

Я Волков Никита Валерьевич, взял тему “Класс, который выполняет функцию маршрутизатора, принимая решения по маршрутизации пакетов в сети”

Цель проекта: Создать класс Router, способный анализировать IP-адрес назначения в сетевом пакете и, используя таблицу маршрутизации, определять и направлять пакет следующему хопу (маршрутизатору или конечному узлу) на пути к месту назначения.
Основные задачи: Таблица маршрутизации и интерфейсы:

Основные задачи:

1. Таблица маршрутизации и интерфейсы

1) Как она будет работать? Будет реализована как структура данных, хранящая информацию о маршрутах.

2) Интерфейсы: Маршрутизатор может иметь несколько сетевых интерфейсов, которые подключены к отдельной сети.

3) interfaces это порты которые подключаются к другим сетям или к устройствам, а для чего еще нужен он определяет, к каким сетям подключен маршрутизатор и какие IP адреса.

Описание: Разработка структуры данных для хранения информации о доступных маршрутах и сетевых интерфейсах маршрутизатора.

4) Разработка структуры данных для хранения информации о доступных маршрутах и сетевых интерфейсах маршрутизатора.

Реализация: Таблица маршрутизации (routing_table):

Будет реализована как словарь (dict), где ключом является IP-адрес сети назначения (в формате строки, например, "[192.168.1.0/24](#)"), а значением – информация о следующем хопе и интерфейсе.

Интерфейсы (interfaces): Маршрутизатор может иметь несколько сетевых интерфейсов, каждый из которых подключен к отдельной сети. interfaces будет представлен в виде словаря, где ключ – идентификатор интерфейса (например, "eth0"), а значение – IP-адрес этого интерфейса.

Назначение interfaces: interfaces определяет, к каким сетям подключен маршрутизатор, и какие IP-адреса он использует для связи с этими сетями

2. Методы (Функции класса Router):

- 1)Описание: Разработка методов, определяющих поведение маршрутизатора. Реализация: `init(self, router_id)` (Инициализация): Отвечает за создание экземпляра класса Router. При создании экземпляра инициализируется таблица маршрутизации (делается пустой) и настраиваются сетевые интерфейсы (или их список остается пустым для дальнейшей настройки). Метод принимает `router_id` – уникальный идентификатор маршрутизатора.
- 2)Назначение `init`: Этот метод, вызываемый при создании нового объекта Router, "включает" маршрутизатор и настраивает его начальное состояние.
- 3)`add_route(self, destination, next_hop, interface)` (Добавление маршрута): Добавляет новую запись в таблицу маршрутизации. Метод принимает IP-адрес назначения (`destination`), IP-адрес следующего хопа (`next_hop`) и идентификатор интерфейса (`interface`).
- 4)Назначение `add_route`: Позволяет маршрутизатору узнавать о новых сетях и путях к ним, указывая, как добраться до определенной сети.
- 5)`remove_route(self, destination)` (Удаление маршрута): Удаляет существующую запись из таблицы маршрутизации. Метод принимает IP-адрес назначения (`destination`), маршрут к которому следует удалить.
- 6)Назначение `remove_route`: Позволяет маршрутизатору "забывать" маршруты, которые больше не актуальны.
3. Управление таблицей маршрутизации:
- 7)Описание: Реализация методов для динамического добавления, удаления и изменения маршрутов в таблице маршрутизации. * Реализация: За это отвечают методы `add_route` и `remove_route`, описанные выше.
- 8)инициализация- он будет отвечать за создание экземпляра класса Router, а также в нем инициализируется таблица маршрутизации и настраиваются сетевые интерфейсы.
- 9) `init self,router id` это такой метод когда я создам новый объект Router потом он включает маршрутизатор, а также он настраивает начальное состояние маршрутизатора.
- 10) `add route` это метод добавляет новую запись в таблицу маршрутизации, типо он говорит маршрутизатору как добраться до

определенной сети, а также он нужен для маршрутизатору узнавать о новых путях к ним.

4) Простейшая обработка ошибок маршрутизации:

В реальных сетях при маршрутизации могут возникать ошибки (например, недоступен следующий хоп).

