

第十三周实验报告

网络路由实验（第二部分）

2015K8009922021

李一苇

一、实验内容

基于实验一，实现路由器计算路由表项的相关操作

二、实验流程

在实验一的基础上，得到了一致的链路状态数据库，本次实验据此生成路由表。

在实验一dump数据库条目的线程里调用 `generate_rib` 函数，该函数按照下面顺序依次执行下面各个子命令

1. 将链路状态数据库抽象成图拓扑

图拓扑的数据结构如下

```
#define MAX_NODE_NUM 255
#define INF 255
int graph[MAX_NODE_NUM][MAX_NODE_NUM]; //图的邻接矩阵
int visited[MAX_NODE_NUM];
int dist[MAX_NODE_NUM]; //从图结点0到结点i的最短跳数
int prev[MAX_NODE_NUM]; //从图结点0到结点i的最短路的最后一跳结点
int node_num; //节点数
u32 dic[MAX_NODE_NUM]; //存储从图的结点i到结点id(dic[i])的映射
```

◦ 图拓扑数据结构的初始化 `init_graph`

生成 `dic` 数组，其中 `dic[0] = instance->router_id` 代表本节点id

◦ 建边 `generate_graph`

注意可以直接建双向边，因为一致链路数据库条目一定是互通的

2. 计算最短路径

- 在基础的dijkstra算法上，增加记录 `prev` 数组记录到结点i的最后一跳的图结点编号

3. 生成网络路由

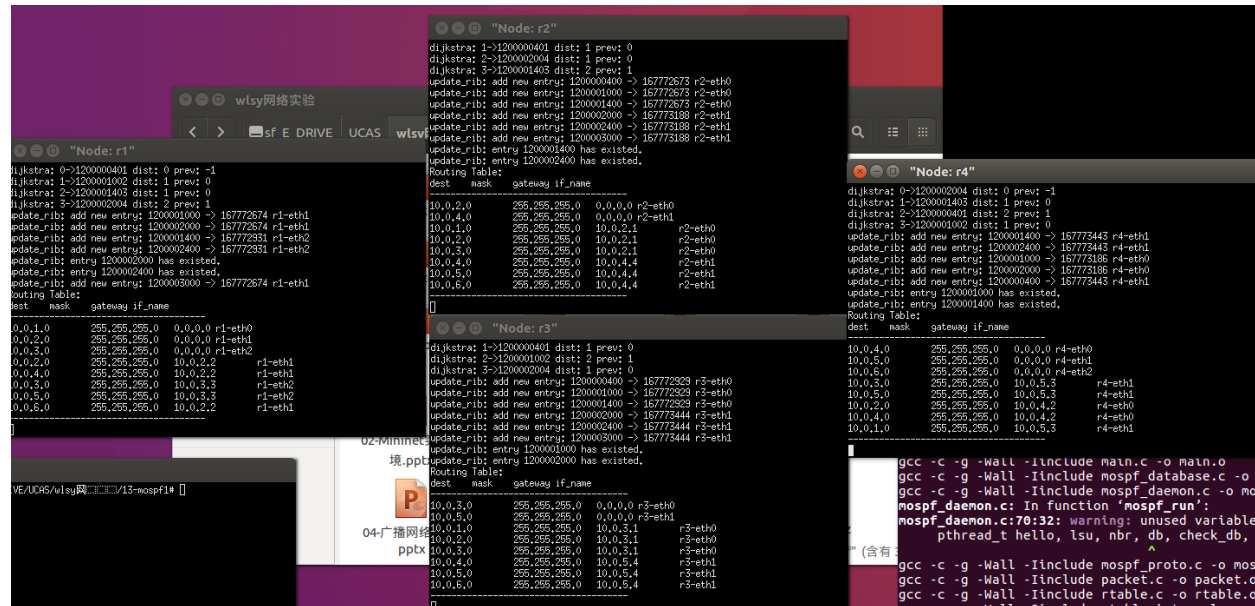
- 清空已有的路由表（除了从内核读到的本地路由条目，区别在于其网关为 `0.0.0.0`）
 - 本质原因在于本实验路由表没有老化机制
- 按照路径长度从小到大遍历各结点，可用类似dijkstra的第一步选顶点的方法得到最近结点k
- 计算结点0到结点k的下一跳结点 `track_path`
 - 本质是递归prev(k)，直到找到第一跳结点k'
 - 找到从结点0到结点k'的链路，方法：遍历邻居表，找到链路的发送端口 `iface` 和结点k'的网关 `gw`
 - 对结点k的每个子网，填充路由表：(entry->array[i].subnet, entry->array[i].mask) -> (gw, iface)

