
生成树学习实验

熊子威

2015K8009915050

1 实验内容

本次实验接上一次交换机的实验。交换机为了能够成功高效地进行消息转发，在转发表未学习到的情况下需要进行广播，但在环路上，这样的广播将导致无限的循环。因此本次实验我们编写学习生成树的代码，让一个交换网络能够得到一个生成树以打破环路。

2 实验流程

2.1 理论流程

理论上需要做的事情不是很多，每个节点只需要不断向外发送消息，告知其他节点自己的身份，认定的根，到达根的代价，然后各个节点依据收到的消息，按照优先级规则更新自己的消息并转发消息，直到整个网络都将同一个节点认可为根节点，我们就得到了一个生成树。

2.2 实践流程

理论上的说明略过了一些细节。例如，生成生成树时各个节点哪些端口要停止发送信息；节点如何利用自己全部端口的信息来知道自己是否真的是根节点；当端口信息发生变化时，节点要如何作出响应？

实际写代码时，依据老师的框架，节点本身维护了一套信息，这套信息是整个节点尺度上的信息，而各个节点则维护了一套端口尺度上的信息。对于节点来说，其最终认可的根节点必定是某一个端口认可的根节点，这个根节点具有网络中最高的优先级。那么我们要做的事情就比较明晰了，我们需要各个端口基于一定的规则，依据收到的消息来维护自己的信息，同时节点本身依据各个端口的变化适当修改自己维护的更大尺度上的信息，当最后所有的端口信息都一致的时候，我们就知道网络已经收敛。

我们可以适当分析一个节点会遇到一些什么情况。首先一个节点的端口一定是可以分类的，因为我们可以想见，生成树生成之后，节点的一些端口能够接收来自根节点的消息，一些端口需要转发各类消息，而一些端口会被阻塞以避免环路。

回顾生成树的原理，各个节点初始化时都需要假装自己是根节点，但是与根节点直接相连的节点将在第一轮消息传递中意识到自己不是根节点，或者说，这些节点域根节点直接相连的端口将获得特殊的身份，即根端口，能够最近地与根节点交流。一旦得到了根端口，我们就知道其他一些端口中一定要有

一些端口承担对外发送真正的根节点是谁的消息，而另一些端口则有可能被阻塞。因此一个节点可能有三种端口：根端口（从根节点获得消息），指定端口（送出根节点的消息），其他端口（暂时不承担任何功能）。于是我们可以让一个节点依据自己的端口们在这三种类型的端口之间转换的情况来确定谁是根节点，自己应该作出什么样的调整。

一开始时，所有的节点都以为自己是根节点，于是很显然，所有的根节点都不会乐意从任何一方那里做一个消息接收者，认为自己所有的端口都是指定端口，为自己传播信息。但是当收到 config 内容之后，优先级的端口很快就发现自己不是真正的指定端口，因此赶紧修改自己的身份并且转告节点，节点意识到自己有一个端口收到了来自更高优先级的消息，也意识到自己不是真正的根节点，因此需要赶紧扫描自己所有的非指定端口，从中选出一个优先级最高的端口，用这个端口和根节点建立连接（这样的端口总能找到，否则这个节点肯定没有接入网络，或者根节点没有接入网络）。这样一轮之后，出了根节点外，其余所有节点都将生成一个根端口。之后节点在依据情况修改各个端口的信息（如果是指定端口，指定端口认定的根要和节点保持一致；否则检查这个端口是不是拥有比节点更好的信息（这个是可能的，因为端口收到更好的信息是就会把这个 config 复制下来，但是节点此时可能还未更新）），最终网络将达到收敛。

3 实验结果

因为电脑屏幕比较小，这一次不方便截图，所以这次的实验报告没有附加的图片结果。但是我修改了 dump_output.sh 文件，使其能够依据出现的.txt 文件数量不同而以不同的长度来 dump 这些文件，老师可以在运行 four_node_ring.py 后光标闪烁四下时 sudo 运行 dump_output.sh 获取四个节点的输出，这个和 ppt 上面的参考是一致的（可能会因为死锁的问题导致有的文件没有 dump 信息，重复运行可以解决），对 six_node_ring.py 参照同样的方法可以获得六个节点时的生成树（在光标闪烁 4 下或者 5 下时 sudo 运行 dump_output.sh，时间过长也可能导致死锁而没有输出）。

4 结果分析

四个节点的和 ppt 参考完全一致，这里略过，只分析六个节点两两连接的情况。

六个节点，b1, b2, b3, b4, b5, b6 按照先上后下先左后右的顺序排列。优先级和名字一致。最终的结果应该是 b1 为根节点，拥有五个指定端口；b2 是第二级的节点，拥有四个指定节点（第一轮广播就能确定）；b3 次一级，因为和 b2 有直接连接，这个端口会变成非指定端口，又因为和 b1 直接连接，因此这个非指定端口不会变成根端口，因此 b3 拥有一个根端口，三个指定端口和一个其他端口。b4, b5, b6 一次可以类推，分别拥有一个根端口，两个指定端口（和 b5 和 b6 链接），两个其他端口；一个根端口，一个指定端口（和 b6 连接），三个其他端口；一个根端口，四个其他端口。于是当前网络拓扑中不再有环路。而代码运行结果与我的描述一致。