

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 23.Б11-мм

Расширение функциональности Miminet: добавление моделирования потери пакетов

Усольцев Данил Вячеславович

Отчёт по учебной практике
в форме «Решение»

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

Санкт-Петербург
2025

Оглавление

Введение	3
1. Постановка задачи	4
2. Обзор	5
2.1. Обзор существующих решений	5
2.2. Выводы	5
3. Описание решения	7
3.1. Обновление структуры конфигурации ребра	7
3.2. Добавление поля для настройки процента потерь пакетов на ребре сети	7
3.3. Интеграция параметра потери пакетов в процесс эмуля- ции сети	9
4. Эксперимент	11
Заключение	12
Список литературы	13

Введение

В сфере информационных технологий, особенно в области компьютерных сетей, практические навыки играют большую роль в подготовке квалифицированных специалистов. В рамках курса «Компьютерные сети» на Математико-механическом факультете Санкт-Петербургского государственного университета используется веб-эмулятор Miminet [6], который позволяет студентам изучать принципы работы сетевых технологий в интерактивной форме.

Одной из важных характеристик сетей является потеря пакетов, которая возникает вследствие таких причин, как перегрузка сети, неисправность оборудования или помехи. Однако в текущей версии Miminet отсутствовала возможность моделирования этой характеристики, что ограничивало возможности практического обучения.

Целью данной работы стало добавление в Miminet функции настройки процента потери пакетов между устройствами.

1. Постановка задачи

Целью работы является улучшение функциональности Miminet путём добавления механизма моделирования потери пакетов между сетевыми устройствами. Для её выполнения были поставлены следующие задачи:

1. Модифицировать структуру конфигурации сетевого ребра, включив в неё параметр, определяющий процент потерь пакетов;
2. Реализовать в веб-интерфейсе поле настройки процента потерь пакетов для сетевого ребра с обеспечением передачи заданного значения на бэкенд;
3. Обеспечить применение параметров потерь пакетов в процессе сетевой эмуляции;
4. Протестировать реализованную функциональность;

2. Обзор

Для реализации функции моделирования потери пакетов в сетевом эмуляторе Mimirnet был проведён анализ существующих решений в популярных сетевых эмуляторах. Основное внимание уделялось возможностям настройки потерь пакетов и простоте использования.

2.1. Обзор существующих решений

- Cisco Packet Tracer [2] — один из самых известных эмуляторов, используемых в образовательных курсах сетевых технологий. Однако его возможности по моделированию потери пакетов ограничены: отсутствует встроенная функция для задания процента потерь. Для создания реалистичных условий потери пакетов требуется использование сторонних инструментов или сложных сценариев, что усложняет процесс обучения.
- GNS3 [3] — гибкий эмулятор, поддерживающий работу с реальными образами IOS Cisco и другими сетевыми устройствами. Возможности имитации потери пакетов реализованы с использованием механизма NETem. GNS3 позволяет настраивать процент потерь, задержку и другие параметры через графический интерфейс или командную строку. Однако для эффективного использования NETem в GNS3 требуются глубокие знания Linux, что создаёт достаточно высокий порог входа.

2.2. Выводы

Анализ популярных сетевых эмуляторов выявил их недостатки для обучения задачам моделирования потери пакетов: несмотря на преимущества Cisco Packet Tracer в простоте базового моделирования, а GNS3 — в точности эмуляции за счёт реальных ОС, все они либо лишены встроенного простого механизма настройки потерь пакетов либо требуют сложной ручной конфигурации через терминал и глубоких знаний

Linux, что создаёт высокий барьер для студентов. Для преодоления этого недостатка в Miminet был реализован веб-интерфейс, предлагающий интуитивную настройку параметров потерь пакетов и доступность для пользователей без высоких знаний Linux.

