实验报告

姓名： 袁峥 学号： 2015K8009929008

一、实验题目： 生成树机制实验

二、实验内容

1、基于已有代码，实现生成树运行机制，对于给定拓扑(four\_node\_ring.py)，计算输出相应状态下的最小生成树拓扑

2、自己构造一个不少于6个节点，链路冗余度不小于2的拓扑，节点和端口的命名规则可参考four\_node\_ring.py，使用stp程序计算输出最小生成树拓扑

三、实验流程

本实验主要实现stp节点在收到config后的处理过程。端口收到config消息后，比较本端口Config与Config消息：

1、如果收到的config比本地端口存储的config优先级更高，说明该网段通过对方端口连接根节点代价更小，则将本端口的Config替换为收到的Config消息，本端口为非指定端口，更新节点状态，更新剩余端口的Config，如果节点由根节点变为非根节点，停止hello定时器，最后将更新后的Config从每个指定端口转发出去

2、否则，说明该网段通过本端口连接根节点代价更小，该端口是指定端口，所以从该端口发送Config消息

以上流程可以分为四个部分。

1、比较两个config的优先级

①如果两者认为的根节点ID不同，则根节点ID小的一方优先级高

②如果两者到根节点的代价不同，则代价小的一方优先级高

③如果两者到根节点的上一跳节点不同，则上一跳节点ID小的一方优先级高

④如果两者到根节点的上一跳端口不同，则上一跳端口ID小的一方优先级高

2、更新节点状态

①遍历所有端口，找出根端口(root port)，满足如下条件

该端口不能是指定端口

该端口的优先级要高于所有其他的端口(①)

②如果不存在根端口，则该节点为根节点(root switch)

stp->designated\_root = stp->switch\_id

stp->root\_path\_cost = 0

③否则，更新节点状态为：

stp->root\_port = root\_port

stp->designate\_root = root\_port->designated\_root

stp->root\_path\_cost = root\_port->designated\_cost +root\_port->path\_cost

3、**更新端口的Config**

①如果一个端口本来就是指定端口，那么只更新如下内容

p->designated\_root = stp->designated\_root

p->designated\_cost = stp->root\_path\_cost

②如果一个端口为非指定端口，且其网段通过本节点到根节点的代价比通过对端节点的代价小，那么该端口成为指定端口，并更新如下内容：

p->designated\_root = stp->designated\_root

p->designated\_cost = stp->root\_path\_cost

p->designated\_switch = stp->switch\_id

p->designated\_port = p->port\_id

4、根节点主动发送config消息

(每个)根节点有一个hello定时器(hello\_timer)

①什么时候启动：stp\_start\_timer(&stp->hello\_timer)

当一个节点是根节点时，包括：初始化时、定时器触发后、(从非根节点变为根节点时)

②什么时候关闭：stp\_stop\_timer(&stp->hello\_timer)

从根节点变为非根节点时

③定时器触发后的操作：stp\_handle\_hello\_timeout(stp);

