

Лабораторная работа № 3. Анализ трафика в Wireshark

3.1. Цель работы

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

3.2. Предварительные сведения

3.2.1. Анализатор протоколов Wireshark. Установка и запуск

Wireshark — анализатор трафика сетей на базе технологии Ethernet (<https://www.wireshark.org/>). По функциональности Wireshark аналогичен утилите tcpdump (<http://www.tcpdump.org/>), но имеет графический интерфейс. В основе используется библиотека Pcap.

Wireshark может использоваться для анализа и устранения неполадок в сети, определения угроз безопасности сети и их источников, выявления некорректной работы сетевых приложений в процессе их разработки, отладки реализации протоколов, для изучения внутренней структуры протоколов.

3.2.1.1. Установка и запуск Wireshark в ОС Windows

Рекомендуется для установки Wireshark в ОС Windows использовать менеджер пакетов Chocolatey (<https://chocolatey.org/>). Потребуется установить Wireshark и драйвер WinPcap:

```
choco install wireshark
choco install winpcap
```

Wireshark для сбора данных в реальном времени использует драйвер WinPcap (NPF), для работы которого требуются права администратора. При установке драйвера WinPcap через Chocolatey он устанавливается для автоматического запуска с полномочиями администратора и не требует дальнейшей настройки.

Если установка Wireshark и WinPcap проводилась вручную, то для запуска потребуется вручную запустить NPT из-под записи администратора:

```
runas /u:administrator "net start npf"
```

После этого можно работать с Wireshark.

Для остановки NPF из-под записи администратора следует ввести:

```
runas /u:administrator "net stop npf"
```

Для запуска драйвера NPF под администратором автоматически при старте системы следует из-под учётной записи администратора ввести:

```
sc config npf start=auto
```

Альтернативный вариант — воспользоваться Device Manager, выбрать View->Show hidden devices, открыть Non-Plug and Play Drivers. Затем настроить запуск NetGroup Packet Filter Driver.

3.2.1.2. Установка и запуск Wireshark в ОС типа Linux

Команды по установке Wireshark в ОС типа Linux зависят от используемого вами дистрибутива:

- Ubuntu:
`sudo apt-get install wireshark`
- Fedora (или CentOS):
`sudo dnf install wireshark`

При работе Wireshark для захвата пакетов требуются административные права. Можно установить ограничения для захвата пакетов для определённой группы пользователей.

Для произвольного дистрибутива Linux по сути необходимо установить права доступа к файлу `/usr/sbin/dumpcap`. Для этого требуется создать группу, например `wireshark`, и добавить в неё пользователя `user_name`, под которым вы работаете:

```
sudo -i
groupadd -s wireshark
gpasswd -a -G wireshark user_name
```

Здесь `user_name` — ваша учётная запись.

Затем измените параметры файла `/usr/sbin/dumpcap`:

```
sudo -i
chgrp wireshark /usr/bin/dumpcap
chmod 750 /usr/bin/dumpcap
```

После этого можно перелогиниться или временно добавить себя в новую группу:

```
newgrp wireshark
```

Для работающих под ОС Ubuntu требуется выполнить следующие действия:

```
sudo -i
apt-get install wireshark
apt-get install pcaputils
dpkg-reconfigure wireshark-common
groupadd wireshark
usermod -a -G wireshark user_name
```

После завершения этих действий можно перелогиниться или временно добавить себя в новую группу:

```
newgrp wireshark
```

3.2.1.3. Запуск Wireshark в MacOS

В MacOS и BSD для захвата пакетов необходим доступ для чтения к устройствам BPF в `/dev/bpf*`.

Возможное решение по запуску Wireshark с привилегиями администратора:

```
sudo /Applications/Wireshark.app/Contents/MacOS/Wireshark
```

или

```
sudo dseditgroup -o edit -a `whoami` -t user_name access_bpf
sudo "/Library/Application
↳ Support/Wireshark/ChmodBPF/ChmodBPF"
```

Поскольку в MacOS используется `devfs`, то изменения не сохраняются между перезагрузками системы.

