МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Гродненский государственный университет имени Янки Купалы»

Факультет математики и информатики

Кафедра современных технологий программирования

КУСТАНОВИЧ АРТУР СЕРГЕЕВИЧ

**Разработка программы для анализа личного сетевого трафика**

**“Packet Sniffer”**

Курсовая работа

студент 2 курса специальности

1-40 01 01 «Программное обеспечение информационных технологий»

дневной формы получения образования

Научный руководитель

??????????????????,

доцент кафедры

современных технологий

программирования

Гродно, 2022

**РЕЗЮМЕ**

Кустанович Артур Сергеевич

Тема курсовой работы

**«Разработка программы для анализа личного сетевого трафика»**

Работа содержит: 24 страницы, 9 рисунков, 8 использованных источников литературы.

Ключевые слова: Java, Java Swing, WireShark, Sniffer, libpcap, WinPcap, jna, api, pcap4j, IP4v, IP6v, DNS, сеть, пакеты внутри сети, анализ

Цель курсовой работы – Разработка приложения, предназначенного, для перехвата пакетов внутри сети и их же анализ в отдельном файле. Приложение предусматривает реализацию механизма выбора файлов, записи файлов, поиска пакетов внутри сети.(разработано в целях анализа)

Предусмотрено решение следующих задач:

1. Выполнить сравнительный анализ аналогов анализаторов пакетов внутри сети;
2. Проектирование приложения на языке программирования Java;
3. Разработка и программная реализация интерфейса приложения.

Объект исследования – анализ трафика.

Предметом исследования настоящей работы является реализация оконного приложения с интуитивно понятным интерфейсом для отображения информации трафика, выбора исполняемых адаптеров и добавления файлов.

В работе были использованы следующие методы: анализ, синтез, системный подход.

Данное приложение разработано с целью предоставить пользователю простое и понятное приложения для анализа сетевого трафика. Это особый инструментарий, позволяющий тестироващикам перехватывать и анализировать запросы, которые проходят через них. Подобные технологии очень полезны в ситуации, когда пользователю нужно получить сведения или сконфигурировать необходимый ответ от сервера.

SUMMARY

Kustanovich Artur Sergeevich

Theme of the course work

"Development of programs for network traffic analysis"

contains: 24 pages, 9 figures, 8 references.

Keywords: Java, Java Swing, WireShark, Sniffer, libpcap, WinPcap, jna, api, pcap4j, IP4v, IP6v, DNS, network, intranet packets, analysis

The purpose of the course work is to develop applications designed to intercept packets within the network and analyze them in a separate file. The application provides for the use of a mechanism for selecting files, writing files, searching for packets within the network. (Designed for analysis purposes)

The solution of the following tasks is provided:

1. Compare packet analyzers within the network;
2. Designing applications in the Java programming language;
3. Development and software implementation of the application interface.

The object of the study is traffic analysis.

The subject of this study is the implementation of a window application with an intuitive interface for displaying traffic information, selecting executable adapters and adding files.

The following methods were used in the work: analysis, synthesis, system approach.

This application is designed to provide the user with a simple and intuitive application for analyzing network traffic. This is a special tool that allows testers to intercept and analyze requests that pass through them. Such technologies are very useful in a situation when the user needs to get information or configure the necessary response from the server.

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 6](#_Toc87812843)

[**ГЛАВА 1** 8](#_Toc87812844)

[**АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 8](#_Toc87812845)

[1.1. Постановка задач для реализации приложения для распределения доступа 8](#_Toc87812846)

[1.2. Обзор существующих решений для распределения доступа 10](#_Toc87812847)

[1.3. Выводы по главе 1 13](#_Toc87812848)

[**ГЛАВА 2** 14](#_Toc87812849)

[**ПРОЕКТИРОВАНИЕ СИСТЕМЫ** 14](#_Toc87812850)

[2.1. Проектирование функций приложения 14](#_Toc87812851)

[2.2. Построение графического интерфейса приложения 15](#_Toc87812853)

[2.3. Выводы по главе 2 16](#_Toc87812854)

[**ГЛАВА 3** 17](#_Toc87812855)

[**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ДЛЯ РАСПРЕДЕЛЕНИЯ ДОСТУПА К ФАЙЛАМ** 17](#_Toc87812856)

[3.1. Выбор программных средств 17](#_Toc87812857)

