

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 23.Б15-мм

Мiminet: добавить отображение РСАР-файла для маршрутизатора, концентратора и коммутатора

Заикин Владимир Алексеевич

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
старший преподаватель кафедры системного программирования, Зеленчук И. В.

Санкт-Петербург
2025

Оглавление

Введение	3
1. Обзор	4
1.1. Используемые технологии	4
1.2. Существующее решение для хоста и сервера	5
1.3. Результат обзора	6
2. Постановка задачи	7
3. Описание решения	8
3.1. Переадресация	8
3.2. Изменение конфигурационного меню	8
Заключение	10
Список литературы	11

Введение

В рамках курса «Компьютерные сети» на Математико-механическом факультете Санкт-Петербургского государственного университета используется веб-эмулятор Miminet¹, разработанный на базе Mininet [4]. Mininet – сетевой эмулятор, который создает сеть из виртуальных хостов, коммутаторов и других устройств сети, используя механизмы ядра Linux. Таким образом, Miminet представляет веб-интерфейс с возможностью в визуальном режиме строить топологию сети, которая при помощи библиотеки Mininet будет эмулироваться в Linux.

Miminet – веб-тренажёр, предназначенный для обучения и проверки знаний в области компьютерных сетей. Для понимания структуры различных сетевых протоколов и получения подробной информации о проходящих пакетах в Miminet встроен Mimishark – анализатор трафика, аналогичный Wireshark [6], с веб-интерфейсом. Wireshark для захвата пакетов использует библиотеку libpcap [8]. Mimishark отображает данные, хранящиеся в PCAP-файлах², получаемых после эмуляции сети на сервере.

Mimishark поддерживает анализ трафика только для таких элементов сети, как хост и сервер [1]. В данной работе будет реализована поддержка Mimishark для маршрутизатора, концентратора и коммутатора.

¹Официальный сайт Miminet – URL: <https://miminet.ru> (дата обращения: 28 марта 2025 г.).

²PCAP-файл (англ. Picture Capture) – файл данных сетевого пакета, который содержит собранные проводные или беспроводные сетевые данные.

1. Обзор

Обзор проводится с целью ознакомления с существующим решением и составления четкого перечня задач.

1.1. Используемые технологии

В данной части обзора описываются технологии, которые используются в существующем решении.

- RabbitMQ [5] – брокер сообщений, который служит посредником для обмена информацией между различными системами. В проекте используется для асинхронного обмена сообщениями, обеспечивая взаимодействие между фронтендом и бэкендом.
- Python – язык программирования, применяемый для веб-разработки, анализа данных и автоматизации. В проекте используется для фронтенда (веб-интерфейс и серверная часть для работы с базой данных) и бэкенда (эмулятор сетей Mininet).
- Flask [2] – фреймворк для разработки веб-приложений и API на языке Python. В проекте Flask используется для обработки HTTP-запросов и взаимодействия с базой данных.
- Jinja2 [3] – встроенный в Flask шаблонизатор, позволяющий динамически генерировать HTML-страницы.
- JavaScript – язык программирования, применяемый для добавления интерактивности на веб-страницах. В проекте используется совместно с AJAX³ для динамической загрузки данных без необходимости полной перезагрузки страницы, что улучшает удобство и скорость работы приложения.

³AJAX (англ. Asynchronous Javascript And Xml) – технология, которая позволяет веб-страницам обновлять данные без перезагрузки.

- jQuery [7] – JavaScript-библиотека, которая облегчает работу с элементами DOM⁴, управление событиями и обработку запросов AJAX. Её использование в проекте также помогает обеспечить кроссбраузерную совместимость и упрощает клиентскую часть приложения.

1.2. Существующее решение для хоста и сервера

1.2.1. Появление PCAP-файла

При запуске эмуляции сети на сайте Miminet сервер ведет себя следующим образом:

1. Запрос обрабатывается Flask-приложением, которое передает его соответствующей Python-функции на стороне фронтенда.
2. Данная функция использует брокер сообщений для постановки задачи в очередь на обработку бэкендом.
3. На бэкенде с использованием библиотеки Mininet:
 - создается виртуальная сеть согласно конфигурации;
 - в процессе работы генерируются PCAPNG-файлы (PCAP Next Generation, дампы сетевого трафика).
4. Содержимое полученных файлов передается на фронтенд с именем сетевого устройства, служащим источником пинга. Функция, переданная через RabbitMQ, как целевая для ответа бэкенда, сохраняет содержимое на фронтенде, создавая папку с PCAP-файлами в соответствующем виде.

