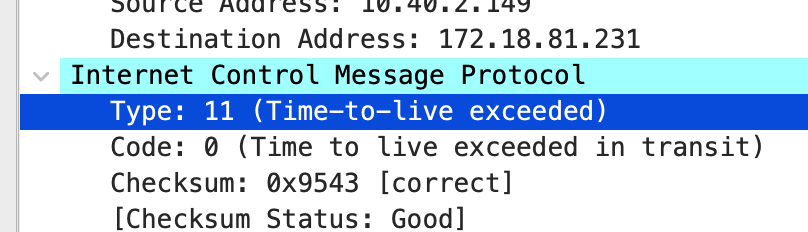
ICMP reply пакеты используются для обеспечения подтверждения успешной доставки пакетов. Примеры ICMP ответов включают Echo Reply, который используется в командах ping для проверки доступности узлов в сети.

ICMP error пакеты используются для сообщения об ошибках, возникших в процессе обработки пакетов в сети. Примеры ICMP ошибок включают Time Exceeded, который сообщает отправителю о том, что TTL пакета истек, и Destination Unreachable, который указывает на то, что пункт назначения недоступен или что пакет не может быть доставлен. ICMP ошибки помогают обнаруживать и реагировать на проблемы в сети, такие как недоступность узлов, сетевые перегрузки и неправильно настроенные маршрутизаторы.

Значение поля TYPE = 0 соответствует сообщению Echo Reply, при тестировании сообщения данного типа не получал.

Значение поля TYPE = 11 соответствует сообщению Time Exceeded (истекло

время)



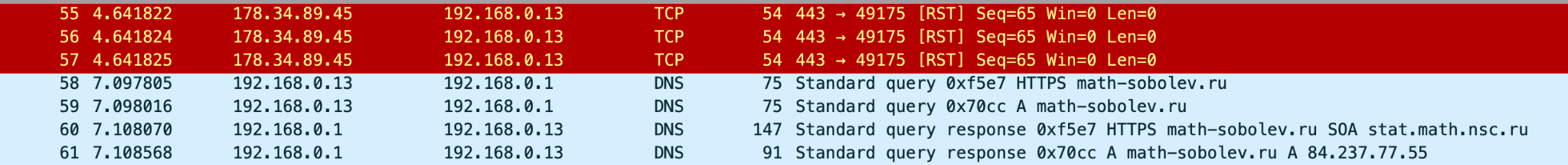
**5. Что изменится в работе tracert, если убрать ключ “-d”? Какой**

