

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б11-мм

Интеграция технологии VXLAN в Mimirnet

Калашников Матвей Михайлович

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 инструментов и технологий	6
2.1. IPMininet и Mininet	6
2.2. VXLAN	7
3. Описание решения	11
3.1. Разработка серверной части	11
3.2. Разработка веб-интерфейса	12
Заключение	16
Список литературы	17

Введение

На Математико-механическом факультете Санкт-Петербургского государственного университета ведется курс «Компьютерные сети». В связи с этим важно предоставить студентам удобные инструменты для выполнения практических заданий по настройке сетей, так как это помогает в усвоении материала. Настройка виртуальной компьютерной сети может быть трудоемкой задачей. Обычно это требует либо развертывания среды с использованием VirtualBox [11] и Linux, либо использования специализированных инструментов для эмуляции сетей, таких как GNS3 [4]. Оба этих подхода требуют много времени для настройки и могут быть избыточно сложны для задачи обучения.

Для решения этих проблем на кафедре системного программирования разрабатывается веб-эмулятор Miminet, основанный на сетевом эмуляторе Mininet [8]. Miminet предоставляет пользователям понятный интерфейс для создания компьютерных сетей, настройки сетевых устройств и протоколов, а также анализа работы сети после завершения эмуляции. Это позволяет студентам на практике понять принципы функционирования сетей и приобрести необходимые навыки. Кроме этого Miminet не зависит от операционной системы и позволяет сосредоточиться на изучении сетевых концепций, не задумываясь о специфике настройки сети для конкретной ОС.

Для соответствия требованиям курса «Компьютерные сети» в Miminet интегрируются различные сетевые технологии. Это обеспечивает обучающимся получение актуальных навыков и знаний, полезных в профессиональной деятельности.

Одной из таких технологий является VXLAN (Virtual Extensible LAN), который описан в RFC 7348 [10]. VXLAN позволяет разделить физическую сеть на 16 миллионов логических сетей, создавая тем самым виртуальные L2-сети поверх существующей L3-инфраструктуры. Она добавляет к оригинальному пакету заголовок VXLAN и отправляет его по L3 сети, используя протокол UDP.

Благодаря своим преимуществам перед VLAN, VXLAN хорошо под-

ходит для построения гибких сетей в центрах обработки данных [3]. Множество инфраструктурных проблем, связанных с высокой нагрузкой со стороны пользователей и большими объемами трафика, может быть решено путем использования VXLAN [1].

Так как добавление возможности конфигурировать сети с использованием VXLAN в Mimirnet способствует формированию у студентов представления о работе современных сетевых инфраструктур, было решено реализовать эту возможность.

1. Постановка задачи

Целью работы является добавление поддержки unicast VXLAN в Miminet. Для её выполнения были поставлены следующие задачи:

1. Изучить принцип работы VXLAN.
2. Определить подходящий способ настройки VXLAN для Miminet.
3. Разработать веб-интерфейс для настройки VXLAN.
4. Реализовать эмуляцию VXLAN на бэкенде.
5. Обеспечить корректное взаимодействие фронтенда и бэкенда.

2. Обзор инструментов и технологий

Miminet представляет собой веб-эмулятор работы сети. Архитектура состоит из двух ключевых компонентов: фронтенда и бэкенда. Фронтенд управляет веб-интерфейсом, обеспечивает пользователю доступ к различным функциям Miminet, включая авторизацию и управление конфигурациями сетей. Под фронтендом в данном случае подразумевается не только пользовательский интерфейс, но и серверная часть, которая хранит пользовательские данные и конфигурации сетей, но непосредственно не участвует в эмуляции.

Сообщения от фронтенда передаются на бэкенд с помощью брокера сообщений RabbitMQ [9]. Такой подход позволяет бэкенду асинхронно обрабатывать запросы на эмуляцию вне зависимости от времени их поступления. Запросы на эмуляцию передаются последовательно, что позволяет бэкенду нормально работать, даже если на обработку поступает одновременно множество запросов. Эмуляция сетей осуществляется с помощью Mininet [8].

