

# 计算机网络 Lab7

---

## 一、实验任务

---

### 实现思路

#### 1.LPR

**基本思路：** LPR（左路径路由）算法的核心思想是，当数据包到达某个交换机时，选择与该交换机相连的编号较小的交换机作为下一跳。具体来说，LPR选择的路径遵循拓扑图中交换机的顺序，优先选择较左侧的路径。

#### 具体实现步骤

1. **接收数据包：** 在 `packet_in_handler` 函数中接收到 `packet_in` 事件，解析数据包，获取源地址 `src_addr` 和目标地址 `dest_addr`，以及数据包触发的交换机 `datapath.id`。
2. **调用 `addr_get` 函数：** 根据源地址和目的地址，调用 `addr_get` 函数来计算下一跳的地址。此时，LPR通过判断与当前交换机相连的交换机编号，选择编号较小的那个。
3. **更新路由表：** 在 `packet_in_handler` 中，更新路由表，设置相应的路由规则，确保数据包能够按照LPR的策略进行转发。
4. **发送数据包：** 计算出下一跳的交换机后，将数据包继续转发到下一跳

---

#### 2.RSR

**基本思路：** RSR（随机化最短路径路由）算法与LPR类似，但在选择下一跳交换机时，加入了随机性。在多个候选路径中，RSR使用 `random.randint` 函数来随机选择一条路径

### 具体实现步骤:

1. **接收数据包:** 在 `packet_in_handler` 中, 解析数据包并提取出源地址 `src_addr` 和目标地址 `dest_addr`, 以及触发事件的交换机编号 `datapath.id`。
  2. **调用 `addr_get` 函数:** 根据源地址和目标地址, 调用 `addr_get` 函数来计算下一跳的地址。与LPR不同的是, RSR在选择候选路径时, 会引入一个随机选择机制, 使用 `random.randint` 在多个候选路径中随机选择一个。
  3. **更新路由表:** 同样需要更新路由表, 确保根据随机选择的路径正确转发数据包
  4. **发送数据包:** 根据RSR选择的路径, 将数据包转发到下一跳
- 

## 3.LLR

**基本思路:** LLR (负载学习路由) 算法的核心是选择负载最小的路径作为数据包的转发路径。在这种方法中, 每个交换机会维护一张 `path_store` 表, 记录路径的负载情况, 动态更新负载信息以实现自学习。

### 具体实现步骤:

1. **接收数据包:** 在 `packet_in_handler` 中解析数据包, 提取源地址 `src_addr` 和目标地址 `dest_addr`, 以及触发事件的交换机编号 `datapath.id`。
2. **调用 `addr_get` 函数:** 根据源地址和目标地址, 调用 `addr_get` 函数。与LPR和RSR不同, LLR在选择路径时, 会根据 `path_store` 中的负载信息选择负载最小的路径。
3. **更新 `path_store`:** 在每次转发数据包时, 更新 `path_store` 中的负载信息, 以便下次选择时可以根据实时的负载情况选择最优路径。
4. **更新路由表:** 在 `packet_in_handler` 中, 更新路由表, 并将最优路径应用到路由规则中

5. **发送数据包**：将数据包转发到负载最小的下一跳交换机。

## 运行命令

```
LPR
ryu-manager LPR.py --observe-links
sudo python3 parallel_traffic_generator.py

RSR
ryu-manager RSR.py --observe-links
sudo python3 parallel_traffic_generator.py

LLR
ryu-manager LLR.py --observe-links
sudo python3 sequential_traffic_generator.py
```

