STP实验报告

1. 实验任务
2. 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.
3. 自己构造一个不少于7个节点，冗余链路不少于2条的拓扑，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.
4. 实验过程
5. 总逻辑

总的流程如下图的流程图所示：

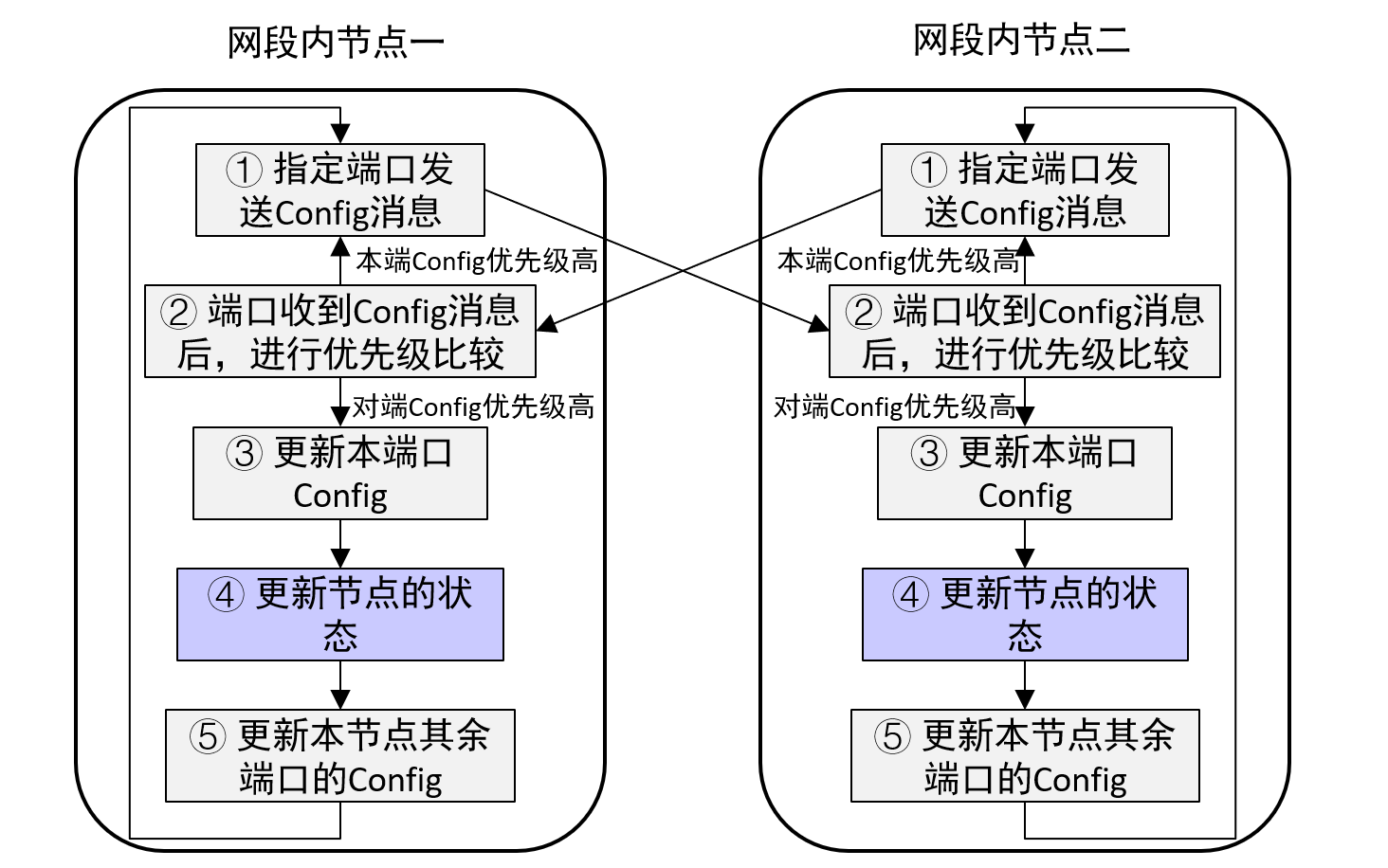


图1：stp算法的流程图

首先我们需要判断收到的config消息和本端口的config消息哪个优先级高，如果本端口优先级高，则说明该网段应该通过本端口存储config对应的端口连接根节点，在代码中体现就是不需要做额外的事情。如果本端口的config消息没有收到的config消息高，则需要更新本端口和本端口节点所对应的状态；并且停止计时器，将自己更新过后的状态转发出去，具体的代码如下图所示：