2.1. IPMininet и Mininet

Mininet [8] – это сетевой эмулятор, который позволяет создать сеть, состоящую из виртуальных хостов, коммутаторов, контроллеров и соединений. Хосты в Mininet запускают стандартное сетевое ПО Linux, а коммутаторы поддерживают OpenFlow. Mininet подходит для задач обучения, разработки, прототипирования и тестирования.

Mininet использует виртуализацию на основе процессов для запуска множества устройств на одном ядре ОС. Для этого используются сетевые пространства имен, которые позволяют настроить сетевое взаимодействие для каждого процесса по отдельности.

IPMininet [5] – это python библиотека, расширяющая функциональность Mininet. Она предназначена для эмуляции сложных IP сетей. Она предоставляет новые классы, такие как маршрутизаторы, и автоматически настраивает свойства, которые не были заданы пользователем, включая IP-адреса и конфигурационные файлы маршрутизаторов.

Этот гибкий и удобный инструмент для моделирования сетевых топологий любой сложности.

2.2. VXLAN

Virtual Extensible Local Area Network (VXLAN) [10] – протокол, расширяющий сеть L2, используя сетевую инфраструктуру L3. Он инкапсулирует Ethernet фреймы в UDP пакеты. Это может быть полезно в датацентрах, где виртуальные машины одного арендатора могут находиться на разных физических серверах или в разных датацентрах. VXLAN обеспечивает L2 связность и изоляцию трафика, предотвращая конфликты MAC адресов и позволяя виртуальным машинам, работающим на одном физическом оборудовании, функционировать независимо друг от друга.

В отличие от VLAN, которые ограничены 4096 идентификаторами, VXLAN использует 24-битный VNI (VXLAN Network Identifier), позволяющий разделить сеть более чем на 16 миллионов сегментов. Это еще одно из преимуществ VXLAN для крупных датацентров. Функционирование VXLAN туннелей обеспечивается специальными устройствами или программными компонентами, которые выступают в роли конечных точек туннеля. Их называют VTEP или VXLAN Tunnel Endpoint. Они отвечают за инкапсуляцию Ethernet фреймов в VXLAN-пакеты и декапсуляцию VXLAN-пакетов обратно в Ethernet фреймы. Таким образом L2 пакеты передаются по широковещательному домену, который пересекает L3 сеть. VTEP может создать множество таких туннелей для виртуальных машин, работающих на одном сервере. Эта технология позволяет, например, обеспечить бесшовную миграцию виртуальных машин с одного физического сервера на другой.

