

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА
ВЕЛИКОГО

ФИЗИКО-МЕХАНИЧЕСКИЙ ИНСТИТУТ

ВЫСШАЯ ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ
ФИЗИКИ

Компьютерные сети

Отчёт по лабораторной работе №2

“Реализация протокола маршрутизации Open Shortest
Path First”

Выполнил:

Студент: Сачук Александр

Группа: 5040102/30201

Принял:

к. ф.-м. н., доцент

Баженов Александр Николаевич

2024 г.

Содержание

1. Постановка задачи	2
2. Теория	2
3. Реализация	2
4. Результаты	3
5. Обсуждение	4
6. Приложения	5

1. Постановка задачи

Требуется реализовать протокол маршрутизации OSPF и проверить работоспособность протокола для следующих видов топологии: линейная, кольцевая, звёздная. Проверить возможность перестройки таблиц достижимости в случае стохастического разрыва связи.

2. Теория

Протокол маршрутизации OSPF (Open Shortest Path First – алгоритм предпочтительного выбора кратчайшего маршрута) предназначен для работы в сетях множественного доступа, т. е. сетях, у которых может быть несколько маршрутизаторов, способных общаться друг с другом. Основой работы данного протокола является представление множества сетей, маршрутизаторов и каналов в виде ориентированного графа. Такое представление позволяет учитывать различные условия и ограничения при выборе кратчайшего пути между любыми двумя маршрутизаторами, а также делить большие системы на области, каждая из которых может обладать своей собственной топологией, условиями выбора маршрутов и другими особенностями.

Принцип работы заключается в следующем:

- После включения маршрутизаторов протокол ищет непосредственно подключённых соседей и устанавливает с ними связь
- Затем они обмениваются друг с другом информацией о подключённых и доступных им сетях. То есть они строят карту сети (граф сети). Данная карта одинакова на всех маршрутизаторах
- На основе полученной информации запускается алгоритм SPF (Shortest Path First, “выбор наилучшего пути”), который рассчитывает оптимальный маршрут к каждой сети. Данный процесс представляет из себя поиск кратчайшего пути в графе, вершинами которого являются доступные сети, а рёбрами – пути между сетями.

3. Реализация

Симуляция системы реализована на языке программирования Python. Для создания симуляции работы протокола маршрутизации реализованы следующие абстракции:

- Router: базовый элемент сети, выполняющий основные элементы симуляции: инициализацию, отправка информации о соседях и её получение, "отключение" и прочее;
- Designated router: выделенный маршрутизатор. Управляет процессом рассылки LSA (link-state advertisement, объявление о состоянии канала) в сети. С

данным маршрутизатором каждый маршрутизатор сети устанавливает отношение смежности для отправки информации об изменении сети;

- Message: структура из пары "данные"+"значение". Формируются роутерами и пересылаются по соединениям;

- Connection: пара очередей, для реализации взаимодействия.

Симуляция реализована на ЯП Python 3.10, где каждый элемент сети работает параллельно. Для ее работы устанавливается время рассылки сообщений (было принято равным 10 секунд), а также вероятность отключения узла сети (было принято равным 0.25).

4. Результаты

Рассмотрим пример работы программы для линейной топологии с 3 узлами. Здесь и далее: узлы – указаны их номера, связи – список номеров соседних узлов на позиции текущего узла.

- Узлы [0, 1, 2]
- Связи [[1], [0, 2], [1]]

К сети подключены все 3 узла. Кратчайшие пути:

- 0: [[0], [0, 1], [0, 1, 2]]
- 1: [[1, 0], [1], [1, 2]]
- 2: [[2, 1, 0], [2, 1], [2]]

От сети отключен 2-ой узел. Новые кратчайшие пути:

- 0: [[0], [0, 1], []]
- 1: [[1, 0], [1], []]
- 2: [[], [], [2]]

Теперь рассмотрим пример работы программы для кольцевой топологии с 4 узлами.

- Узлы [0, 1, 2, 3]
- Связи [[3, 1], [0, 2], [1, 3], [2, 0]]

К сети подключены все 4 узла. Кратчайшие пути:

- 0: [[0], [0, 1], [0, 1, 2], [0, 3]]
- 1: [[1, 0], [1], [1, 2], [1, 0, 3]]
- 2: [[2, 1, 0], [2, 1], [2], [2, 3]]
- 3: [[3, 0], [3, 0, 1], [3, 2], [3]]

От сети отключен 0-ой узел. Новые кратчайшие пути из соответствующих узлов:

- 1: $[\]$, $[1]$, $[1, 2]$, $[1, 2, 3]$
- 2: $[\]$, $[2, 1]$, $[2]$, $[2, 3]$
- 3: $[\]$, $[3, 2, 1]$, $[3, 2]$, $[3]$

Наконец, рассмотрим пример работы программы для звездной топологии с 5 узлами. Центр в узле с индексом 0.

- Узлы $[0, 1, 2, 3, 4]$
- Связи $[[1, 2, 3, 4], [0], [0], [0], [0]]$
- 0: $[[0], [0, 1], [0, 2], [0, 3], [0, 4]]$
- 1: $[[1, 0], [1], [1, 0, 2], [1, 0, 3], [1, 0, 4]]$
- 2: $[[2, 0], [2, 0, 1], [2], [2, 0, 3], [2, 0, 4]]$
- 3: $[[3, 0], [3, 0, 1], [3, 0, 2], [3], [3, 0, 4]]$
- 4: $[[4, 0], [4, 0, 1], [4, 0, 2], [4, 0, 3], [4]]$

От сети отключен 3-ий узел. Новые кратчайшие пути:

- 0: $[[0], [0, 1], [0, 2], [\], [0, 4]]$
- 1: $[[1, 0], [1], [1, 0, 2], [\], [1, 0, 4]]$
- 2: $[[2, 0], [2, 0, 1], [2], [\], [2, 0, 4]]$
- 4: $[[4, 0], [4, 0, 1], [4, 0, 2], [\], [4]]$

От сети отключен 0-ой, центральный узел. Новые кратчайшие пути отсутствуют, т.к. узлы стали отдельными областями связности:

- 1: $[\]$, $[1]$, $[\]$, $[\]$, $[\]$
- 2: $[\]$, $[\]$, $[2]$, $[\]$, $[\]$
- 3: $[\]$, $[\]$, $[\]$, $[3]$, $[\]$
- 4: $[\]$, $[\]$, $[\]$, $[\]$, $[4]$

5. Обсуждение

Был реализован и протестирован на различных топологиях протокол OSPF.

Данная программа была проверена на трех разных топологиях в течение длительного времени. Ввиду сходящихся с теорией результатов, можно сделать вывод о корректной работе симуляции на представленных топологиях (линейная, кольцо, звезда).

6. Приложения

1. Репозиторий с кодом программы и кодом отчёта:

<https://github.com/AS2/comp_networks/tree/develop/lab2/>