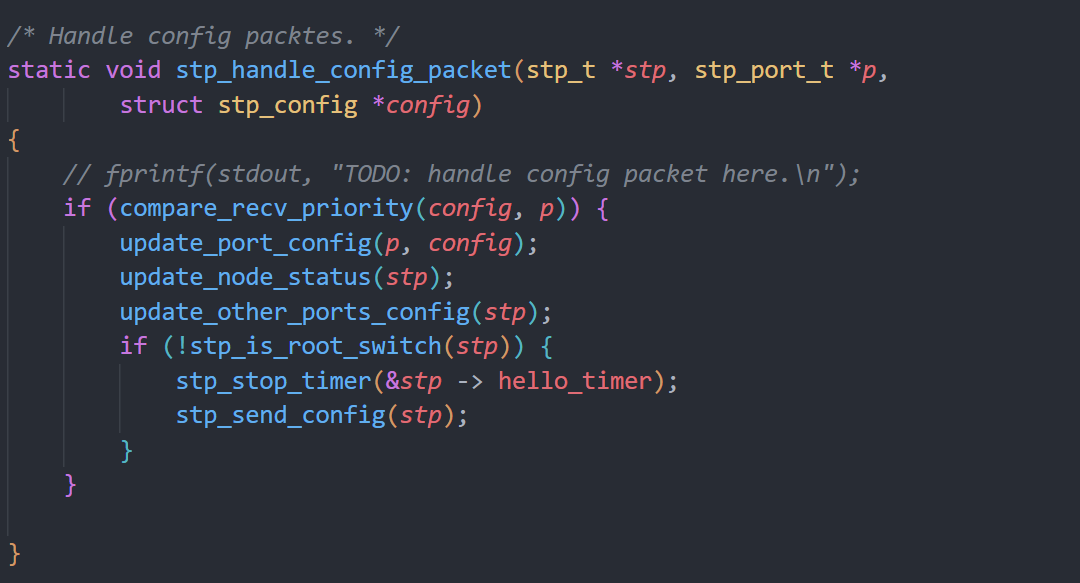


图2：总的处理逻辑

之后我们会一个一个函数的进行说明。

1. Config之间比较优先级

首先我们需要比较config之间的优先级，这里有两种比较，第一种是比较两个port的config的优先级，第二个是比较收到的config和端口config的优先级，两者比较逻辑大同小异，都遵循下图的原则：

