

Санкт-Петербургский государственный университет

Кафедра Системного программирования

Группа 22.Б10-мм

Доработка Miminet

Писарев Никита Павлович

Отчёт по учебной практике
в форме «Производственное задание»

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

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
3. Метод	8
3.1. Исправление дефектов в интерфейсе	8
3.2. Роутеры и VLAN	9
3.3. Потеря пакетов	10
4. Эксперимент	11
Заключение	12
Список литературы	13

Введение

В сфере информационных технологий, особенно в области компьютерных сетей, обучение и практический опыт играют ключевую роль в формировании квалифицированных специалистов. В рамках курса «Компьютерные сети» на Математико-механическом факультете Санкт-Петербургского государственного университета важно предоставить студентам инструменты для понимания теоретических и практических аспектов сетевых технологий. Miminet [4], веб-эмулятор компьютерных сетей, служит образовательным инструментом, демонстрирующим студентам ключевые аспекты функционирования сетей.

На данный момент Miminet активно развивается и в проект поступают задачи от учебных заведений и компаний. Так компания YADRO¹ в ходе одной из встреч заметил, что в текущей версии эмулятора роутеры не могут проводить трафик с метками VLAN² (аббр. от англ. Virtual Local Area Network), что ограничивает возможности практического изучения этой важной технологии.

VLAN, или виртуальная локальная сеть, — это технология, позволяющая создать логически обособленные сети поверх одной физической сети. Она обеспечивает изоляцию трафика между различными сегментами сети, что увеличивает безопасность и гибкость управления сетью. Каждый VLAN идентифицируется уникальным номером, и сетевые устройства, такие как коммутаторы и маршрутизаторы, используют эти номера для разделения и обработки трафика.

Кроме того, интерфейс VLAN в текущей версии Miminet содержит некоторые дефекты, которые могут затруднять обучение и работу с эмулятором. Исправление этих ошибок позволит улучшить пользовательский опыт и повысить эффективность учебного процесса.

Ещё одной важной функцией, отсутствующей в Miminet, является возможность моделирования потери сетевых пакетов. Потеря пакетов происходит, когда один или несколько пакетов данных, передаваемых

¹Российская IT-компания <https://yadro.com/> [Дата обращения: 15 июня 2024 г.].

²<https://standards.ieee.org/ieee/802.1Q/6844/> [Дата обращения: 15 июня 2024 г.].

по сети, не достигают своего назначения. Это может происходить по различным причинам, включая перегрузку сети, ошибки на сетевых устройствах или помехи в канале связи. Внедрение возможности моделирования потери пакетов позволит студентам изучить влияние этого явления на качество сетевых соединений.

Таким образом, необходима разработка и интеграция новых функций в Miminet. Предполагается, что это улучшит понимание студентами принципов работы механизмов VLAN и особенностей настройки сетей с учетом потери пакетов.

1. Постановка задачи

Целью работы является улучшение функциональности Mimirnet путём внедрения новых возможностей. Для её выполнения были поставлены следующие задачи:

1. выполнить обзор существующих технологий, с акцентом на возможность роутерам принимать пакеты с VLAN и потерю пакетов;
2. добавить возможность роутерам принимать пакеты с VLAN;
3. внедрить механизм моделирования потери пакетов, позволяющего задавать процент потерь для каждого сетевого соединения;
4. исправить дефекты в интерфейсе VLAN;
5. протестировать реализованную функциональность.

2. Обзор

Были рассмотрены наиболее популярные сетевые эмуляторы с точки зрения возможности добавления сабинтерфейсов для роутеров и моделирования потери пакетов.