[3.2. Описание структуры проекта 18](#_Toc87812858)

[3.3. Работа с приложением 20](#_Toc87812859)

[3.5. Открытие файла после дампа с помощью WireShark 21](#_Toc87812859)

[3.4. Выводы по главе 3 22](#_Toc87812860)

[ЗАКЛЮЧЕНИЕ 23](#_Toc87812861)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc87812862)

ВВЕДЕНИЕ

Анализатор трафика — это программа для перехвата и анализа сетевого трафика(своего или чужого) позволяющий тестироващикам перехватывать и анализировать запросы, которые проходят через них. Подобные технологии очень полезны в ситуации, когда пользователю нужно получить сведения или сконфигурировать необходимый ответ от сервера.

Цель данной работы – разработка приложения, предназначенного, для перехвата пакетов внутри сети и их же анализ в отдельном файле. Приложение предусматривает реализацию механизма выбора файлов, записи файлов, поиска пакетов внутри сети.(разработано в целях анализа)

Предусмотрено решение следующих задач:

1. Выполнить сравнительный анализ аналогов анализаторов пакетов внутри сети;
2. Проектирование приложения на языке программирования Java;
3. Разработка и программная реализация интерфейса приложения.

В первой главе «Анализ предметной области» дается краткий обзор предмета исследования и выбранного языка программирования, описываются изучение проблемы и возможные решения поставленной задачи.

Во второй главе «Проектирование утилиты, связанной с распределением доступа» подробно описываются механизмы построения данного проекта, раскрываются проблемы, с которыми можно столкнуться при реализации.

«Реализация утилиты, связанной с распределением доступа» – последняя глава курсовой работы. В ней приведены части исходного кода программы, которые необходимы для реализации механизмов, определенных в главе «Проектирование утилиты, связанной с распределением доступа».

В приложении к пояснительной записке прикреплены основные части кода. Также в этом разделе приведены снимки экрана, показывающие работу программы.

ГЛАВА 1

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Постановка задач для реализации анализатора трафика

Задача курсового проекта состоит в разработке анализа сетевого трафика.

Для подобного рода манипуляций необходима клиентская программа, так называемая утилита. Отличие от обыкновенного изменения уровня доступа через консоль здесь в том, что работать приходиться в гораздо более приятном специализированном приложении. Утилита предоставляет такие возможности как: просмотреть найденные пакеты в таблице, вывести информацию отдельно и создания dump файла.

Исходя из всего вышесказанного, требуется реализовать приложение, данное приложение будет иметь одну панель.

1.2. Обзор существующих решений анализа сетевого трафика

**1. SolarWinds Network Bandwidth Analyzer**

Данное решение позиционируется производителем как программный пакет из двух продуктов — Network Performance Monitor (базовое решение) и NetFlow Traffic Analyzer (модульное расширение). Как заявляется, они имеют схожие, но все же отличающиеся функциональные возможности для анализа сетевого трафика, дополняющие друг друга при совместном использовании сразу двух продуктов.  
  
Network Performance Monitor, как следует из названия, осуществляет мониторинг производительности сети и станет для вас заманчивым выбором, если вы хотите получить общее представление о том, что происходит в вашей сети. Покупая это решение, вы платите за возможность контролировать общую работоспособность вашей сети: опираясь на огромное количество статистических данных, таких как скорость и надежность передачи данных и пакетов, в большинстве случаев вы сможете быстро идентифицировать неисправности в работе вашей сети. А продвинутые интеллектуальные возможности программы по выявлению потенциальных проблем и широкие возможности по визуальному представлению результатов в виде таблиц и графиков с четкими предупреждениями о возможных проблемах, еще больше облегчат эту работу.  
  
Модульное расширение NetFlow Traffic Analyzer больше сконцентрировано на анализе самого трафика. В то время, как функциональность базового программного решения Network Performance Monitor больше предназначена для получения общего представления о производительности сети, в NetFlow Traffic Analyzer фокус внимания направлен на более детальный анализ процессов, происходящих в сети. В частности, эта часть программного пакета позволит проанализировать перегрузки или аномальные скачки полосы пропускания и предоставит статистику, отсортированную по пользователям, протоколам или приложениям. Обратите внимание, что данная программа доступна только для среды Windows.