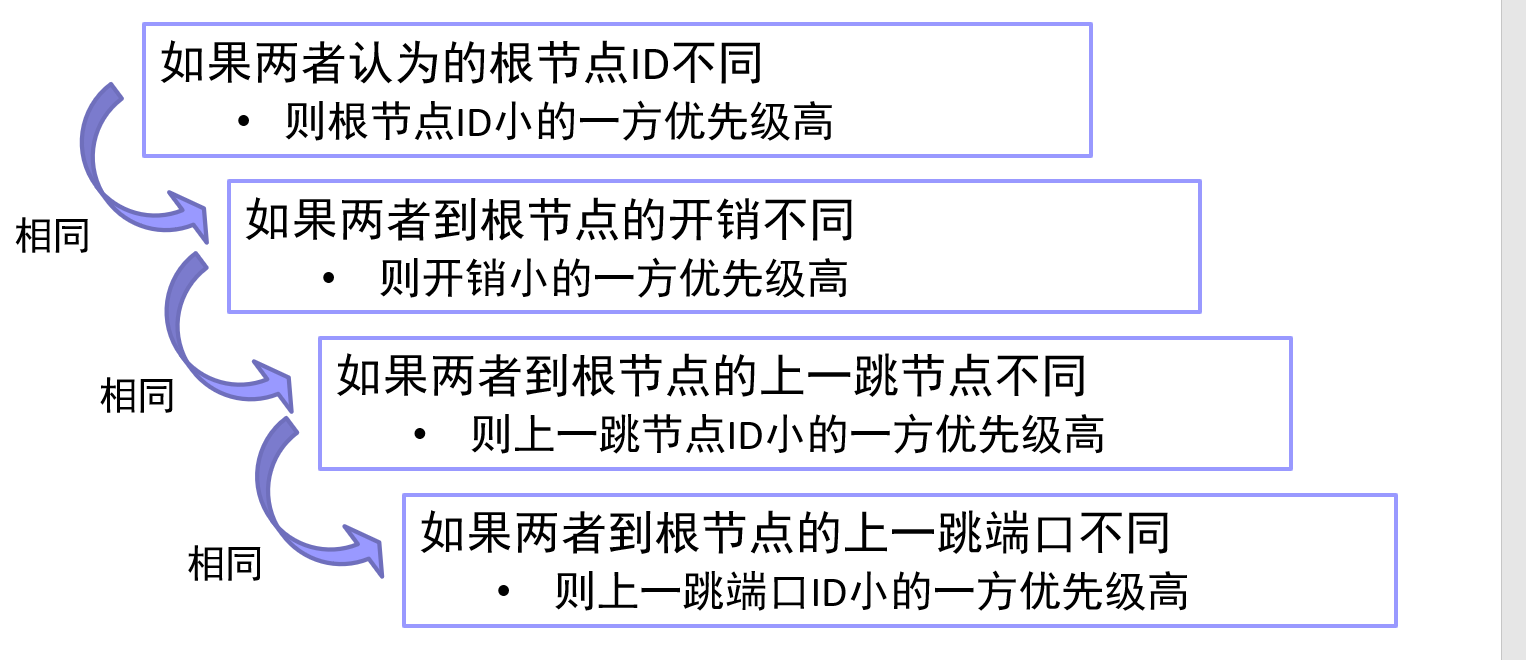


图3：config之间的比较逻辑

具体到代码来说即下图：

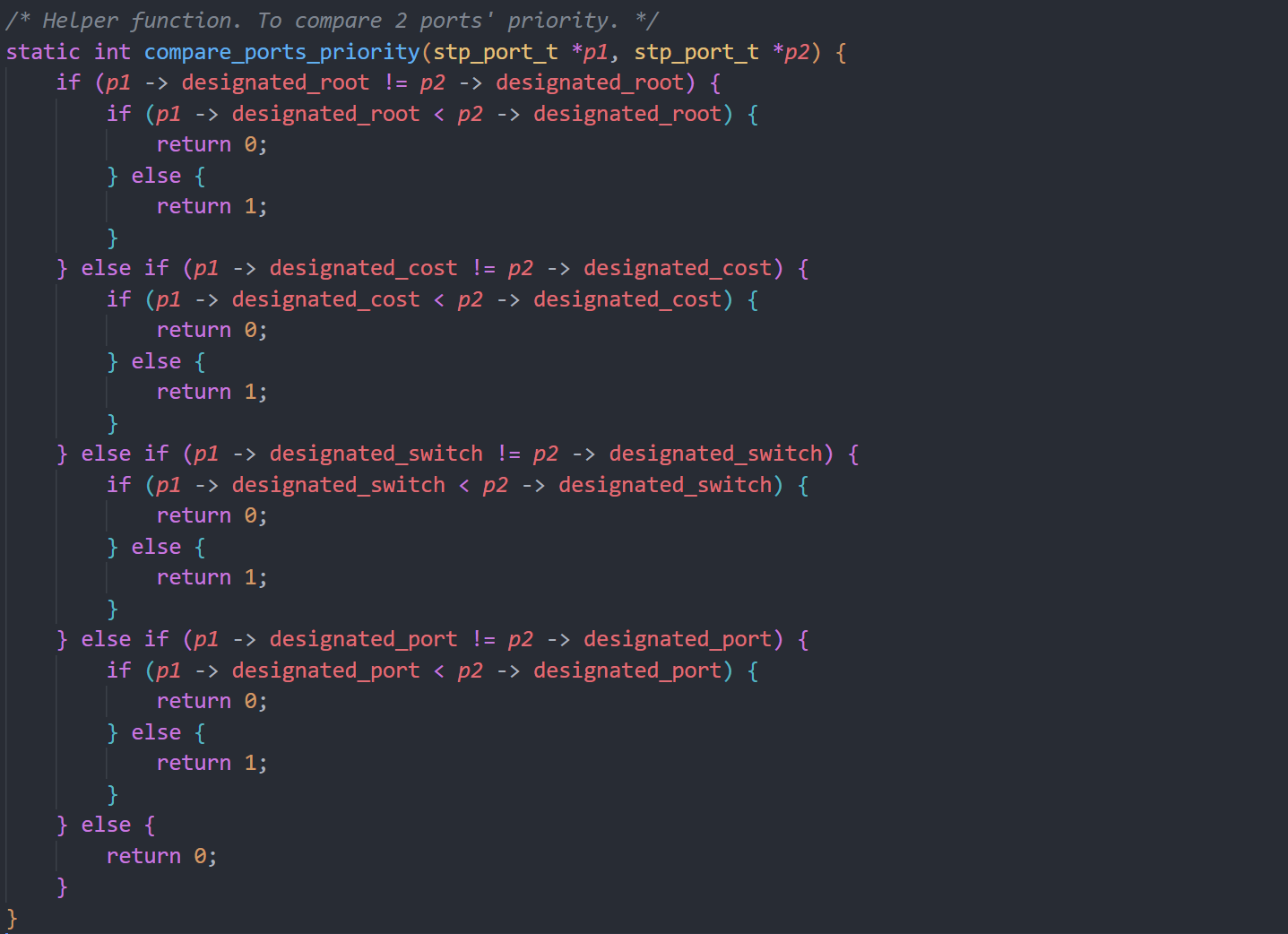


图4：比较两个端口config优先级的代码

注意的是，我们进行端口接收的config和端口的config比较的时候，要对接收端口的相应字段进行字节序转化，如下图：

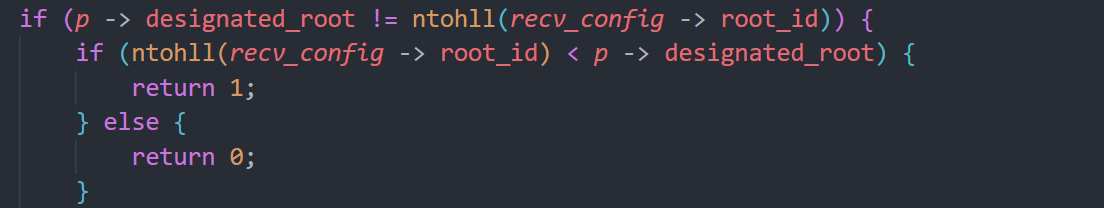


图5：字节序的转换

1. 替换端口的config

如果我么发现端口接受的config比自己原本的config优先级要高，这个时候我们就需要把这个端口的config替换为这个优先级更高的config，具体的逻辑很简单，把端口的config每个字段更新即可：

