### **Задание 1. Анализ захвата трафика**

Описание задания Вам поставили задачу проанализировать результаты захвата трафика сетевого интерфейса. Это базовый навык при работе с сетью. В будущем он пригодится вам для того, чтобы выявлять источники проблем в сети и проверять проблемы безопасности (траблшутинг).

### **Требования к результату**

* Вы должны отправить скриншоты захваченного пакета. Пример вы найдёте по ссылке.
* К скриншотам необходимо приложить комментарии с информацией, какие протоколы и уровни модели OSI вы обнаружили.

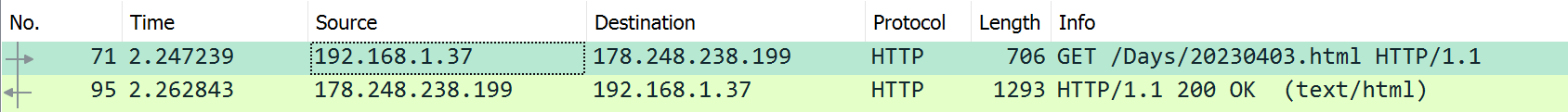
### **Процесс выполнения**

1. Откройте Wireshark.
2. Запустите захват трафика с сетевого интерфейса.
3. Запустите браузер и зайдите на любой сайт.
4. Выберите какой-нибудь пакет из захваченного трафика.
5. Посмотрите разные уровни и разверните параметры при необходимости.
6. Напишите, какие протоколы и уровни модели OSI вы видите. Ответ внесите в комментарии к решению задания в личном кабинете Нетологии.

**Решение:**

Для простоты выбрал сайт со статической html страницей.

Запросил сайт - зафиксировались два пакета:

