Санкт-Петербургский политехнический университет Петра Великого
Высшая школа прикладной математики и вычислительной физики
Кафедра прикладной математики

Лабораторная работа
по дисциплине «Компьютерные сети»
на тему

Реализация протокола динамической маршрутизации Open Shortest Path First

Выполнил
студент гр. 5040102/10201 Долгий В.С. /_____/

Руководитель
доцент, к.ф.-м.н. Баженов А.Н. /_____/

Санкт-Петербург
2022

Постановка задачи

Реализовать систему роутеров, объединяющихся в сеть с помощью протокола Open Shortest Path First [1]. Рассмотреть таблицы кратчайших путей в сетях с линейной, звездной и кольцевой топологией. Также рассмотреть перестроение таблиц в случае выхода из строя одного или нескольких роутеров.

Реализация

Модель реализована на языке Python. Все роутеры работают в отдельных потоках, создаваемых с использованием модуля threading.

Для обмена сообщениями о состоянии каналов (LSU) используется выделенный Designated Router (DR), с которым связаны все остальные маршрутизаторы. Каналы связи будем считать ориентированными, стоимость передачи данных по всем каналам – одинаковой.

Код проекта выложен на GitHub:

https://github.com/Ragnarok7861/CompNET

Результаты

Для начала рассмотрим процесс формирования линейной топологии. В ней роутеры соединены в цепочку, у каждого роутера, за исключением первого и последнего по 2 соседа, у первого и последнего – по одному. Ниже представлены сообщения, получаемые каждым роутером от DR.

```
router0 got: "LSU 1->0, 1->2"

router0 got: "LSU 2->1, 2->3"
router1 got: "LSU 2->1, 2->3"

router0 got: "LSU 3->2, 3->4"
router1 got: "LSU 3->2, 3->4"
router2 got: "LSU 3->2, 3->4"

router0 got: "LSU 4->3"
router1 got: "LSU 4->3"
router2 got: "LSU 4->3"
router3 got: "LSU 4->3"
```

Роутеры подключались по очереди в порядке возрастания индексов. В силу особенностей реализации можно заметить, что при «знакомстве» с соседями роутерам не приходили сообщения с текущим состоянием полной таблицы достижимости на момент их включения, так как она уже была им известна при включении.

Рассмотрим таблицу кратчайших путей в случае линейной топологии для пяти роутеров:

```
shortest ways from 0:
(0, 0): 0
```

```
(0, 1): 0 -> 1
(0, 2): 0 -> 1 -> 2
(0, 3): 0 -> 1 -> 2 -> 3
(0, 4): 0 -> 1 -> 2 -> 3 -> 4
shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): 1 -> 2 -> 3 -> 4
shortest ways from 2:
(2, 0): 2 -> 1 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): 2 -> 3 -> 4
shortest ways from 3:
(3, 0): 3 -> 2 -> 1 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): 3 -> 4
shortest ways from 4:
(4, 0): 4 -> 3 -> 2 -> 1 -> 0
(4, 1): 4 -> 3 -> 2 -> 1
(4, 2): 4 -> 3 -> 2
(4, 3): 4 -> 3
(4, 4): 4
```

Между любой парой роутеров существует связь, при этом, как и ожидалось, для передачи сообщений между роутерами на разных концах цепочки потребуется существенно больше промежуточных звеньев, чем между соседними.

Теперь смоделируем отключение первого роутера:

```
router0 got: "LSU 1 dropped"
router2 got: "LSU 1 dropped"
router3 got: "LSU 1 dropped"
router4 got: "LSU 1 dropped"
```

Рассмотрим новую таблицу кратчайших путей:

```
shortest ways from 0:
(0, 0): 0
(0, 1): None
(0, 2): None
(0, 3): None
(0, 4): None
shortest ways from 2:
(2, 0): None
(2, 1): None
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): 2 -> 3 -> 4
shortest ways from 3:
(3, 0): None
(3, 1): None
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): 3 -> 4
shortest ways from 4:
(4, 0): None
```

```
(4, 1): None
(4, 2): 4 -> 3 -> 2
(4, 3): 4 -> 3
(4, 4): 4
```

Нулевой роутер оказался изолирован от сети, и потерял связь со всеми остальными. Второй, третий и четвёртый при этом сохранили связь между собой, но потеряли связь с первым, и как следствие, с нулевым.

Теперь также выведем из строя последний роутер:

```
router0 got: "LSU 4 dropped"
router2 got: "LSU 4 dropped"
router3 got: "LSU 4 dropped"
```

```
shortest ways from 0:
(0, 0): 0
(0, 1): None
(0, 2): None
(0, 3): None
(0,  4):  None
shortest ways from 2:
(2, 0): None
(2, 1): None
(2, 2): 2
(2, 3): 2 -> 3
(2,  4):  None
shortest ways from 3:
(3, 0): None
(3, 1): None
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): None
```

Как и ожидалось, в сети осталось только 3 рабочих роутера, при этом нулевой всё ещё изолирован, а второй и третий могут обмениваться информацией между собой.

Восстановим работу первого роутера:

```
router0 got: "LSU 1->0, 1->2"
router0 got: "LSU 2->1"
router1 got: "LSU 2->1"
router1 got: "LSU 0->1"
router1 got: "LSU 4 dropped"
router2 got: "LSU 1->0, 1->2"
router2 got: "LSU 0->1"
router3 got: "LSU 1->0, 1->2"
router3 got: "LSU 2->1"
router3 got: "LSU 0->1"
```

Убедимся, что таблица кратчайших путей корректно перестроилась:

```
shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 1
(0, 2): 0 -> 1 -> 2
(0, 3): 0 -> 1 -> 2 -> 3
(0,  4):  None
shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): None
```

```
shortest ways from 2:
(2, 0): 2 -> 1 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2,  4):  None
shortest ways from 3:
(3, 0): 3 -> 2 -> 1 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): None
```

Связь между всеми роутерами, кроме неработающего последнего, восстановилась.

