Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б07-мм

Добавление DHCP сервера в Miminet

Диденко Андрей Антонович

Отчёт по учебной практике
в форме «Решение»

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

Санкт-Петербург
2024

Оглавление

Введение

В современных компьютерных сетях, где подключение к сетевым ресурсам и обмен данными становятся неотъемлемой частью повседневной работы, эффективное управление IP-адресами и другими сетевыми параметрами является ключевым аспектом обеспечения стабильного функционирования сетевой инфраструктуры. Одним из фундаментальных протоколов, отвечающих за автоматизированную настройку сетевых параметров устройств, является DHCP (Dynamic Host Configuration Protocol) [7].

Miminet [12] - это веб-эмулятор компьютерной сети, разработанный с целью обучения. Это веб-приложение обеспечивает виртуальное окружение, где пользователи могут создавать, анализировать и тестировать различные конфигурации сетей, не прибегая к использованию физического оборудования.

Miminet представляет собой графическое представление IPMininet [11], широко распространенного инструмента в операционной системе Linux для создания виртуальных сетевых топологий. IPMininet создает виртуальные сети с ограниченными ресурсами, предназначенные для обучения и тестирования различных аспектов сетевых технологий, обеспечивая при этом быстроту и простоту использования.

В дополнение к существующим функциям, в разработке Miminet планируется внедрение нового инструмента, способного наглядно демонстрировать процессы DHCP (Протокол динамической конфигурации хоста) в виртуальных сетях, что позволит пользователям лучше понять и визуализировать принципы этого протокола.

В рамках текущей учебной практики предлагается разработать веб-интерфейс, который будет выполнять следующие функции:

- Выдача IP-адресов [8]: Основная задача DHCP сервера — автоматическое присвоение IP-адресов устройствам в сети. Это позволяет избежать конфликтов IP и упрощает процесс подключения новых устройств к сети.

- Выдача подсети и шлюза по умолчанию: DHCP также предоставляет информацию о подсети и шлюзе по умолчанию. Эти параметры важны для правильного функционирования устройств в сети, позволяя им обмениваться данными с другими сетями.

Дата сборки: 8 января 2024 г.

1. Постановка задачи

Целью работы является внедрение протокола DHCP в веб-приложение Miminet. Для её выполнения были поставлены следующие задачи:

1. Изучение принципов работы DHCP сервера.
2. Определение основных параметров и настроек DHCP.
3. Разработка архитектуры DHCP сервера.
4. Разработка веб-интерфейса.
5. Внедрение разработанного веб-интерфейса в структуру Miminet.

2. Обзор

2.1. ISC DHCP

ISC DHCP (Internet Systems Consortium Dynamic Host Configuration Protocol) [4] - это программное обеспечение, предоставляющее сервер DHCP.

- Сложность настройки: Некоторые пользователи отмечают, что настройка ISC DHCP может быть сложной, особенно для тех, кто не имеет опыта работы с DHCP-серверами.
- Ресурсоемкость: В сравнении с некоторыми легковесными решениями, ISC DHCP может потреблять больше системных ресурсов, что может быть проблемой для устройств с ограниченными ресурсами.

2.2. UDHCPD

UDHCPD (Micro DHCP Daemon) [3] - это легковесный DHCP-сервер.

- Ограниченная функциональность: По сравнению с некоторыми полнофункциональными DHCP-серверами, UDHCPD может предоставлять ограниченный набор функций, что может быть недостаточным для сложных сетевых сценариев.
- Отсутствие активной поддержки: В некоторых случаях, UDHCPD может иметь ограниченную или отсутствующую активную поддержку, что может затруднить получение помощи при возникновении проблем.

2.3. Kea DHCP

Kea DHCP (Kea) [5] - это современный и гибкий DHCP-сервер, разрабатываемый организацией Internet Systems Consortium (ISC).

- Требовательность к ресурсам: Kea, хотя и более мощный и гибкий, чем некоторые легковесные решения, может потреблять больше системных ресурсов, что может быть проблемой для устройств с ограниченными ресурсами.

- Ограниченная поддержка: Некоторые сообщества пользователей отмечают, что у Kea может быть ограниченная активная поддержка или нехватка документации по сравнению с другими DHCP-серверами.

2.4. dnsmasq

dnsmasq [9] - это легковесный и простой в настройке DNS и DHCP сервер, предназначенный для использования в малых и домашних сетях. Его основная цель - обеспечить удобство в управлении сетью, предоставляя функционал DNS-сервера для разрешения имен хостов в IP-адреса и DHCP-сервера для автоматической выдачи IP-адресов устройствам в сети.