3. Описание решения

3.1. Обновление структуры конфигурации ребра

Класс, описывающий сетевое ребро, был расширен новым атрибутом `loss_percentage`, хранящим процент потерь пакетов. Это позволило сохранять данный параметр в JSON-конфигурации сети. Для обеспечения обратной совместимости с существующими конфигурациями, при отсутствии поля `loss_percentage` в JSON-файле, его значение автоматически инициализируется нулём.

```
@dataclass
class EdgeData:
    id: str
    source: str
    target: str
    loss_percentage: int = 0
```

3.2. Добавление поля для настройки процента потерь пакетов на ребре сети

В форме редактирования параметров ребра было внедрено поле ввода, отвечающее за процент потерь пакетов. Реализована автоматическая коррекция вводимых значений через JavaScript-обработчик [5], который фильтрует нечисловые символы и автоматически ограничивает диапазон (от 0 до 100), а также обеспечено визуальное отображение текущего значения параметра потерь пакетов через динамическую привязку к данным модели.

Механизм сохранения настроек активируется при нажатии кнопки "Сохранить", иницируя AJAX-запрос [1] через функцию `UpdateEdgeConfiguration`. При этом система автоматически выполняет два действия: временно блокирует элементы управления и отправляет параметры на сервер.

Также была реализована динамическая смена цвета соединения при установке процента потерь пакетов больше 0, что даёт пользователю возможность мгновенно выявлять проблемные участки сети.

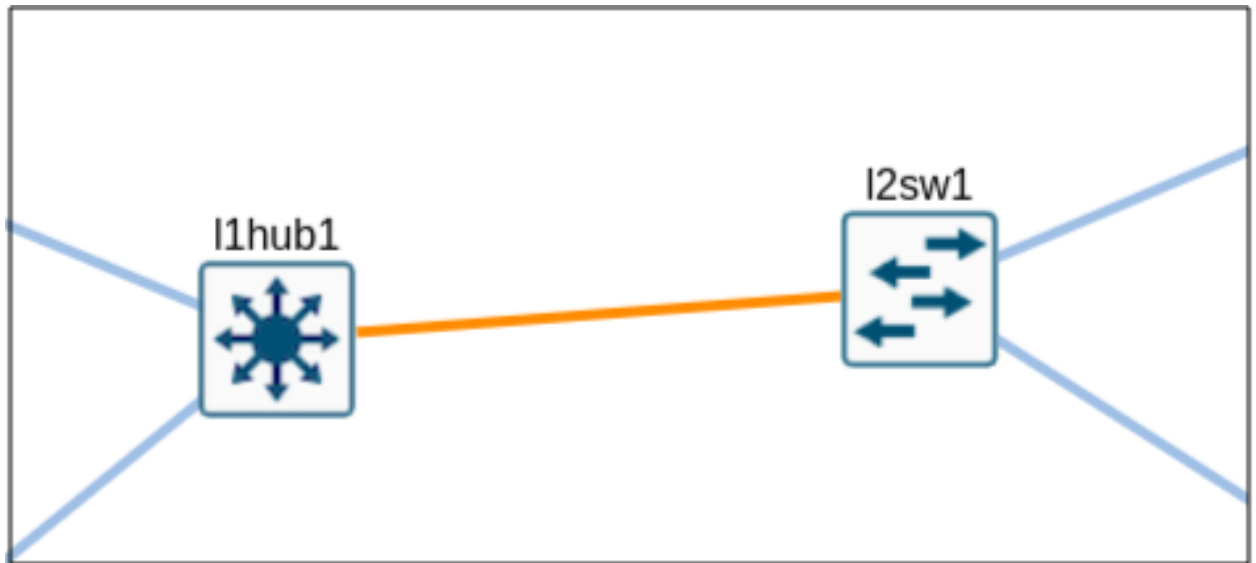


Рис. 1: Сетевое ребро, на котором установлена потеря пакетов

Ниже слева и справа представлены прошлое меню конфигурационного ребра и новое соответственно.