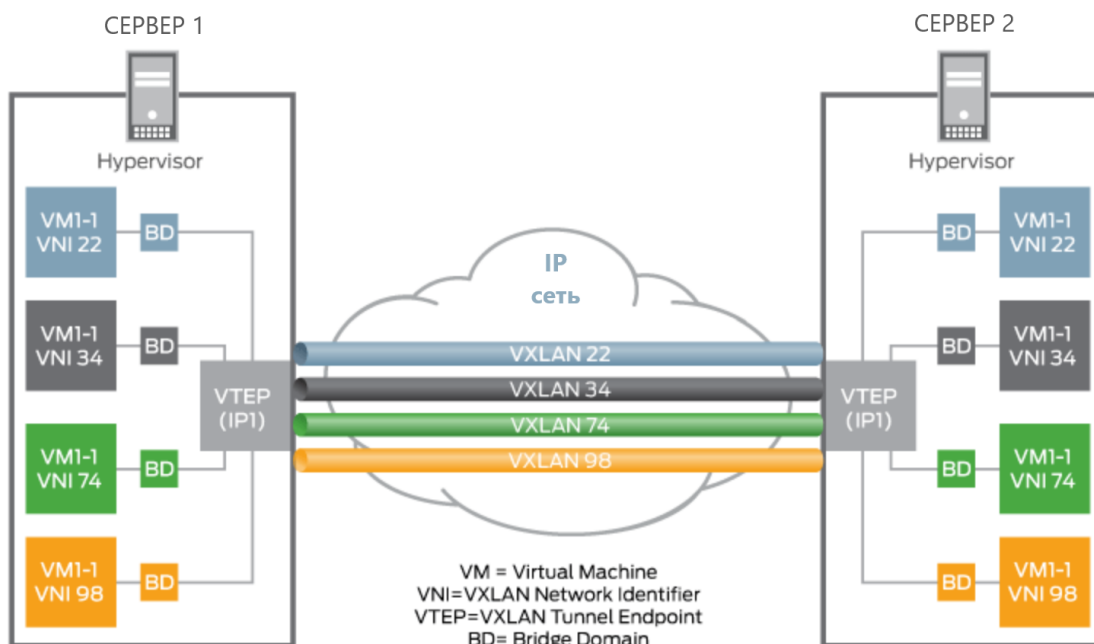


Рис. 1: Схема использования VXLAN

Современные датацентры используют VXLAN в сочетании с другими технологиями, такими как Software Defined Networking (SDN) и BGP EVPN (Border Gateway Protocol Ethernet VPN), что упрощает развертывание и управление сетью.

2.2.1. Формат пакета

Пакет VXLAN состоит из внешних заголовков, которые обеспечивают передачу пакета по сети, заголовка VXLAN который содержит флаги, VNI и зарезервированные поля, а также из оригинального Ethernet фрейма. Во внешнем IP заголовке указывается адрес VTEP назначения. Для заголовка UDP порт отправления вычисляется динамически, а порт назначения по умолчанию равен 4789.

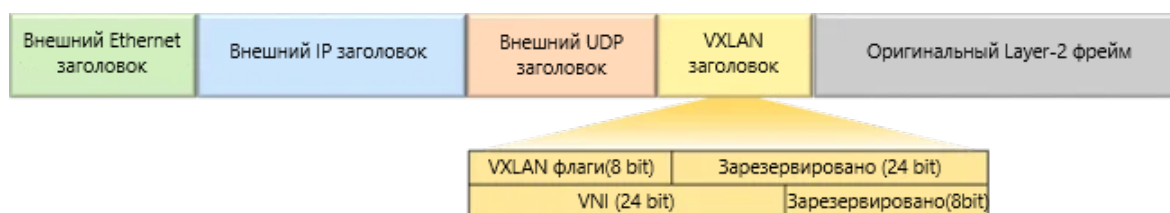


Рис. 2: Формат VXLAN пакета

2.2.2. Методы настройки VXLAN

Существует несколько методов настройки VXLAN, среди которых: unicast VXLAN, multicast VXLAN, EVPN VXLAN и VXLAN с использованием контроллера.

Unicast VXLAN – это самый простой вариант настройки VXLAN. В данном случае при настройке VTEP, необходимо для каждого VNI статически задать адреса всех удаленных VTEP, клиенты которых используют тот же самый VNI. Таким образом при отправке пакетов VTEP будет указывать в IP заголовке адрес удаленного VTEP. Если удаленных VTEP будет больше одного, то один и тот же пакет будет отправлен каждому VTEP по отдельности. Несмотря на то что VTEP может ассоциировать туннели с MAC адресами устройств и в будущем отправлять только один пакет, это является одним из недостатков, так как при большом количестве удаленных VTEP повышается нагрузка на сеть. Также при добавлении или удалении нового VTEP необходимо вручную изменять конфигурацию всех остальных VTEP.

Multicast VXLAN может автоматически находить удаленные VTEP в отличие от unicast VXLAN. При настройке VTEP для каждого VNI необходимо задать мультикаст группу. Каждый VTEP принимает пакеты только с необходимой ему мультикаст группой. Одним из преимуществ этого подхода является то, что при наличии нескольких удаленных VTEP нагрузка на сеть будет ниже в сравнении с unicast VXLAN. Для работы данного режима необходима поддержка мультикаст трафика.