## 输出截图

### LPR

```
hyf@ubuntu:~/ComputerNetwork/lab7$ ryu-manager LPR.py --observe-links
loading app LPR.py
loading app ryu.controller.ofp_handler
loading app ryu.topology.switches
loading app ryu.controller.ofp_handler
instantiating app LPR.py of ProjectControllerRSR
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app ryu.topology.switches of Switches
Path: 10.0.0.16 -> 10.0.0.4 via DPID 8
Path: 10.0.0.16 -> 10.0.0.5 via DPID 8
Path: 10.0.0.16 -> 10.0.0.5 via DPID 15
Path: 10.0.0.16 -> 10.0.0.4 via DPID 16
Path: 10.0.0.16 -> 10.0.0.4 via DPID 20
Path: 10.0.0.16 -> 10.0.0.5 via DPID 17
Path: 10.0.0.16 -> 10.0.0.4 via DPID 10
Path: 10.0.0.16 -> 10.0.0.5 via DPID 11
Path: 10.0.0.16 -> 10.0.0.5 via DPID 3
Path: 10.0.0.16 -> 10.0.0.4 via DPID 2
Path: 10.0.0.16 -> 10.0.0.5 via DPID 8
Path: 10.0.0.16 -> 10.0.0.5 via DPID 15
Path: 10.0.0.16 -> 10.0.0.5 via DPID 17
Path: 10.0.0.16 -> 10.0.0.5 via DPID 11
Path: 10.0.0.16 -> 10.0.0.4 via DPID 8
Path: 10.0.0.16 -> 10.0.0.5 via DPID 3
Path: 10.0.0.16 -> 10.0.0.4 via DPID 15
Path: 10.0.0.16 -> 10.0.0.4 via DPID 17
```

## RSR

```
hyf@ubuntu:~/ComputerNetwork/lab7$ ryu-manager RSR.py --observe-links
loading app RSR.py
loading app ryu.controller.ofp_handler
loading app ryu.topology.switches
loading app ryu.controller.ofp_handler
instantiating app RSR.py of ProjectControllerRSR
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app ryu.topology.switches of Switches
Path: 10.0.0.16 -> 10.0.0.4 via DPID 8
Path: 10.0.0.16 -> 10.0.0.5 via DPID 8
Path: 10.0.0.16 -> 10.0.0.4 via DPID 16
Path: 10.0.0.16 -> 10.0.0.5 via DPID 16
Path: 10.0.0.16 -> 10.0.0.4 via DPID 20
Path: 10.0.0.16 -> 10.0.0.4 via DPID 10
Path: 10.0.0.16 -> 10.0.0.5 via DPID 19
Path: 10.0.0.16 -> 10.0.0.5 via DPID 12
Path: 10.0.0.16 -> 10.0.0.5 via DPID 3
Path: 10.0.0.16 -> 10.0.0.4 via DPID 2
Path: 10.0.0.16 -> 10.0.0.4 via DPID 8
Path: 10.0.0.16 -> 10.0.0.4 via DPID 16
Path: 10.0.0.16 -> 10.0.0.4 via DPID 20
Path: 10.0.0.16 -> 10.0.0.4 via DPID 10
Path: 10.0.0.16 -> 10.0.0.5 via DPID 8
Path: 10.0.0.16 -> 10.0.0.5 via DPID 15
Path: 10.0.0.16 -> 10.0.0.5 via DPID 18
Path: 10.0.0.16 -> 10.0.0.4 via DPID 2
Path: 10.0.0.16 -> 10.0.0.5 via DPID 11
```

## LLR

```
hyf@ubuntu:~/ComputerNetwork/lab7$ ryu-manager LLR.py --observe-links
loading app LLR.py
loading app ryu.topology.switches
loading app ryu.controller.ofp_handler
instantiating app LLR.py of ProjectControllerRSR
instantiating app ryu.topology.switches of Switches
instantiating app ryu.controller.ofp_handler of OFPHandler
ARP 10.0.0.1 -> 10.0.0.5 path = 1
ARP 10.0.0.5 -> 10.0.0.1 path = 3
IP 10.0.0.1 -> 10.0.0.5 path = 1
ARP 10.0.0.1 -> 10.0.0.6 path = 1
ARP 10.0.0.6 -> 10.0.0.1 path = 3
IP 10.0.0.1 -> 10.0.0.6 path = 1
ARP 10.0.0.2 -> 10.0.0.6 path = 1
ARP 10.0.0.6 -> 10.0.0.2 path = 3
ARP 10.0.0.3 -> 10.0.0.7 path = 2
ARP 10.0.0.7 -> 10.0.0.3 path = 4
```