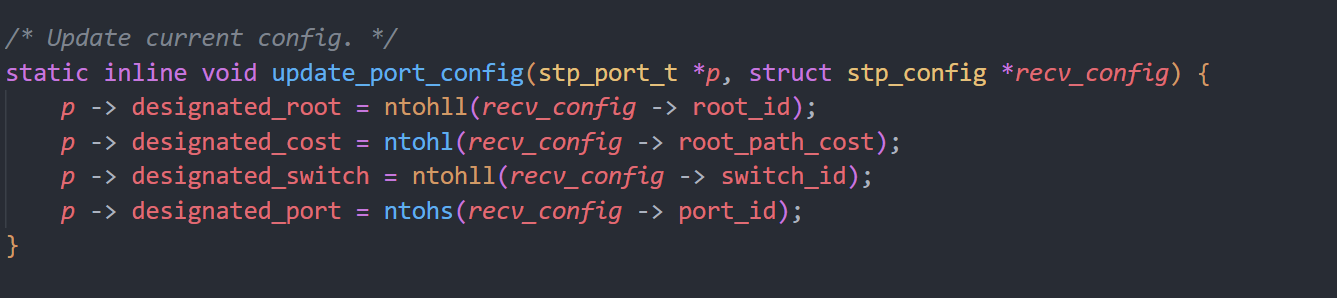


图6：更新端口的config

1. 更新节点状态

这一部分主要有两个任务，第一个是找到root端口，第二步是更新节点状态，选择通过root\_port连接到root端口。

首先我们先要找到root端口，需要满足两个条件：第一是该端口为非指定端口，第二为该端口的优先级要高于其它非指定端口。

具体我们的实现细节是，先遍历所有端口找到所有非指定端口，再找到非指定端口中优先级最高的一个，具体的代码逻辑如下图所示：



图7：找到root port的代码

接下来我们需要更新节点的状态，这里分为两种情况，如果我们没有找到root port（比如这个节点为root node），我们就认为root\_node还为本节点，cost为0；如果有root node，我们利用root port的config信息更新我们节点的状态，主要是root node的id和cost，代码如下图所示：