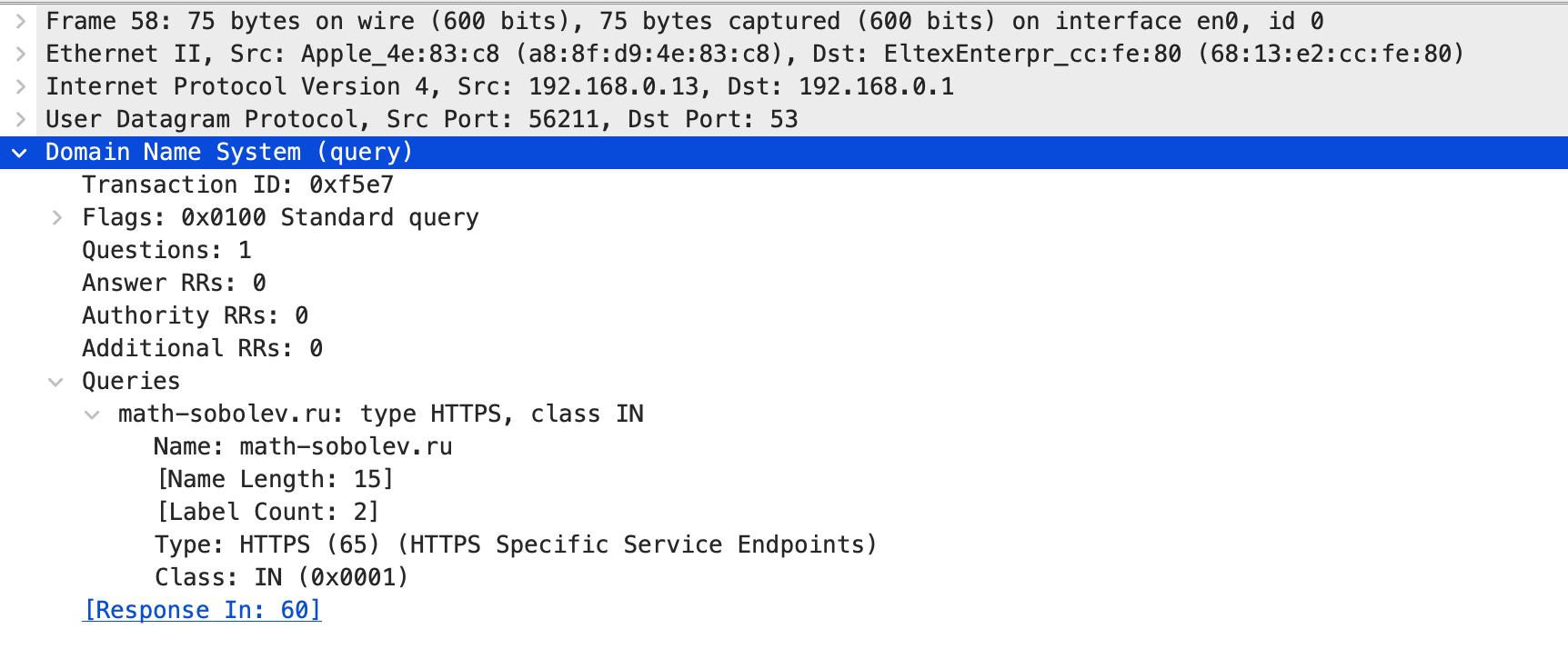
**дополнительный трафик при этом будет генерироваться?**

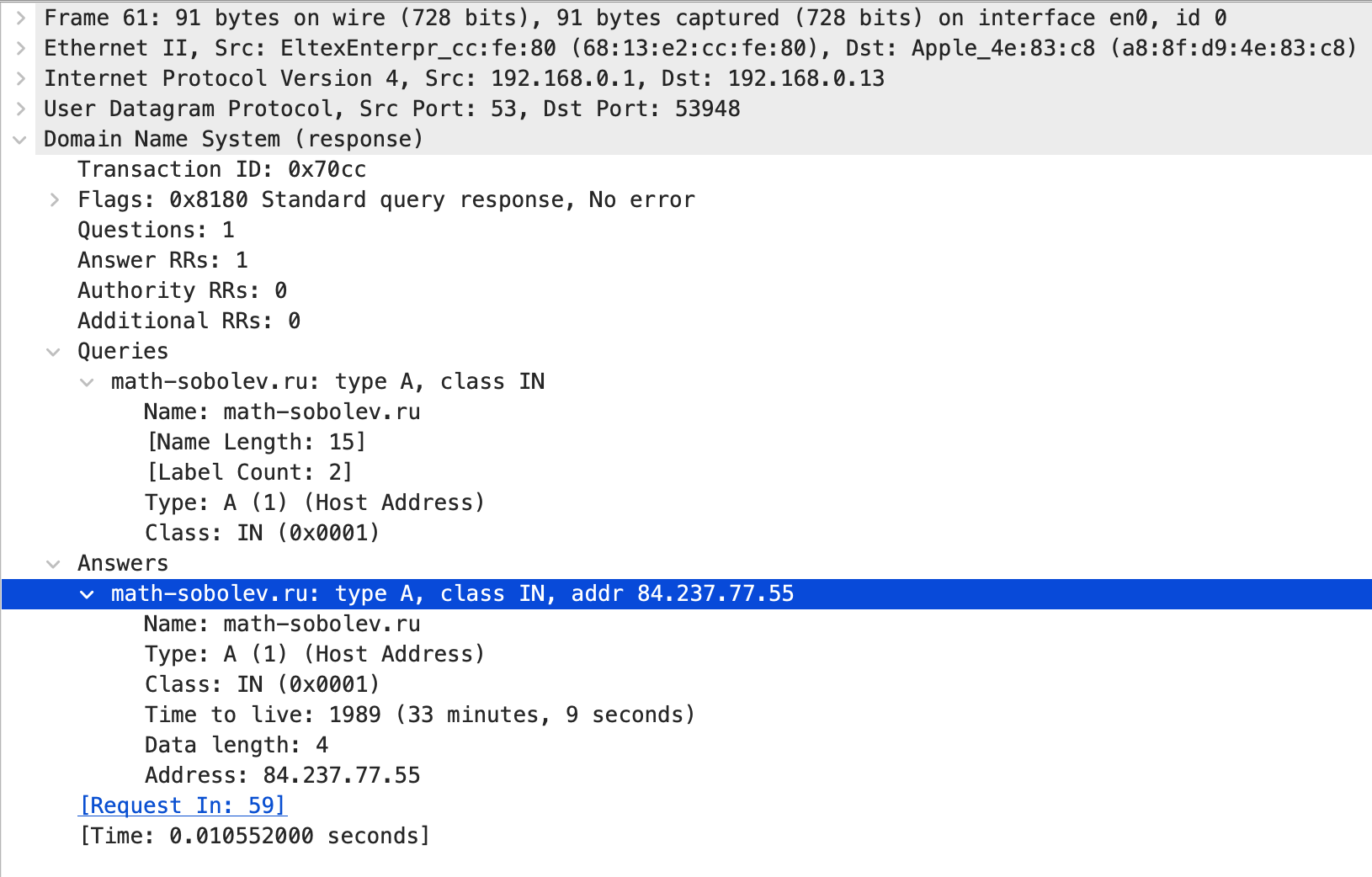
Ключ -d предотвращает попытки команды tracert разрешения IP-адресов промежуточных маршрутизаторов в имена, то есть в выводе будут отсутствовать имена хостов, через которые проходит IP-пакет.

