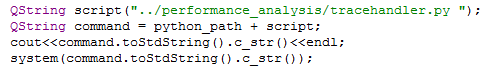
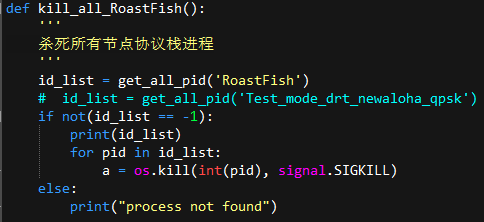


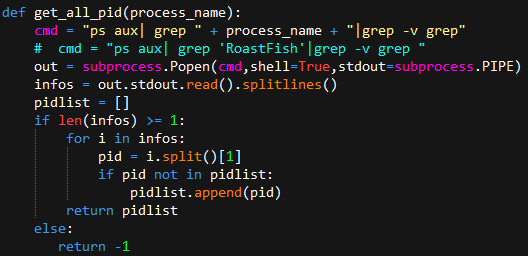
**Show Log**：是直接启动性能分析模块。



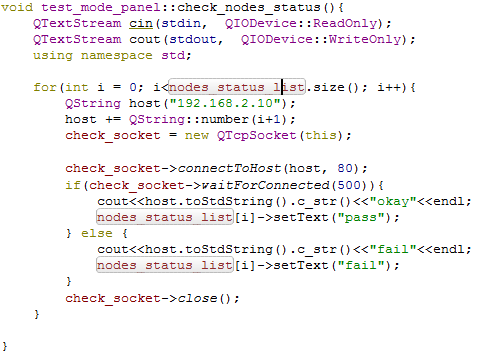
**Stop Test：**调用python函数kill\_all\_RoastFish.py来杀进程。

（主要是执行ps aux | grep 'RoastFish' | grep -v grep ，即找到含有RoastFish，但是不含grep的进程。grep -v grep是反向查找，查找不含grep的进程。）



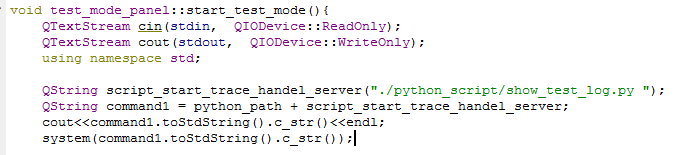


**检查连通button**：使用QTCPSocket类。分别对5个节点建立tcp连接，建立成功就在text上写pass，否则fail。（nodes\_status\_list是一个vector，保存Qlabel。）

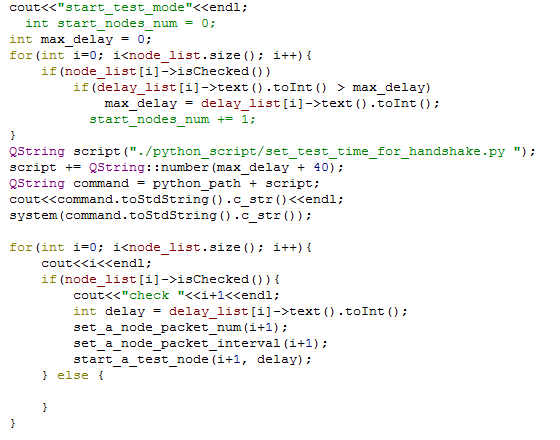


**开始测试button：（槽函数）**

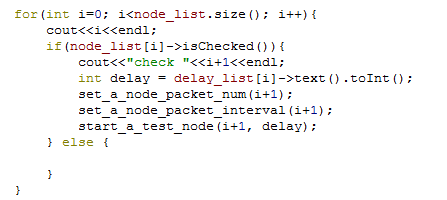
点击test mode按钮，先启动性能分析模块：（这里show\_test\_log和直接启动tracehandler.py是一样的。）



