# 第十二周实验报告

# 网络路由实验

2015K8009922021

李一苇

# 一、实验内容

基于已有代码框架,实现路由器生成和处理mOSPF Hello/LSU消息的相关操作,构建一致性链路状态数据库

#### 二、实验流程

按照下面顺序依次实现各个函数

1. 发送mOSPF Hello消息: mospf\_send\_hello

在每个接口(iface)里由外向内依次封装ether包头、ip包头、mOSPF包头、mOSPF Hello包并发出目的的MAC地址和IP地址都是已知的专用常量

后两者可由给出的init函数实现:

mospf\_init\_hdr(mospf, MOSPF\_TYPE\_HELLO, MOSPF\_HDR\_SIZE + MOSPF\_HELLO\_SIZE, instance>router\_id, instance->area\_id);

mospf\_init\_hello(hello, iface->mask); 关键是发出自己接口的掩码信息

通过 iface\_send\_packet 函数直接发送包

2. 接收mOSPF Hello消息: handle\_mospf\_hello

当某接口收到Hello包之后,进行解析:

- o 如果之前收到过这个rid的Hello包,只更新邻居表里的相应项目
- o 否则,新增把Hello包里的rid作为新邻居表项插入
  - 因为邻居表项的结构发生变化,所以触发发送LSU包

注意, 在更新邻居表时应该加锁

- 3. 发送mOSPF LSU消息: mospf\_send\_1su
  - o 汇总各个接口的各个邻居的消息,生成LSA
    - 枚举各接口,累加 iface->num\_nbr
    - 申请总邻居数长度的LSA项,重新枚举接口的各邻居,填充LSA项
  - o 在各个接口上生成完整的LSU包,并从所有接口,向所有接口的所有邻居都发出
    - 包的IP部分,源地址为 iface->ip ,目的地址为 nbr->nbr\_ip
    - 通过 ip\_send\_packet 发送包

#### 注意:

第一步应该上锁,防止在汇总 num\_nbr 之后邻居项再发生变化,可能导致栈的溢出

第二步中只需申请ether包头的内存空间,无需填充包头,因为在发送函数中,已经有通过ARP机制找MAC地址的过程

第一步汇总过程中,如果某接口的num\_nbr为0,代表这个接口没有收到任何Hello包,即该接口没有邻接的路由器,可能该接口闲置或连接普通主机。此时,按照老师讲义的规范,仍然生成一条LSA条目,邻居ip为0.0.0.0

# 4. 接收mOSPF LSU消息: handle\_mospf\_lsu

- o 如果在路由器的LSU数据库里有rid的信息且LSU包的 sequence\_num 更大,把rid对应的所有项更新为包里的新消息
- o 如果数据库里没有这个路由器rid的信息,则把rid对应的所有项作为新数据库条目插入
- o 只要对数据库进行了更新,则根据洪泛法则,向除接收外的所有接口转发该LSU包

## 注意:

在更新数据库时,应该上锁,防止在后续实验,生成路由表时出现意想不到的错误

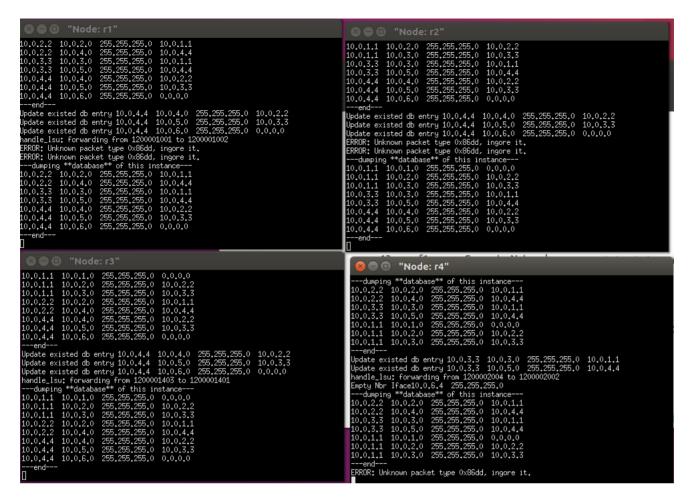
在转发操作时,也不应该对包rid路由器对应的接口发送消息,防止路由器接收到邻居为自己的LSU包(因为这个包最初就是由rid发送的,所以没必要接收自己最初发出的包)

## 三、实验结果和分析

## 验证方法:

- 在 mospf\_run 中新注册一条线程 generating\_rib\_from\_mospf\_db , 每5秒执行一次操作, dump 数据库 条目
- 运行网络拓扑(topo.py)
  - o 脚本会在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
- 在路由器结点上运行./mospfd,等待一段时间,使得各个节点生成一致的链路状态数据库并输出数据库所有 条目

# 结果如下:



可以看到,在一段时间(启动后加一次LSU主动发送的间隔)后,所有路由器都生成了同样的数据库条目你,为下一阶段生成统一路由表做好了铺垫。