3.2.2. Интерфейс Wireshark

После запуска Wireshark необходимо выбрать интерфейс для прослушивания входящего и исходящего трафика (рис. 3.1). В случае стационарного компьютера это может быть интерфейс сетевого адаптера, в случае ноутбука — интерфейс беспроводного адаптера (например, `wlan`).

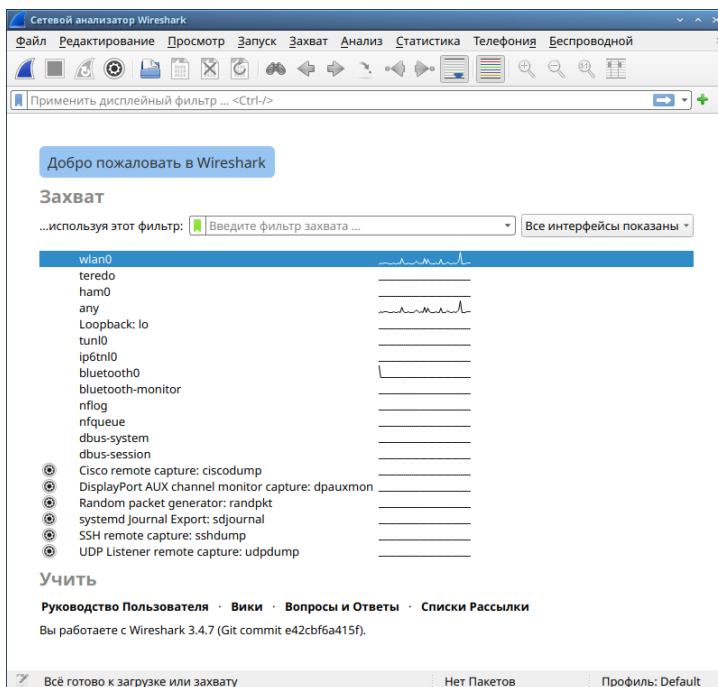


Рис. 3.1. Стартовое окно Wireshark. Выбор интерфейса для прослушивания

Основное меню Wireshark традиционно располагается в верхней части окна программы (рис. 3.1).

Основные пункты меню Wireshark:

- *File (Файл)* содержит пункты для открытия и объединения файлов захвата, вывода на печать, экспорта и сохранения файлов, выход из программы;
- *Edit (Редактирование)* содержит пункты поиска пакетов по файлу захвата, пункты привязки пакетов по времени, конфигурационные профили и параметры настройки программы;

- *View (Просмотр)* позволяет управлять отображением информации о захваченных пакетах;
- *Go (Запуск)* осуществляет навигацию по пакетам;
- *Capture (Захват)* позволяет запустить или остановить захват трафика, установить фильтры и параметры захвата;
- *Analyze (Анализ)* содержит элементы для обработки фильтров отображения информации о протоколах;
- *Statistics (Статистика)* позволяет отобразить различные статистические данные, включая сводку захваченных пакетов, отображение статистики иерархии протоколов и многое другое;
- *Telephony (Телефония)* позволяет отобразить различные статистические данные, связанные с телефонией, включая анализ медиафайлов, блок-схемы, статистику иерархии протоколов отображения;
- *Wireless (Беспроводной)* позволяет отобразить статистику о работе протоколов беспроводной связи Bluetooth и IEEE 802.11;
- *Tools (Инструменты)* содержит доступные в Wireshark инструменты, например по созданию правил списков контроля доступа межсетевых экранов;
- *Help (Помощь)* содержит локальную справку по программе и ссылки на онлайн ресурсы и материалы по работе с программой.

Под основным меню программы расположена панель инструментов с наиболее часто используемыми кнопками запуска, остановки и перезапуска захвата пакетов, настройки опций захвата, действий с файлом захвата, поиска и навигации по пакету и т.п.