1.2.2. Отображение конфигурационного меню

На странице сети при нажатии на устройство появляется конфигурационное меню, в котором перечислены все связи с другими устройствами

⁴DOM (англ. The Document Object Model) – представление данных об объектах, которые составляют структуру и содержимое документа в Интернете.

с помощью библиотеки jQuery. После успешной эмуляции сети рядом с каждым соединением появляется кнопка "(pcap)", обеспечивающая переход на страницу Mimishark с анализом трафика между соответствующей парой устройств с помощью Flask-переадресации.

1.2.3. Отображение PCAP-файла

При первом обращении к Mimishark для анализа трафика конкретного устройства проверяется наличие заранее сформированного JSON-файла с данными о сетевом трафике. Если такой файл существует, благодаря возможностям фреймворка Flask и шаблонизатора Jinja2 рендерится HTML-страница с содержимым данного файла. В случае отсутствия JSON-файла, парсится соответствующий PCAP-файл устройства, преобразуются полученные данные в JSON-формат, после чего снова проверяется наличие JSON-файла.

1.3. Результат обзора

В заключении обзора существующего решения выявлено, что появление PCAP-файла реализовано независимо от типа устройства сети, поэтому реализация поддержки Mimishark для маршрутизатора, концентратора и коммутатора требует только изменение конфигурационного меню и добавления переадресации для этих устройств. Важно отметить, что данная доработка будет выполнена по аналогии с уже реализованным решением для хоста и сервера, путем избавления от существующего дублирующего кода и выделения более общих функций.

2. Постановка задачи

Цель данной работы — реализация отображения PCAP-файла для маршрутизатора, концентратора и коммутатора в Mimishark. Для её выполнения были поставлены следующие задачи:

1. Изменить конфигурационное меню для маршрутизатора, концентратора и коммутатора.
2. Добавить переадресацию на Mimishark для маршрутизатора, концентратора и коммутатора.

3. Описание решения

3.1. Переадресация

Функция, вызывающая рендеринг страницы Mimishark, не зависит от типа устройства, так как в ее логику вложена только проверка наличия PCAP-файла и передача его данных в JSON-формате, то для всех устройств можно использовать эту функцию, которая уже реализована в Miminet. Для переадресации были добавлены Flask-правила для маршрутизатора, концентратора и коммутатора.

Листинг 1: Переадресация.

```
app.add_url_rule(
    "/router/mimishark",
    methods=['GET'],
    view_func=mimishark_page
)
```

3.2. Изменение конфигурационного меню

Для отображения устройства, с которым связан хост, сервер и маршрутизатор, уже были реализованы JavaScript-функции с похожей реализацией, поэтому была создана общая функция, которая принимала тип устройства дополнительным аргументом. Поскольку конфигурационное меню концентратора и коммутатора изначально не отображало подключенные устройства, была доработана функция отображения меню для данных устройств, путем добавления цикла, который обрабатывает каждое соединение и генерирует соответствующие поля интерфейса. Обработка соединений для всех типов устройств сети имела одинаковую реализацию, поэтому была вынесена в отдельную функцию. Такой подход позволил избавиться от дублирующего кода и добиться поддержки концентратора и коммутатора.

При проектировании интерфейса конфигурационного меню для концентратора и коммутатора необходимо было учитывать их специфику

- отсутствие IP-адреса и маски сети. За основу был взят интерфейс хоста, но с удалением соответствующих полей под именами связанных устройств. Дополнительно был добавлен отступ перед кнопкой "Сохранить", компенсирующий отсутствие поля "Шлюз по умолчанию". Такой подход обеспечил единообразие интерфейса для каждого типа сетевого устройства. Пример отображения конфигурационного меню коммутатора показан на Рис. 1.