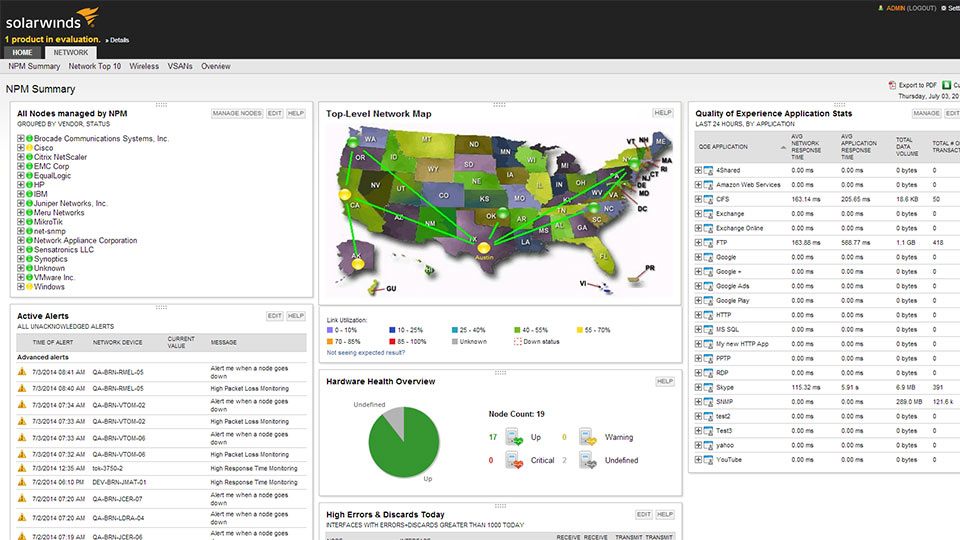
****

Рис.1.1 Интерфейс программы SolarWinds

**Достоинства:**

* расширенный функционал;
* огромное количество статистических данных;
* идентификация неисправности в работе сети
* интеграция с Linux;

**Недостатки:**

* только платная версия;
* сложность начальной настройки.

**2.**  **tcpdump**

Анализатор трафика tcpdump выглядит как некий древний инструмент, и, если уж быть до конца откровенным, с точки зрения функциональности работает о так же. Несмотря на то, что со своей работой он справляется и справляется хорошо, причем используя для этого минимум системных ресурсов, насколько это вообще возможно, многим современным специалистам будет сложно разобраться в огромном количестве «сухих» таблиц с данными. Но бывают в жизни ситуации, когда использование столь обрезанных и неприхотливых к ресурсам решений может быть полезно.

Изначально программное решение tcpdump разработано для среды \*NIX, но на данный момент он также работает с несколькими портами Windows. Он обладает всей базовой функциональностью, которую вы ожидаете увидеть в любом анализаторе трафика — захват, запись и т.д

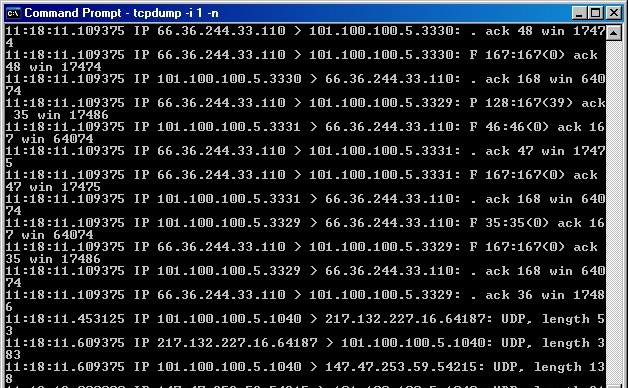


Рис.1.2. Интерфейс программы tcpdump

**Достоинства:**

* простой интерфейс;
* абсолютно не требователен к ресурсам компьютера;
* базовые фукнции, запись и захват;
* интеграция с Linux;

**Недостатки:**

* современно устаревший;
* малофункциональный

**3.**  **Kismet**

Анализатор трафика Kismet — не просто анализирует сетевой трафик, он предоставляет гораздо более расширенные функциональные возможности. К примеру, он способен проводить анализ трафика скрытых сетей и даже беспроводных сетей, которые не транслируют свой идентификатор SSID! Подобный инструмент для анализа трафика может быть чрезвычайно полезен, когда в вашей беспроводной сети есть что-то, вызывающее проблемы, но быстро найти их источник вы не можете. Kismet поможет вам обнаружить неавторизированную сеть или точку доступа, которая работает, но имеет не совсем правильные настройки.