- Cisco Packet Tracer [1]: Это известный эмулятор сети, широко используемый в образовательных целях, особенно в программах Cisco Networking Academy. Cisco Packet Tracer предлагает обширные возможности для настройки VLAN, включая выбор создание VLAN и назначение портов к VLAN. Однако добавление настроек по обработке VLAN пакетов на роутеры требует выполнения множества команд через консоль, что может усложнить обучение для студентов. Возможности моделирования потери пакетов в Cisco Packet Tracer ограничены: в базовой версии эмулятора нет встроенных инструментов для задания процента потерь пакетов. Для создания таких условий необходимо использовать сторонние симуляторы или сетевые инструменты, что делает этот процесс менее интуитивным и сложным для студентов.
- GNS3 [3]: Это современный эмулятор, отличающийся своей гибкостью и способностью работать с реальными образами IOS от Cisco³. Он предоставляет широкие возможности для настройки роутеров и управления VLAN, что делает его подходящим инструментом для глубокого изучения сетевых технологий. Тем не менее, сложность настройки и необходимость в обширных технических знаниях могут быть препятствием для использования GNS3 в образовательных целях. Возможности моделирования потери пакетов в GNS3 присутствуют, но требуют сложной конфигурации и дополнительных плагинов, таких как NETem (Network Emulator)⁴, который позволяет управлять параметрами сети, включая потерю пакетов, задержку и вариации задержки. Настройка NETem тре-

³Cisco IOS. <https://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-110/13178-15.html> [Дата обращения: 15 июня 2024 г.].

⁴NETem <https://www.gns3.com/marketplace/appliances/netem> [Дата обращения: 15 июня 2024 г.].

бует особых знаний, знаний Linux и работы с командной строкой, что может затруднить процесс обучения.

- CORE (Common Open Research Emulator) [2]: Это сетевой эмулятор с открытым исходным кодом, который позволяет детально настраивать и управлять сетевыми сценариями. Он поддерживает настройку роутеров и различные режимы работы портов, что позволяет пользователям осваивать сложные сетевые сценарии. Однако, как и в случае с GNS3, настройка через консоль может затруднить обучение. Возможности моделирования потери пакетов в CORE нет.

В заключении анализа сетевых эмуляторов с возможностью добавления возможности роутерам принимать пакеты с VLAN и моделирования потери пакетов выявлено, что большинство все из них требуют сложных консольных настроек, что затрудняет их использование в образовательных целях. В связи с отсутствием удобного графического интерфейса и ограниченными возможностями моделирования потери пакетов было решено разработать и интегрировать свой интерфейс в Mimirnet, обеспечив более интуитивный и удобный инструмент для обучения студентов.

3. Метод

3.1. Исправление дефектов в интерфейсе

Ранее таблица устройств VLAN отображала только класс устройства (host, switch и т.д.), что затрудняло идентификацию. Теперь таблица выводит полное имя устройства, заданное пользователем. Это позволяет пользователю легче находить нужные устройства и упрощает процесс управления сетью.

Также при просмотре сети от имени гостя все кнопки, включая кнопку VLAN, блокировались, что не позволяло просматривать конфигурацию. Теперь кнопка VLAN доступна для просмотра, что позволяет гостям видеть конфигурации сети без возможности внесения изменений.

Устройство	VLAN ID	Тип подключения
user_2	10	Access
user_3	20	Access
I2	10,20	Trunk

Рис. 1: Модальное окно конфигурации VLAN

3.2. Роутеры и VLAN

Для реализации поддержки роутерами обработки пакетов с VLAN мы рассмотрели два возможных подхода: аппаратная поддержка VLAN и использование сабинтерфейсов, это логические интерфейсы, которые создаются на основе физических интерфейсов роутеров. Каждый сабинтерфейс может быть настроен для работы с определённым номером VLAN. Аппаратная поддержка включало бы модификацию стандартного роутера ipmininet, чтобы он поддерживал обработку VLAN на уровне аппаратного обеспечения. Это более сложный подход, так как требует изменения базового кода и архитектуры роутера. А Логические интерфейсы могут быть легко изменены или удалены без изменения основной конфигурации физического интерфейса, что намного гибче, поэтому было решено реализовывать сабинтерфейсы.

Для создания сабинтерфейсов мы использовали стандартную утилиту ip из набора утилит iproute2 [6].

В рамках Mininet была добавлена новая job — команда, которая выполняется во время конфигурации сети. В ней прописаны консольные команды, которые будут выполнены на указанном роутере, с которого выполняется команда.

