

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа компьютерных технологий и информационных систем

Отчёт по лабораторным работам

Дисциплина: Технологии компьютерных сетей.

Выполнил студент гр. 5130901/10101 _____ Д.Л. Симоновский
(подпись)

Руководитель _____ Н.В. Богач
(подпись)

“10” сентября 2024 г.

Санкт-Петербург

2024

Оглавление

1. Лабораторная работа 1. Wireshark: Введение.....	2
1.1. Цель работы:	2
1.2. Ход работы:	2
1.3. Вывод:	5
2. Приложение:	6

1. Лабораторная работа 1. Wireshark: Введение.

1.1. Цель работы:

В этой лабораторной работе мы познакомимся с программой Wireshark, которая используется для анализа сетевого трафика путем перехвата пакетов данных и изучения их структуры. Wireshark является мощным инструментом для мониторинга сетевой активности, позволяющим наблюдать за обменом сообщениями между протоколами, такими как HTTP, FTP, TCP, UDP, DNS, и IP, на различных уровнях сетевой архитектуры.

Цель данной работы — углубить понимание работы сетевых протоколов, увидеть их в действии, анализируя последовательности пакетов, передаваемых между устройствами в сети. Мы будем наблюдать, как протоколы, используемые нашим компьютером, обмениваются данными с удаленными узлами сети Интернет, что позволит лучше понять работу сетевых приложений.

В ходе лабораторной работы мы научимся использовать Wireshark для захвата сетевых пакетов, интерпретации их структуры и анализа различных полей протокольных сообщений. Особое внимание будет уделено тому, как сообщения верхних уровней, такие как HTTP-запросы, инкапсулируются в кадры канального уровня и передаются по сети. Это позволит понять, как данные путешествуют через сетевые уровни от приложения до физического канала связи.

Таким образом, мы увидим на практике, как работают сетевые протоколы, и получим ценные навыки работы с инструментами анализа сетевого трафика, что является важной частью изучения современных компьютерных сетей.

1.2. Ход работы:

Перейдем непосредственно в Wireshark и посмотрим на главный экран:

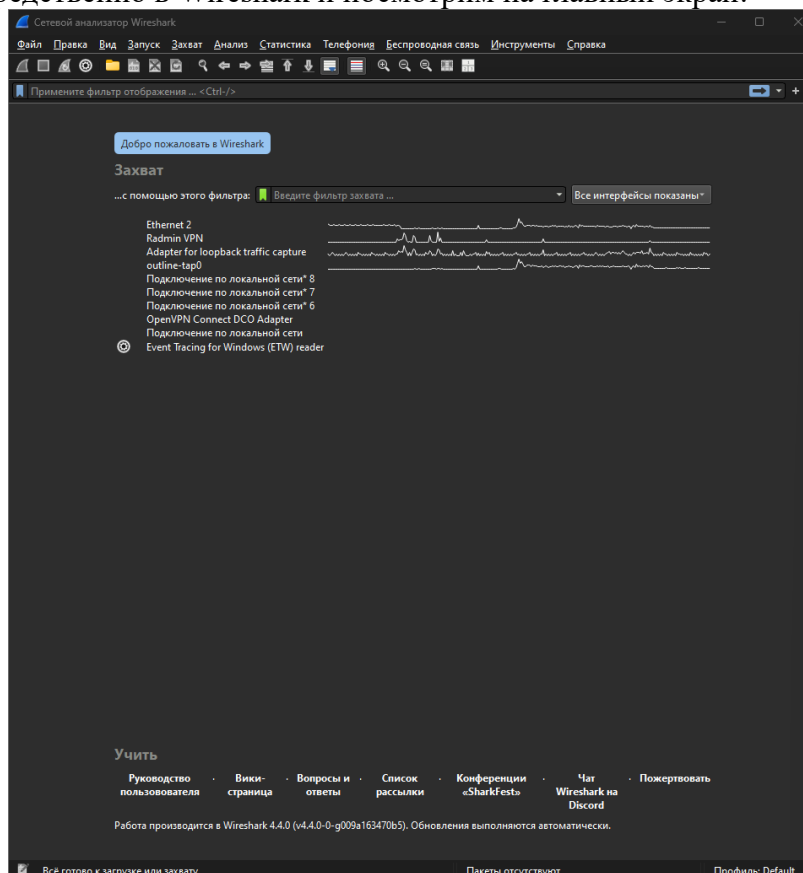


Рис. 1.1. Главный экран Wireshark.