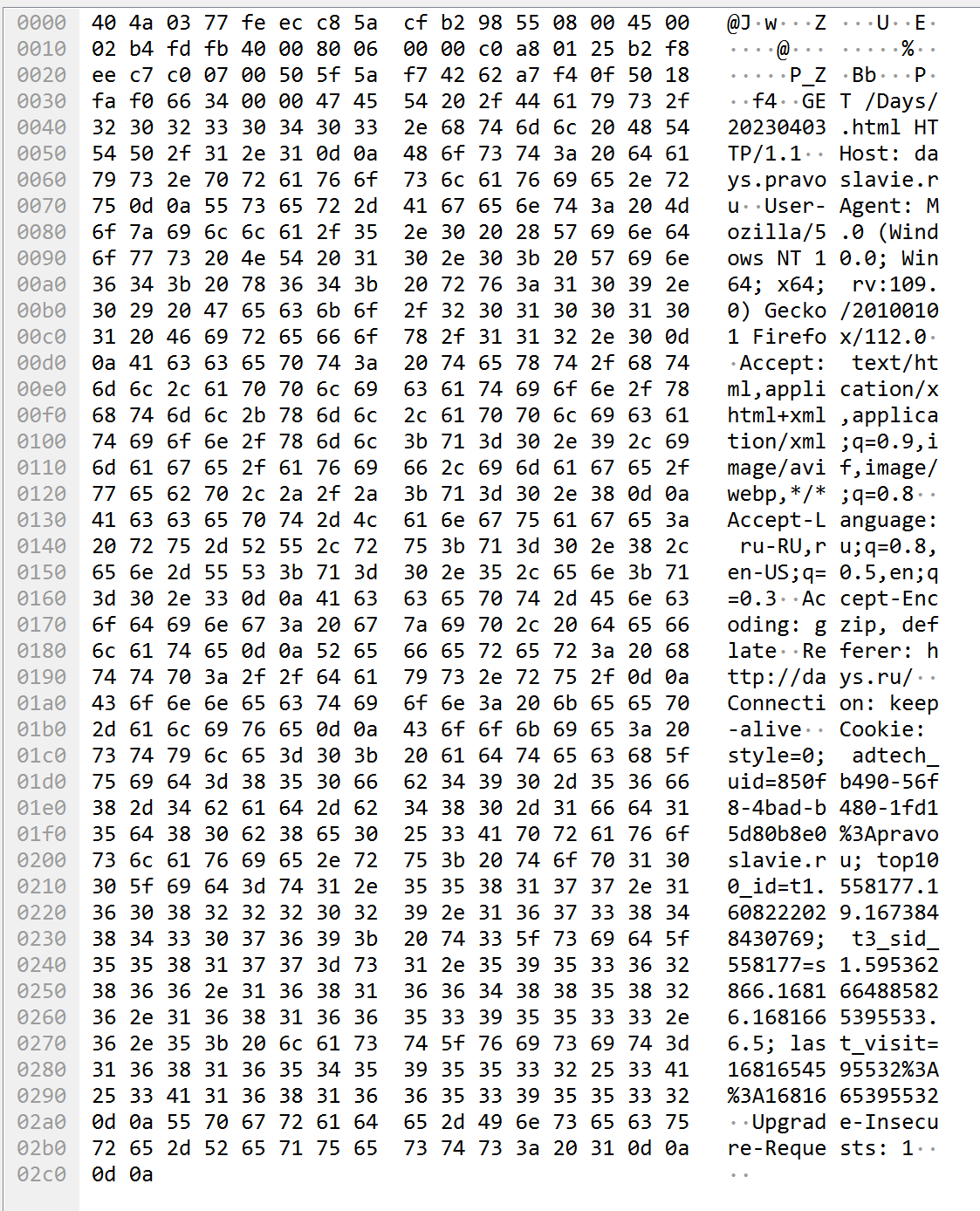


1 - это мой запрос к сайту.

2 - ответ от сайта.

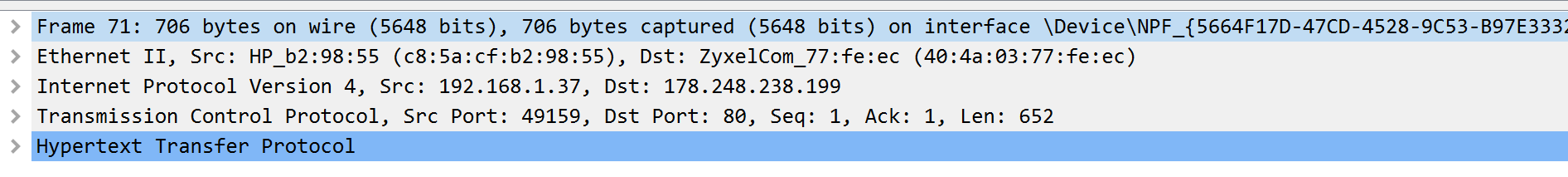
**Смотрим первый запрос детальнее.**

1. захваченный трафи - запрос с моего ПК:

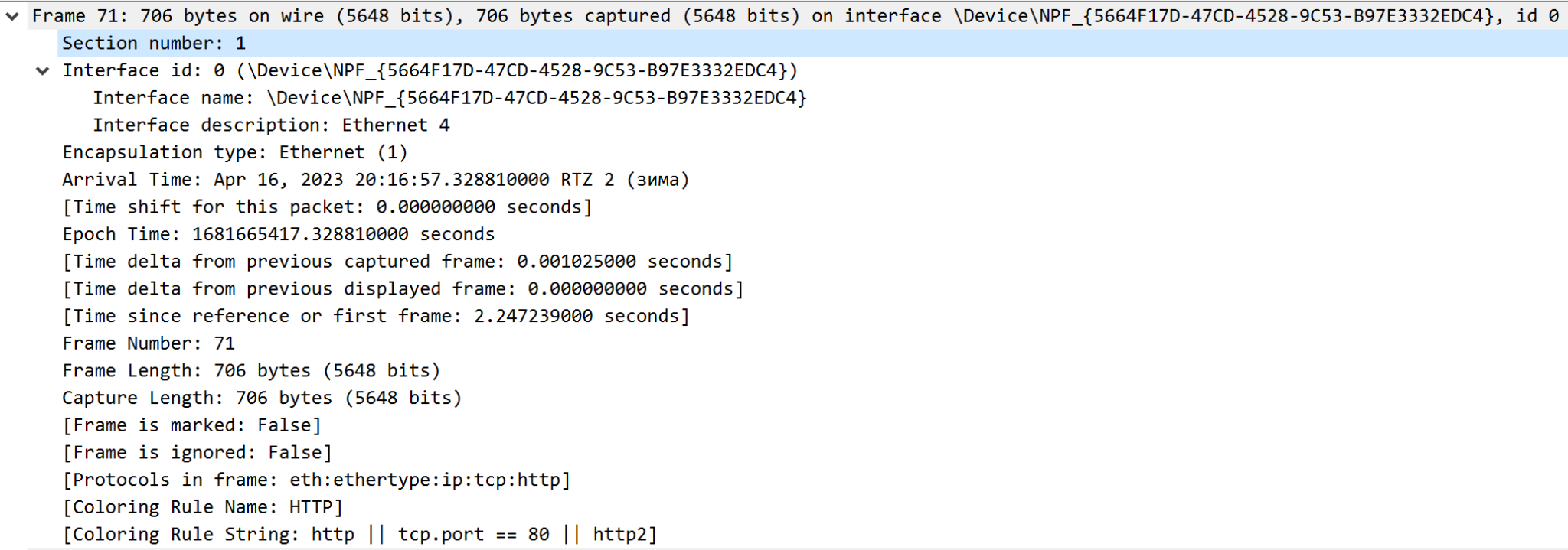


* указна каждый байт в 16-ти ричной форме (hex) и в символьной.
* не все коды корректно отображаются в символе, поэтому бывает нужно и hex формат смотреть.

Вся информация сгруппирована на разделы:

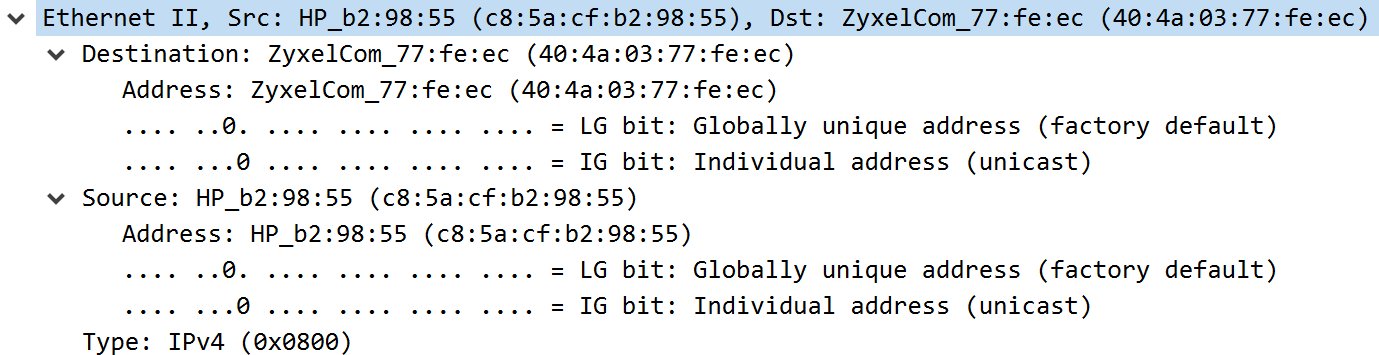


Смотрим первый раздел “Frame…” - дана общая информация о запросе:



* Указана общая информация сколько байт захвачено на каком интерфейсе.
* Данные о времени.
* Encapsulation type: Ethernet (1) - это видимо что данные инкапсулированы в Ethernet протокол.
* Всего фреймов было 71 - кусков данных для пересылки. и передано 706 байтов.
* [Protocols in frame: eth:ethertype:ip:tcp:http] - выявлены эти протоколы.
* [Coloring Rule String: http || tcp.port == 80 || http2] **-** это видимо так - что на прикладном уровне - http протокол, стандарт http2, передано через tcp протокол - запрос на удаленный порт 80.

**Рассматриваем следующий раздел:**

****

* тут информация о канальном уровне OSI:
* Ethernet II, Src: HP\_b2:98:55 (c8:5a:cf:b2:98:55), Dst: ZyxelCom\_77:fe:ec (40:4a:03:77:fe:ec) - между какими устройствами с какими MAC адресами были запросы. Это ПК и Роутер (NAT) “у клиента” до “Интернет”.
* В виде байтов указаны прям эти MAC адреса:  
  Получателя:



Отправителя:



Эти адреса совпадают с:

* адрес получателя можно посмотреть через arp -a - получаем таблицу ARP сопоставления MAC и Ethernet адресов. Цитирую:



* адрес роутера и его MAC адрес - совпадает первыми байтами трафика.