В класс Router можно добавить простейшую обработку таких ошибок. Например, если в `routing_table` указан `next_hop`, который не является IP-адресом другого маршрутизатора в моделируемой сети, можно вывести сообщение об ошибке и отбросить пакет.

В более сложной реализации можно имитировать отправку ICMP-сообщения "Destination Unreachable" отправителю пакета

- 5) `remove router` это удаление маршрута, он нужен для того, чтобы удалять из таблицы маршрутизации.

3. Управление таблицей маршрутизации

- 1) Этот метод для добавление и удаления и изменения маршрутов в таблице маршрутизации

4. Маршрутизации пакетов;

Описание: Разработка основного алгоритма маршрутизации, определяющего, куда отправить полученный пакет.

Реализация: Метод `receive_packet` получает пакет, анализирует IP-адрес назначения и ищет подходящий маршрут в таблице маршрутизации. На основе найденного маршрута вызывается метод `forward_packet` для пересылки пакета следующему хопу.

5. Представление пакетов;

- 1) Обработка пакетов с неизвестным назначением (Default Gateway): В реальных сетях не всегда есть явный маршрут для каждого возможного IP-адреса назначения. Для обработки таких ситуаций в класс Router может быть добавлена поддержка шлюза по умолчанию (`default_gateway`).

Если при получении пакета не удастся найти маршрут в таблице маршрутизации, пакет отправляется на IP-адрес, указанный в `default_gateway`.

Это позволяет маршрутизатору перенаправлять трафик в вышестоящую сеть (например, к провайдеру интернет-услуг) для дальнейшей обработки.

Для реализации этой функции в класс Router можно добавить поле `default_gateway` (тип `string`, хранящий IP-адрес) и метод `set_default_gateway(self, gateway_ip)`, позволяющий установить шлюз по умолчанию. В методе `receive_packet` следует добавить проверку: если маршрут не найден, пакет направляется на `default_gateway` (если он установлен).

2) Простейшая имитация "отправки" пакета:

В текущей реализации метод `forward_packet` просто выводит сообщение о пересылке пакета. Можно расширить его функциональность, добавив простую имитацию "отправки" пакета.

Например, можно добавить проверку: если `next_hop` совпадает с IP-адресом одного из интерфейсов текущего маршрутизатора, то можно считать, что пакет достиг конечного пункта назначения (хоста, подключенного к этому интерфейсу). В этом случае можно вывести сообщение об успешной доставке пакета.

класса – принятие решений о маршрутизации сетевых пакетов на основе таблицы маршрутизации. Это позволит моделировать сетевую инфраструктуру и исследовать принципы маршрутизации в сетях.

3) Создание класса `Packet`, который представляет собой пакет данных, проходящий через маршрутизатор, а также этот класс может содержать IP адрес источника и еще данные.

4) `Time-To-Live` – количество "прыжков", которое пакет может совершить, прежде чем будет отброшен. Это позволяет предотвратить заикливание пакетов в сети.

В методе `receive_packet` можно добавить проверку `ttl`: если `ttl` равен 0, пакет отбрасывается.

5) Реализация метода `repr(self)`:

Помимо `str(self)`, можно реализовать метод `repr(self)`. `repr` предназначен для создания строкового представления объекта, которое можно использовать для его воссоздания. В простом случае `repr` может возвращать строку, содержащую имя класса и `router_id` (например, `"Router(router_id='Router1')"`).

6. Моделирование сети;

1. Создание дополнительных классов для представления сетевых узлов компьютеров и маршрутизаторов. По этому позволит моделировать более сложную сеть.

7. Поля класса Router;

1) RoutingTable: Dictionary<string, string>-таблица маршрутизации сеть назначения следующий хоп. Interfaces: List string или List Interface список IP адресов интерфейсов или список объектов Interface

2) interfaces: List string еще можно List Interface IP адресов интерфейсов объектов Interface.

8. Поля класса;

1) router id это Идентификатор маршрутизатор

он выполняет задачу идентифицирует каждый маршрутизатор в сети. Он нужен для того, чтобы различать маршрутизаторы когда их несколько пример Router1 или Router2

2) interfaces это порты которые подключаются к другим сетям или к устройствам, а для чего еще нужен он определяет, к каким сетям подключен маршрутизатор и какие IP адреса.

[Volkov - Google Документы](#)