EVPN VXLAN представляет собой продвинутый метод настройки VXLAN, основанный на использовании EVPN [2]. EVPN основан на BGP и используется для изучения MAC адресов между VTEP и поиска соседей. Таким образом список удаленных VTEP формируется с помощью EVPN. Использование этого подхода значительно сокращает количество трафика при изучении MAC адресов по сравнению с unicast VXLAN. Обычно именно этот подход используется для настройки сети в датацентрах, из за своей гибкости и масштабируемости.

VXLAN с использованием контроллера это режим в котором для настройки VTEP достаточно указать адрес контроллера, а вся остальная функциональность, такая как поиск соседей реализуется контроллером.

Наименее подходящими для задач обучения и внедрения в Mimirnet являются методы, использующие контроллер или EVPN. Это связано с их сложностью и абстракцией, которые могут скрывать важные детали, необходимые для понимания основ работы VXLAN. В то же время, unicast VXLAN является наилучшим вариантом для обучающих целей. Его простота в реализации и понимании делает его отличным выбором. Этот метод наглядно демонстрирует, как пакеты отправляются ко всем указанным VTEP, что помогает лучше усвоить материал. Кроме того, в Mimirnet ожидается работа с сетями небольшого размера, поэтому возможная проблема ручной настройки VTEP здесь теряет актуальность.

3. Описание решения

Для того чтобы реализовать технологию VXLAN в Mimirnet необходимо выбрать устройство, которое будет выступать в роли VTEP, спроектировать веб интерфейс и реализовать настройку VTEP на сервере.

Несмотря на то что в датацентрах в качестве VTEP в большинстве случаев выступают коммутаторы [1], в реализации для Mimirnet в качестве VTEP будут выступать маршрутизаторы. Дело в том, что для датацентров использование коммутаторов обусловлено архитектурой сети, а также физическим расположением серверов, а в случае Mimirnet мы не имеем таких ограничений. Также в Mimirnet коммутаторы это L2 устройства, интерфейсы которых не имеют IP адреса, а для настройки VTEP необходимо чтобы интерфейс, от имени которого отправляются инкапсулированные пакеты, имел IP адрес.

3.1. Разработка серверной части

После того как пользователь запустил эмуляцию, на сервер передается конфигурация сети для эмуляции. Информация, необходимая для настройки VXLAN содержится в следующих полях интерфейса.

```
vxlan_vni: Optional[int] = None
vxlan_connection_type: Optional[int] = None
vxlan_vni_to_target_ip: Optional[list[list[str]]] = None
```

Поле `vxlan_connection_type` обозначает роль которую выполняет интерфейс. Если оно равно 1, то интерфейс отвечает за взаимодействие с внешней сетью и за прием и отправку инкапсулированного трафика, а в поле `vxlan_vni_to_target_ip` содержится список IP удаленных VTEP, ассоциированных с VNI. Если оно равно 0, то это интерфейс который соединен с клиентским устройством и обменивается с ним деинкапсулированными пакетами, а в `vxlan_vni` содержится VNI с которым ассоциирован этот клиент.

Код настройки VTEP в Mimirnet выполняется перед началом эмуля-

ции, на этапе настройки сети. Сначала с помощью утилиты `iproute2` [7] выполняется настройка всех интерфейсов с `connection_type` равным 1. Для каждого VNI на интерфейсе выполняются следующие команды.

```
ip link add {vxlan_name} type vxlan id {vni} local {local_ip} \
dstport 4789 dev {intf}
ip link set {vxlan_name} up
brctl addbr {bridge_name}
brctl addif {bridge_name} {vxlan_name}
ip link set {bridge_name} up
```

Для каждого VNI создается отдельный виртуальный VXLAN интерфейс и мост, с которым он соединяется. После этого программа проходит по списку `vxlan_vni_to_target_ip` и добавляет в `fdb` (Forwarding Database) адреса удаленных VTEP с помощью следующей команды.

```
bridge fdb append 00:00:00:00:00:00 dev {vxlan_name} dst {target_ip}
```