Процесс создания сабинтерфейсов для пользователя: пользователь выбирает роутер, указывает физический интерфейс, задает IP-адрес и номер VLAN, который будет фильтровать сабинтерфейс.

The image shows a web-based configuration interface for a network device. It contains several input fields and dropdown menus:

- Имя роутера** (Router name): A text input field containing "router_1".
- Выполнить команду** (Execute command): A dropdown menu with "Добавить сабинтерфейс" (Add sub-interface) selected.
- Имя интерфейса** (Interface name): A dropdown menu with "I2sw1" selected.
- IP-адрес / Маска** (IP address / Mask): A split input field with "10.10.1.2" in the left part and "24" in the right part.
- VLAN**: A text input field containing "10".

Рис. 2: Конфигурация сабинтерфейса

3.3. Потеря пакетов

Для реализации функциональности потери пакетов использовалась стандартная утилита `tc` (traffic control) [5] из набора утилит `iproute2`, через дисциплину очередности `qdisc`. Реализована серверная часть, которая принимает сетевое соединение и процент потери пакетов. Затем на вспомогательном коммутаторе между конечными устройствами создается новая очередь, со специальным правилом обработки, с заданным процентом он откидывает пакеты. Вспомогательный коммутатор служит для прозрачной обработки трафика между устройствами, создавая реалистичные условия для моделирования потерь пакетов. Этот подход позволяет студентам на практике увидеть, как потеря пакетов может повлиять на работу сети, что делает процесс обучения более наглядным и понятным.

4. Эксперимент

Для обеспечения качества разработанной функциональности были написаны модульные тесты с использованием библиотеки Pytest⁵. Эти тесты охватывают различные сценарии использования, обеспечивая проверку новой функциональности. Помимо этого, команда разработчиков регулярно проводила встречи для ручного тестирования новых функций интерфейса. Такой подход позволял не только демонстрировать прогресс, но и эффективно обнаруживать и устранять возможные недочеты, улучшая общее качество и работоспособность проекта.

Дополнительно, компания YADRO, предложившая одну из задач, приняла работу и использовала ее в своих тестовых заданиях. Это подтверждает практическую ценность и надежность разработанных решений.

⁵Pytest - фреймворк для тестирования Python-приложений <https://docs.pytest.org/en/7.4.x>
[Дата обращения: 15 июня 2024 г.].

Заключение

В ходе выполнения курсовой работы были достигнуты следующие результаты:

- Проведен анализ популярных эмуляторов сетей, с акцентом на возможность роутерам принимать пакеты с VLAN и потерю пакетов.
- Добавлена возможность роутерам принимать пакеты с VLAN тегом.
- Исправлены дефекты в пользовательском интерфейсе управления VLAN.
- Реализована серверная часть возможности потери пакетов.
- Проведено тестирование реализованных функций управления VLAN, подтверждено их корректное функционирование.

Код доступен в репозитории GitHub.⁶

Функциональность сабинтерфейсов доступна на сайте Miminet.⁷

⁶<https://github.com/mimi-net/miminet/pulls/NikitaPisarev> [Дата обращения: 15 июня 2024 г.].

⁷<https://miminet.ru> [Дата обращения: 15 июня 2024 г.].

Список литературы

- [1] Cisco Packet Tracer. — URL: <https://www.netacad.com/courses/packet-tracer> (дата обращения: 4 июня 2024 г.).
- [2] Common Open Research Emulator. — URL: <https://coreemu.github.io/core/index.html> (дата обращения: 4 июня 2024 г.).
- [3] Graphical Network Simulator-3. — URL: <https://www.gns3.com> (дата обращения: 4 июня 2024 г.).
- [4] IPmininet. — URL: <https://ipmininet.readthedocs.io/en/latest> (дата обращения: 4 июня 2024 г.).
- [5] Linux traffic control. — URL: https://access.redhat.com/documentation/ru-ru/red_hat_enterprise_linux/8/html/configuring_and_managing_networking/linux-traffic-control_configuring-and-managing-networking (дата обращения: 15 июня 2024 г.).
- [6] iproute2. — URL: <https://wiki.linuxfoundation.org/networking/iproute2> (дата обращения: 15 июня 2024 г.).