计算机网络实验报告-06

阮星程
2015K8009929047

一、 实验题目

生成树机制实验。

二、 实验内容

理解生成树协议的内容和原理，构建生成树程序，在一个简单的 4 交换机环路上自动生成生成树；

构建一个不少于 6 个节点，冗余度不小于 2 的拓扑，利用代码自动构建生成树。

三、 实验过程

开始逐步构建各个函数

优先级函数，对比 config 与当前端口存储的信息的优先级。返回负数则表明 config 的优先级要更高。巧妙地利用了 4 种信息各自的优先级，对应了 8,4,2,1，避免了大量的连续 if 的出现。

```
//compare the priority between config and port
int get_port_priority(struct stp_config * config, stp_port_t * p)
{
    int tmp;
    int priority = 0;
    if(tmp = (config->root_id - p->designated_root))
        priority += 8 * tmp/abs(tmp);
    if(tmp = (config->root_path_cost - p->designated_cost))
        priority += 4 * tmp/abs(tmp);
    if(tmp = (config->switch_id - p->designated_switch))
        priority += 2 * tmp/abs(tmp);
    if(tmp = (config->port_id - p->designated_port))
        priority += 1 * tmp/abs(tmp);
    return priority;

}
```

开始构建 config 处理程序

首先将 config 中的网络序信息转换为主机序，获取信息的优先级。

```
static void stp_handle_config_packet(stp_t *stp, stp_port_t *p,
        struct stp_config *config)
{
    int priority = 0;
    static int root = 1;

    //transfer config info from network form to host form
    config->root_id = ntohll(config->root_id);
    config->root_path_cost = ntohl(config->root_path_cost);
    config->switch_id = ntohll(config->switch_id);
    config->port_id = ntohs(config->port_id);

    //compare priority between config and current port
    priority = get_port_priority(config, p);
```

若当前端口优先级更高，当当前端口设置为指定端口，若相同则将其转发出去

```
//if receiving packet priority is lower than the current one, change it to designated port
if(priority > 0)
{
    p->designated_port = p->port_id;
    p->designated_switch = stp->switch_id;
    print_info(config, p, priority);
    return ;
}

//if it is the same, send it
if(priority == 0)
{
    stp_send_config(stp);
    return ;
}
```

若端口优先级低，则开始相应处理

```
//change to non-root switch
if(root == 1)
{
    root = 0;
    stp_stop_timer(&stp->hello_timer);
}

//update config info
print_info(config, p, priority);
p->designated_port = config->port_id;
p->designated_switch = config->switch_id;
p->designated_cost = config->root_path_cost;
p->designated_root = config->root_id;

//if root_port has higher priority or the same
if(stp->root_port)
    if(get_port_priority(config, stp->root_port) >= 0)
        return ;


//change root port
stp->root_port = p;
stp->designated_root = config->root_id;
stp->root_path_cost = p->designated_cost + p->path_cost;

//update port info
for(int i=0; i < stp->nports; i++)
{
    if(stp->ports[i].port_id != p->port_id)
    {
        stp->ports[i].designated_cost = stp->root_path_cost;
        stp->ports[i].designated_root = stp->designated_root;
    }
}

//send new info from designated ports
stp_send_config(stp);
```

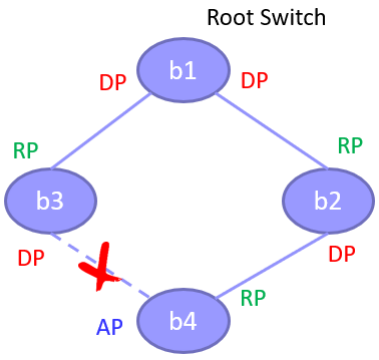
至此所有函数实现完毕，开始进行虚拟机测试，运行 python 脚本，启动终端

```
ruan@ruan-VirtualBox:/mnt/06-stp$ sudo python four_node_ring.py
mininet> xterm b1 b2 b3 b4
mininet> xterm b1
```