Рис.1.3. Интерфейс программы Kismet

**Достоинства:**

* высокая производительность;
* многофункциональность;
* быстрая скорость исполнения;
* анализ беспроводного трафика
* интеграция Linux, macOS

**Недостатки:**

* чрезвычайно высокая стоимость;
* сложность начальной настройки.

1.3. Выводы по главе 1

В первой главе были рассмотрены наиболее подходящие решения разработки программы для анализа сетевого трафика. Были разобраны несколько программ таких, как «SolarWinds», «tcpdump», «Kismet». Каждое решение было описано, для каждого решения были отмечены как достоинства, так и недостатки. Приведённые программы имеют схожий функционал:

1. Быстрая скорость исполнения;
2. Простой и интуитивно понятный интерфейс;
3. Наличие основных функций просмотра, записи и захвата;
4. Наличие статистических данных

К Перечисленным минусам я свою разработку не приписываю, она абсолютно для базового анализа сетевого трафика, а главное безобидного.

ГЛАВА 2

ПРОЕКТИРОВАНИЕ ПРОГРАММЫ

2.1. Проектирование функций приложения

В первой главе были разобраны реализации решений по данной тематике. Сейчас необходимо разработать набор функций программы, которые пользователь сможет использовать. Так как наша цель - создание оконного приложения для клиентской, значит, необходимо разработать набор функций на основе реализованных проектов в данной сфере, для того чтобы пользователю было интуитивно понятно, что и как работает в приложении.

Прежде всего, следует отметить, что программа должна быть приятным для конечного пользователя.

2.2. Построение графического интерфейса приложения

Для данного приложения необходимо реализовать интерфейс пользователя: форма для основной работы с функционалом. Ее интерфейс содержат такие элементы как – список доступных адаптеров, кнопки для запуска и остановки поиска пакетов, для записи в дамп файла, очистки таблицы программа представлена на рисунке 2.1.

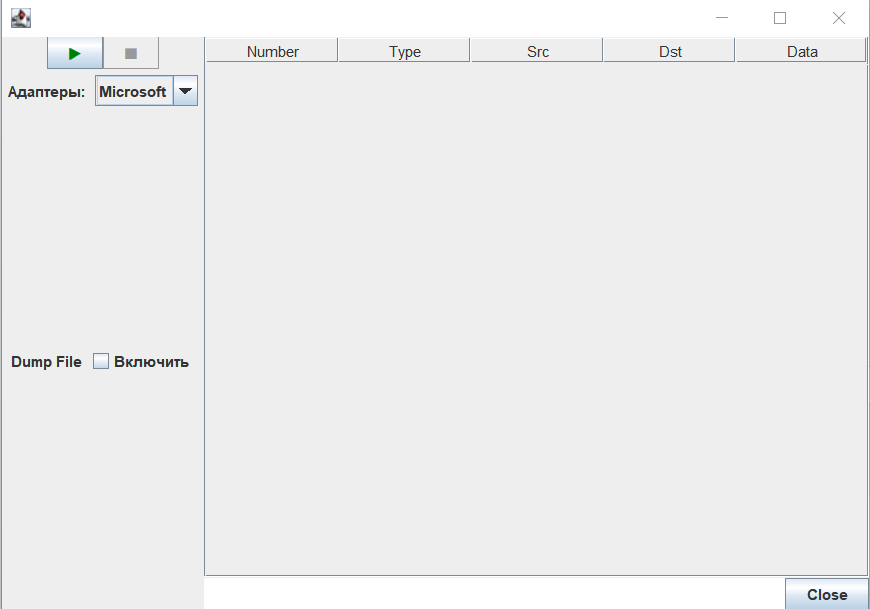


Рис.2.1. Макет графического интерфейса

Первое, и единственное, что увидит пользователь, это основная форма Здесь от может выбрать доступные адаптеры. Также, при клике по кнопке “Dump File - Включить”, он может выбрать в какой файл записать все захваченные пакеты, после выбора всех необходимых функций, в конечном итоге пользователь должен нажать на запуск основной функции(поиска пакетов) - кнопка Пуск.

2.3. Выводы по главе 2

Таким образом, в процессе проектирования были составлены требования к функциональным характеристикам. Из них стало известно, что приложение должно быть удобным для конечного пользователя, иметь возможность просмотреть информацию в реальном времени, записывать их в файл, проводить необходимые операции над ними.

