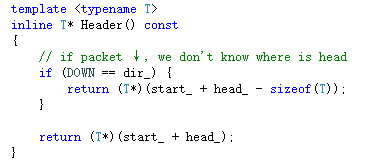
解封装：

通过Header函数，如果是往上传的话，直接返回起始指针+下层封装大小。（这里主要是为了获得该层的封装信息）

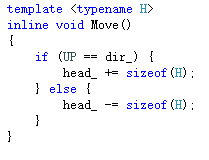




由Packet类来完成解封装：（这里主要是为了移动“指针”head\_的大小）



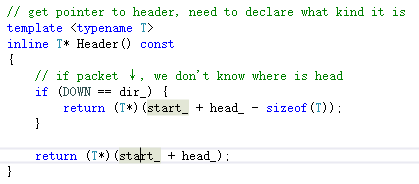
传入模板参数是协议头，会讲head\_加上当前传入的协议头的大小。（这里的主要目的是该head\_的大小，后面需要对Packet对象进行操作的时候就可以知道包头的位置）



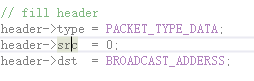
封装：



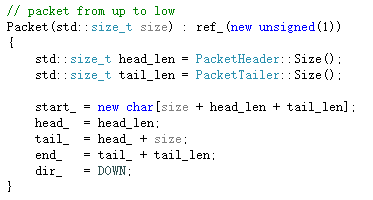
Packet对象会把添加完头部大小之后的地址返回，在这之后，需要协议强转成该协议头结构体的格式，然后改参数。



协议中的改参数：



注意：从上层往下传的时候，会申请纯数据+总头部的大小。



从下层往上传的时候，不会额外申请头部大小，因为只需要解封装。（并且初始化的时候就已经把方向也初始化好了）