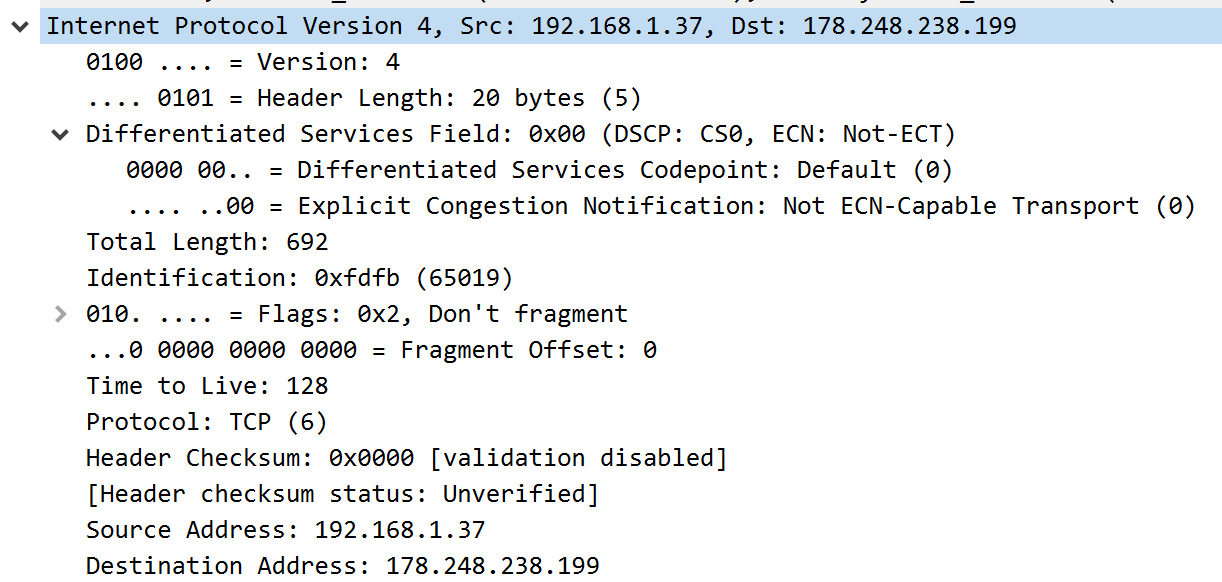
Далее MAC адрес отправителя:  
- совпадает из ipconfig /all - на Windows:



* это как раз следующие байты в трафике захваченном.
* Type: IPv4 (0x0800) - указание, что на следующем уровне будет протокол IPv4.



**Далее начинается уровень - “сетевой”:**



* указаны IP адреса и версия протокола



* где:
* 4 - это 0100 .... = Version: 4
* 5 это .... 0101 = Header Length: 20 bytes (5)
* заголовок из 20 байтов.

Далее указаны некоторые параметры протокола:

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

Длина данных:

Total Length: 692

и др.

Далее сказано что будет TCP протокол на следующем уровне:

Protocol: TCP (6)

Далее видимо указано что без контрольной суммы на следующем транспортном уровне:

Header Checksum: 0x0000 [validation disabled]