После этого настраиваются клиентские интерфейсы с `connection_type` равным 0. Они просто подключаются к уже существующему мосту, который связан с виртуальным VXLAN интерфейсом.

```
brctl addif {bridge_name} {intf}
```

Одной из проблем стало то, что маршрутизаторы обмениваются между собой IGMP [6] пакетами, которые предназначены для обмена информацией о принадлежности к мультикаст группам. Они не несут никакой полезной информации для пользователя и поэтому их нужно фильтровать из результатов эмуляции. В Mimirnet реализован механизм фильтрации таких пакетов, но он не может фильтровать IGMP внутри VXLAN. Для решения этой проблемы был добавлен класс описывающий VXLAN пакет. С его помощью удалось отфильтровать IGMP пакеты внутри VXLAN пакетов.

3.2. Разработка веб-интерфейса

Веб-интерфейс разрабатывался с помощью Bootstrap, поскольку этот фреймворк уже используется в проекте. Для настройки VXLAN было

решено реализовать отдельную форму, потому что требуется предоставить пользователю удобный способ ввода множества параметров, таких как идентификатор VXLAN, адреса удаленных VTEP и VNI.

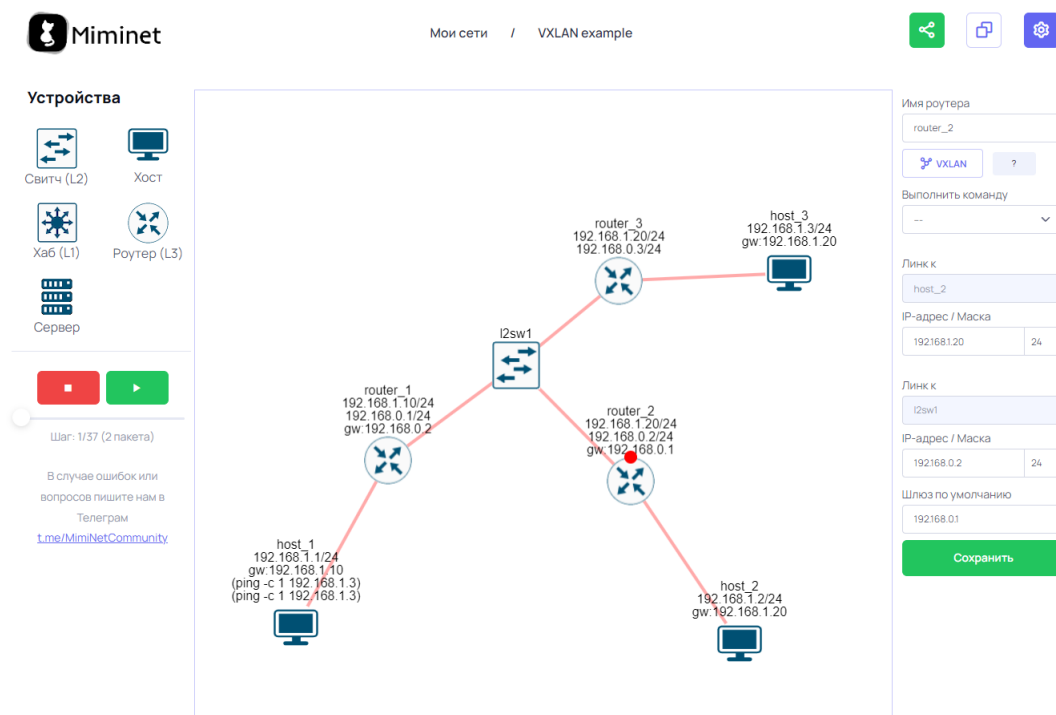


Рис. 3: Кнопка открытия формы на панели настройки

Настройка VXLAN

Неверный IP-адрес. Пожалуйста, введите действительный IPv4 адрес.

