在Server端（P2PServer.cpp）中，首先创建并初始化套接字，绑定端口，然后调用InitializeCriticalSection初始化g\_PeerListLock这个客户列表，让这个列表成为一个临界资源对象，从而能够做到线程间同步互斥访问。然后创建一个I/O线程，从而使得I/O操作能够与定时向客户发询问包验证客户是否在线（类似心跳功能）操作并行执行。

在这个I/O线程中，一直不断循环，从而不断接收所有客户端发来的请求包。通过pMsg这个CP2PMessage对象中的nMessageType变量，来得知客户端发送过来的消息的种类。如果是USERLOGIN（即101）,那么就是有用户登录，对这个新用户存到用户列表中，同时发送确认消息，将用户的公共地址传递过去。如果是USERLOGOUT（即102），那么就是有用户登出，调用g\_PeerList.DeleteAPeer删除该用户。如果GETUSERLIST（即104），那么就是有用户输入了”getu”，要求获得用户列表。于是利用循环将g\_PeerList.m\_pPeer这个数组中的所有用户的名称发送过去。如果是P2PCONNECT（即108），那么就是用户要求与另一个用户建立连接，于是在EnterCriticalSection和LeaveCriticalSection这两个加锁/解锁语句中间调用g\_PeerList.GetAPeer获得另一个用户的信息，加锁是为了互斥，向用户发送CP2PMessage类型的消息。如果是USERACTIVEQUERYACK（即107），那么就是在main()函数中不断向所有客户端发送询问消息的应答，证明用户仍然存活，更新该用户dwLastActiveTime（即最后验证的存活时间）。

在Client端（P2PClient.cpp）中，初始化一个CMyP2P对象，由用户在终端窗口输入服务器的IP和自己的用户名，调用Login登录服务器。由于是第一次登录，需要调用GetUserList发送GETUSERLIST请求，更新用户列表，并向用户的终端窗口输出状态信息。然后不断循环，等待用户向终端窗口输入命令。

如果用户输入getu，那么调用GetUserList获得用户列表并向终端打印。如果用户输入send和对方的用户名以及要发送的消息，那么调用SendText函数，构建封包，首先直接发向目标，给peerAddr这个地址，然后通过服务器转发，给serverAddr这个地址，请求对方向自己打洞，最后等待对方的P2PCONNECTACK消息，打洞完成。如果用户输入exit，那么结束退出。

同时，在CMyP2P类中还有static DWORD WINAPI RecvThreadProc(LPVOID lpParam)，创建了一个接收线程，与主线程并发执行，不断接收服务器或其他节点发来的请求包。当接收到请求包时，调用HandleIO函数，查看CP2PMessage对象的nMessageType，得知其请求包的类型。如果是USERACTIVEQUERY，那么就是服务器询问本用户是否还存活，发送m\_LocalPeer（本用户信息）通知服务器仍然存活。如果是GETUSERLIST，那么就是通过”getu”获得的用户列表。如果是P2PCONNECT，就是服务器或者某个节点发送过来的打洞信息。若是节点发送过来的，那么取得本用户信息后发送ackMsg。若是服务器转发的，那么向所有用户的peerAddr发送打洞消息。如果是P2PCONNECTACK，那么就是接收到了节点的打洞消息的应答，在这里设置它的P2P通信地址。

150420120 王晓玮