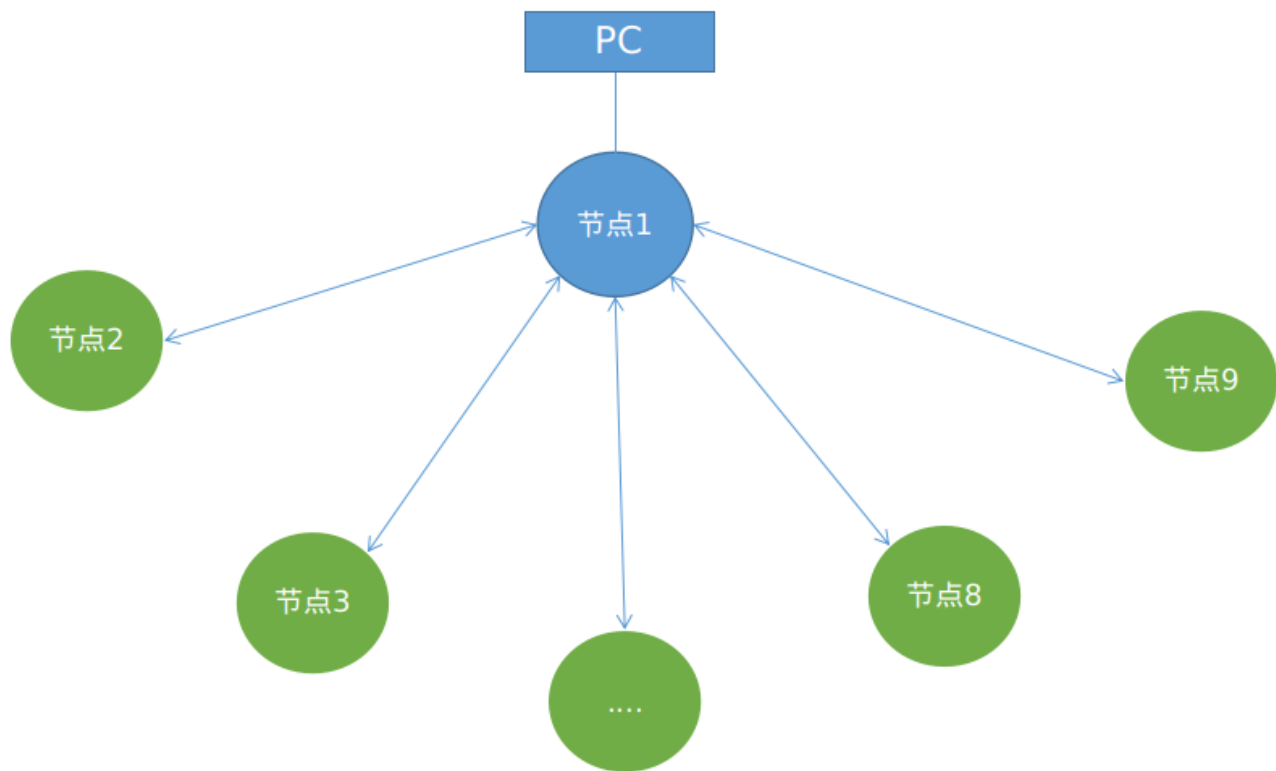


telink_mesh稳定性测试

测试方案

将灯节点分为两类，一类负责向mesh中所有节点发送开关命令，统计发出的命令次数 `counts1`，并统计其他节点响应开关的次数 `counts2`，这类节点只设置一个，即为节点1，节点1将数据通过uart传到PC端以便于统计。另一类节点有多个，接收节点1的开关命令，做出响应，并将本节点的开关次数传回节点1，节点2~9属于这类节点。



测试结果及问题

测试结果

大致测试了24小时左右，节点1发送开关命令的频率为1s一次。测试结果如下表所示，多数节点没有遗漏或者丢失消息，有一个节点少接收了一条消息。相较于之前nordic方案，比较稳定。五一期间，可以进行更长时间的测试。

| 节点 | 时间 (h) | 命令数 |
|----|--------|--------------|
| 1 | 24 | 87451 |
| 2 | 24 | 87451 |
| 3 | 24 | 87451 |
| 4 | 24 | 87451 |
| 5 | 24 | 87451 |
| 6 | 24 | 87451 |
| 7 | 24 | 87450 |
| 8 | 24 | 87451 |
| 9 | 24 | 87451 |

测试中发现的问题

1. 意博mesh的app (ios版) 在节点数量较多时, 手机发热严重, 且出现卡死。五个节点时, 还可以接受, 当九个节点都入网后, 卡顿严重。
2. 当多个节点同时向节点1发送消息时, 该节点会收到并处理多个相同消息。原因可能是由于mesh的泛洪消息机制造成的, 一个节点发出的消息会被多个节点收到并转发, 所以到达目的节点的消息有多条同样消息。目的节点根据消息中的sno值来判断此条消息是否已经收到了, 如果没收到则处理, 否则丢弃。但是, 当多个节点同时向节点1发送数据时, 节点1在同一时间, 收到了大量消息, 比如三个节点同时向节点1发送一条消息时, 节点1实际上可能收到了20条数据, 节点1收到4条来自于同一节点的同一条消息, 且来不及判断sno值, 导致节点1把该消息处理了4次。