Теперь рассмотрим таблицу кратчайших путей для звездной топологии. В ней все роутеры соединены с центральным. В нашем случае центральным является роутер под номером два:

```
shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 2 -> 1
(0, 2): 0 -> 2
(0, 3): 0 -> 2 -> 3
(0, 4): 0 -> 2 -> 4
shortest ways from 1:
(1, 0): 1 -> 2 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): 1 -> 2 -> 4
shortest ways from 2:
(2, 0): 2 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): 2 -> 4
shortest ways from 3:
(3, 0): 3 -> 2 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): 3 -> 2 -> 4
shortest ways from 4:
(4, 0): 4 -> 2 -> 0
(4, 1): 4 -> 2 -> 1
(4, 2): 4 -> 2
(4, 3): 4 -> 2 -> 3
(4, 4): 4
```

Видим, что любой роутер может обмениваться сообщениями с центральным напрямую, а любая другая пара роутеров – через центральный.

Смоделируем отключение центрального роутера:

```
router0 got: "LSU 2 dropped"
router1 got: "LSU 2 dropped"
router3 got: "LSU 2 dropped"
router4 got: "LSU 2 dropped"
```

```
shortest ways from 0:
(0, 0): 0
(0, 1): None
(0, 2): None
```

```
(0, 3): None
(0,  4):  None
shortest ways from 1:
(1, 0): None
(1, 1): 1
(1, 2): None
(1, 3): None
(1,  4):  None
shortest ways from 3:
(3, 0): None
(3, 1): None
(3, 2): None
(3, 3): 3
(3,  4):  None
shortest ways from 4:
(4, 0): None
(4, 1): None
(4, 2): None
(4, 3): None
(4, 4): 4
```

Как и ожидалось, вся сеть развалилась на 4 независимых роутера, которые не в состоянии обмениваться сообщениями друг с другом.

Теперь проведём аналогичный эксперимент в случае кольцевой топологии. Она совпадает с линейной за исключением того, что первый и последний роутеры в цепочке также соединены. Так выглядит таблица кратчайших путей для кольцевой топологии:

```
shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 1
(0, 2): 0 -> 1 -> 2
(0, 3): 0 -> 4 -> 3
(0, 4): 0 -> 4
shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): 1 -> 0 -> 4
shortest ways from 2:
(2, 0): 2 -> 1 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): 2 -> 3 -> 4
shortest ways from 3:
(3, 0): 3 -> 4 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): 3 -> 4
shortest ways from 4:
(4, 0): 4 -> 0
(4, 1): 4 -> 0 -> 1
(4, 2): 4 -> 3 -> 2
(4, 3): 4 -> 3
(4, 4): 4
```

Теперь опять смоделируем отключение второго роутера:

```
shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 1
(0, 2): None
(0, 3): 0 -> 4 -> 3
(0, 4): 0 -> 4
shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): None
(1, 3): 1 -> 0 -> 4 -> 3
(1, 4): 1 -> 0 -> 4
shortest ways from 3:
(3, 0): 3 -> 4 -> 0
(3, 1): 3 -> 4 -> 0 -> 1
(3, 2): None
(3, 3): 3
(3, 4): 3 -> 4
shortest ways from 4:
(4, 0): 4 -> 0
(4, 1): 4 -> 0 -> 1
(4, 2): None
(4, 3): 4 -> 3
(4, 4): 4
```

Видим, что связность в сети сохранилась, любые два работающих роутера всё ещё в состоянии обмениваться сообщениями, хотя пути между первым и третьим стали после отключения второго стали длиннее.

Теперь отключим ещё четвертый роутер:

```
shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 1
(0, 2): None
(0, 3): None
(0, 4): None
shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): None
(1, 3): None
(1,  4):  None
shortest ways from 3:
(3, 0): None
(3, 1): None
(3, 2): None
(3, 3): 3
(3, 4): None
```

В результате третий роутер оказался изолирован, а поддерживать связь могут только нулевой и первый.

Выводы

В рамках работы была реализована программная модель, позволяющая объединять произвольное количество роутеров в сети с помощью протокола Open Shortest Path First. Корректность работы алгоритма построения кратчайших путей подтверждена тестированием на трёх видах топологий сети.

По результатам сравнения свойств различных топологий, в предположении равенства весов всех каналов связи, можно сделать следующие выводы:

- При линейной топологии максимальная длина пути между узлами может достигать $n - 1$, где n – число роутеров в сети. При этом выход из строя любого узла, кроме крайних, нарушает связность сети, и разбивает её на две компоненты.
- При звездной топологии максимальная длина пути при любом числе узлом не превосходит 2. Выход из строя любого нецентрального узла не нарушает связность, но при отключении центрального роутера сеть полностью «разваливается», и все остальные узлы оказываются изолированными.
- При кольцевой топологии максимальная длина пути в 2 раза меньше, чем в линейной, и может быть порядка $^{n}/_{2}$. При этом в отличие от линейно топологии, выход из строя одного любого узла не нарушает связность, но при выходе из строя двух узлов сеть также распадается на две компоненты.

Использованная литература

1. А.Н. Баженов, Компьютерные сети, курс лекций