- Простота использования: Dnsmasq обладает простым конфигурационным файлом, что делает его легким в настройке и управлении, особенно для пользователей без глубоких знаний в области сетевых технологий.

- Легковесность: Этот сервер является легковесным и эффективным с точки зрения ресурсов, что позволяет использовать его на устройствах с ограниченными ресурсами, таких как маршрутизаторы или встраиваемые системы.

- Автоматическое обновление DNS: Dnsmasq обновляет DNS записи при изменениях в DHCP, обеспечивая актуальность данных в сети.

Было решено выбрать dnsmasq из нескольких вариантов реализации DHCP-сервера. Этот выбор обоснован тем, что dnsmasq представляет собой легковесное и эффективное решение, идеально подходящее для

устройств с ограниченными ресурсами, таких как маршрутизаторы и встроенные системы.

3. Обзор используемых инструментов

Поскольку Miminet в своей реализации уже использует Flask для обеспечения серверной части и Bootstrap для стилизации интерфейса, данные инструменты были использованы в рамках учебной практики.

3.1. Flask

Flask [6] представляет собой инструмент для создания веб-приложений на языке программирования Python и использует компоненты Werkzeug и шаблонизатор Jinja2.

Этот фреймворк обеспечивает гибкость и легкость в разработке веб-приложений на Python, предоставляя набор базовых инструментов. Он является легким по сравнению с Django, предоставляя только необходимые базовые функциональности.

3.2. Bootstrap

Bootstrap [2] представляет собой свободный набор инструментов, предназначенных для создания сайтов и веб-приложений. Включая в себя HTML- и CSS-шаблоны для различных компонентов интерфейса, таких как типографика, веб-формы, кнопки, метки, блоки навигации, а также JavaScript-расширения. Использование Bootstrap обеспечивает быстрое и эффективное создание привлекательного дизайна веб-приложения.

3.3. Miminet и IPMininet

Mininet [1] - это программное средство, предназначенное для эмуляции сетей в виртуальной среде. Оно позволяет создавать виртуальные сетевые топологии, состоящие из хостов, коммутаторов и контроллеров, которые могут быть запущены на одном физическом компьютере. Mininet позволяет студентам, разработчикам и исследователям изучать

и тестировать сетевые конфигурации без использования реального оборудования.

IPMininet [11] - это расширение Mininet, специально ориентированное на поддержку сетей с применением протокола IP (Internet Protocol). Он предоставляет дополнительные возможности для создания и настройки сетей с использованием IP-адресации, маршрутизации и других сетевых аспектов. IPMininet упрощает процесс моделирования и тестирования IP-сетей в виртуальной среде.

Оба инструмента активно применяются в образовательных и исследовательских целях. Они обеспечивают простой способ создания различных сетевых сценариев для изучения и экспериментов, а также позволяют обходить ограничения, связанные с использованием реального оборудования.

3.4. dnsmasq

Для решения задачи автоматической настройки DHCP-сервера и DNS-прокси в сетевом окружении, был выбран dnsmasq [9]. Dnsmasq является легковесным и простым в использовании инструментом, предоставляющим DNS- и DHCP-сервер в одном пакете. Он включает в себя возможности трансляции имен хостов в MAC-адреса, фильтрации контента и другие функции. Выбор dnsmasq обусловлен его легкостью настройки, что делает его идеальным решением для создания небольших сетей или встроенных систем с ограниченными ресурсами.

4. Реализация

В рамках выполнения учебной практики были сформулированы две основные задачи:

- Изучение принципов DHCP и его конфигураций.
- Реализация соответствующей функции, с последующим внедрением в Miminet.

4.1. Тестирование DHCP в виртуальной сети Mininet

Для успешной интеграции DHCP в обширный проект было необходимо провести тщательное изучение его функциональности и освоить процесс настройки конфигураций. Однако, перед тем как внедрить DHCP в реальный проект, требовалось провести тестирование его работы в контролируемой среде.

Для этого была создана небольшая тестовая сеть с использованием инструмента Mininet, включающая два хоста. На одном из хостов был настроен и запущен DHCP-сервер, а также конфигурационный файл был подстроен под требования проекта. Далее, на втором хосте был осуществлен запрос на выдачу IP-адреса с использованием DHCP-клиента.