Здесь мы видим все подключенные интернет адаптеры к компьютеру, в том числе различные VPN подключения. Ethernet 2 является основным адаптером, поэтому выберем именно его. Тогда окно изменится следующим образом:

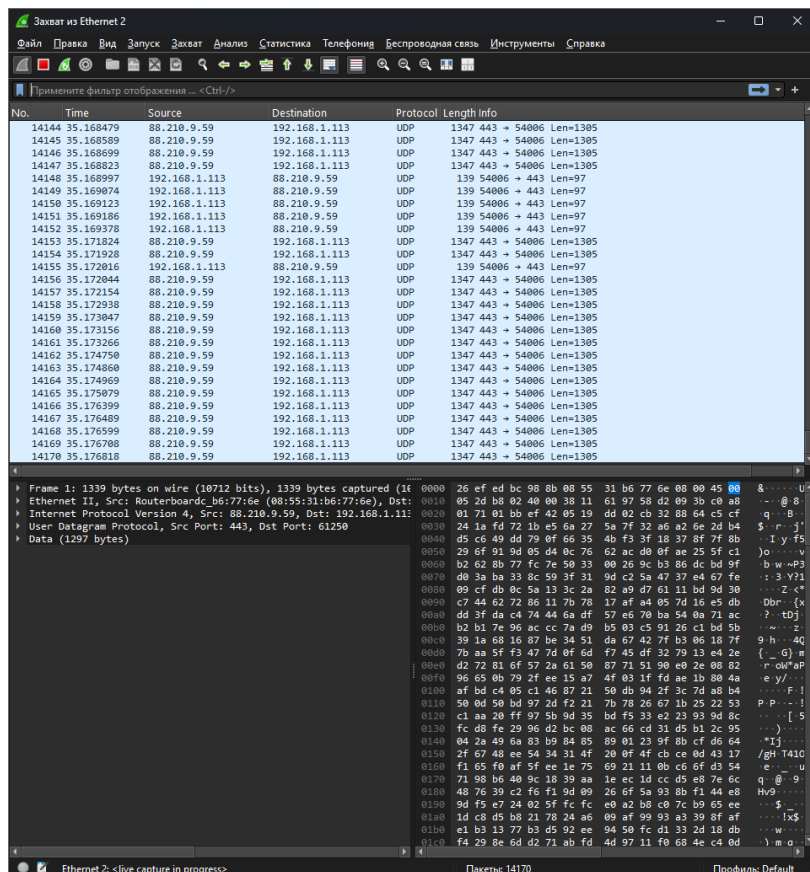


Рис. 1.2. Wireshark. Захват из Ethernet 2.

На рис. 1.2. мы видим следующие поля в интерфейсе:

- Командные меню — меню для сохранения, открытия, захвата данных и др.
- Поле фильтра — фильтрация пакетов по протоколам и критериям.
- Окно списка пакетов — список перехваченных пакетов с фильтрацией.
- Окно деталей заголовка — детальная информация по выбранному пакету.
- Окно содержимого пакета — данные пакета в шестнадцатеричном и ASCII формате.

Как мы видим по рисунку 1.2. наш компьютер непрерывно обменивается множеством различных пакетов с ресурсами, которые мы даже не запускали. Попробуем отследить конкретно какой-то пакет. Например, по протоколу http, для этого в поле фильтров напишем http. Теперь нам будут отображаться только http пакеты. Попробуем посмотреть на них, для этого перейдем в браузер и зайдём на сайт <https://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html>. В браузере отобразится следующее окно:

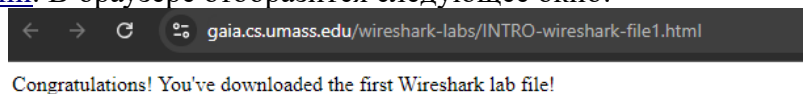


Рис. 1.3. Результат перехода на сайт.