Была намечена определена архитектура приложения и выделены преимущества в ее использовании, заключающиеся в улучшении структуризации кода, упрощении поддержки, отладки и тестирования. На основе анализа существующих решений, проведенного в главе 1, были спроектированы интерфейсы пользователя.

ГЛАВА 3

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ДЛЯ РАСПРЕДЕЛЕНИЯ ДОСТУПА К ФАЙЛАМ

3.1. Выбор программных средств

В качестве средств реализации был выбран язык программирования Java.

Java — строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems. Разработка ведётся сообществом, организованным через Java Community Process; язык и основные реализующие его технологии распространяются по лицензии GPL.

Java относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и C#.

Для реализации интерфейса был выбрана библиотека Swing.

Swing в Java является частью базового класса Java, который является независимым от платформы. Он используется для создания оконных приложений и включает в себя такие компоненты, как кнопка, полоса прокрутки, текстовое поле и т. д. .

Swing в Java – это легкий инструментарий с графическим интерфейсом, который имеет широкий спектр виджетов для создания оптимизированных оконных приложений. Это часть JFC (Java Foundation Classes). Он построен на основе AWT API и полностью написан на Java. Он не зависит от платформы в отличие от AWT и имеет легкие компоненты.Создавать приложения становится проще, поскольку у нас уже есть компоненты GUI, такие как кнопка, флажок и т. д.

3.2. Описание структуры проекта

Основой приложения является класс class PacketSniffer(рисунок 3.1), который содержит все необходимые методы:

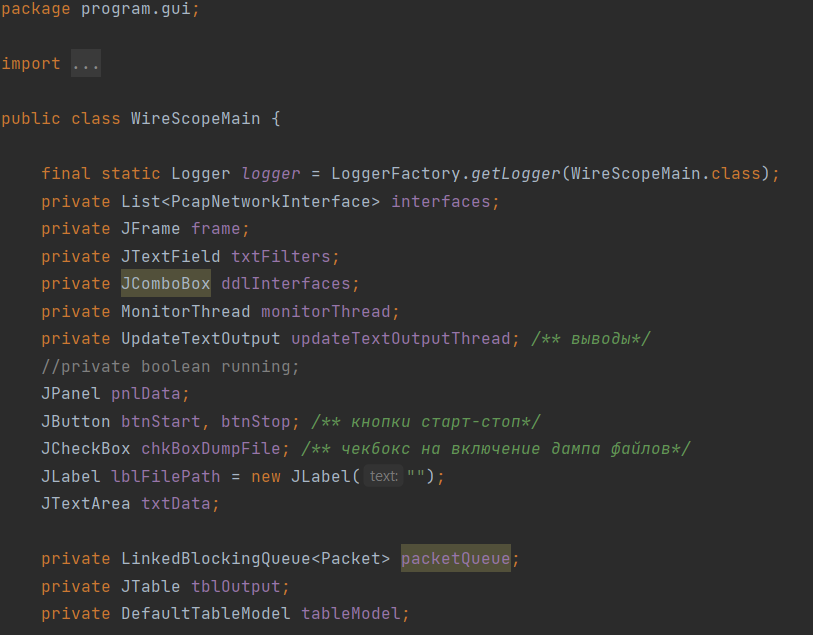


Рис.3.1. Главный класс приложения

В данном классе содержаться все необходимые для работы приложения методы и поля. Поле UpdateTextOutput, например, отвечает за весь вывод пакетов в программе, со всей информацией. Поле MonitorThread за окончательный вывод после окончания процесса поиска пакетов – количество захваченных пакетов, количество записанных пакетов.

3.3. Работа с приложением

С учетом требований, перечисленных в главе 2, было спроектировано приложение, скриншоты которого представлены на рисунках 3.2 – 3.3.