Под панелью инструментов располагается строка редактирования фильтров отображения. В ней можно или просто указать протокол, информацию, по которому требуется проанализировать, или прописать более сложные правила отображения интересующей вас информации.

Под строкой фильтра располагается основное окно Wireshark, разделённое на три области (рис. 3.2).

В верхней части окна показана информация о пакетах, проходящих через прослушиваемый интерфейс (время, адреса источника и получателя пакета, задействованный протокол, размер пакета, информация о содержимом пакета). В нижней части окна даётся информация о пакете в виде набора шестнадцатиричных чисел — последовательности байтов. В средней части окна приводится панель сведений о пакете — пояснение по структуре передаваемых данных, о заголовках протоколов, инкапсулированных в пакет.

При выборе в средней части окна заголовка интересующего вас протокола или его части информация в нижней части окна подсвечивается, показывая её место в структуре пакета. И наоборот, отметив курсором часть байтов в нижней части окна, в средней части можно получить пояснение об их назначении в пакете.

Из примера, приведённого на рис. 3.3, можно увидеть, что анализируется пакет, переданный с локального хоста во внешнюю сеть по протоколу TCP; в пакет включена информация протокола Ethernet II, содержащая в частности, MAC-адрес источника информации; в нижней части выделены те байты, которые содержат информацию о MAC-адресе.

