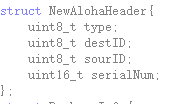
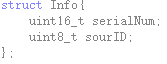
收到从物理层传来的MsgRecvDataNtf时，就会调用SendUp函数。（物理的上传来的Packet对象一般都是Up的，因此调用Header就会得到当前指针，强转成MAC层的封装头部即可找到对应的头部信息。）



有应答Aloha的头部有数据类型（1表示ACK，0表示DATA），还有源目标MAC地址，和串号（只是为了保证发出的MAC数据包的唯一性，MAC层每发出一个包就**该节点**的MAC串号加一。）



当收到Data数据的时候，会创建一个Info结构体，放入收到Data数据的串号和源MAC地址（以便万一再次收到该Data的时候，又往网络层传一次这个Data。只要确定串号和源MAC地址即可确定是不是同一个MAC数据包）



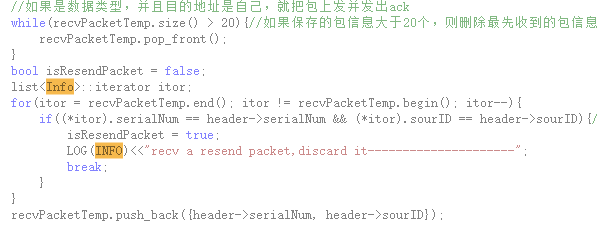
协议类中的私有成员，一个队列，保存了所有收到数据的串号和源MAC地址。



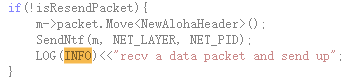
当获得了所有头部信息之后，协议通过判断type是DATA还是ACK。

一、如果**是DATA**，就判断目标MAC地址是不是自己，或者是不是广播包（其他情况都不处理）

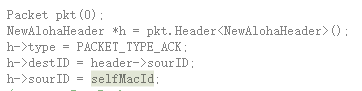
1. 假如MAC地址是自己的：先判断recvPacketTemp中是否已经满了（20个），如果满了就剔出第一个，然后从队列中找是否有对应的包（串号和源MAC都一样），再设置标志位（如果是重复的DATA，则置1）。



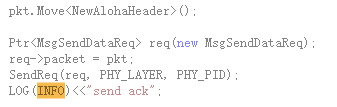
如果收到重复的DATA，就不往上传，否则往上传。



但不论是否重复DATA，都要发送ACK。（生成的是一个只有所有协议头的Packet对象，这时没有上层的封装，但是当MAC层收到这个包的时候，就知道这时一个ACK包，因此不再往上传）







2. 假如收到的是一个广播包：直接上传，不需要做其他操作（不需要发ACK，也不需要重传）



二、如果是**ACK包**：需要判断ACK包的目标MAC是不是自身，并且串号和源MAC是否能对应上。（存在的话，只要把packetGroup中对应的包删去即可，**没有往网络层发其他消息**。这里定时器事件应该还是会发生的，只是在重传的时候，在packetGroup中找不到该数据包，就没有发出去了。）