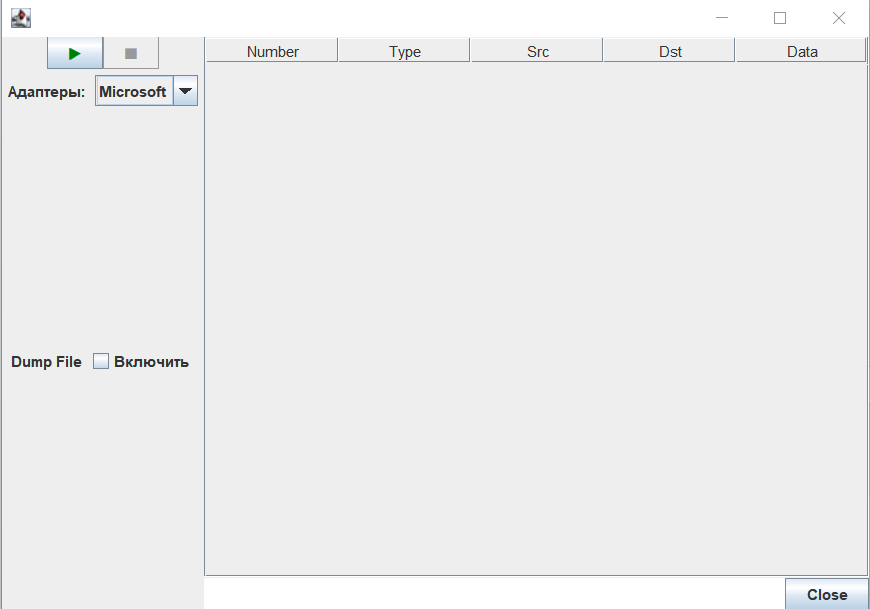


Рисунок 3.2. - главная страница приложения

Главная страница приложения(рис. 3.2) включает в себя просмотр всех захваченных пакетов, а также кнопки для управления. Кнопка «Start» отвечает за запуск основной функции(поиска пакетов), кнопка “Stop” отвечает за полную остановку процессов, кнопка “Close” отвечает за очистку таблицы.

После нажатия кнопки «Start» с заранее выбранным нужным адаптером, начнется основной процесс поиска трафика и запись в таблицу.

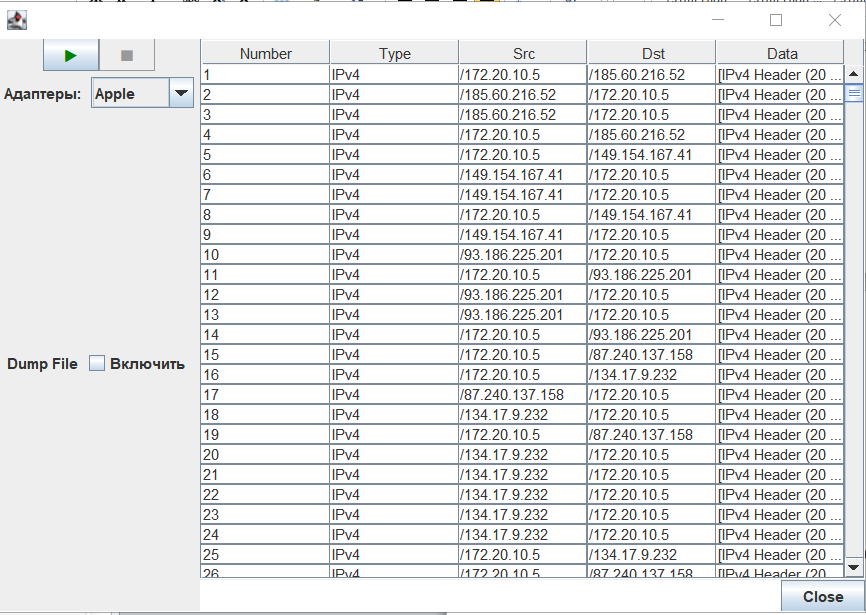


Рисунок 3.3 – отображение информации о захваченном пакете

Приложение содержит в себе все необходимые функции поиска, захвата, записи и конечный пользователь может в полной мере управлять ими. Приложение имеет интуитивно понятный интерфейс и качественную обработку ошибок, которые могу возникнуть в процессе работы с системой Windows.