Пример конфигурационного файла dhcp:

```
# Файл конфигурации dnsmasq.conf

# Интерфейс, на котором слушает DHCP сервер
interface=eth0

# Разрешить использование DHCP сервера
dhcp-authoritative

# Диапазон IP-адресов для выдачи
dhcp-range=192.168.1.50,192.168.1.100,12h

# Адрес шлюза по умолчанию
dhcp-option=3,192.168.1.1
```

Рис. 1: Пример конфигурационного файла dhcp

В этом конфигурационном файле сосредотачиваются ключевые параметры, определяющие поведение сервера в сетевой среде. Это включает в себя указание диапазонов адресов, параметры аренды IP, настройки опций сетевого протокола и другие важные параметры, которые определяют функциональность DHCP-сервера.

По завершении успешного запуска DHCP-сервера и успешного получения вторым хостом IP-адреса, был произведен переход на второй этап.

Второй этап включает в себя несколько подзадач:

4.2. Определение основных параметров DHCP

После обсуждения с научным руководителем было решено использовать следующие конфигурации DHCP сервера:

- Диапазон IP-адресов: Определение диапазона IP-адресов, которые DHCP-сервер может динамически выделять устройствам в сети. Этот параметр определяет доступные адреса для присвоения клиентам в рамках сети.

- Шлюз (Gateway): Указание IP-адреса шлюза по умолчанию, который предоставляется клиентам через DHCP. Этот параметр определяет маршрут по умолчанию для устройств в сети.

- Маска подсети: Установка маски подсети, которая определяет размер подсети и, следовательно, количество доступных IP-адресов в диапазоне. Этот параметр важен для правильного выделения адресов и организации сетевого трафика.

4.3. Разработка архитектуры DHCP сервера

После обсуждения с научным руководителем было принято решение интегрировать DHCP сервер на хосте (рассматривался также вариант добавление DHCP на элемент ”сервер”), представляя его в качестве сервера в разделе ”Выбрать команду”. В этом разделе пользователь может

указать три основных параметра: диапазон IP-адресов, маску подсети и шлюз, которые предназначены для автоматического выделения на других хостах в сети.

Введена функциональность автоматического добавления IP-адресов на хосты для предоставления DHCP-сервером своих функций. Для этого был добавлен чекбокс для удобства пользовательских настроек (отображается только при добавлении команды запуска DHCP-сервера), который при активации предоставляет доступ к данной функциональности, а также отключал возможность редактирования некоторых полей (ip-адрес, маска и шлюз по умолчанию). Это позволяет пользователям легко определить, хотят ли они автоматически применять настройки DHCP сервера к выбранному устройству в сети.

Для правильной симуляции работы DHCP сервера был разработаны функции для добавления конфигураций сервера и последующего взаимодействия с ними.

Листинг 1: **Запуск DHCP сервера на хосте**

```
def dhcp_server(job: Job, job_host):
    job_host.cmd('service dnsmasq stop')
    ip_range = job.arg_1
    mask = job.arg_2
    gw = job.arg_3
    mask = mask_to_byte(mask)
    job_host.cmd(f"dnsmasq —dhcp–range={ip_range},
    {mask} —dhcp–option=3,{gw}")
```

Эта функция предназначена для управления DHCP-сервером. В ней два аргумента - объект ‘Job‘ и хост, на котором выполняется данная задача. Внутри функции происходят следующие действия:

- Команда ‘job_host.cmd('service dnsmasq stop')‘ останавливает службу dnsmasq на хосте. Это используется для предварительной остановки DHCP-сервера перед его переконфигурированием.
- Получение аргументов из объекта ‘job‘ (ip_range, mask и gw).

- Вызов команды конфигурирования DHCP-сервера с использованием утилиты ‘dnsmasq‘. Команда содержит параметры, такие как ‘–dhcp-range‘ с указанием диапазона IP-адресов, а также ‘–dhcp-option‘ с указанием шлюза.

Эта команда используются для запуска и настройки DHCP-сервера на указанном хосте в рамках виртуальной среды эмуляции.

Далее представлена функция запроса IP-адреса:

Листинг 2: **Запрос IP-адреса у DHCP сервера**

```
def dhcp_client(job: Job, job_host):
    job_host.cmd(f'ifconfig {job_host.intf().name} 0')
    job_host.cmd(f'timeout -k 0 30 dhclient -v -4
    {job_host.intf().name}')
    ip, netmask, gateway =
    parse_ip_route_show_output(job_host.cmd('ip route show'))
    job_host.setIP(f'{ip}/{netmask}')
    job_host.cmd(f'route add default gw {gateway}')
```

Эта функция предназначена для управления DHCP-клиентом на хосте.