Настройки	Настройки
Из l1hub1	Из l2sw1
В l2sw1	В l1hub1
	Потери пакетов (%) 0
	Сохранить

Рис. 2: Конфигурационное меню ребра

3.3. Интеграция параметра потери пакетов в процесс эмуляции сети

Для обеспечения эмуляции сетевых условий параметр потерь пакетов интегрирован на всех уровнях системы:

1. **Инициализация соединений:** При создании виртуальных соединений с помощью IPMininet [4] значение `loss_percentage` передаётся в метод `addLink` топологии, что позволяет динамически конфигурировать параметры сетевых интерфейсов:

```
for edge in self.__network.edges:
    # Some code
    loss_percentage = (
        edge.data.loss_percentage
        if edge.data.loss_percentage is not None
        else 0
    )
    self.addLink(..., loss_percentage=loss_percentage)
```

2. **Низкоуровневая настройка:** В реализации `addLink` параметр используется для настройки дисциплины очередей через механизм NETem:

```
link_opts_src = {"params2":
    {"delay": delay, "loss": loss_percentage}}
link_opts_dst = {"params1":
    {"delay": delay, "loss": loss_percentage}}
```

Данная конфигурация обеспечивает применение потерь к обоим направлениям трафика (исходящему и входящему).

3. **Визуализация:** При генерации анимации параметр передаётся в парсер пакетов, что позволяет реалистично отображать потери пакетов в UI:

```
packets = create_pkt_animation(...,  
    loss_percentage=loss_percentage  
)
```

4. Эксперимент

Основным методом верификации функциональности выступило ручное тестирование. Для оценки корректности работы в условиях стабильной сети существующие тесты были модифицированы путём введения параметра `loss_percentage`, моделирующего уровень потерь пакетов в канале связи и равного 0 (%).

В ходе тестирования установлено, что при значениях параметра `loss_percentage` от 1% до 99% процесс эмуляции потерь пакетов приводит к вариативности результатов выполнения тестов при одинаковых входных данных. Данный фактор ограничивает возможность прямого сравнения результатов тестирования и получение строго воспроизводимых результатов для каждого теста.

В связи с этим основной акцент в экспериментальной оценке был сделан на ручном исследовании поведения системы.

Заключение

В течение семестра были достигнуты следующие результаты:

- Проведён обзор популярных сетевых эмуляторов (Cisco Packet Tracer, GNS3) с акцентом на поддержку моделирования потери пакетов.
- Был реализован пользовательский интерфейс позволяющий задавать/изменять значение параметра `loss_percentage` для выбранных сетевых рёбер.
- В существующую конфигурацию ребра был добавлен новый параметр, отвечающий за процент потерь пакетов, а также обеспечена корректная передача обновленных конфигурационных данных ребра с фронтенда на бэкенд для последующего применения в эмуляции.
- Проведено ручное тестирование системы, а также модифицированы уже имеющиеся тесты с учетом обновленной конфигурации ребра.

Реализация доступна в пулл-реквесте Github - URL: <https://github.com/mimi-net/miminet/pull/262>

Функциональность доступна на сайте Miminet - URL: <https://miminet.ru/>

Список литературы

- [1] AJAX. — URL: <https://ru.wikipedia.org/wiki/AJAX> (дата обращения: 17 июня 2025 г.).
- [2] Cisco Packet Tracer. — URL: https://ru.wikipedia.org/wiki/Cisco_Packet_Tracer (дата обращения: 17 июня 2025 г.).
- [3] GNS3. — URL: <https://www.gns3.com/> (дата обращения: 17 июня 2025 г.).
- [4] IPMininet. — URL: <https://ipmininet.readthedocs.io/en/latest/> (дата обращения: 17 июня 2025 г.).
- [5] JavaScript. — URL: <https://ru.wikipedia.org/wiki/JavaScript> (дата обращения: 17 июня 2025 г.).
- [6] Miminet. — URL: <https://miminet.ru/> (дата обращения: 17 июня 2025 г.).