Однако для нас представляет интерес, что мы увидим в wireshark:

No.	Time	Source	Destination	Protocol	Length Info
150	4.377738	192.168.1.120	128.119.245.12	HTTP	548 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
161	4.498666	128.119.245.12	192.168.1.120	HTTP	492 HTTP/1.1 200 OK (text/html)
165	4.602549	192.168.1.120	128.119.245.12	HTTP	494 GET /favicon.ico HTTP/1.1
180	4.727888	128.119.245.12	192.168.1.120	HTTP	538 HTTP/1.1 404 Not Found (text/html)

Рис. 1.4. Окно Wireshark после открытия страницы.

Можно увидеть 2 GET запроса и 2 ответа. Первый – запрос непосредственно веб-страницы, по протоколу HTTP, а второй запрос какого-то файла иконки, который не был успешно получен. Рассмотрим первый запрос подробнее:

No.	Time	Source	Destination	Protocol	Length Info
150	4.377738	192.168.1.120	128.119.245.12	HTTP	548 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
161	4.498666	128.119.245.12	192.168.1.120	HTTP	492 HTTP/1.1 200 OK (text/html)
165	4.602549	192.168.1.120	128.119.245.12	HTTP	494 GET /favicon.ico HTTP/1.1
180	4.727888	128.119.245.12	192.168.1.120	HTTP	538 HTTP/1.1 404 Not Found (text/html)

Рис. 1.5. Пакет запроса в Wireshark.

Как мы видим, здесь есть вся информация о нашем запросе, в том числе метод (GET) и другая служебная информация, такая как User-Agent и др.

Далее рассмотрим ответ, который, как видно по рисунку, идет следующим:

No.	Time	Source	Destination	Protocol	Length Info
150	4.377738	192.168.1.120	128.119.245.12	HTTP	548 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
161	4.498666	128.119.245.12	192.168.1.120	HTTP	492 HTTP/1.1 200 OK (text/html)
165	4.602549	192.168.1.120	128.119.245.12	HTTP	494 GET /favicon.ico HTTP/1.1
180	4.727888	128.119.245.12	192.168.1.120	HTTP	538 HTTP/1.1 404 Not Found (text/html)

Рис. 1.6. Пакет ответа в Wireshark.