**Анализ DNS-трафика**

Очистим кэш dns - sudo dscacheutil -flushcache; sudo killall -HUP mDNSResponder







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

Для того, чтобы получить IP адрес сайта отправляется запрос на DNS сервер, который возращает IP адрес уже сайта. На рисунке выше мы можем видеть, что получается IPv4 адрес необходимой страницы. Адресом назначения указан адрес маршрутизатора.

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

* Прямой — запрос на преобразование имени (символьного адреса) хоста в его IP-адрес.
* Обратный — запрос на преобразование адреса хоста в его имя.
* Рекурсивный - DNS-сервер опрашивает серверы (в порядке убывания уровня зон в имени), пока не найдёт ответ или не обнаружит, что домен не существует.
* Нерекурсивный (итеративный) - DNS-сервер либо возвращает данные о зоне, за которую он ответственен, либо возвращает ошибку.

Также выделяют запросы типа:

* A — получение IPv4
* AAAA — получение IPv6
* CNAME — получение канонического имени
* MX — получение информации о почтовых серверах, ответственных за обработку почту для данного домена
* NS (Name Server) — вернуть список DNS серверов, ответственных за данный домен
* PTR - обратная DNS-запись или запись указателя связывает IP-адрес хоста с его каноническим именем.

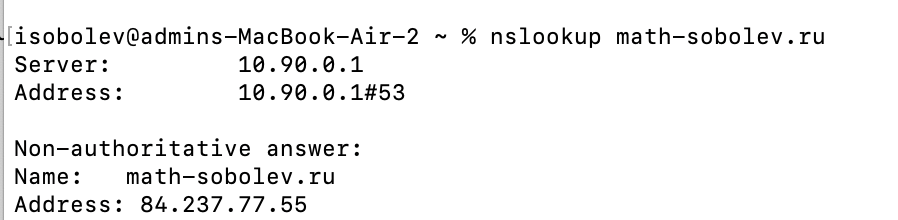
**3. В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?**

