

Разработка алгоритма поиска в таблице классификации сетевого процессора

1 Постановка задачи

Основным функциональным элементом сетевых устройств (коммутаторов, маршрутизаторов) является специализированный процессор для обработки трафика – сетевой процессор. Сетевой процессор выполняет классификацию пакета (т.е. идентификацию по значениям полей заголовка), его модификацию и пересылку на основе принадлежности к какому-либо потоку (классу) пакетов. Сетевой процессор хранит таблицы классификации, содержащие правила, устанавливающие соответствие наборов полей заголовка пакета и действий, которые необходимо выполнить над пакетом.

В настоящее время получили активное развитие программируемые сетевые процессоры, позволяющие менять как алгоритм обработки пакетов, так и поддерживаемые поля заголовков. Разработка таких устройств актуальна в свете развивающихся и часто сменяющих друг друга протоколов, применяемых, например, в центрах обработки данных, при разработке сетей 5G и интернета вещей.

Сетевые устройства содержат также управляющий процессор общего назначения, на котором запущена операционная система устройства. Управляющий процессор реализует функцию управления передачей данных (в традиционных сетях) или реализует интерфейс связи с контроллером, реализующим эту функцию в программно-конфигурируемых сетях (ПКС). При этом управляющий процессор выполняет настройку сетевого процессора, т.е. модифицирует алгоритм обработки пакетов, поддерживающие заголовки, и управляет наборами правил в таблицах классификации сетевого процессора.

Основным в работе сетевого процессора является процесс классификации, то есть

поиск одного из полей заголовка пакета по некоторым правилам в таблице классификации.

В рамках задания предлагается реализовать алгоритм поиска в таблице классификации, представленной в указанной структуре данных. Для проверки алгоритма поиска (классификации) необходимо реализовать минимальное тестовое окружение, которое на вход будет получать набор сетевых пакетов, а на выходе сохранять классифицированные сетевые пакеты.

2 Алгоритмы работы сетевого процессора

2.1 Базовая маршрутизация

Опишем работу простейшего маршрутизатора. Маршрутизатор должен отправить пакет на следующее устройство на маршруте, устройства идентифицируются по IP адресам. Алгоритмы построения маршрутов находятся в ведении контроллера сети, и к данному заданию не относятся. Программа для сетевого процессора маршрутизатора должна работать следующим образом:

1. Разрешается обработка пакетов, содержащих заголовок IPv4 (заголовок сетевого уровня), остальные пакеты сбрасываются.
2. Определение маршрута пакетов осуществляется путём поиска по наибольшему совпадению префикса поля в заголовке сетевого уровня (например DST IP) с ключом в таблице классификации, по результатам поиска должен IP адрес следующего на маршруте устройства (IP next hop), структуру записи такой таблицы можно посмотреть в разделе 2.3.2.
3. В таблице классификации должна быть задана запись «по умолчанию», которая должна выбираться, если длина искомого префикса равна 0. Тогда IP next hop устанавливается в 0, а пакет сразу отправляется на порт связи с контроллером (порт управления).
4. В случае выбора записи не «по умолчанию» по результатам поиска в таблице классификации, необходимо определить MAC адрес next hop путём поиска в таб-

лице соответствия IP адресов и MAC адресов. Поиск производится по точному совпадению IP next hop и ключа в таблице ARP (раздел 2.3.3).

5. При неуспешном поиске MAC next hop пакет отправляется на порт управления.
6. При успешном поиске MAC адреса next hop значение поля в заголовке канального уровня (DST MAC) пакета должно быть изменено на MAC адрес next hop. Таблица классификации не должна содержать двух полей с одинаковыми ключами.
7. Далее необходимо определить номер выходного порта, на который следует отправить сетевой пакет, путём поиска по точному совпадению в таблице MAC-VLAN (Раздел 2.3.1).

2.2 Методы поиска по таблицам классификации

Основными методами поиска по таблицам классификации являются:

- Поиск по наибольшему совпадению префикса. Результатом такого поиска является запись, ключ которой имеет наибольшее совпадение по префиксу с искомым значением. Если несколько правил являются результатом поиска, выбирается первое правило по порядку вхождения в таблицу классификации.
- Поиск по точному совпадению. Результатом такого поиска является запись, ключ которой имеет полное совпадение с искомым значением.

2.3 Таблицы классификации

Для работы в сети для маршрутизации трафика используются таблицы классификации, которые представляют из себя таблицу, в которой есть поля идентификаторов (поля ключей обозначаются через $<\dots>$), и поля результата (искомое значение, обозначаются через $\{ \dots \}$).

2.3.1 Таблица MAC-VLAN

Данная таблица используется в, когда необходимо выполнить коммутацию пакета. Таблица состоит из записей вида: <MAC адрес>, <VLAN ID>, {<Номер выходного порта>}.

- MAC адрес — 48 битный адрес устройства (IEEE 802.3);
- VLAN ID — метка ID виртуального канала канального уровня (IEEE 802.1Q);
- Номер выходного порта — Идентификатор выходного порта, куда необходимо отправить пакет, если он соответствует данному правилу.