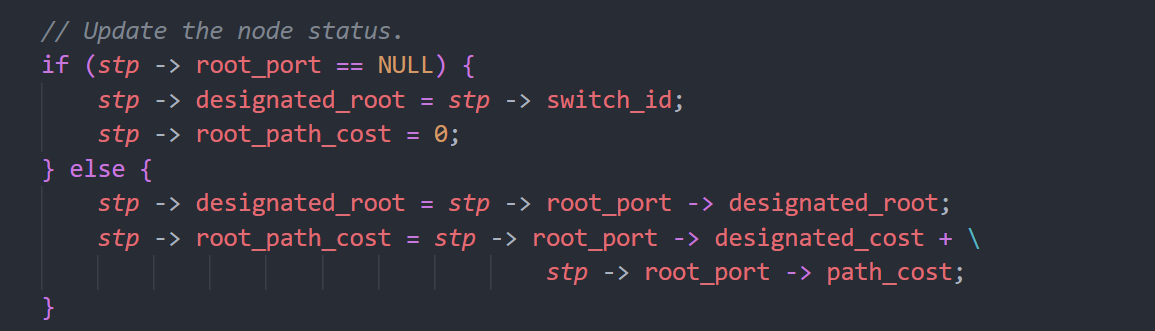


图8：更新node的状态

1. 更新端口的config

我们首先需要处理非指定端口变为指定端口的情况，这种情况会在当一个端口为非指定端口，且其config较网段内其他端口优先级更高，具体的代码逻辑如下：



图9：非指定端口变为指定端口

接着我们对于每个指定端口，更新认为的根节点和路径开销：

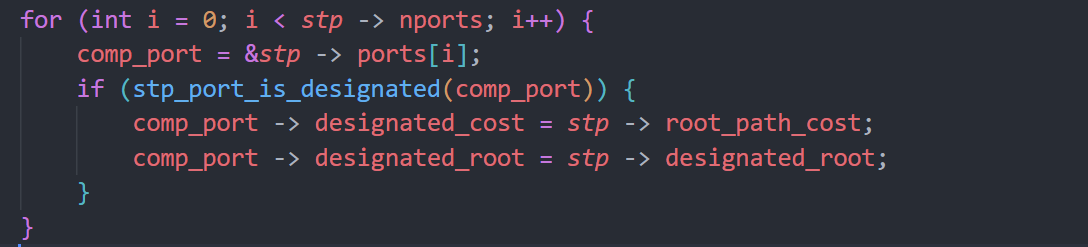


图10：更新指定端口的状态

1. 生成最小生成树拓扑的例子

我们建立最初的拓扑结构：

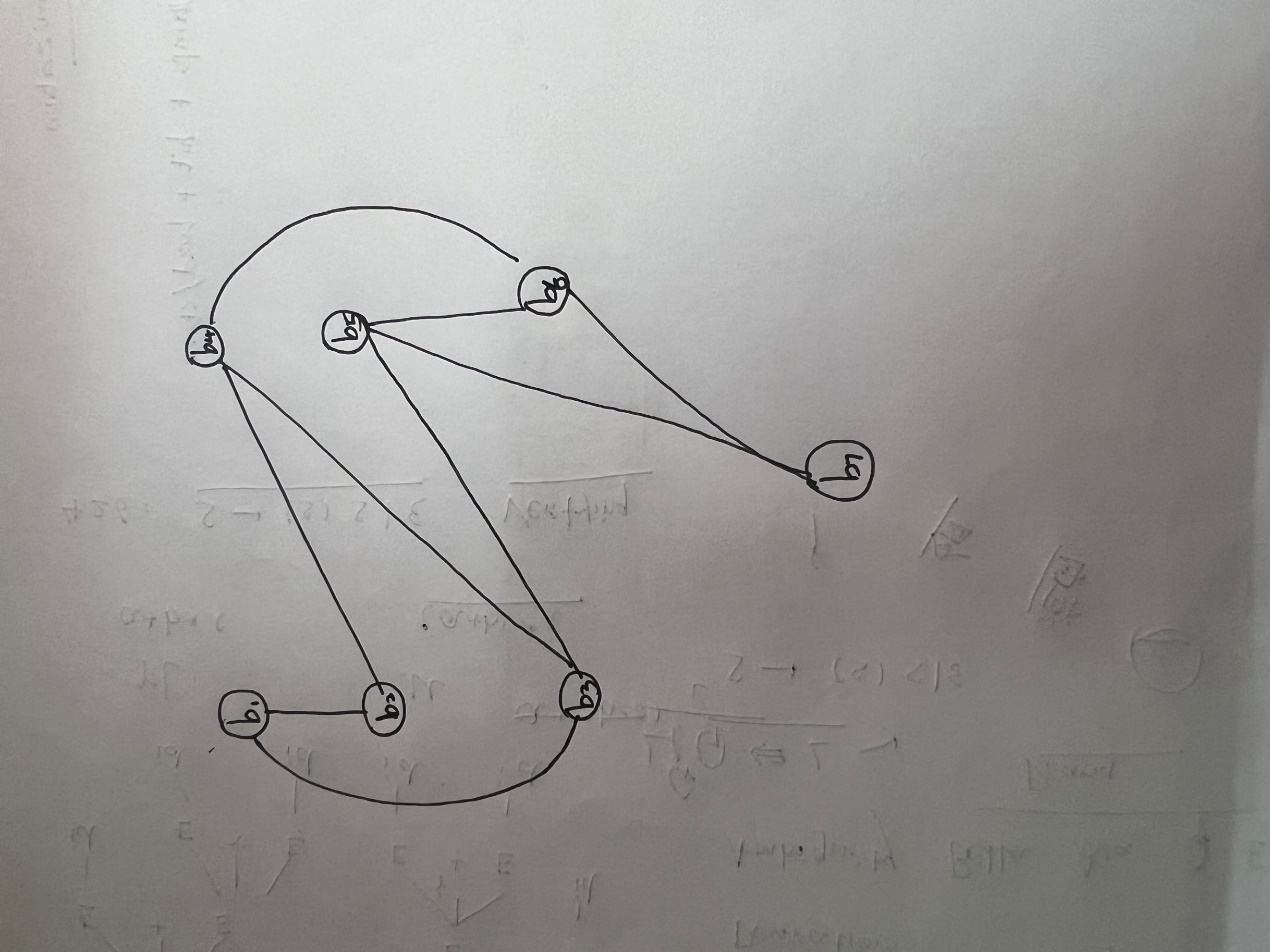
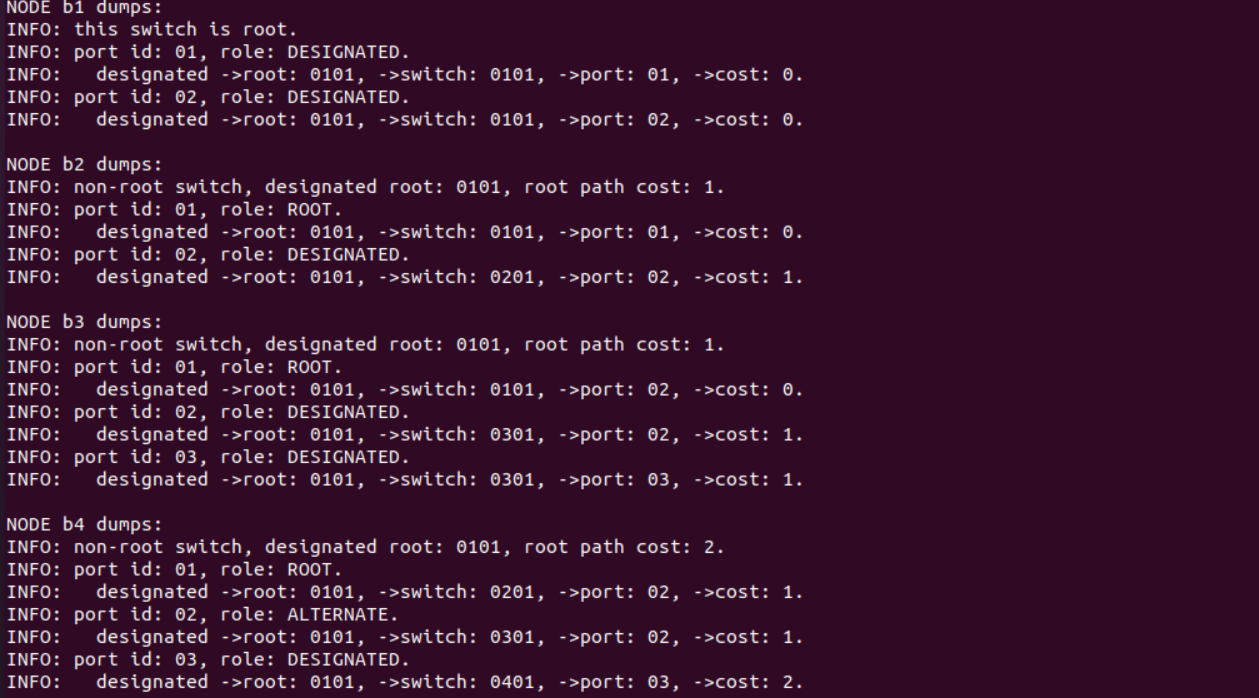