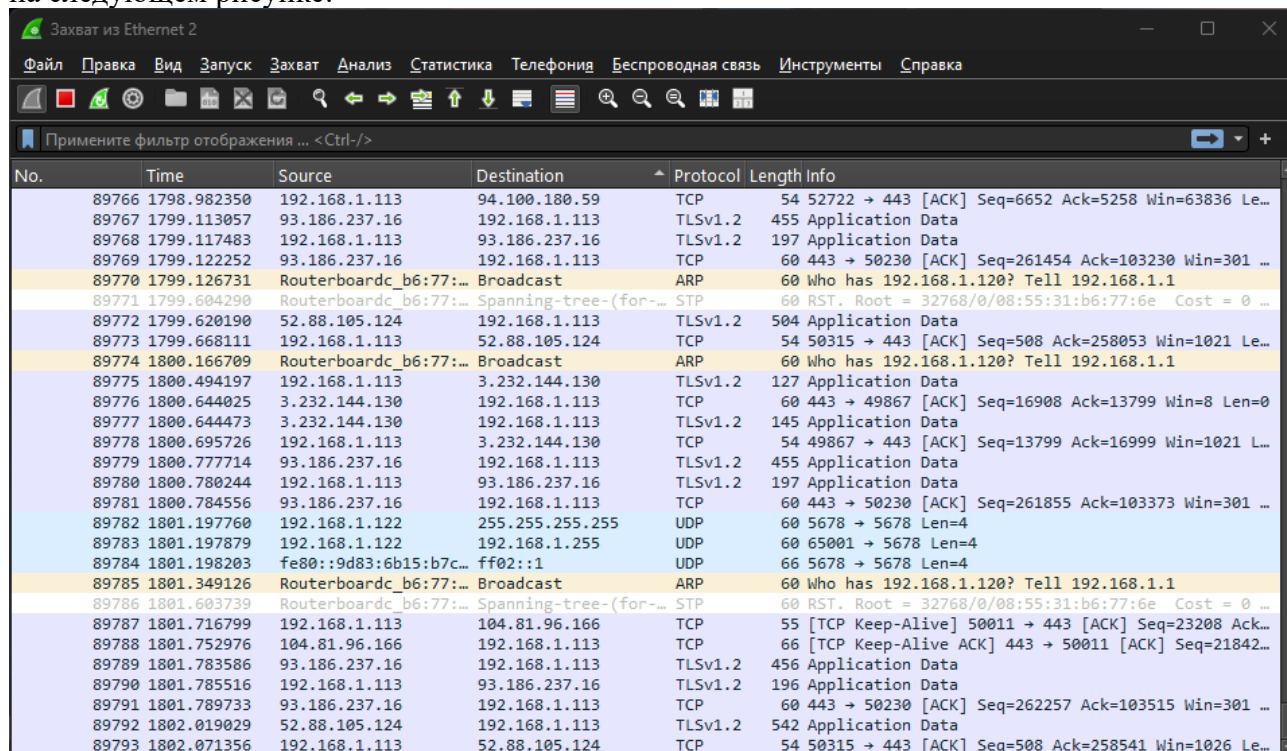
Как видим, тут есть как тело ответа с веб-страницей, так и описание самого ответа, в том числе его код и другая служебная информация.

Также используя Wireshark можно обнаружить IP-адрес своего компьютера, он находится в столбце Source (Отправитель) у отправляемого пакета и наоборот в столбце назначение (Destination) у принимаемого, в моем случае это 192.168.1.120, это IP-адрес внутри моей локальной подсети. Также можно посмотреть на IP-адрес сервера, куда отправляется запрос, в столбце назначение (Destination) у отправляемого пакета и наоборот в столбце назначение Source (Отправитель) у принимаемого, в данном случае сервер имеет IP-адрес 128.119.245.12.

Еще можно посмотреть время между принятием и отправкой пакета, для этого обратимся к столбцу time, по которому видно, что между запросом и ответом прошло около 120 мс (0.12 сек), что достаточно быстро т. к. принимаемый пакет достаточно маленький.

Сохраним пакеты, их можно будет найти в репозитории лабораторной или по следующей ссылке: github.com/DafterT/TKS_Labs/tree/main/Лабораторная_1

Wireshark позволяет анализировать не только HTTP-трафик, но и множество других, что видно на следующем рисунке:



No.	Time	Source	Destination	Protocol	Length	Info
89766	1798.982350	192.168.1.113	94.100.180.59	TCP	54	52722 → 443 [ACK] Seq=6652 Ack=5258 Win=63836 Le...
89767	1799.113057	93.186.237.16	192.168.1.113	TLSv1.2	455	Application Data
89768	1799.117483	192.168.1.113	93.186.237.16	TLSv1.2	197	Application Data
89769	1799.122252	93.186.237.16	192.168.1.113	TCP	60	443 → 50230 [ACK] Seq=261454 Ack=103230 Win=301 ...
89770	1799.126731	Routerboardc_b6:77:...	Broadcast	ARP	60	Who has 192.168.1.120? Tell 192.168.1.1
89771	1799.604290	Routerboardc_b6:77:...	Spanning-tree-(for-...	STP	60	RST. Root = 32768/0/08:55:31:b6:77:6e Cost = 0 ...
89772	1799.620190	52.88.105.124	192.168.1.113	TLSv1.2	504	Application Data
89773	1799.668111	192.168.1.113	52.88.105.124	TCP	54	50315 → 443 [ACK] Seq=508 Ack=258053 Win=1021 Le...
89774	1800.166709	Routerboardc_b6:77:...	Broadcast	ARP	60	Who has 192.168.1.120? Tell 192.168.1.1
89775	1800.494197	192.168.1.113	3.232.144.130	TLSv1.2	127	Application Data
89776	1800.644025	3.232.144.130	192.168.1.113	TCP	60	443 → 49867 [ACK] Seq=16908 Ack=13799 Win=8 Len=0
89777	1800.644473	3.232.144.130	192.168.1.113	TLSv1.2	145	Application Data
89778	1800.695726	192.168.1.113	3.232.144.130	TCP	54	49867 → 443 [ACK] Seq=13799 Ack=16999 Win=1021 L...
89779	1800.777714	93.186.237.16	192.168.1.113	TLSv1.2	455	Application Data
89780	1800.780244	192.168.1.113	93.186.237.16	TLSv1.2	197	Application Data
89781	1800.784556	93.186.237.16	192.168.1.113	TCP	60	443 → 50230 [ACK] Seq=261855 Ack=103373 Win=301 ...
89782	1801.197760	192.168.1.122	255.255.255.255	UDP	60	5678 → 5678 Len=4
89783	1801.197879	192.168.1.122	192.168.1.255	UDP	60	65001 → 5678 Len=4
89784	1801.198203	fe80::9d83:6b15:b7c...	ff02::1	UDP	66	5678 → 5678 Len=4
89785	1801.349126	Routerboardc_b6:77:...	Broadcast	ARP	60	Who has 192.168.1.120? Tell 192.168.1.1
89786	1801.603739	Routerboardc_b6:77:...	Spanning-tree-(for-...	STP	60	RST. Root = 32768/0/08:55:31:b6:77:6e Cost = 0 ...
89787	1801.716799	192.168.1.113	104.81.96.166	TCP	55	[TCP Keep-Alive] 50011 → 443 [ACK] Seq=23208 Ack...
89788	1801.752976	104.81.96.166	192.168.1.113	TCP	66	[TCP Keep-Alive ACK] 443 → 50011 [ACK] Seq=21842...
89789	1801.783586	93.186.237.16	192.168.1.113	TLSv1.2	456	Application Data
89790	1801.785516	192.168.1.113	93.186.237.16	TLSv1.2	196	Application Data
89791	1801.789733	93.186.237.16	192.168.1.113	TCP	60	443 → 50230 [ACK] Seq=262257 Ack=103515 Win=301 ...
89792	1802.019029	52.88.105.124	192.168.1.113	TLSv1.2	542	Application Data
89793	1802.071356	192.168.1.113	52.88.105.124	TCP	54	50315 → 443 [ACK] Seq=508 Ack=258541 Win=1026 Le...

Рис. 1.7. Окно Wireshark с различными пакетами.

Здесь можно заметить такие протоколы, как UDP, ARP, TCP и некоторые другие, этот список далеко не полный и Wireshark позволяет работать со множеством других интерфейсов, что будет рассмотрено в последующих лабораторных.

1.3. Вывод:

В ходе лабораторной работы была успешно изучена программа Wireshark, которая используется для анализа сетевого трафика. Мы научились захватывать пакеты данных, анализировать их структуру и извлекать полезную информацию о работе сетевых протоколов. На практическом примере с HTTP-запросом и ответом мы рассмотрели, как сетевые протоколы передают данные между устройствами. Было продемонстрировано, как данные инкапсулируются и передаются через различные уровни сетевой архитектуры, от прикладного до канального.

Мы также научились использовать фильтры Wireshark для отслеживания конкретных протоколов, что упрощает анализ большого объема трафика. Определение IP-адресов отправителей и получателей, а также анализ времени между запросами и ответами помогли глубже понять процессы взаимодействия в сети.

Таким образом, данная работа позволила на практике увидеть, как функционируют сетевые протоколы и как передаются данные в сети. Мы освоили базовые навыки работы с инструментами анализа сетевого трафика, что является важным шагом в изучении принципов

работы современных компьютерных сетей. Полученные знания помогут лучше понимать сетевые процессы и взаимодействие между устройствами в сети.

2. Приложение:

Ссылка на репозиторий с исходными кодами: https://github.com/DafterT/TKS_Labs