☒ VXLAN

VNI:

Клиентский линк:

Добавить устройство

Линк к: host_2, VNI: 1000 Удалить

VNI:

IP целевого VTEP:

Сетевой линк:

Добавить интерфейс

Линк к: router_1, VNI: 1000, Удаленный IP: 192.168.0.1 Удалить

Заккрыть и сохранить

Рис. 4: Форма настройки VXLAN

В верхней части формы можно увидеть переключатель, с подписью VXLAN. Он позволяет включить VXLAN на маршрутизаторе, а также отчистить настройки. Верхняя форма позволяет задать интерфейс, к которому подключено клиентское устройство и ассоциировать его с определенным VNI. Клиентскими являются устройства, которые отправляют на VTEP пакеты для транспортировки по туннелю, а также принимают декапсулированные пакеты.

Нижняя форма позволяет создать интерфейс для отправки и получения пакетов через туннель. Это работает следующим образом:

- Вы можете задать IP-адрес удаленного VTEP и ассоциировать его с VNI.
- Если VNI клиентского устройства совпадает с VNI, привязанным

к IP-адресу удаленного VTEP, устройство может отправлять пакеты на этот удаленный VTEP.

Таким образом, устройство с определенным VNI получает доступ к отправке данных на соответствующий удаленный VTEP.

Добавленные настройки отображаются под формой и могут быть удалены с помощью соответствующей кнопки. Выбирать интерфейсы в форме позволяет выпадающий список, содержащий имена соседних устройств. Также в форме реализована проверка ввода, которая запрещает добавлять записи с некорректными IP адресами и VNI и выводит соответствующее предупреждение.

Заключение

В результате работы над учебной практикой в течение весеннего семестра были выполнены следующие задачи:

- Изучен принцип работы VXLAN.
- Unicast VXLAN выбран, как оптимальный способ настройки VXLAN для Mimirnet.
- Разработана форма для настройки VXLAN.
- На бэкенде реализована эмуляция VXLAN.
- После внесенных изменений фронтенд и бэкенд взаимодействуют корректно.

Исходный код можно увидеть в GitHub репозитории¹. Был выполнен PR в основной репозиторий².

¹<https://github.com/foreverjun/mimirnet>

²<https://github.com/mimi-net/mimirnet/pull/144>

Список литературы

- [1] A Brief Overview of VXLAN EVPN.— URL: https://www.researchgate.net/publication/352999040_A_Brief_Overview_of_VXLAN_EVPN (дата обращения: 22 сентября 2024 г.).
- [2] EVPN RFC.— URL: <https://www.rfc-editor.org/rfc/rfc9135.html> (дата обращения: 22 сентября 2024 г.).
- [3] Enhancing Data Center Performance using Virtual Extensible LAN (VXLAN) Emerging Technology.— URL: https://www.researchgate.net/publication/383272592_Enhancing_Data_Center_Performance_using_Virtual_Extensible_LAN_VXLAN_Emerging_Technology (дата обращения: 22 сентября 2024 г.).
- [4] GNS3.— URL: <https://www.gns3.com/> (дата обращения: 22 сентября 2024 г.).
- [5] IPMininet.— URL: <https://github.com/cnp3/ipmininet> (дата обращения: 22 сентября 2024 г.).
- [6] Internet Group Management Protocol.— URL: <https://datatracker.ietf.org/doc/html/rfc3376> (дата обращения: 22 сентября 2024 г.).
- [7] Iproute2.— URL: <https://wiki.linuxfoundation.org/networking/iproute2#documentation> (дата обращения: 22 сентября 2024 г.).
- [8] Mininet.— URL: <https://github.com/mininet/mininet> (дата обращения: 22 сентября 2024 г.).
- [9] RabbitMQ.— URL: <https://www.rabbitmq.com/> (дата обращения: 22 сентября 2024 г.).
- [10] VXLAN RFC.— URL: <https://www.rfc-editor.org/rfc/rfc7348.txt> (дата обращения: 22 сентября 2024 г.).

[11] VirtualBox. — URL: <https://www.virtualbox.org/> (дата обращения: 22 сентября 2024 г.).