第六周实验报告

生成树机制实验

2015K8009922021

李一苇

**一、实验内容**

* 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
* 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

**二、实验流程**

本实验只需实现stp.c里的stp\_handle\_config\_packet函数

如果STP包优先级比端口config高：

* + 替换端口的config信息
  + 更新节点的根端口和cost信息
  + 更新剩余端口的config信息
  + 停止节点的hello计时器
  + 每个指定端口转发新的config包

如果STP包优先级更低：

* + 直接从本端口转发自己的config信息

需要注意的点：

* 所有涉及Config的比较都需要用ntoh系列函数进行大小端转化

优先级函数实现如下：

bool is\_config\_higher(stp\_port\_t \*p, struct stp\_config \*config) {

if (ntohll(config->root\_id) < p->designated\_root) return true;

else if (ntohll(config->root\_id) > p->designated\_root) return false;

else if (ntohl(config->root\_path\_cost) < p->designated\_cost) return true;

else if (ntohl(config->root\_path\_cost) > p->designated\_cost) return false;

else if (ntohll(config->switch\_id) < p->designated\_switch) return true;

else if (ntohll(config->switch\_id) > p->designated\_switch) return false;

else if (ntohs(config->port\_id) < p->designated\_port) return true;

else if (ntohs(config->port\_id) > p->designated\_port) return false;

else log(INFO, "config equally comparing: impossible");

return false;

}

* 如何判定指定端口：在非指定端口中，其网段通过本节点到根节点的开销比通过对端节点的开销小。前者可由stp->root\_path\_cost得到，后者无法得知。但如果条件成立，则一定有stp->root\_path\_cost < port\_i->designated\_cost。因为网段的耗费一定是二者的最小值。
* 何时停掉hello计时器：因为初识情况下都是根节点，如果某端口遇到更优的config信息，则证明自己一定不是根节点，所以只要config优先级高，则停掉hello计时器，这一句话一定不会错。
* 不会形成STP包的转发风暴：如果收到的config优先级低，的确存在回传config请求，以便确认对端更新为高优先级；当收到config优先级高时，主机从所有DP转发config，而生成树的叶子主机不存在DP，因此转发中止。

1. **实验结果和分析**

用给定的四结点环路验证STP函数如下

NODE b1 dumps:

INFO: this switch is root.

INFO: port id: 01, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0101, ->port: 01, ->cost: 0.

INFO: port id: 02, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0101, ->port: 02, ->cost: 0.

NODE b2 dumps:

INFO: non-root switch, designated root: 0101, root path cost: 1.

INFO: port id: 01, role: ROOT.

INFO: designated ->root: 0101, ->switch: 0101, ->port: 01, ->cost: 0.

INFO: port id: 02, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0201, ->port: 02, ->cost: 1.

NODE b3 dumps:

INFO: non-root switch, designated root: 0101, root path cost: 1.

INFO: port id: 01, role: ROOT.

INFO: designated ->root: 0101, ->switch: 0101, ->port: 02, ->cost: 0.

INFO: port id: 02, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0301, ->port: 02, ->cost: 1.

NODE b4 dumps:

INFO: non-root switch, designated root: 0101, root path cost: 2.

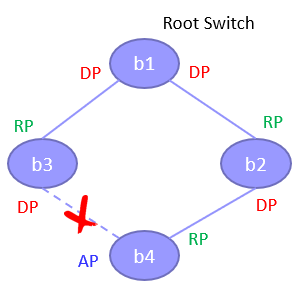
INFO: port id: 01, role: ROOT.

INFO: designated ->root: 0101, ->switch: 0401, ->port: 02, ->cost: 1.

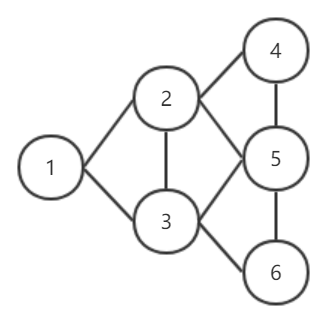
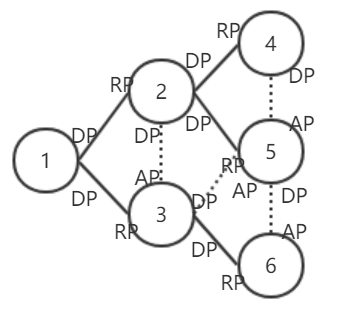
INFO: port id: 02, role: ALTERNATE.

INFO: designated ->root: 0101, ->switch: 0301, ->port: 02, ->cost: 1.

即得到课件中的网络结构：



自己构造六结点拓扑结构如下左图，显然包含两个以上的冗余度。

STP程序给出的输出如下，等价于右上的拓扑图

NODE b1 dumps:

INFO: this switch is root.

INFO: port id: 01, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0101, ->port: 01, ->cost: 0.

INFO: port id: 02, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0101, ->port: 02, ->cost: 0.

NODE b2 dumps:

INFO: non-root switch, designated root: 0101, root path cost: 1.

INFO: port id: 01, role: ROOT.

INFO: designated ->root: 0101, ->switch: 0101, ->port: 01, ->cost: 0.

INFO: port id: 02, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0201, ->port: 02, ->cost: 1.

INFO: port id: 03, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0201, ->port: 03, ->cost: 1.

INFO: port id: 04, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0201, ->port: 04, ->cost: 1.

NODE b3 dumps:

INFO: non-root switch, designated root: 0101, root path cost: 1.

INFO: port id: 01, role: ROOT.

INFO: designated ->root: 0101, ->switch: 0101, ->port: 02, ->cost: 0.

INFO: port id: 02, role: ALTERNATE.

INFO: designated ->root: 0101, ->switch: 0201, ->port: 02, ->cost: 1.

INFO: port id: 03, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0301, ->port: 03, ->cost: 1.

INFO: port id: 04, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0301, ->port: 04, ->cost: 1.

NODE b4 dumps:

INFO: non-root switch, designated root: 0101, root path cost: 2.

INFO: port id: 01, role: ROOT.

INFO: designated ->root: 0101, ->switch: 0201, ->port: 03, ->cost: 1.

INFO: port id: 02, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0401, ->port: 02, ->cost: 2.

NODE b5 dumps:

INFO: non-root switch, designated root: 0101, root path cost: 2.

INFO: port id: 01, role: ROOT.

INFO: designated ->root: 0101, ->switch: 0201, ->port: 04, ->cost: 1.

INFO: port id: 02, role: ALTERNATE.

INFO: designated ->root: 0101, ->switch: 0301, ->port: 03, ->cost: 1.

INFO: port id: 03, role: ALTERNATE.

INFO: designated ->root: 0101, ->switch: 0401, ->port: 02, ->cost: 2.

INFO: port id: 04, role: DESIGNATED.

INFO: designated ->root: 0101, ->switch: 0501, ->port: 04, ->cost: 2.

NODE b6 dumps:

INFO: non-root switch, designated root: 0101, root path cost: 2.

INFO: port id: 01, role: ROOT.

INFO: designated ->root: 0101, ->switch: 0301, ->port: 04, ->cost: 1.

INFO: port id: 02, role: ALTERNATE.

INFO: designated ->root: 0101, ->switch: 0501, ->port: 04, ->cost: 2.