- ‘job_host.cmd(f’ifconfig job_host.intf().name 0’)‘: Эта команда устанавливает интерфейс хоста в состояние ”down” (выключено) с использованием ‘ifconfig‘. Это делается для очистки текущих настроек интерфейса перед запросом нового IP-адреса от DHCP-сервера.

- ‘job_host.cmd(f’timeout -k 0 30 dhclient -v -4 job_host.intf().name’)‘: Эта команда использует утилиту ‘dhclient‘ для запроса IP-адреса от DHCP-сервера. Опция ‘-v‘ отвечает за подробный вывод, а ‘-4‘ указывает, что следует использовать только IPv4.

- ‘ip, netmask, gateway = parse_output(job_host.cmd(’ip route show’))‘: Получение вывода команды ‘ip route show‘ для дальнейшего разбора и извлечения информации о IP-адресе, маске подсети и шлюзе.

- ‘job_host.setIP(f’ip/netmask’)‘: Установка нового IP-адреса на хосте с использованием извлеченной информации.
- ‘job_host.cmd(f’route add default gw gateway’)‘: Добавление шлюза по умолчанию для нового IP-адреса.

В целом, эта функция выполняет процесс запроса и установки нового IP-адреса с использованием DHCP-клиента на указанном хосте в виртуальной среде эмуляции.

На этом листинге приведена функция передачи данный о нажатом чекбоксе:

Листинг 3: **Передача состояния чекбокса и отправка запроса IP-адреса хосту**

```
if(host_checkbox_value=='1'):
    node["config"]["checkbox"] = 1

    existing_job_105 = next((job for job in
    jnet.get("jobs", []) if job.get('job_id') == 105 and
    job.get('host_id') == node['data']['id']), None)

    if(existing_job_105 == None):
        if not jnet.get("jobs"):
            jnet["jobs"] = []

        job_level = len(jnet["jobs"])
        try:
            jnet['jobs'].append({'id': job_id_generator(),
                'level': job_level,
                'job_id': 105,
                'host_id': node['data']['id'],
                'print_cmd': ''})
        except Exception as e:
            print(e)
            ret.update({'warning': ' Something went wrong '})
```

```
else:
    node["config"]["checkbox"] = 0
```

В этой функции происходит:

- Проверка значения переменной host_checkbox_value, которая, содержит информацию о состоянии чекбокса на веб-странице.
- Если host_checkbox_value равно 1, то устанавливается значение node["config"]["checkbox"] в 1. Это используется для отслеживания состояния чекбокса.
- Проверяется наличие задачи с идентификатором 105 (запрос IP-адреса у DHCP) для текущего хоста в структуре данных jnet["jobs"].
- Если задачи с идентификатором 105 для текущего хоста нет, то создается новая задача с этим идентификатором, уникальным идентификатором и пустыми аргументами. Эта задача добавляется в список задач jnet["jobs"].
- В противном случае, если host_checkbox_value не равно 1, устанавливается значение node["config"]["checkbox"] в 0.

Этот код обрабатывает запросы, связанные с чекбоксом на веб-странице и управляет добавлением задачи с идентификатором 105 в список задач.

4.4. Разработка веб-интерфейса

Для реализации графической части проекта был использован инструмент Figma. С его помощью был создан макет, охватывающий детали взаимодействия между DHCP-сервером и DHCP-клиентом.

Для реализации интерфейса клиент-сервер обсуждались различные варианты, включая использование кнопки. Однако пришлось отказаться от этого варианта в пользу чекбокс, так как этот элемент управления успешно применяется в интерфейсах таких компаний, как Microsoft, и

в продуктах, например, Cisco Packet Tracer, что подтверждает его эффективность и широкое распространение в индустрии.