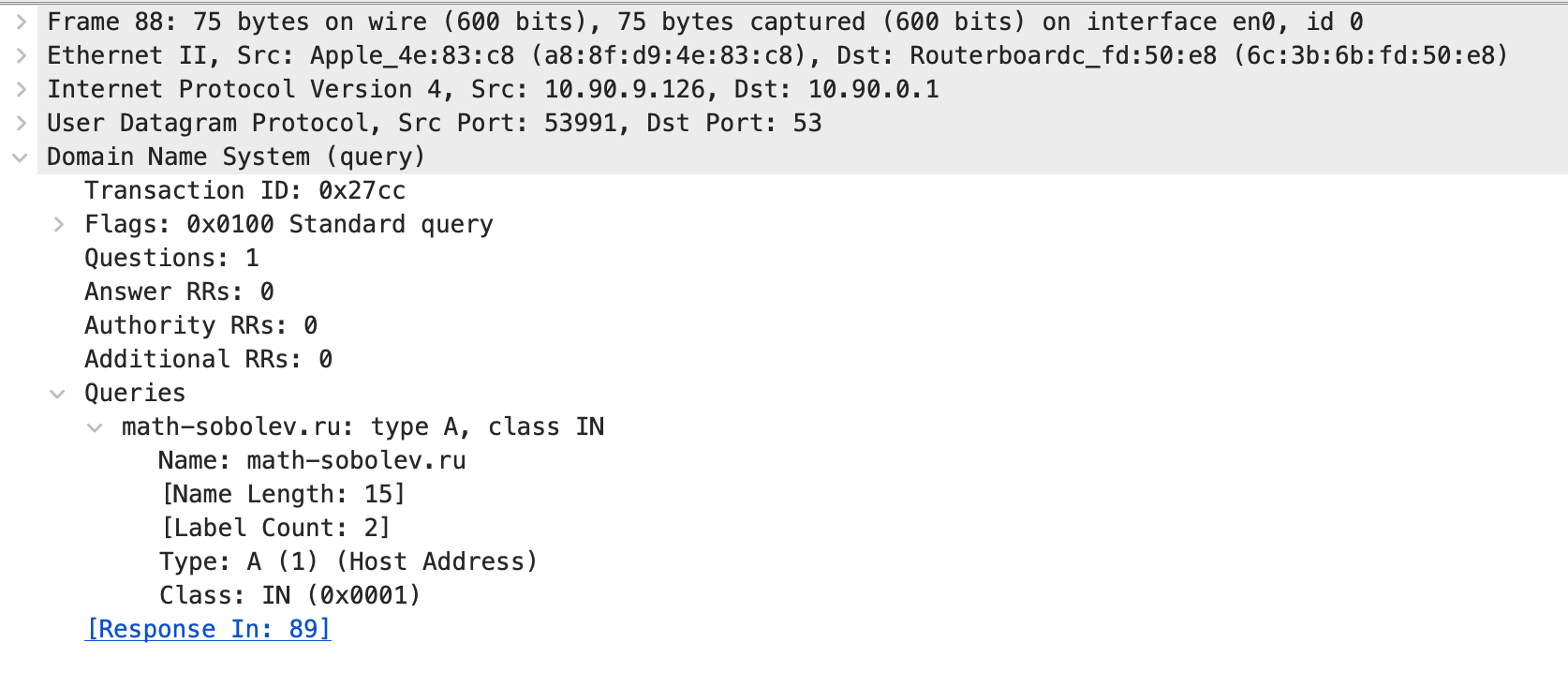
Выполнение дополнительных DNS-запросов необходимо, если изображения, содержащиеся на сайте, находятся на другом доменном имени. Также такая ситуация возможна в случае, если сайт использует CDN (Content Delivery Network) для ускорения загрузки сайта. С его помощью данные передаются быстро, независимо от местонахождения хоста. Это возможно благодаря распределенным по всему миру кэширующим серверам.