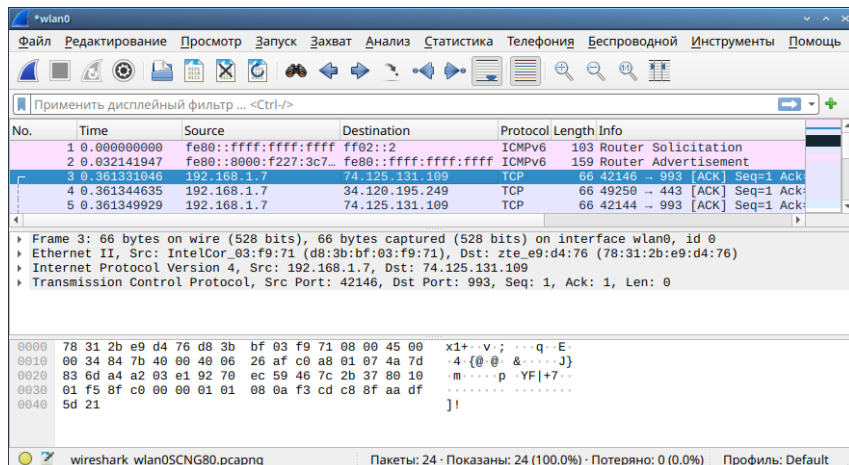


Рис. 3.2. Рабочая область Wireshark

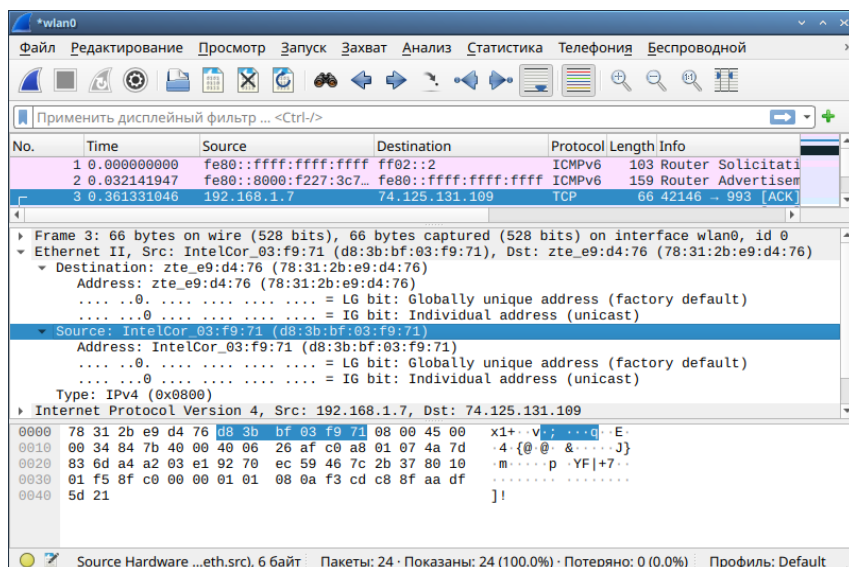


Рис. 3.3. Пример определения физического адреса источника передаваемой информации

3.2.3. Трёхступенчатый handshake TCP

На транспортном уровне семейства протоколов TCP/IP применяются два основных протокола — ориентированный на соединение протокол TCP (Transmission Control Protocol) и не требующий соединения протокол UDP (User Datagram Protocol).

Важной концепцией служб транспортного уровня семейства протоколов TCP/IP является концепция портов, представляющих собой 16-битный номер и идентифицирующих службу прикладного уровня стека протоколов TCP/IP. С номерами портов, соотнесёнными с протоколами и службами, можно ознакомиться на сайте Internet Assigned Numbers Authority (IANA) — <https://www.iana.org/>.

Протокол UDP используют приложения, которым требуется передавать датаграммы последовательно, например, протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP), служба именования доменов (Domain Name Service, DNS), простой протокол управления сетью (Simple Network Management Protocol, SNMP) и др.

Протокол TCP имеет средства управления потоком и коррекции ошибок, ориентирован на установление соединения до начала передачи данных.

Установление связи клиент-сервер в TCP осуществляется в три этапа (трёхступенчатый handshake) (рис. 3.4).

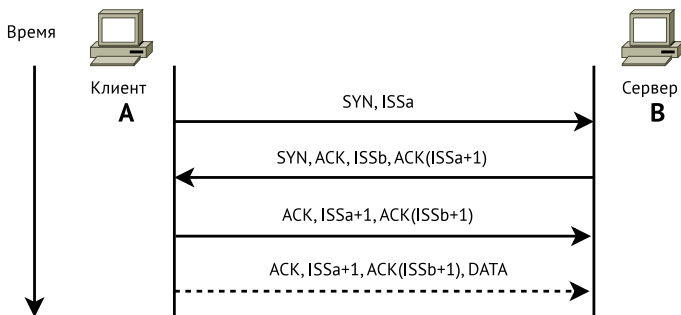


Рис. 3.4. Трёхступенчатый handshake TCP

Пусть хост А создаёт соединение с хостом В.

1. *Режим активного доступа (Active Open)*. Клиент посылает сообщение **SYN, ISSa**, т.е. в передаваемом сообщении установлен бит SYN (Synchronize Sequence Number), а в поле *Порядковый номер (Sequence Number)* — начальное 32-битное значение ISSa (Initial Sequence Number).
2. *Режим пассивного доступа (Passive Open)*. Сервер откликается, посылая сообщение **SYN, ACK, ISSb, ACK(ISSa+1)**, т.е. установлены биты SYN и ACK; в поле *Порядковый номер (Sequence Number)* хостом В устанавливается начальное значение счётчика — ISSb; поле *Номер подтверждения (Acknowledgment Number)* содержит значение ISSa, полученное в первом пакете от хоста А и увеличенное на единицу.

3. *Завершение рукопожатия.* Клиент отправляет подтверждение получения SYN-сегмента от сервера с идентификатором, равным $ISN(\text{сервера})+1$: $ACK, ISSb+1, ACK(ISSb+1)$. В этом пакете установлен бит ACK, поле *Порядковый номер (Sequence Number)* содержит $ISSa+1$, поле *Номер подтверждения (Acknowledgment Number)* содержит значение $ISSb+1$. Псылкой этого пакета заканчивается трёхступенчатый handshake, и TCP-соединение считается установленным.
4. Теперь клиент может посылать пакеты с данными на сервер по только что созданному виртуальному TCP-каналу: $ACK, ISSa+1, ACK(ISSb+1); DATA$.
Из рассмотренной выше схемы создания TCP-соединения видно, что единственными идентификаторами TCP-абонентов и TCP-соединения являются два 32-битных параметра *Порядковый номер (Sequence Number)* и *Номер подтверждения (Acknowledgment Number)*.