图11：最初的拓扑结构

跑完stp算法，我们得到如下结果：



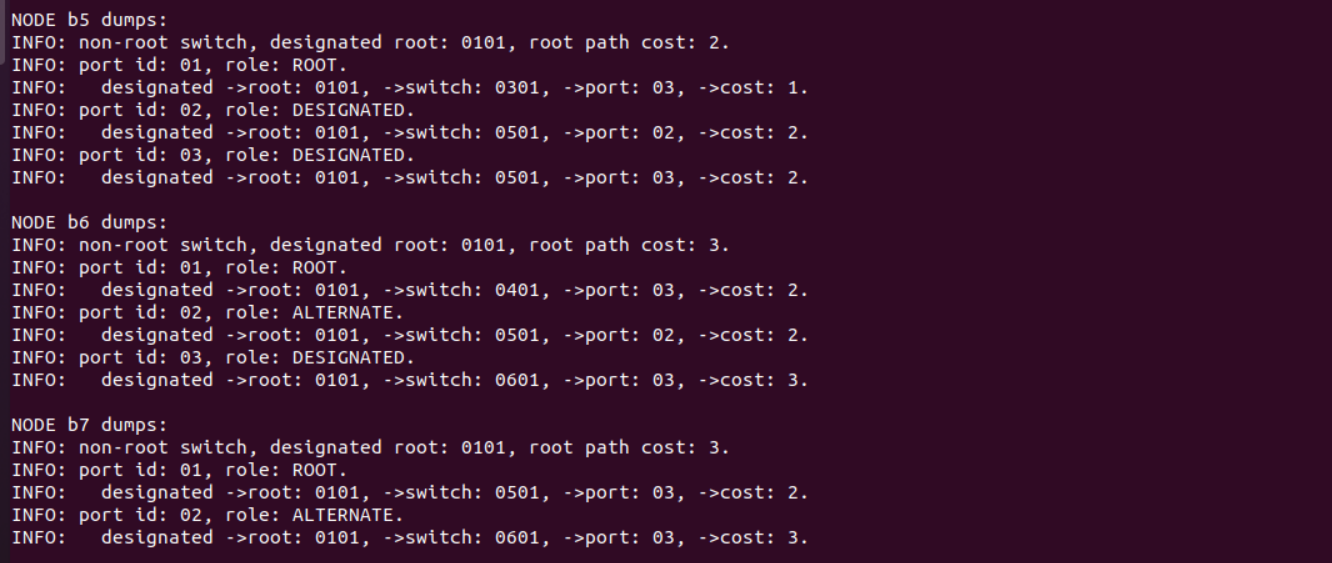


图12：最后生成树结果

我们具体化一些：

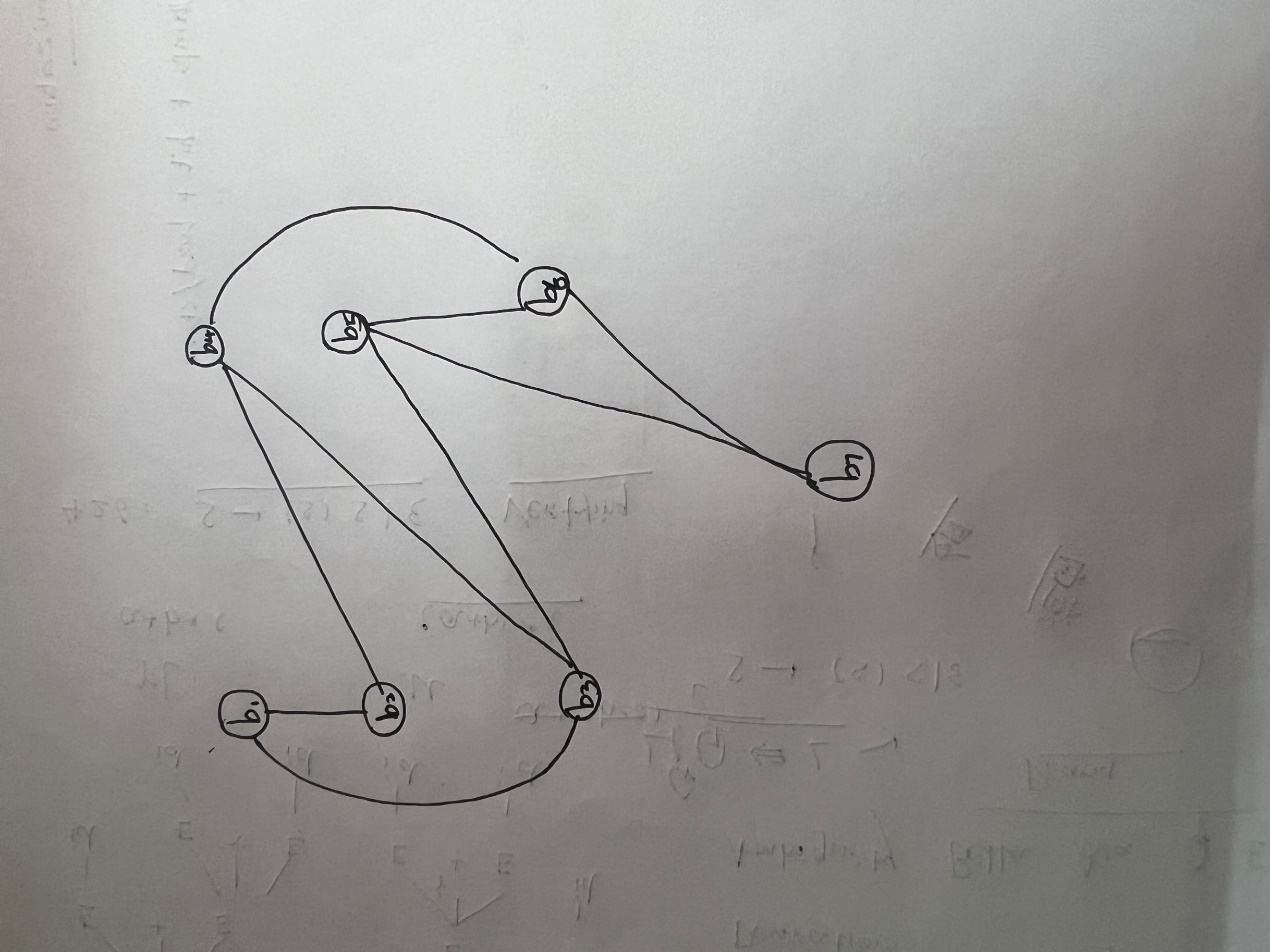
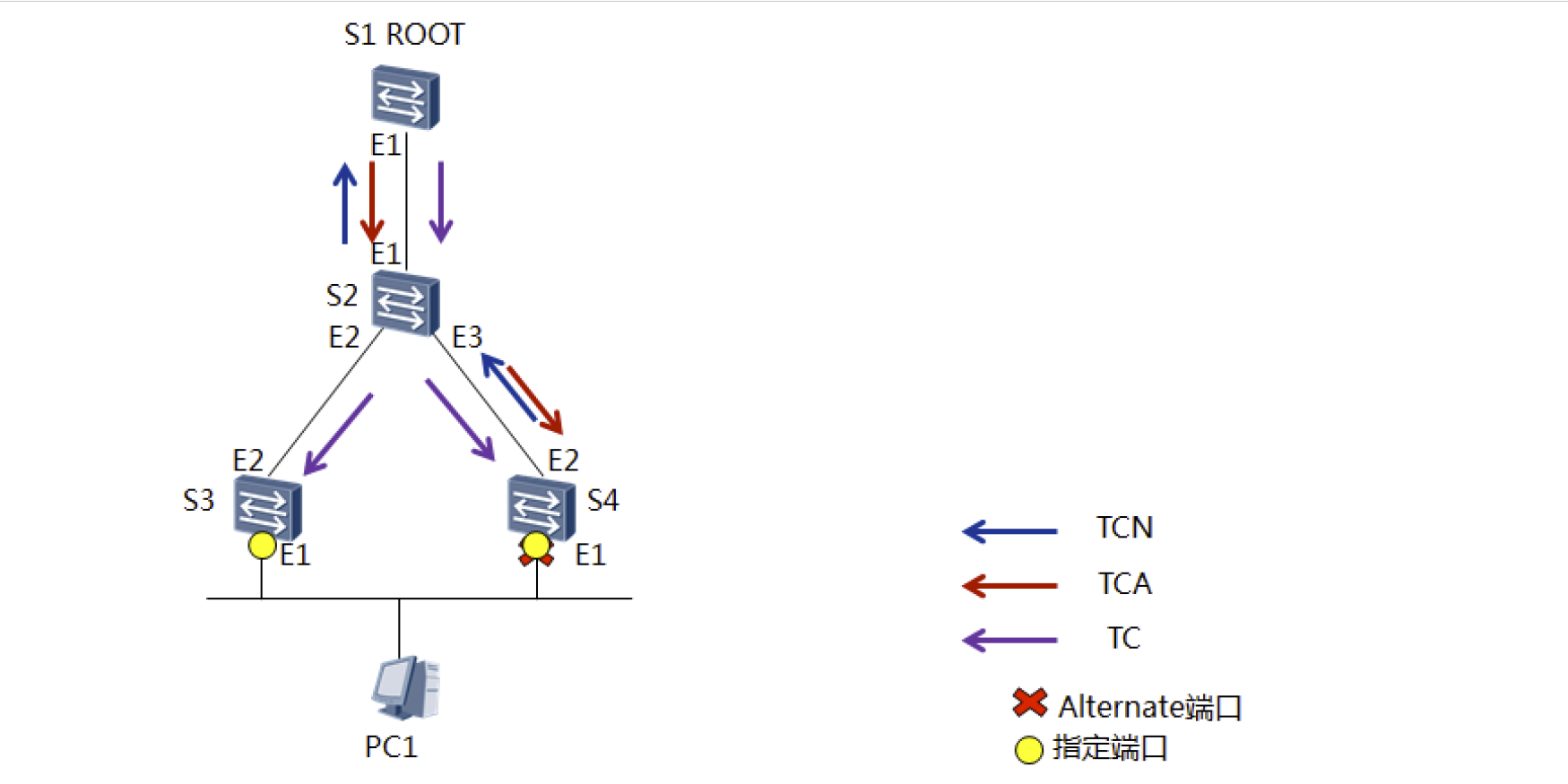


图13：最后的生成树结果（图示）

1. 实验思考
2. 标准生成树协议中，如何处理网络拓扑变动的情况？

节点加入或者离开拓扑结构时候，都要重新进行STP计算，更新节点状态和端口的配置信息，如果生成树拓扑变化，发送数据的路径也会发生变化。拓扑变化时，应该及时更新MAC地址表项。拓扑变化时候，发生改变的交换机通过TCN BPDU报文(Topology Change Notification)告知根桥生成树拓扑里发生了变化，只有指定端口会处理TCN BPDU。上游设备收到TCN BPDU，回复TCA置位的配置BPDU，用于终止下游设备发送TCN BPDU。上游继续向根桥方向发送TCN BPDU，直到根桥收到后。根桥回复TCA+TC，TC全网泛洪，用于清除交换机上的mac地址表。TC的作用为：避免旧的mac表造成数据的转发错误。



