关于SSTP协议

SSTP(Simple STP这是我自己取的名字)。我之所以这么称呼它是因为它没有很完善的标准，我也无法用数学来证明它的正确性(我认为这是一件很困难的事情,所以我的SSTP可能会有bug)。在 SSTP这个协议中所有的端口只有两种状态：block和transmit。端口的类型有三种：根端口，指定端口，非指定端口。交换机可以分为两种：root和非root。数据包只有一种BPDU。每个下面我来解释这个SSTP这个协议的实现过程。

1. 所有交换机上电选择自己为根桥。
2. 根桥上所有端口都为指定端口，没有根端口。
3. 根桥每两秒转发一次BPDU包。
4. 非根桥交换机只有一个根端口，且根端口是非根桥交换机到根桥交换机最短的路径的端口。
5. 非根桥把根端口接收的BPDU从所有的指定端口转发出去。
6. 当非根桥交换机的根端口在20s内收不到从根桥来的BPDU，交换机重新选自己为根桥，把所有的端口设置为转发状态，并且发送自己的BPDU包。
7. 根桥、根端口、非指定端口、指定端口的选举与STP一样。这里不多解释。
8. 当非根桥交换机的连着根端口的线出现故障或者根桥出现故障时，可用第6条规则来进行修复。
9. 当非根交换机不是连着根端口的线出现故障，该故障可以忽略，因为它不会影响生成树的结构。

BPDU的格式：

|  |  |
| --- | --- |
| 根桥的优先级 | int BID\_priority; |
| 根桥的MAC地址 | QString BID\_MAC; |
| 到根桥的cost | int cost; |
| 发送者的优先级 | int sender\_priority; |
| 发送者的MAC | QString sender\_MAC; |

关于SSTP的问题(这些问题仅仅是SSTP的问题，也有可能在STP中发生)：

1. 不能在删完线后添加一条比原来线的cost值更大的线

dwb