Макет dhcp сервера:

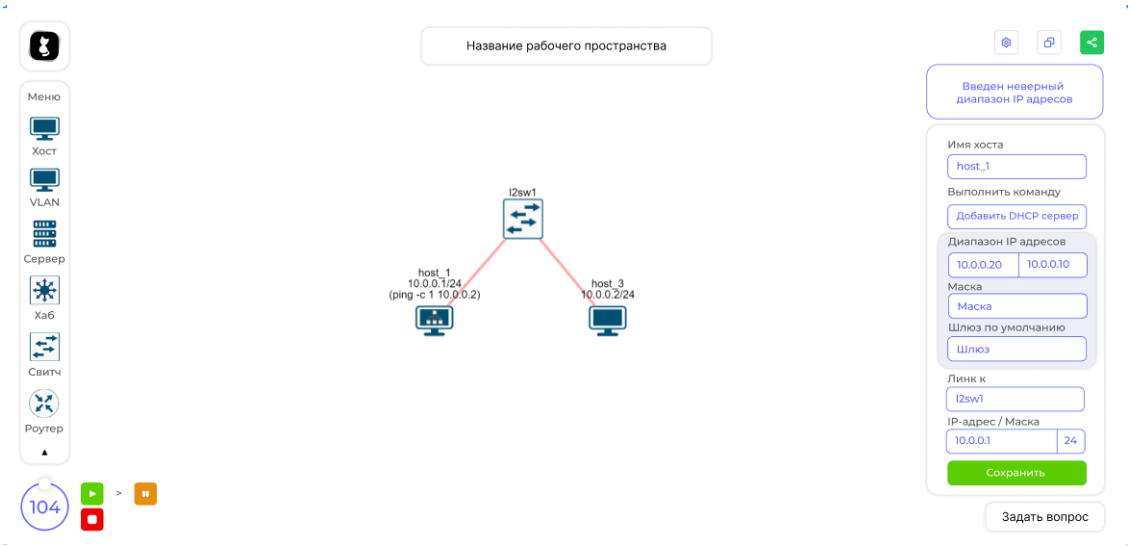

Рис. 2: Макет dhcp сервера

Макет клиента DHCP:

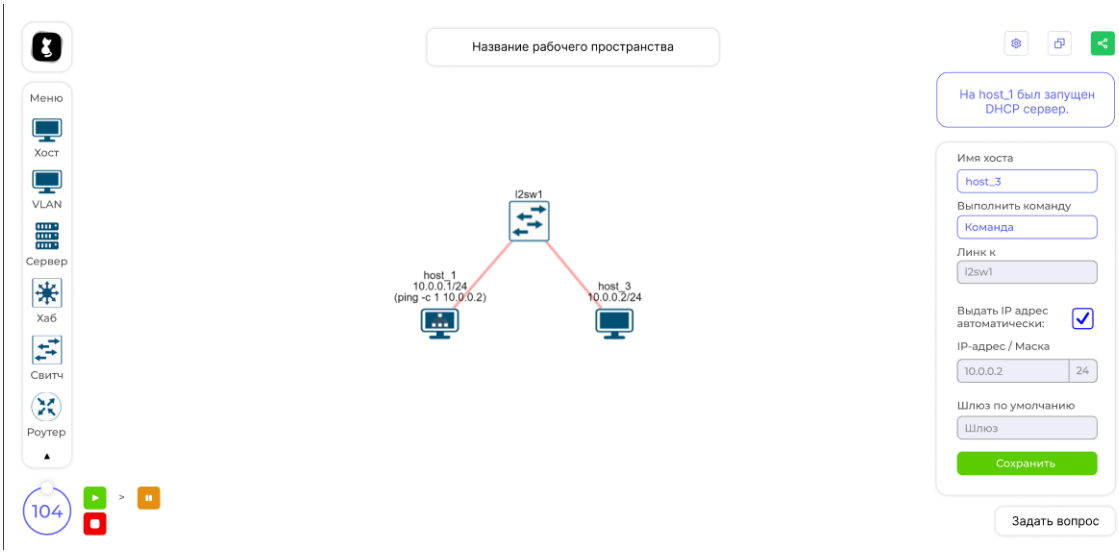

Рис. 3: Макет клиента DHCP

Кроме того, при разработке веб-сайта активно использовался JavaScript для внедрения функциональности чекбокса, а также для сохранения состояния этого чекбокса.

