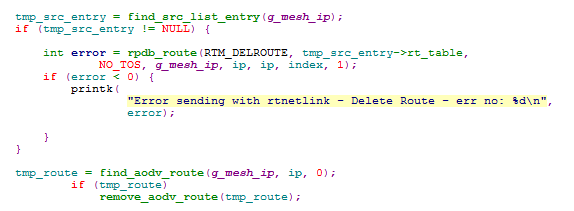
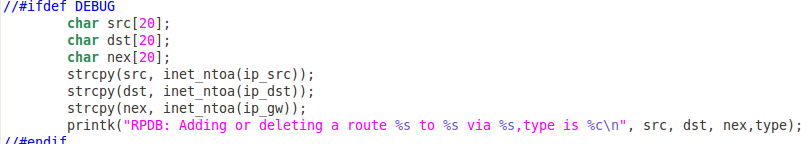
**断路机制验证**

1. **源代码修改处**

aodv-Android工程仅在当邻居节点失去联系（即维护邻居的任务超时）时删除邻居节点并产生断路包，在原工程中，在删除时即删除路由导致gen\_rerr()函数无法正常进行。故在aodv\_neigh.c中，将delete\_aodv\_neigh()函数中的以下代码注释掉即可。



在rpdb.c文件的rpdb\_route()函数中，为调试信息加入处理路由的类型以辨别是增加还是删除路由条目。



1. **实验结果**

节点3，4，5，6的IP分别为192.168.1.3，192.168.1.4，192.168.1.5，和192.168.1.6。

**模型1.一字型**

节点3只能通过4连接5，通过4，5连接6，节点4只能经过5到达6，模型完全对称。

3

4

5

6

实验（1）——删除节点6

数据流：在实验过程中3不停地往6发送数据。

|  |  |  |
| --- | --- | --- |
| 节点 | 行为 | 路由表操作 |
| 5 | 删除邻居节点6，并调用gen\_rerr(192.168.1.6)产生断路包 | 删除6->3 via 4  删除 3->6 via 6 |
| 4 | 接收到来自5的rerr包recv\_rerr | 删除6->3 via 3  删除3->6 via 6 |
| 3 | 接收到来自4的rerr包recv\_rerr | 删除3->6 via 4 |

实验（2）——删除节点5

数据流：在实验过程中3不停地往6发送数据。

|  |  |  |
| --- | --- | --- |
| 节点 | 行为 | 路由表操作 |
| 6 | 删除邻居节点5，并调用gen\_rerr(192.168.1.5)产生断路包 | 删除6->3 via 5 |
| 4 | 删除邻居节点5，并调用gen\_rerr(192.168.1.5)产生断路包 | 删除6->3 via 3  删除3->6 via 5 |
| 3 | 接收到来自4的rerr包recv\_rerr | 删除3->6 via 4 |

删除节点3或者4行为与5，6类似。

**模型2.Y字型**

节点3和4不互为邻居，节点3，4均只能经过5到达6，节点6只能经过5到达节点3，4。

3

4

5

6

实验（1）——删除节点6

数据流：节点3和4不停地给节点6发送数据。

|  |  |  |
| --- | --- | --- |
| 节点 | 行为 | 路由表操作 |
| 5 | 删除邻居节点6，并调用gen\_rerr(192.168.1.6)产生断路包 | 删除6->3 via 3  删除6->4 via 4 |
| 4 | 接收到来自5的rerr包recv\_rerr | 删除4->6 via 5  删除6->3 via 5 |
| 3 | 接收到来自5的rerr包recv\_rerr | 删除3->6 via 5  删除 6->4 via 5 |

实验（2）——删除节点5

数据流：节点3和4不停地给节点6发送数据。

|  |  |  |
| --- | --- | --- |
| 节点 | 行为 | 路由表操作 |
| 6 | 删除邻居节点5，并调用gen\_rerr(192.168.1.5)产生断路包 | 删除6->4 via 5  删除6->3 via 5 |
| 4 | 删除邻居节点5，并调用gen\_rerr(192.168.1.5)产生断路包 | 删除4->6 via 5 |
| 3 | 删除邻居节点5，并调用gen\_rerr(192.168.1.5)产生断路包 | 删除3->6 via 5 |

实验（3）——删除节点4

数据流：节点3不停地给节点6发送数据，6则不停地给4发送数据。

|  |  |  |
| --- | --- | --- |
| 节点 | 行为 | 路由表操作 |
| 5 | 删除邻居节点4，并调用gen\_rerr(192.168.1.6)产生断路包 | 删除4->6 via 6  删除3->6 via 6  删除6->4 via 4  删除6->3 via 3  此后若3有数据发往6，则重建路由6->3 via 3 |
| 3 | 后重建3->6 via 5 | 删除4->6 via 5(?) |
| 6 | 接收到来自5的rerr包recv\_rerr | 删除6->4 via 5  删除6->3 via 5 |

删除节点3的行为与4的类似。