2.3.2 Аппаратная таблица маршрутизации (Forwarding Information Base, FIB)

Необходима для определения IP next hop. Таблица состоит из записей вида: <IP-адрес получателя>, <длина префикса>, {<IP next-hop>, <VLAN ID>}

- IP-адрес получателя — IPv4 адрес получателя;
- длина префикса — точная длина префикса IPv4 адреса получателя, по которому производится поиск;
- {<IP next-hop>, <VLAN ID>} — IP адрес следующего узла, куда необходимо отправить пакет и метка VLAN.

2.3.3 Таблица ARP

Необходима для получения MAC next-hop по заданному IP next-hop, найденному в таблице FIB. Содержит записи вида <IP next-hop>, {<MAC next-hop>}.

3 Задача

3.1 Основное задание

Данное задание является обязательным и оценивается в 5 баллов. Реализовать алгоритм поиска по таблице классификации на любом языке программирования (рекомендуются Python3/C++/Rust) и тестовое окружение. Реализация должна включать в себя:

- Функциональность для получения на вход PCAP файлов с сетевыми пакетами.

- Функциональность для получения на вход JSON файлов описывающих таблицы классификации.
- Структуру данных: бинарное дерево поиска (https://en.wikipedia.org/wiki/Binary_search_tree) для хранения таблицы классификации MAC-VLAN (раздел 2.3.1) и поиска по ней.
- Алгоритм поиска по структуре данных по точному совпадению поля DST MAC:VLAN заголовка канального уровня сетевого пакета в таблице MAC-VLAN,
- Функциональность вывода в PCAP файлы классифицированных сетевых пакетов, в соответствие с найденными в таблице номерами выходных портов.
- Тесты для реализованного алгоритма поиска: набор таблиц классификации MAC-VLAN, набор входных и выходных сетевых пакетов, которые подтверждают корректную работу реализованного алгоритма.

3.2 Дополнительное задание

Данное задание является НЕ обязательным и оценивается в 5 баллов.

В качестве дополнения к основному заданию предлагается расширить реализованный алгоритм поиска следующим образом:

- Реализовать структуру данных: В-дерево для хранения таблиц классификации MAC-VLAN(раздел 2.3.1), FIB (раздел 2.3.2), ARP(раздел 2.3.3).
- Реализовать возможность поиска по наибольшему совпадению префикса поля DST IP заголовка IPv4 сетевого пакета и ключа в таблице FIB.
- Сравнить производительность бинарного дерева поиска по сравнению с реализованной структурой данных.
- Разработать тесты для реализованного алгоритма поиска: набор таблиц классификации MAC-VLAN, FIB, ARP, набор входных и выходных сетевых пакетов, которые подтверждают корректную работу реализованного алгоритма.

Реализовать возможность классификации по нескольким таблицам классификации, а именно:

1. поиск по наибольшему совпадению префикса поля IP DST заголовка IPv4 сетевого пакета с ключом поиска таблицы FIB для получения IP next-hop,
2. поиск по точному совпадению IP next hop и ключу поиска в таблице ARP для получения MAC next hop,
3. поиск по точному совпадению MAC next hop и ключу поиска в таблице MAC-VLAN для получения номера выходного порта.

4 Формат входных данных

Входными данными разработанного алгоритма являются таблицы классификации в формате JSON и входящие пакеты для программы в формате PCAP. Примеры доступны по ссылке: https://github.com/Rav263/pri_task. Для создания пакетов можно воспользоваться библиотекой Scapy (<https://scapy.net/>). Для просмотра PCAP файлов можно пользоваться утилитой tcpdump (<https://www.tcpdump.org/manpages/tcpdump.1.html>) или средством Wireshark (<https://www.wireshark.org/>).

5 Формат выходных данных

Выходными данными разработанного алгоритма являются файлы PCAP вида «port_<X>.pcap», содержащие набор пакетов, которые по итогам классификации были отправлены на порт с номером <X> сетевого процессора. Для сброшенных пакетов следует создать отдельный PCAP файл, «dropped.pcap»

5.1 Требования к реализации тестов

Предлагается разработать систему автоматического тестирования, которая будет составлять отчёт в зависимости от корректности прохождения того или иного теста. Как пример библиотеки для создания такой системы можно указать для Python3: pytest <https://docs.pytest.org/en/stable/>; для C++: Google Test (GTest); для Rust: cargo test.

6 Требования к оформлению выполненного задания

1. Описание разработанного алгоритма классификации, таблиц классификации, разработанных тестов и процесса запуска алгоритма поиска с разработанными тестами представлено в файле текстового формата .pdf или .doc.
2. Исходный код не должен выкладываться в публичные репозитории (github и пр.).
3. Результатом выполнения работы является архив со всеми исходниками, тестами, описанием процесса запуска и инструкцией по установке дополнительных пакетов (при их использовании).

Баллы за основное и дополнительные задания суммируются. Если при проверке задания выявлен плагиат, оценка за все части задания обнуляется; проверяющие не проводят анализ, чье задание является «первоисточником», а чье – «займствованием».

При выполнении заданий необходимо разделять основное и дополнительные задания. Для простоты выполнения и проверки заданий тесты также рекомендуется разделять.

7 Отправка работ

1. Результаты выполнения заданий отправляются на адрес: asvk-nabor@asvk.cs.msu.ru
2. Тема письма должна иметь вид: [номер группы] – NPU3 – Фамилия Имя, например: [201] – NPU3 – Самоваров Иван.
3. Материалы основного и дополнительных (при наличии) заданий должны быть оформлены в виде отдельных архивов формата .tar.bz2 или .zip.