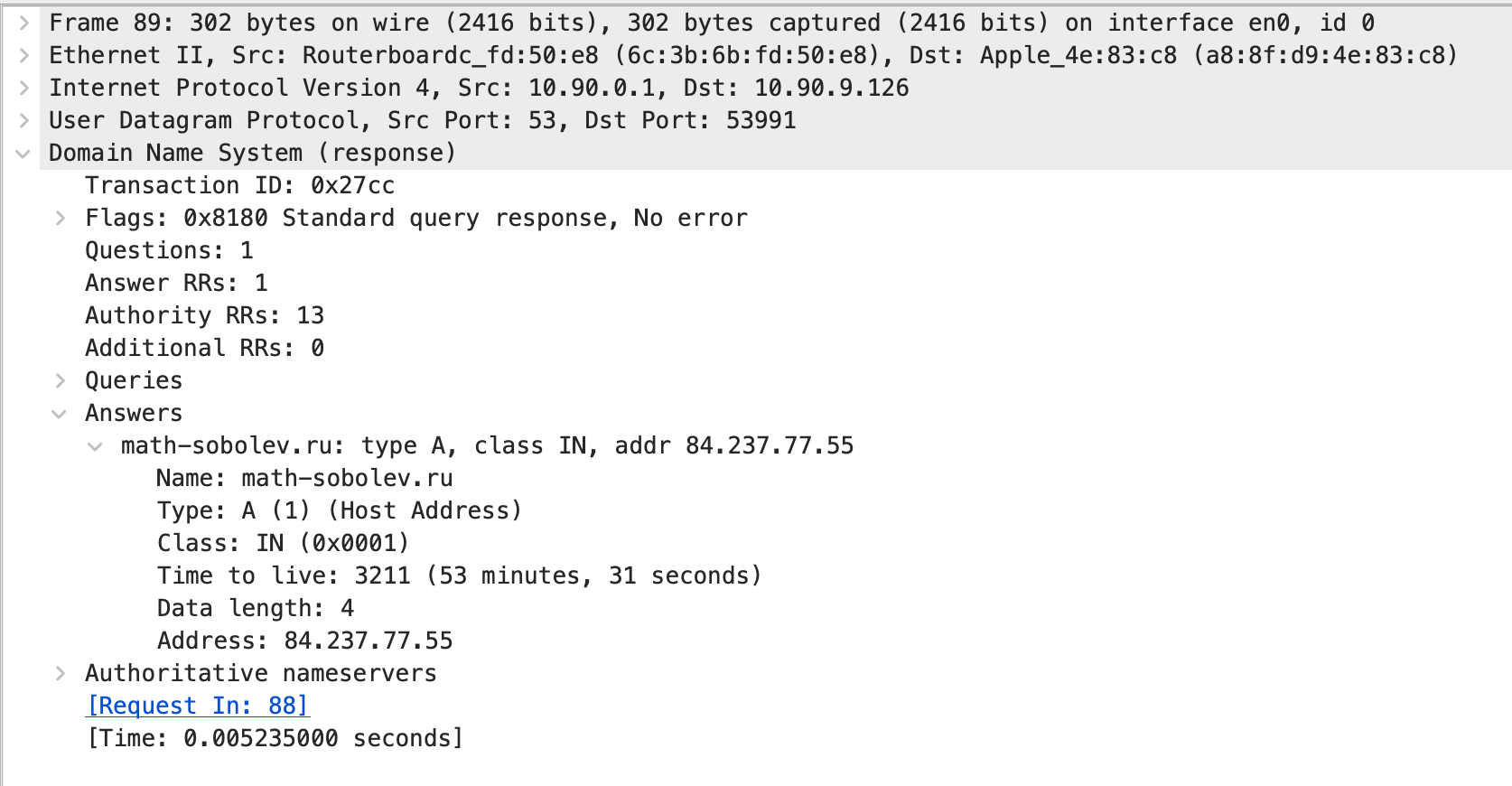
**Анализ трафика утилитой nslookup**

Утилита nslookup предназначена для получения различной информации с DNS серверов.