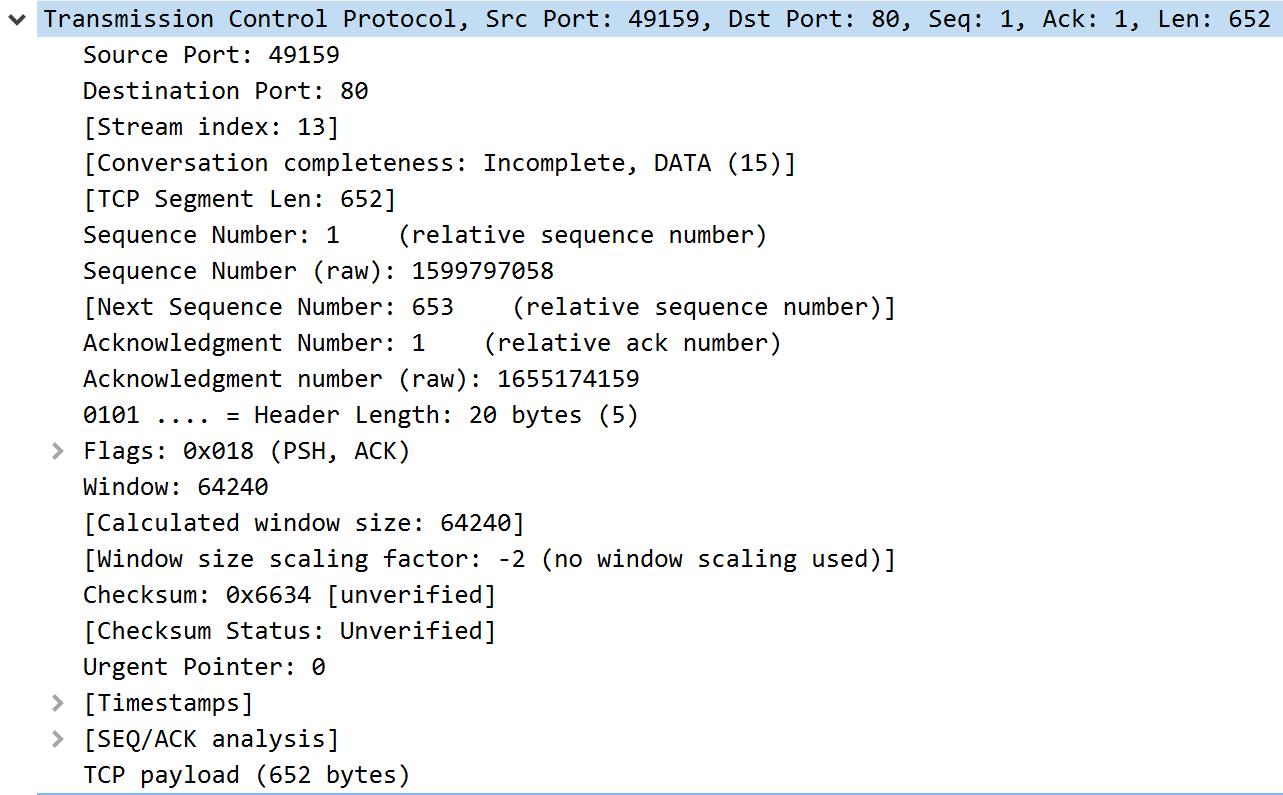
[Header checksum status: Unverified]

и байты IP получателя и отправителя:

Source Address: 192.168.1.37

Destination Address: 178.248.238.199

**Далее - транспортный уровень:**



* указано с какого порта на какой порт и др. информация.
* далее сама посылка из 652 байта:
* TCP payload (652 bytes)

**Прикладной уровень**

Далее Wireshark распознает эту посылку, и понимает, что это http протокол на прикладном уровне:



* тут показано: куда идет запрос, метод запроса, адрес URI (ссылка), версия протокола, хост, информация о браузере и др. информация, данные куков:
* Cookie: style=0; adtech\_uid=850fb490-56f8-4bad-b480-1fd15d80b8e0%3Apravoslavie.ru; top100\_id=t1.558177.1608222029.1673848430769; t3\_sid\_558177=s1.595362866.1681664885826.1681665395533.6.5; last\_visit=1681654595532%3A%3A1681665395532\r\n

По первому заданию - конец.

### **Задание 2. Инкапсуляция данных**

Описание задания Вам поставили задачу найти различия в инкапсуляции данных разных протоколов и технологий. Это также базовый навык при работе с сетью. Вы сможете понимать, как одни приложения и технологии зависят от других протоколов и приложений. Это позволит эффективнее решать сетевые проблемы.

### **Требования к результату**

* Вы должны отправить скриншоты захваченных пакетов.
* В комментариях к скриншоту необходимо указать, чем они различаются по уровням модели OSI.

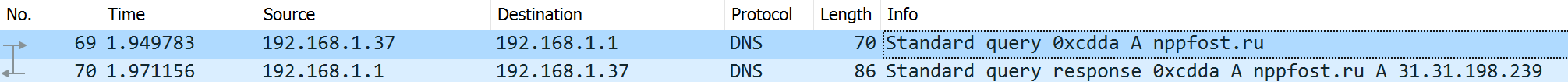
### **Процесс выполнения**

1. Откройте Wireshark.
2. Запустите захват трафика с сетевого интерфейса.
3. Запустите браузер и зайдите на любой сайт.
4. Сначала установите фильтр захваченного трафика по технологии DNS, затем выберите любой HTTP-поток.
5. Сравните пакеты между собой. В чём отличия с точки зрения модели OSI? Ответ внесите в комментарии к решению задания в личном кабинете Нетологии.