从所有端口发送config消息，重新设置定时器

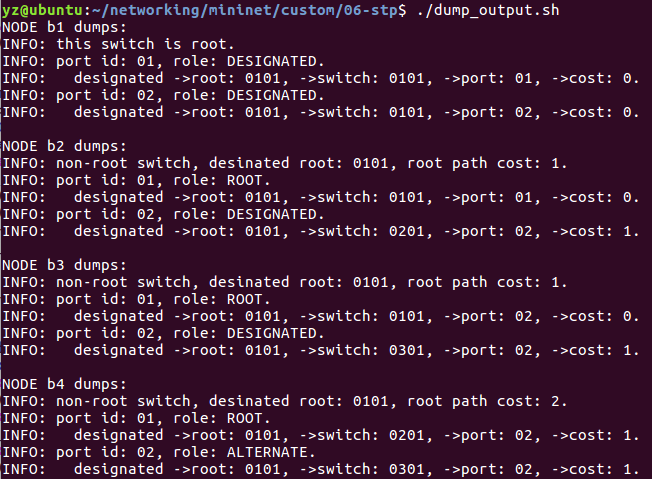
四、实验结果

1、four\_node\_ring.py

生成树拓扑结构：



运行结果：

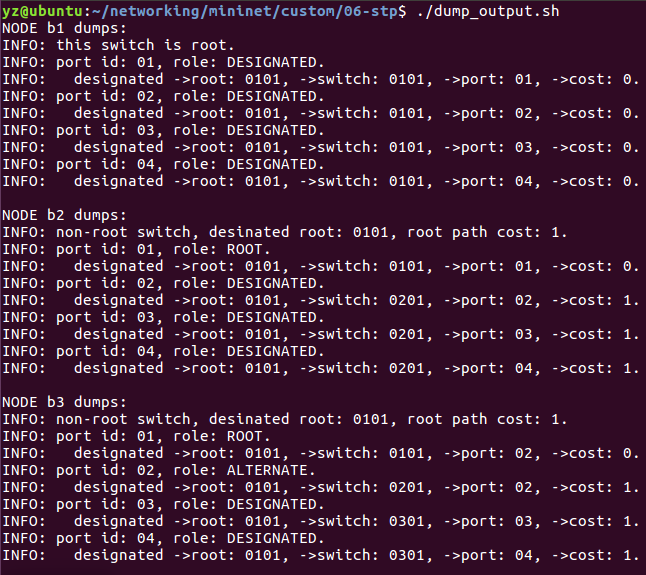


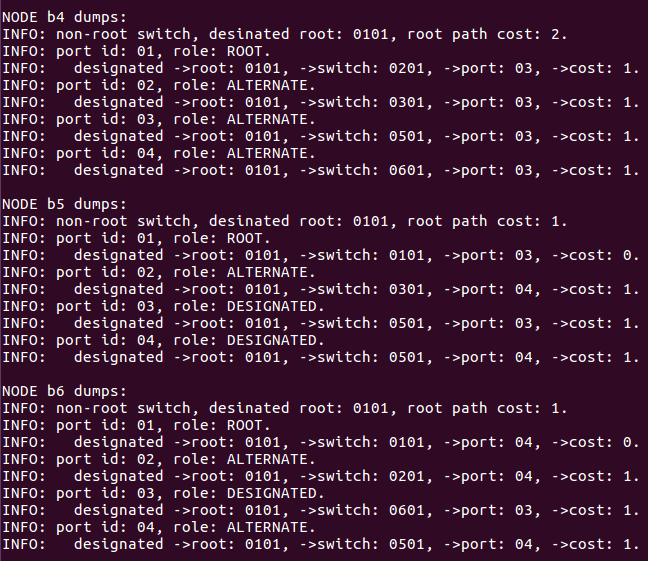
2、six\_node\_ring.py

生成树拓扑结构：



运行结果：





五、结果分析

根据实验输出，分别在两张拓扑图上复原了不同端口的属性。

1、four\_node\_ring.py



2、six\_node\_ring.py



可以看到根据实际输出复原的拓扑图的各个端口属性符合实际情况，因此实验结果正确。

六、实验总结

本次实验只需要完成stp.c文件中的stp\_handle\_config\_packet这一个函数，从代码量来说不大，但是在熟悉实验流程上面花费了一些时间。先反复浏览了课件，理清了跟端口、指定端口和非指定端口的关系，然后根据课件上给出的实验步骤大致明白了代码的框架。在编写代码的时候，也根据课件将流程分为了四步，这样在结构上也较为清晰。

在写完代码后自己进行测试的时候，感觉最初课件上的有些原理有一些问题，按照最初的规则生成的树并不唯一，而且生成的结果可能会不符合逻辑。后来在老师更新课件后，终于理顺了整个实验中的细节，也完全明白了整个步骤。

在调试的过程中，第一遍输出的节点和端口信息非常奇怪，在阅读其他代码后发现，在处理config消息时需要从网络序转为本地序。后来发现在运行four\_node\_ring.py的时候4个节点中只有两个节点有输出信息，于是又重新仔细根据实验流程检查了一遍，发现是遗漏了转发config这个步骤。

总体来说本次实验完成的较为顺利，和前两次的实验一起来看，三次实验一步一步深入，让我们逐步熟悉了在网络拓扑中消息的传输机制。