3.2.4. Протокол QUIC

В сети Интернет для обеспечения транспорта передаваемых по HTTP данных может применяться протокол QUIC (Quick UDP Internet Connections), разработанный компанией Google и стандартизованный в RFC-9000 в 2021 г. Этот протокол позволяет мультиплексировать несколько потоков данных между двумя компьютерами, работая поверх протокола UDP, и содержит возможности шифрования, эквивалентные TLS и SSL.

QUIC является протоколом с установлением соединения, обеспечивающим взаимодействие между клиентом и сервером с сохранением состояния.

Рукопожатие QUIC объединяет согласование криптографических и транспортных параметров. Рукопожатие в QUIC структурировано таким образом, чтобы можно было как можно скорее обменяться данными приложения.

Оконечные устройства взаимодействуют, обмениваясь пакетами QUIC. Большинство пакетов содержат кадры, которые несут управляющую информацию и данные приложения между оконечными устройствами. QUIC проверяет подлинность каждого пакета целиком и шифрует каждый пакет настолько, насколько это возможно. Пакеты QUIC передаются в дейтаграммах UDP, чтобы облегчить развёртывание в существующих системах и сетях.

Протоколы приложений обмениваются информацией через соединение QUIC через потоки, которые представляют собой упорядоченные последовательности байтов. Могут быть созданы два типа потоков: двунаправленные потоки, которые позволяют обоим конечным точкам отправлять данные; и однонаправленные потоки, которые позволяют одной конечной точке отправлять данные.

Кроме того, QUIC обеспечивает необходимую обратную связь для реализации надёжной доставки и контроля перегрузки. Также соединения QUIC не привязаны строго к одному сетевому пути. При переносе подключений используются идентификаторы подключения, чтобы позволить подключениям переходить на новый сетевой путь.

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.
2. Определите MAC-адреса сетевых интерфейсов на вашем компьютере. Подтвердите свой ответ скриншотом.
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.

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.

3.3.2.2. Порядок выполнения работы

1. Установите на вашем устройстве Wireshark.
2. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата трафика.
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).
4. На вашем устройстве в консоли с помощью команды `ping` адрес_шлюза пропингуйте шлюз по умолчанию. Для остановки процесса используйте комбинацию клавиш  или изначально при помощи параметров команды `ping` задайте число сообщений эхо-запроса.

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.
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.
 - 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.
7. Изучите кадры данных протокола ARP. Изучите данные в полях заголовка Ethernet II.
8. Начните новый процесс захвата трафика в Wireshark. На вашем устройстве в консоли пропируйте по имени какой-нибудь известный вам адрес, например `ping rdn.ru`.
9. В Wireshark остановите захват трафика. Изучите запросы и ответы протоколов ARP и ICMP. Определите MAC-адреса источника и получателя, определите тип MAC-адресов.

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.
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.
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.
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.

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.
6. Остановите 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.

3.3.4.2. Порядок выполнения работы

1. Запустите Wireshark. Выберите активный на вашем устройстве сетевой интерфейс. Убедитесь, что начался процесс захвата 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.
4. В Wireshark в меню «Статистика» выберете «График Потока». В отчёте приведите пояснения по изменениям значений соответствующих сообщений при установлении соединения по TCP.
5. Остановите захват трафика в Wireshark.

3.4. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
 - скриншоты (снимки экрана), фиксирующие выполнение лабораторной работы;
 - пояснения по отображаемой информации согласно заданию
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.