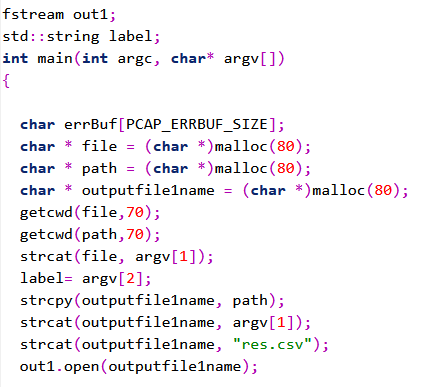
**文件一 main.cpp**

**1 数据包获取功能**

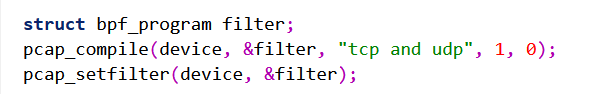
通过调库实现数据包获取：

**1）打开离线文件**

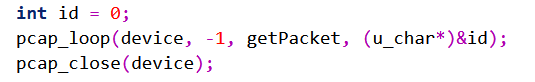


通过命令行获取文件名及标签。输出流和文件标签以全局变量的方式共享。

**2）设置过滤器**



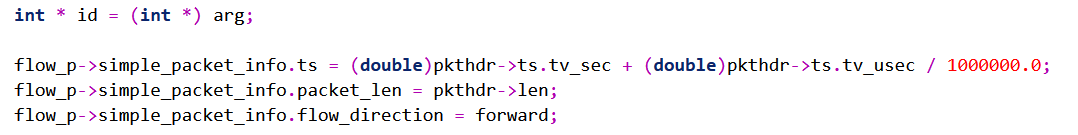
**3）开始循环抓包**



**2 首部信息提取功能**

在main文件里全局定义了flow\_process类对象flow\_p，flow\_p中simple\_packet\_info成员暂存数据包首部信息。暂存的信息中，流标记用于判断归属与方向，其它的首部信息后续传递给流对象用于特征提取。

**1）在每个包到达时首先更新抓包相关信息**



pkthdr为libpcap定义的结构体，包含结构信息和包长度。

**2）逐层获取首部**

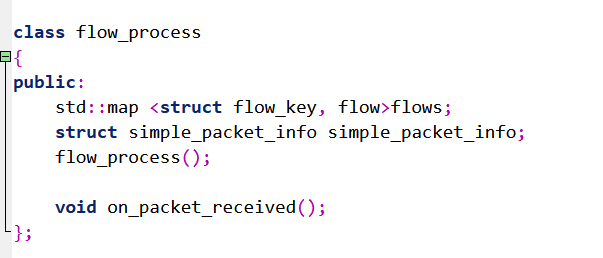
getPacket最后一个参数packet是完整数据包内容。（在flow.h中）定义各层次首部结构体，保证每个字段都和协议首部字段一致，那么将packet和带便宜的packet（如packet+14，packet+34）不同层次结构体，就可以直接截出来首部信息。

**3）告知flow\_p新包到达**



**文件二 flow\_process.h flow\_process.cpp**

**1)类定义**



包含一个流标记到流的映射，一个暂存首部信息的结构体。方法on\_packet\_receive()完成四个任务：包所属判断，包方向判断，流是否反向的判断，流是否超时的判断

**2）包方向和归属判断**

通过正反流标记（反指源、目的端口号、IP地址交换）进行比对，判断包基于流的方向。一致为正向，否则为反向。流的方向等同于该流第一个包的方向（后续可能改变）。

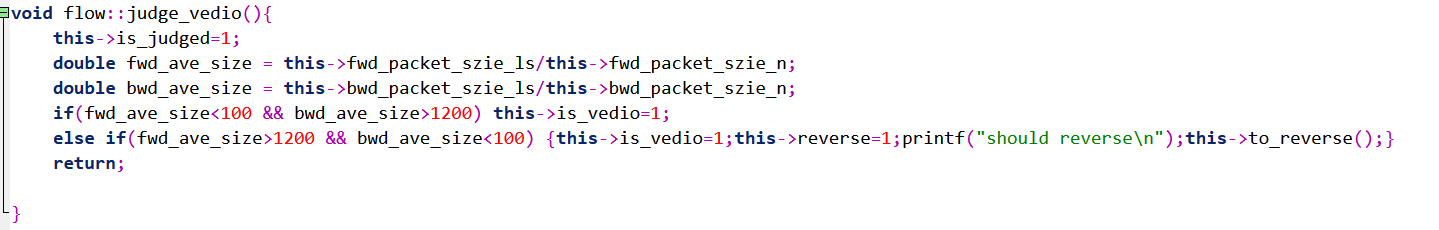
**3）流反向和超时的判断**

如果流反了，把包的方向反转.如果流超时了，把它从映射中删除，在新建具有相同流标记的流。

**文件三 flow.h flow.cpp**

**1）视频流量判断与方向判断相关**

视频流量的特点是上行流量只有段请求（约500B，低频）和ACK（约80B，高频），下行流量只有视频（长度=MTU）。因此上行流量的平均长度在一百以内，下行流量平均长度在1200以上。flow\_process中的方向由流的第一个数据包的方向确定。而flow中将反向包当作下行流量处理。所以可能出现判断反了的状况。对于视频流量，需要分别以正反和反正向平均包长与上述平均长度进行匹配。



反向有100个包，且没有判断过的时候进行判断。（粗糙）



若视频反向，则应交换其正、反向流量并清空段特征。

**2）特征计算相关**

特征分为两种。第一是直接计算特征，比如带宽，吞吐量。对于这种特征，需要记录部分实时数据并进行计算。第二是需要计算统计数据的特征，仅统计他们的加权个数（\_n后缀），加权和（\_ls后缀），加权平方和(\_ss后缀)，用以计算加权平均值和加权方差。

**3）特征获取**

获取三类共八种特征，分别为：

下行：包长sd，包间隔id，吞吐量t，带宽b

上行：请求间隔riu

块：块大小tsb，块持续时间db，块中包数pnb

**4）部分分辨率指标实时计算**

1.停滞事件

当块中包数发生显著下降时并维持低水平时，认为发生卡顿。标记卡顿包并记录卡顿时长。