Ниже представлен код реализации автоматической выдачи адресов при нажатии на чекбокс:

Листинг 4: **Чекбокс**

```
const ConfigHostCheckbox = function(name, checkbox) {
    let elem = document.getElementById
    ('config_host_checkbox_script').innerHTML;
    (elem).insertBefore('#config_host_ip_label_example');
    (elem).insertBefore('#config_host_ip_label_' + name);

    ("#auto_ip").prop('checked', (checkbox == 1) ? true :false);
    const disableInputs = [('#config_host_ip_' + name),
    ('#config_host_mask_' + name),
    ('#config_host_default_gw')]
    const holderInputs = [('#config_host_ip_example'),
    ('#config_host_mask_example'),
    ('#config_host_default_gw_holder')]
```

```
    ("#auto-ip").on("change", function () {
        if ((this).is(':checked')) {
            holderInputs.forEach(input => {
                input.removeAttr("hidden");
            });
            disableInputs.forEach(input => {
                input.attr('hidden', true);
            });
        }
        else {
            disableInputs.forEach(input => {
                input.removeAttr('hidden');
            });
             holderInputs.forEach(input => {
                input.attr("hidden", true);
            })
        }

    });
}
```

4.5. Внедрение готового веб-интерфейса в Miminet

Завершающим этапом проекта стала интеграция разработанного продукта в приложение Miminet. Этот процесс включал в себя модификацию исходных файлов самого Miminet.

В ходе выполнения учебной практики произошло завершение проекта [10] участника команды, направленного на объединение фронтенда и бэкенда в один репозиторий. В связи с этим была проведена миграция в новый репозиторий, а также внесены соответствующие изменения в разработанную архитектуру для адаптации к новой структуре репозитория(переписана структура чекбокса и сохранение его состояния, а

также команда о добавлении DHCP сервера на хост). При миграции на новый репозиторий также необходимо было произвести установку DHCP в контейнере Docker.

Проблема заключалась в успешном переносе не только структуры файлов, но и в согласованной работе фронтэнда и бэкэнда в новом репозитории, так как изначально в проекте не предусматривалась возможность передачи конфигураций хостов из бэкенда во фронтенд, поэтому было необходимо обеспечить корректную передачу данных между этими двумя компонентами, чтобы сохранить функциональность и стабильность работы приложения.

Во время интеграции DHCP сервера были определены и реализованы следующие функции:

- Чекбокс появляется у хостов только после добавления команды о запуске DHCP сервера.
- При удалении команды о запуске DHCP сервера чекбокс также пропадает у хостов.
- Если добавлена команда о запуске DHCP сервера, она выполняется первой, чтобы избежать бесконечного ожидания IP-адреса хостом на незапущенном сервере.
- В эмуляцию добавлены команды остановки DHCP сервера по завершении эмуляции, а также установлен таймер ожидания IP-адреса от DHCP сервера.

Заключение

В результате работы DHCP сервер был внедрен в веб-приложение Miminet. Были выполнены поставленные задачи:

- Изучены принципы работы DHCP сервера.
- Определены основные параметры и настройки DHCP.
- Разработана архитектура DHCP сервера.
- Разработан веб-интерфейс.
- Разработанный веб-интерфейс был внедрен в структуру Miminet.

Всю проделанную работу можно увидеть на GitHub репозитории [1].

[1] https://github.com/K0lba/miminet (Дата доступа: 2023-12-14)

Список литературы

[1] Bob Lantz Brandon Heller. Mininet. — URL: https://github.com/mininet/mininet (дата обращения: 2024-01-04).

[2] Bootstrap. — URL: https://getbootstrap.com/ (дата обращения: 2023-05-09).

[3] BusyBox. UDHCPD. — URL: https://busybox.net/ (дата обращения: 2023-12-14).

[4] Consortium Internet Systems. ISC DHCP. — URL: https://www.isc.org/dhcp/ (дата обращения: 2023-12-14).

[5] Consortium Internet Systems. Kea DHCP. — URL: https://www.isc.org/kea/ (дата обращения: 2023-12-14).

[6] Flask. — URL: https://flask.palletsprojects.com/en/2.3.x/ (дата обращения: 2023-05-09).

[7] Force Internet Engineering Task. DHCP. — URL: https://www.ietf.org/ (дата обращения: 2024-01-04).

[8] IP. — URL: https://tools.ietf.org/html/rfc791 (дата обращения: 2024-01-04).

[9] Kelley Simon. dnsmasq. — URL: https://dnsmasq.org/ (дата обращения: 2023-12-14).

[10] Miminet. Новый репозиторий Miminet. — URL: https://github.com/mimi-net/miminet (дата обращения: 2024-01-04).

[11] cnp3. IPMininet. — URL: https://github.com/cnp3/ipmininet (дата обращения: 2024-01-04).

[12] Зеленчук И. В. Эмулятор компьютерных сетей Miminet. — URL: https://miminet.ru/ (дата обращения: 2023-05-14).