TCN消息处理过程：

1. 在网络拓扑发生变化后，有端口转为转发状态的下游设备会不间断地向上游设备发送TCN BPDU报文。
2. 上游设备收到下游设备发来的TCN BPDU报文后，只有指定端口处理TCN BPDU报文。其它端口也有可能收到TCN BPDU报文，但不会处理。
3. 上游设备会把配置BPDU报文中的Flags的TCA位设置1，然后发送给下游设备，告知下游设备停止发送TCN BPDU报文。
4. 上游设备复制一份TCN BPDU报文，向根桥方向发送。
5. 重复步骤1、2、3、4，直到根桥收到TCN BPDU报文。
6. 根桥收到TCN BPDU后，会将下一个要发送的配置BPDU中的TCA位置位，作为对收到的TCN的确认，还会将该配置BPDU报文中的Flags的TC位置1，用于通知所有网桥拓扑发生了变化。
7. 根桥在之后的max（age+forwardingdelay）时间内，将发送BPDU中的TC置位的报文，收到该配置BDPU的网桥，会将自身MAC地址老化时间缩短为forwarding delay。
8. 标准生成树协议如何在构建生成树的过程中保持网络联通？



图14：端口状态表

在构建生成树的过程中，STP会将部分冗余链路强制转化为阻塞状态，其余则链路处于转发状态。当处于转发状态的链路不可用时，STP可以重新配置网络，集合合适的备用链路，恢复部分冗余链路的转发状态来确保网络的连通性。

1. 调研说明快速生成树机制的原理。

**RSTP概述：**

快速生成树协议RSTP（Rapid Spanning Tree Protocol）在STP基础上实现了快速收敛，并增加了边缘端口的概念及保护功能。

RSTP在STP基础上新增加了2种端口角色：Backup端口和边缘端口。通过端口角色的增补，简化了生成树协议的理解及部署。

Backup端口：由于学习到自己发送的配置BPDU报文而阻塞的端口，指定端口的备份，提供了另外一条从根节点到叶节点的备份通路。

边缘端口：如果端口位于整个交换区域边缘，不与任何交换设备连接，这种端口叫做边缘端口。边缘端口一般与用户终端设备直接连接。

边缘端口的特点

1. 边缘端口会节省30S的延时，端口UP后会立即进入转发状态。

2. 边缘端口的UP/DOWN不会触发拓扑改变。

3. 边缘端口收的TC置为的配置BPDU报文不会将MAC地址的老化时间设置为15s。

4. 边缘端口如果收到配置的BPDU报文会马上变为一个普通端‘’口，进行STP的收敛

5. 边缘端口也会发送配置BPDU报文。

6. PA协商不会阻塞边缘端口。

**RSTP的端口状态**

RSTP的端口状态在STP的基础上进行了改进。由原来的五种缩减为三种。

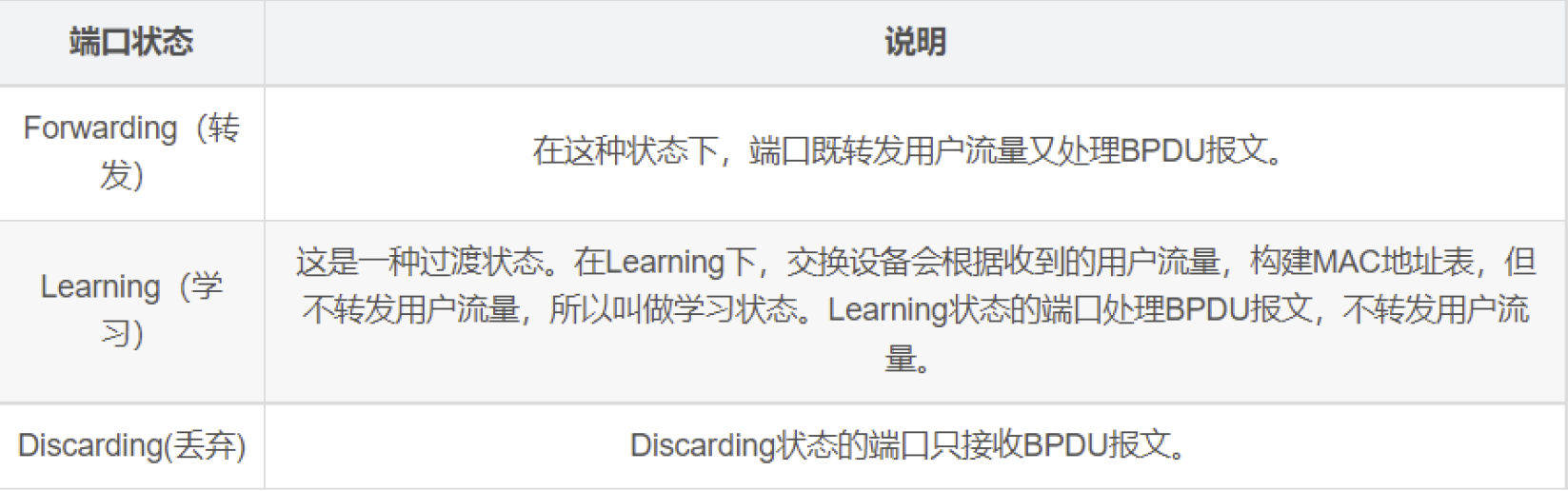


图15：RSTP端口状态

RSTP中只有一种BPDU包，称为RST BPDU。对配置BPDU的flags字段进行了填充，类型值做了改变，如下：

1. Type：RST BPDU包此字段类型是2

2. Flags字段,RST填充了STP BPDU Flags中的保留字段，填充了如下信息：

（1）Aggrement: 确认标识位，用于RSTP中定义的Proposal/Aggrement机制，对于Proposal报文的确认

（2）Forwarding: 转发状态标识位，1: 表示发送该BPDU报文的端口处于Forwarding状态

（3）Learning：学习状态标识位，1：表示发送该BPDU报文的端口处于Learning状态