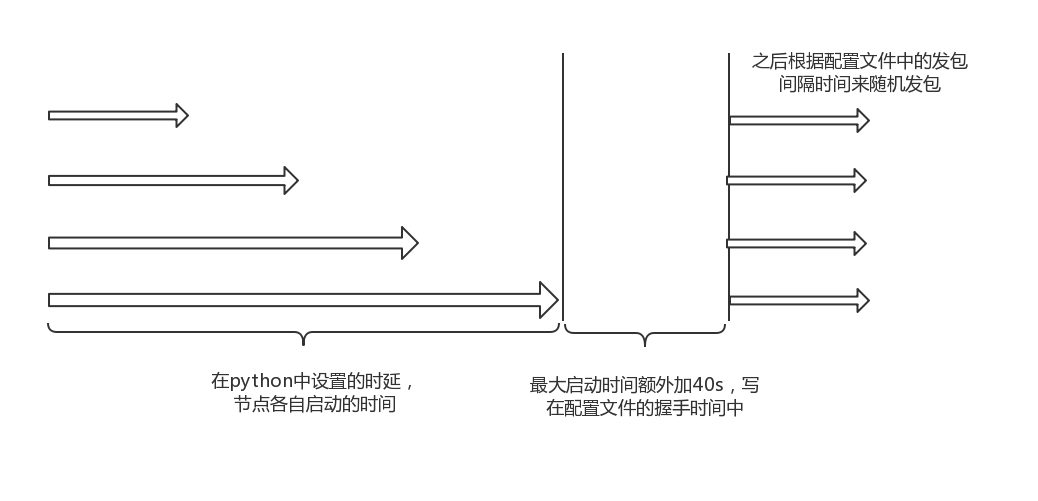
从所有时延里面找到最大的时延，然后加40s，作为握手时间，修改配置文件：（握手时间是测试模式的配置数据，主要是给demo用的。测试模式下设置了定时时间，在这个时间之后才开始发data包）



然后一个个节点启动：

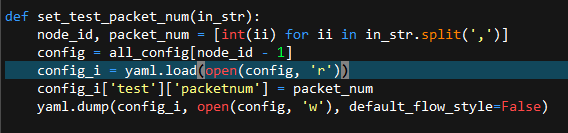


set\_a\_node\_packet\_num是设置测试模式的发包数，set\_a\_node\_packet\_interval是设置测试模式的发包间隔（mintime-maxtime），start\_a\_test\_node是设置测试模式下的节点启动时延（注意这里不是修改配置文件，而是在python中添加的额外添加的时延），并且启动节点：

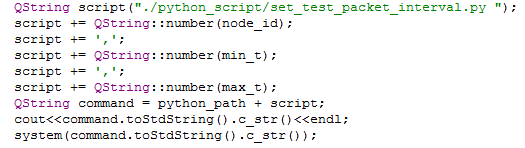


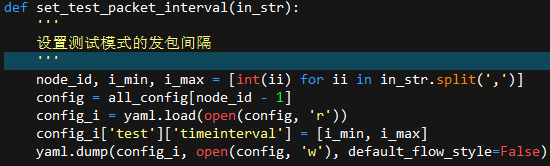
设置发包数。（也是testmode的参数，提供给demo。并且每个节点发包数可以不同，在界面设置。）



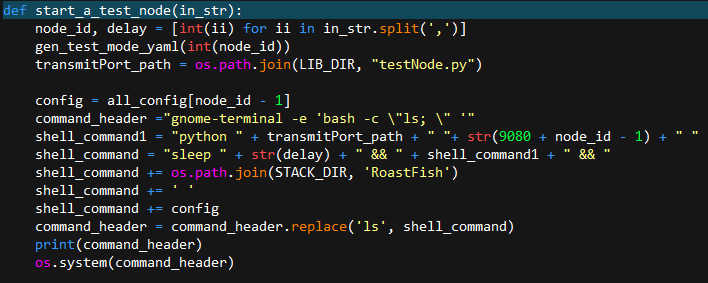


设置发包间隔。（发包间隔有两个值，最小最大，demo从中间找到随机值，然后设置定时器。）





启动节点：（测试模式，需要传入节点ID和时延，并且启动testNode.py，通知性能分析模块与协议栈进程建立套接字连接。如果是chat模式，直接启动即可。）



testNode.py，往性能分析进程（16888）发送应该连接的端口。