**Решение:**

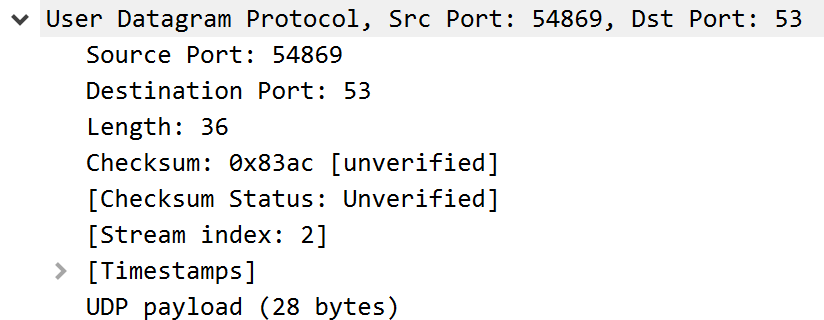
Фильтр по DNS:

было много пакетов запросов и ответов DNS рассмотрю только первый запрос и ответ:



На канальном и сетевом уровне примерно одинаково, я их уже подробно рассмотрел в задании 1, тут не вижу смысла рассматривать подробно.

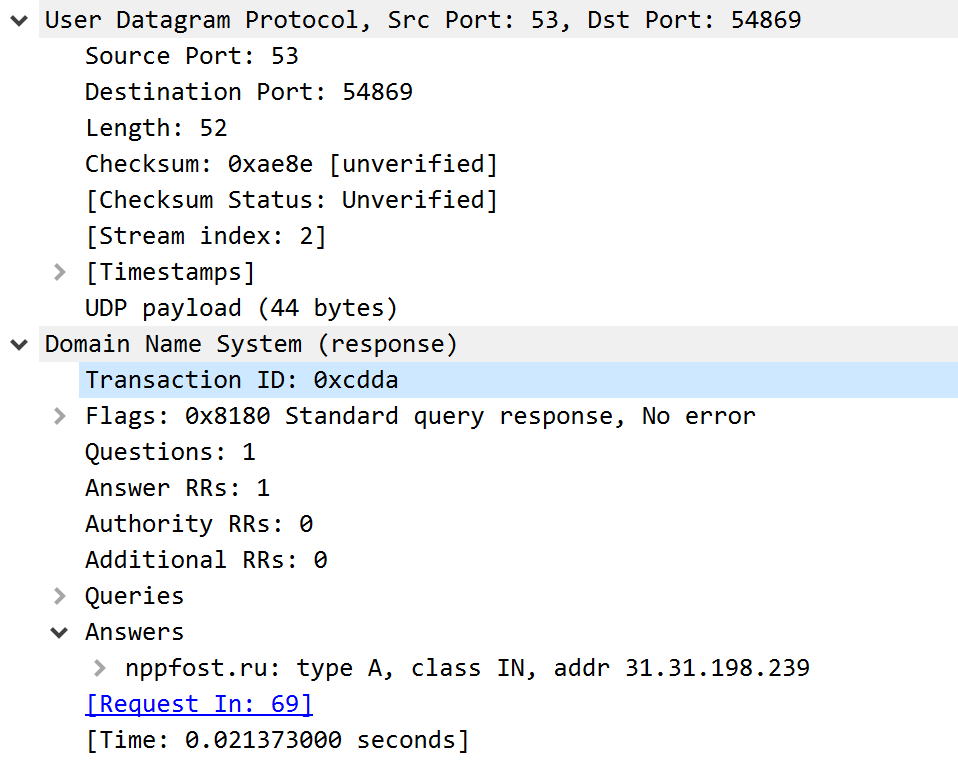
На транспортном уровне:  
 Для DNS используется UDP протокол:



На прикладном уровне - DNS протокол:

На этот запрос приходит ответ.

Канальный и сетевой не буду рассматривать, а сразу Трнаспортный и Прикладной уровни:



Итого на прикладном уровне мы запросили запрос IP для доменного имени в первом запросе (посылке) и получили ответ - IP в посылке ответе.

Подробно рассматривать http посылки думаю смысла нет, т.к. уже их рассмотрел в задании 1.

По поводу:

Сравните пакеты между собой. В чём отличия с точки зрения модели OSI?

* Отличия, существенные в протоколах:
  + на прикладном уровне. для DNS протокола - суть запрос IP для доменного имени и получение ответа. А в HTTP запрос данных с сайта и получение страницы.
  + разница и на транспортном уровне - для DNS там на транспортном уровне используется UDP протокол, а для HTTP - TCP протокол.