三、实验结果和分析

验证方法:

- 运行网络拓扑(topo.py)
 - 在各个路由器节点上执行disable_arp.sh, disable_icmp.sh, disable_ip_forward.sh), 禁止协议栈的相应功能
- 运行./mospfd, 使得各个节点生成一致的链路状态数据库, 并等待一段时间

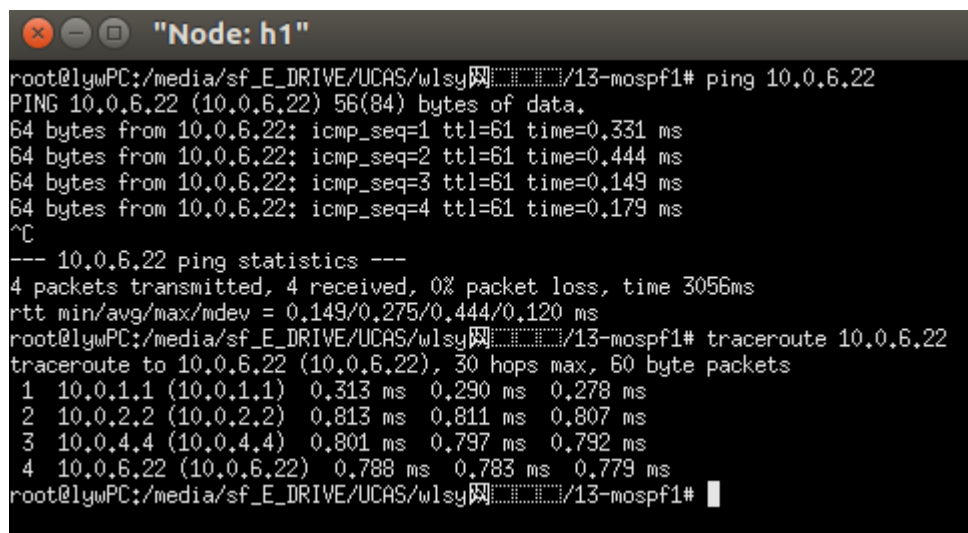
0. 路由表dump结果



可以看到四个结点的路由表均和预期一致

1. 路由互通检测: 在节点h1上ping/traceroute节点h2

结果如下, 正常连通:



2. 路由算法健壮性检测

- 在结点r2上关掉路由程序, 等待一个主动LSU发送周期后, 重新ping/traceroute r2

```
"Node: h1"
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/13-mospf1# ping 10.0.6.22
PING 10.0.6.22 (10.0.6.22) 56(84) bytes of data.
64 bytes from 10.0.6.22: icmp_seq=1 ttl=61 time=0.193 ms
64 bytes from 10.0.6.22: icmp_seq=2 ttl=61 time=0.256 ms
64 bytes from 10.0.6.22: icmp_seq=3 ttl=61 time=0.152 ms
64 bytes from 10.0.6.22: icmp_seq=4 ttl=61 time=0.174 ms
64 bytes from 10.0.6.22: icmp_seq=5 ttl=61 time=0.150 ms
^C
--- 10.0.6.22 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4073ms
rtt min/avg/max/mdev = 0.150/0.185/0.256/0.038 ms
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/13-mospf1# traceroute 10.0.6.22
traceroute to 10.0.6.22 (10.0.6.22), 30 hops max, 60 byte packets
 1 10.0.1.1 (10.0.1.1) 0.056 ms 0.014 ms 0.012 ms
 2 10.0.3.3 (10.0.3.3) 0.051 ms 0.036 ms 0.035 ms
 3 10.0.5.4 (10.0.5.4) 0.076 ms 0.064 ms 0.064 ms
 4 10.0.6.22 (10.0.6.22) 0.126 ms 0.086 ms 0.072 ms
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/13-mospf1#
```