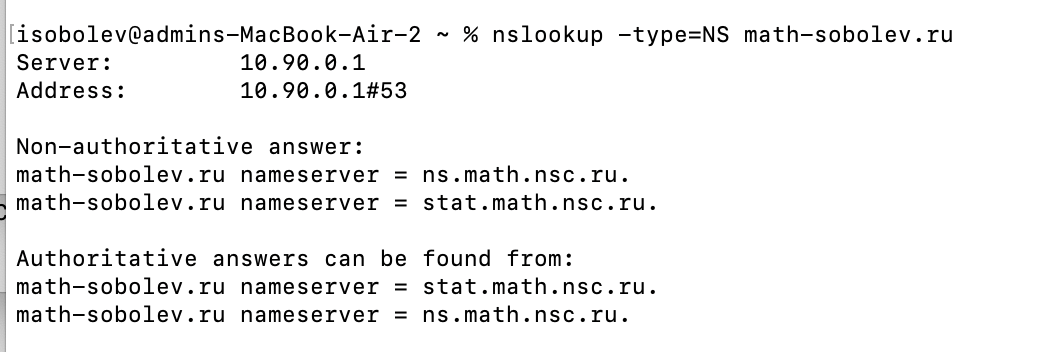
Первая команда, которая была послана для получения dns имени по ip адресу. Тип запроса — PTR, то есть получаем по известному ip адресу dns имя. Ниже представлено содержимое dns сообщения отправителя и получателя.