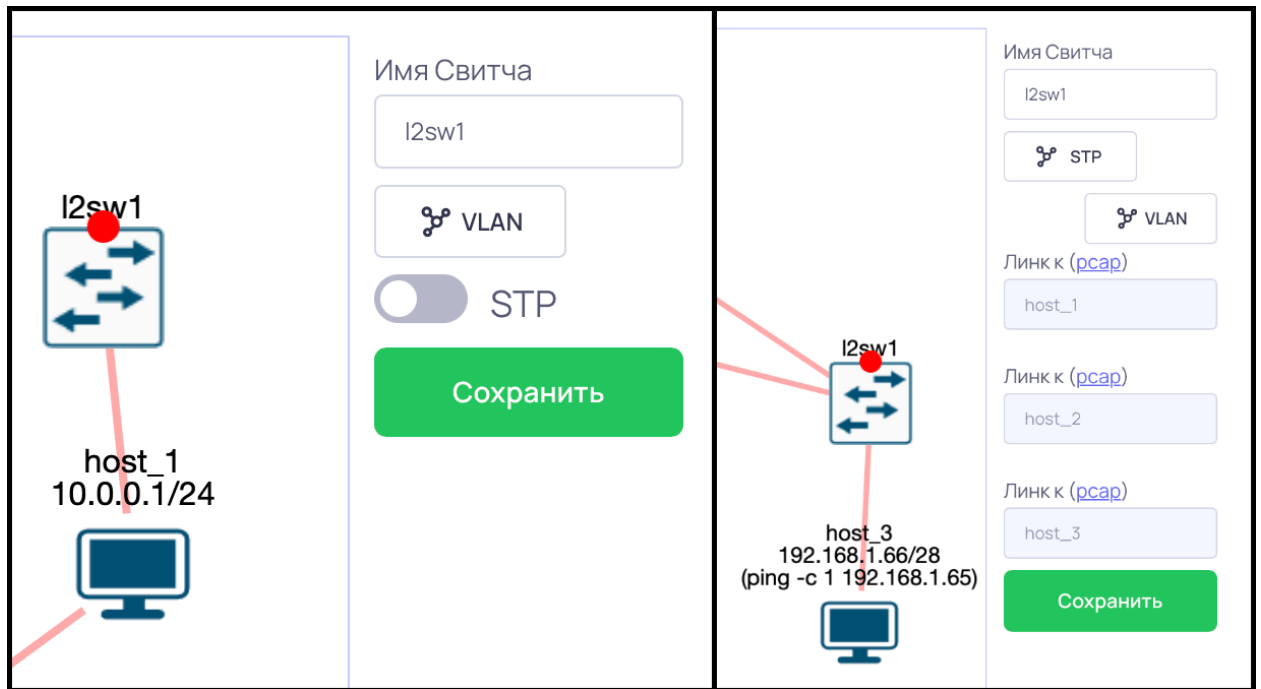


Рис. 1: Конфигурационное меню коммутатора.

Заключение

В ходе выполнения данной работы были выполнены следующие задачи:

- проанализирована реализация похожей функциональности для хоста и сервера;
- изменено конфигурационное меню для маршрутизатора, концентратора и коммутатора;
- добавлена переадресация на Mimishark для маршрутизатора, концентратора и коммутатора.

На момент написания данной работы ведется активное внедрение тестов для фронтенда, и пока что нет возможности запустить эмуляцию сети при тестировании. Учитывая логику появления кнопки переадресации на страницу Mimishark, это накладывает ограничения на проверку предоставленной функциональности. Стоит отметить, что данное ограничение носит временный характер и будет устранено по мере развития тестовой инфраструктуры проекта.

Код доступен на GitHub репозитории проекта Miminet⁵. Реализация отображения PCAP-файлов доступна в pull request⁶.

⁵Официальный репозиторий Miminet – URL: <https://github.com/mimi-net/miminet> (дата обращения: 28 марта 2025 г.).

⁶Pull request с реализацией – URL: <https://github.com/mimi-net/miminet/pull/207> (дата обращения: 28 марта 2025 г.).

Список литературы

- [1] Диденко А. А. Реализация Mimishark. — URL: https://se.math.spbu.ru/thesis/texts/Didenko_Andrej_Spring_practice_2nd_year_2023_text.pdf (дата обращения: 5 апреля 2025 г.).
- [2] Официальный сайт Flask. — URL: <https://flask.palletsprojects.com/en/stable/> (дата обращения: 5 апреля 2025 г.).
- [3] Официальный сайт Jinja2. — URL: <https://jinja.palletsprojects.com/en/stable/templates/> (дата обращения: 5 апреля 2025 г.).
- [4] Официальный сайт Mininet. — URL: <https://mininet.org> (дата обращения: 28 марта 2025 г.).
- [5] Официальный сайт RabbitMQ. — URL: <https://www.rabbitmq.com> (дата обращения: 5 апреля 2025 г.).
- [6] Официальный сайт Wireshark. — URL: <https://www.wireshark.org> (дата обращения: 28 марта 2025 г.).
- [7] Официальный сайт jQuery. — URL: <https://jquery.com> (дата обращения: 5 апреля 2025 г.).
- [8] Официальный сайт libpcap. — URL: <https://www.tcpdump.org> (дата обращения: 28 марта 2025 г.).