r1的包尝试走r3到r4到达目的地

- 打开结点r2，关掉r3和r4的链路（通过执行 `mininet> link r3 r4 down`）

```
"Node: h1"
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/13-mospf1# ping 10.0.6.22
PING 10.0.6.22 (10.0.6.22) 56(84) bytes of data.
64 bytes from 10.0.6.22: icmp_seq=1 ttl=61 time=0.193 ms
64 bytes from 10.0.6.22: icmp_seq=2 ttl=61 time=0.256 ms
64 bytes from 10.0.6.22: icmp_seq=3 ttl=61 time=0.152 ms
64 bytes from 10.0.6.22: icmp_seq=4 ttl=61 time=0.174 ms
64 bytes from 10.0.6.22: icmp_seq=5 ttl=61 time=0.150 ms
^C
--- 10.0.6.22 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4073ms
rtt min/avg/max/mdev = 0.150/0.185/0.256/0.038 ms
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/13-mospf1# traceroute 10.0.6.22
traceroute to 10.0.6.22 (10.0.6.22), 30 hops max, 60 byte packets
 1 10.0.1.1 (10.0.1.1) 0.056 ms 0.014 ms 0.012 ms
 2 10.0.3.3 (10.0.3.3) 0.051 ms 0.036 ms 0.035 ms
 3 10.0.5.4 (10.0.5.4) 0.076 ms 0.064 ms 0.064 ms
 4 10.0.6.22 (10.0.6.22) 0.126 ms 0.086 ms 0.072 ms
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/13-mospf1# traceroute 10.0.6.22
traceroute to 10.0.6.22 (10.0.6.22), 30 hops max, 60 byte packets
 1 10.0.1.1 (10.0.1.1) 0.073 ms 0.013 ms 0.011 ms
 2 10.0.2.2 (10.0.2.2) 0.053 ms 0.036 ms 0.036 ms
 3 10.0.4.4 (10.0.4.4) 5.300 ms 5.554 ms 5.696 ms
 4 10.0.6.22 (10.0.6.22) 5.882 ms 6.105 ms 9.046 ms
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/13-mospf1#

"Node: r3"
Update existed db entry 10.0.2.2 10.0.2.0 255.255.255.0 10.0.1.1
Update existed db entry 10.0.2.2 10.0.4.0 255.255.255.0 10.0.4.4
Send raw packet failed: Network is down
---dumping **database** of this instance---
10.0.1.1 10.0.1.0 255.255.255.0 0.0.0.0
10.0.1.1 10.0.2.0 255.255.255.0 10.0.2.2
10.0.1.1 10.0.3.0 255.255.255.0 10.0.3.3
10.0.4.4 10.0.4.0 255.255.255.0 10.0.2.2
10.0.4.4 10.0.5.0 255.255.255.0 0.0.0.0
10.0.4.4 10.0.6.0 255.255.255.0 0.0.0.0
10.0.2.2 10.0.2.0 255.255.255.0 10.0.1.1
10.0.2.2 10.0.4.0 255.255.255.0 10.0.4.4
---end---
dijkstra: shortest path:
dijkstra: 0->1200001403 dist: 0 prev: -1
dijkstra: 1->1200000401 dist: 1 prev: 0
dijkstra: 2->1200002004 dist: 3 prev: 3
dijkstra: 3->1200001002 dist: 2 prev: 1
update_rib: add new entry: 1200000400 -> 167772929 r3-eth0
update_rib: add new entry: 1200001000 -> 167772929 r3-eth0
update_rib: add new entry: 1200001400 -> 167772929 r3-eth0
update_rib: entry 1200001000 has existed.
update_rib: add new entry: 1200002000 -> 167772929 r3-eth0
update_rib: entry 1200002000 has existed.
update_rib: add new entry: 1200002400 -> 167772929 r3-eth0
update_rib: add new entry: 1200003000 -> 167772929 r3-eth0
Routing Table:
dest mask gateway if_name
-----
10.0.3.0 255.255.255.0 0.0.0.0 r3-eth0
10.0.5.0 255.255.255.0 0.0.0.0 r3-eth1
10.0.1.0 255.255.255.0 10.0.3.1 r3-eth0
10.0.2.0 255.255.255.0 10.0.3.1 r3-eth0
10.0.3.0 255.255.255.0 10.0.3.1 r3-eth0
10.0.4.0 255.255.255.0 10.0.3.1 r3-eth0
10.0.5.0 255.255.255.0 10.0.3.1 r3-eth0
10.0.6.0 255.255.255.0 10.0.3.1 r3-eth0
^C
root@lywPC:/media/sf_E_DRIVE/UCAS/wlsy网[REDACTED]/13-mospf1#
```

可以看到在r3到r4的Hello包发送失败（屏幕显示 `Send raw packet failed: Network is down`）之后

r3的数据库条目里r4的10.0.5.0已经连不到任何路由器，抽象的拓扑里已经不存在环路了，因而生成的路由表全是从一个接口发送。