* 1. Работа функции дамп файла

После того как пользователь захочет записать все захваченные пакеты, он обязан нажать на кнопку “Dump File – Включить”, перед ним откроется окно выбора файла с расширением .pcap, главное чтобы он заранее был создан, неважно пустой он или записан с какой то информацией

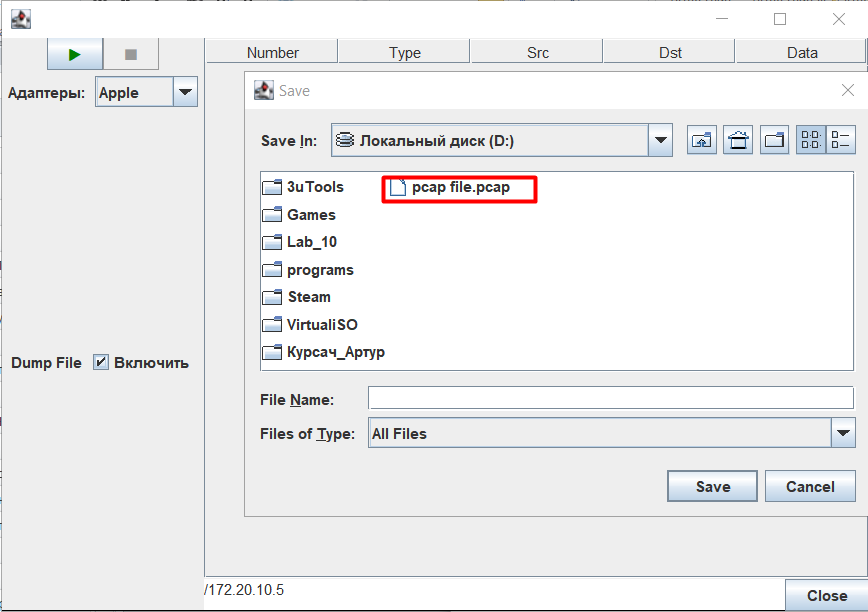


Рисунок 3.4 – запись всех пакетов на отдельный файл

После того как пользователь выбрал нужный файл, ему необходимо нажать на кнопку “Save” и запустить процесс поиска пакетов сетевого трафика.

* 1. Открытие файла после дампа

После завершения поиска, захвата и записи, и дампа файла

Файл полностью записан с нужными пакетами, которая захватила программа. Для открытия файла и дальнейшей работой с пакетами, а если точнее полным анализом мы воспользуемся программой WireShark и откроем ДампФайл