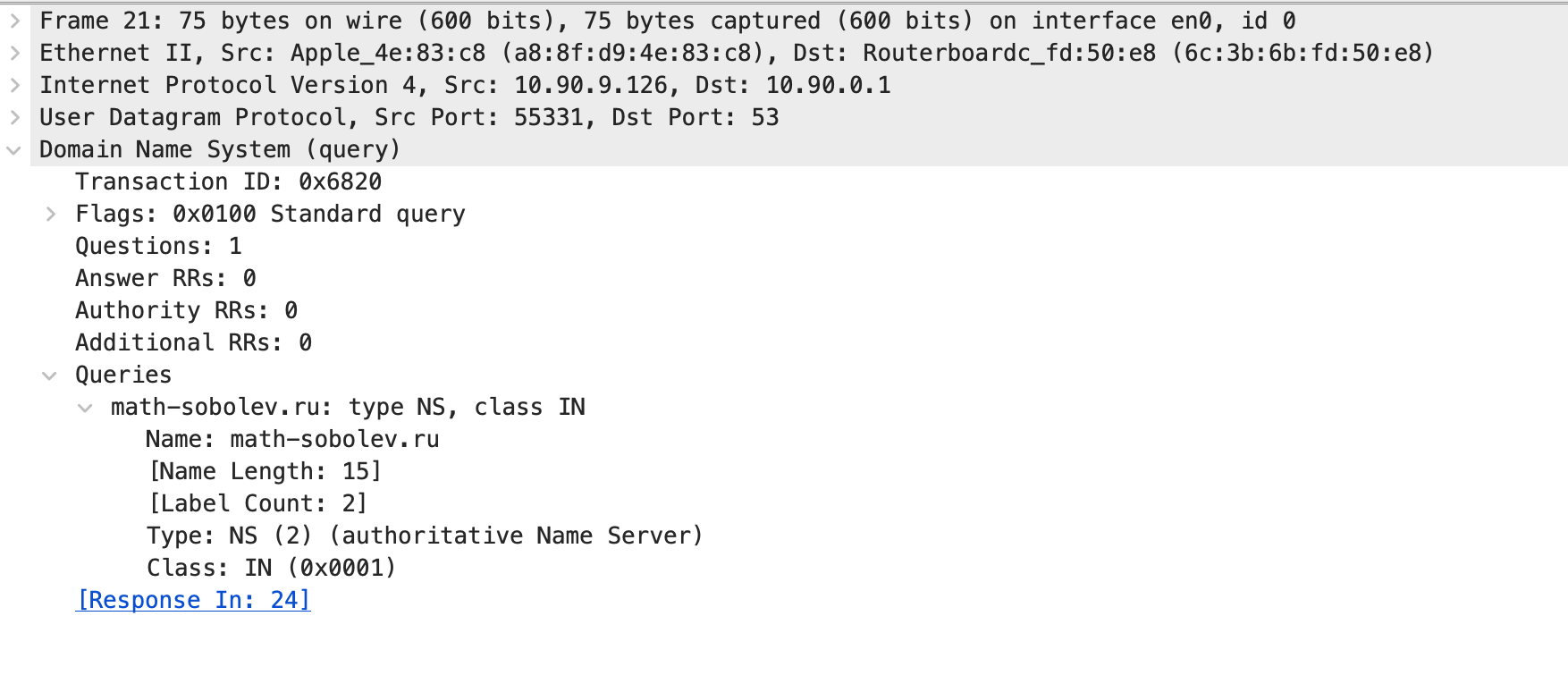




Далее запускаем утилиту с типом NS.

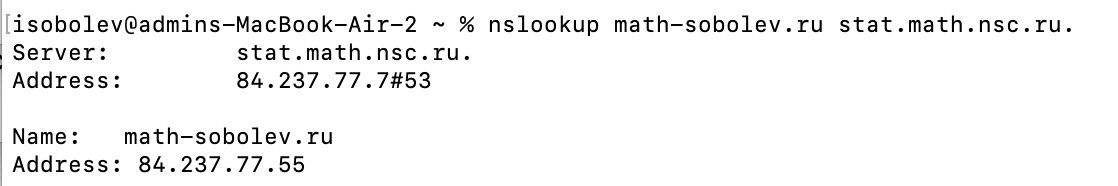


Данный запрос — запрос типа NS, то есть для получения DNS-серверов, которые отвечают за разрешение ip адреса и dns имени для данного доменного имени. Таким образом, определели, что авторитативный ответ может быть получен от DNS-серверов ns.math.nsc.ru. и stat.math.nsc.ru.





Неавторитативная запись означает, что соотношение было получено нет от dns-сервера, который отвечает за данный сектор доменных имен. Так, например, если мы попросим авторитетный dns-сервер предоставить эту информацию, то этого предупреждения не будет.



**Ответы на вопросы**

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

Сначала nslookup посылает запрос на рекурсивный DNS-сервер с типом запроса A, чтобы узнать IP-адрес сервера по его url. DNS-сервер в своем ответе присылает IP-адрес узла. После этого мы посылаем DNS-запрос с type=NS, чтобы получить IP-адрес авторитарного сервера, на котором хранится информация о всех IP-адресах нашей зоны. В отчет получает IP-адрес авторитативного сервера. После этого посылаем DNS-запрос уже на авторитарный сервер по адресу 194.85.61.20 и получаем IP-адрес сайта.

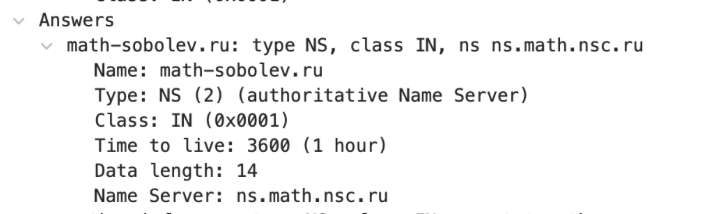
При запуске в п.2 утилита ищет IP-адрес хоста (запись типа A (IPv4) или AAAA

(IPv6)).

При запуске в п.4 утилита ищет Name Server для запрашиваемого хоста.

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

Поле Answer содержит имя хоста, тип и класс записи, TTL, длину поля и IP-адрес запрашиваемого ресурса.



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

С помощью утилиты nslookup было определено, что авторитативный ответ может вернуть сервер ns.math.nsc.ru. и stat.math.nsc.ru.

**Вывод**

В ходе выполнения лабораторной работы был проведен анализ сетевого трафика с использованием программы Wireshark. Было проведено исследование пакетов, передаваемых при выполнении утилит ping и traceroute, изучено их содержимое и информация, которую они несут. Также был проанализирован трафик DNS запросов и запросов, генерируемых утилитой nslookup. Оказалось, что кэширование также влияет на работу DNS.