运行 stp 程序，开始发送和转发 config 信息，结果见下节。

四、 实验结果

4 节点环路拓扑的结果应如下图：



实际结果如下，与预期结果一致：

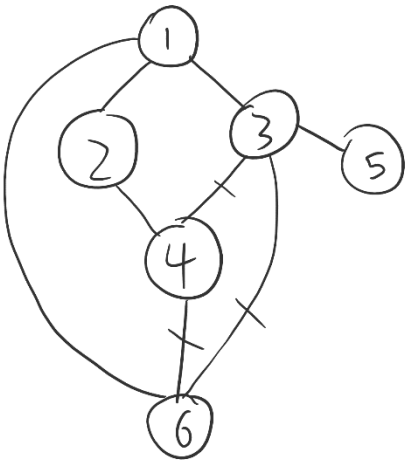
```
NODE b1 dumps:
INFO: this switch is root.
INFO: port id: 01, role: DESIGNATED.
INFO:   designated ->root: 0101, ->switch: 0101, ->port: 01, ->cost: 0.
INFO: port id: 02, role: DESIGNATED.
INFO:   designated ->root: 0101, ->switch: 0101, ->port: 02, ->cost: 0.

NODE b2 dumps:
INFO: non-root switch, desinated root: 0101, root path cost: 1.
INFO: port id: 01, role: ROOT.
INFO:   designated ->root: 0101, ->switch: 0101, ->port: 01, ->cost: 1.
INFO: port id: 02, role: DESIGNATED.
INFO:   designated ->root: 0101, ->switch: 0201, ->port: 02, ->cost: 1.

NODE b3 dumps:
INFO: non-root switch, desinated root: 0101, root path cost: 1.
INFO: port id: 01, role: ROOT.
INFO:   designated ->root: 0101, ->switch: 0101, ->port: 02, ->cost: 1.
INFO: port id: 02, role: DESIGNATED.
INFO:   designated ->root: 0101, ->switch: 0301, ->port: 02, ->cost: 1.

NODE b4 dumps:
INFO: non-root switch, desinated root: 0101, root path cost: 2.
INFO: port id: 01, role: ROOT.
INFO:   designated ->root: 0101, ->switch: 0201, ->port: 02, ->cost: 2.
INFO: port id: 02, role: ALTERNATE.
INFO:   designated ->root: 0101, ->switch: 0301, ->port: 02, ->cost: 2.
```

自己构建的 6 节点拓扑结构如下图，预期结果也如下：



实际结果见下图，与预期相符：

```
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.
INFO: port id: 03, role: DESIGNATED.
INFO:   designated ->root: 0101, ->switch: 0101, ->port: 03, ->cost: 0.

NODE b2 dumps:
INFO: non-root switch, desinated 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, desinated 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.
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, desinated root: 0101, root path cost: 2.
INFO: port id: 01, role: ROOT.
INFO:   designated ->root: 0101, ->switch: 0201, ->port: 02, ->cost: 1.
INFO: port id: 02, role: ALTERNATE.
INFO:   designated ->root: 0101, ->switch: 0301, ->port: 02, ->cost: 1.
INFO: port id: 03, role: ALTERNATE.
INFO:   designated ->root: 0101, ->switch: 0603, ->port: 03, ->cost: 1.

NODE b5 dumps:
INFO: non-root switch, desinated root: 0101, root path cost: 2.
INFO: port id: 01, role: ROOT.
INFO:   designated ->root: 0101, ->switch: 0301, ->port: 03, ->cost: 1.

NODE b6 dumps:
INFO: non-root switch, desinated root: 0101, root path cost: 1.
INFO: port id: 01, role: ROOT.
INFO:   designated ->root: 0101, ->switch: 0101, ->port: 03, ->cost: 0.
INFO: port id: 02, role: ALTERNATE.
INFO:   designated ->root: 0101, ->switch: 0301, ->port: 04, ->cost: 1.
INFO: port id: 03, role: DESIGNATED.
INFO:   designated ->root: 0101, ->switch: 0603, ->port: 03, ->cost: 1.
```

五、 实验总结

通过本次实现，我对生成树协议的内容有了更加具体细致的了解，不再仅仅局限于各节点之间的信息交互，而是理解了各节点间，节点与端口间的信息交互细节，收获满满。

我在本次实验中遇到了一些障碍。开头听过老师的讲解后便开始书写代码了，其实那时也还懵懵懂懂，不太清楚整个的处理流程，光看 PPT 也没能很好地理解节点信息和各端口信息之间的关系。这样写出来的代码自然是会出现各种各样的问题，比如说 4 节点环路中 4 的 root 端口被设置成了与 3 连接的那个。经历了一步步的试错，一步步的更改，一遍遍的阅读，最后才理解了各步究竟该干什么，才最终完成了实验。最终和 PPT 核对后其实每一步老师都有写，只是写的时候各步之间的顺序没有那么明确，各步的作用也没有怎么提及，希望老师能整改下 PPT 的组织结构，这样应该对学弟学妹完成实验有很大的帮助。