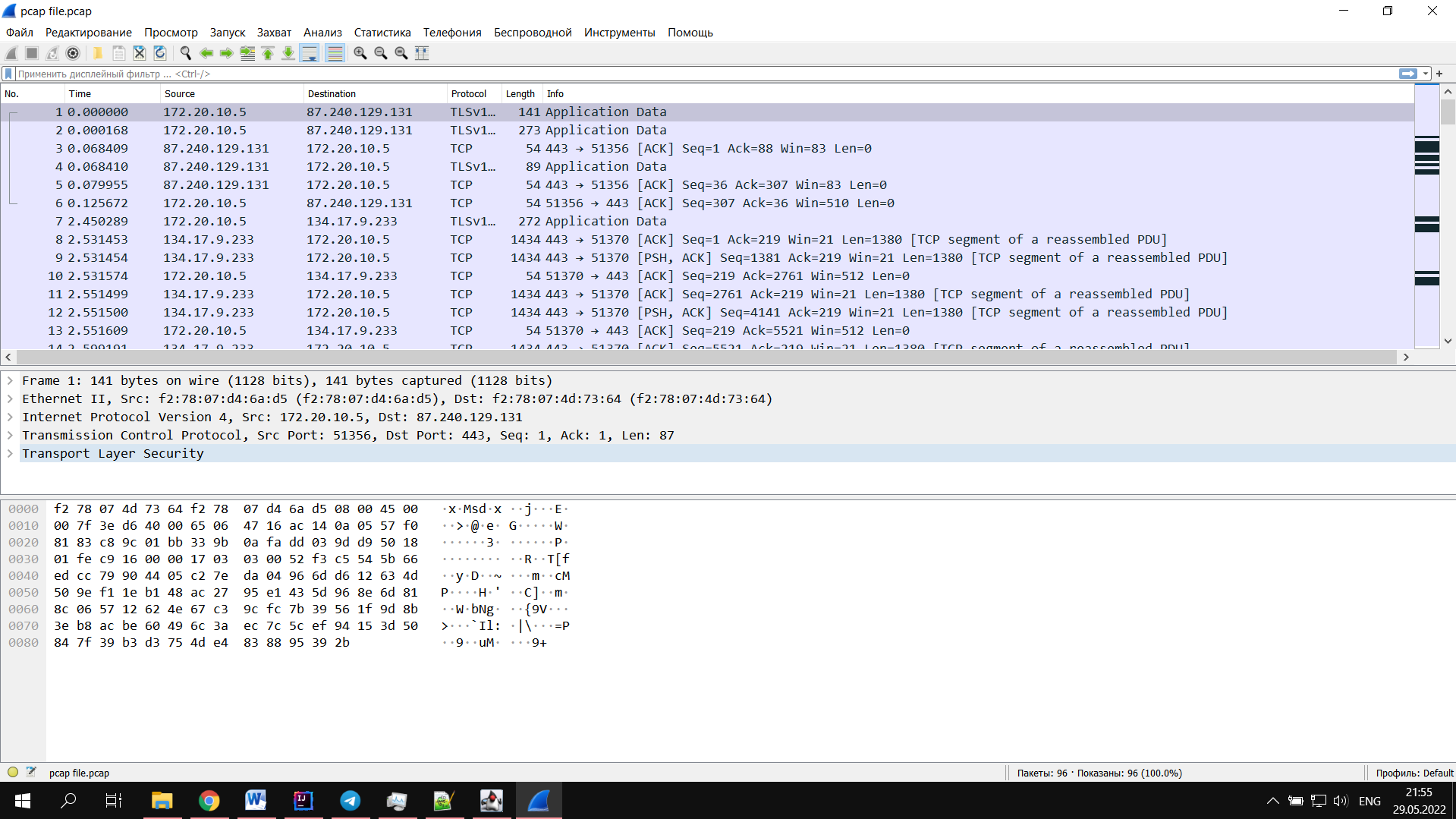


Рисунок 3.5 – открытие ДампФайла с помощью программы WireShark

3.6. Выводы по главе 3

На основании результатов проектирования приложения в главе 2, для его разработки были выбраны оптимальные основные средства: язык программирования Java для работы с логикой приложения. Для работы с интерфейсом программы была выбрана библиотека Swing. Для работы самих пакетов были выбраны библиотеки WinPcap, pcap4j, libpcap. Приложение было разделено на компоненты, каждый из которых находится в собственной зоне ответственности. В ходе работы была успешно реализована программная часть приложения, а также разработан пользовательский интерфейс. Основным требованием к дизайну являлась его интуитивная понятность. Результат работы представлен на скриншотах выше.

ЗАКЛЮЧЕНИЕ

Цель курсовой работы заключалась в разработке программы для анализа сетевого трафика, предназначенного, для поиска, захвата, вывода и записи. Приложение предусматривает реализацию механизма выбора файлов и функции записи/захват.

Результатом данной работы является полноценное приложение «Анализатор сетевого трафика». Приложение позволяет беспрепятственно просмотреть информацию о пакетах, а также работать с ними. Разработан простой и удобный пользовательский интерфейс. В ходе разработки данного приложения была изучена предметная область данной темы, а именно проведена исследовательская работа по изучению основных подходов к разработке оконного приложения, а конкретно аналогов анализатора сетевого трафика. Оценены основные способы воплощения данного проекта в готовое решение. Использованы технологии такие, как язык программирования Java, библиотека Swing.

Был реализован следующий функционал:

* просмотр пакетов в таблице;
* возможность дамп файла;
* возможность захватить различные типы пакетов.

На основе данного заключения можно сказать, что представленное приложение будет эффективным помощником пользователям, которым нужно получить сведения или сконфигурировать необходимый ответ от сервера

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Культин, Н., изучаем Swing в задачах и примерах// Н. Культин. – М.: Издательский дом «Питер», 2016. -352с.
2. Шилдт, Г., Java: полное руководство// Г. Шилдт. – М.: Издательский дом «Питер», 2016. – 572с.
3. https://prog-top.net/utilits – сборник программ [Электронный ресурс] – Режим доступа: https://prog-top.net/utilits – Дата доступа: 22.11.2021
4. https://www.informatique-mania.com – сборник программ [Электронный ресурс] – Режим доступа: https://www.informatique-mania.com – Дата доступа: 22.11.2021
5. https://www.comss.ru – сборник программ [Электронный ресурс] – Режим доступа: https://www.comss.ru – Дата доступа: 22.11.2021
6. https://metanit.com/java – Java [Электронный ресурс] – Режим доступа: https://metanit.com/java – Дата доступа: 25.04.2022
7. <https://kaitoy.github.io/pcap4j> - Uses of interface [Электроный ресурс]
8. <https://guthub.com> – [Электронный ресурс]