eMULE 源码分析

Author: 刘刚 ganghust@gmail.com MSN：ganghust@hotmail.com
博客：http://hustlg.bokee.com
部分翻译和内容材料来源于网络，一并向原作者表示感谢。

已经查看的源代码的版本包括：
1。eMule 0.42b VeryCD0229
2。eMule(电骡) v0.45b 源码
3。eMule0.47a-Sources
4。eMule0.47b-Sources
5。eMule-0.47c-VeryCD1215-Src
从 0。42B 到 0。47 版本主要增强和修改的地方包括：
（1）Web 管理功能的增强；
（2）对服务器探测时间参数的优化；
（3）客户端上传队列的过程进行了优化：排队机制和规则上的修改；
（4）文件缓存 cache 部分改进；
（5）KAD 网络中搜索功能的改进；
源代码版本：0.47cVeryCD 版

1 目录结构：

模块序号	模块大小	模块名称	备注说明
1		src	eMule 源代码主要工程
2	Debug:245K Release:105K	Zlib	数据压缩支持库，传输过程中支持数据压缩

3	Debug:4404K Release:2171K	Id3lib	是用于读、写和操纵 ID3v1 和 ID3v2 标签的 对于媒体类型的文件，它能够调用 id3lib 库来获取诸如作者，唱片发行年代，风格等 tag 信息。如果是视频媒体文件，它还会去抓图
4	Debug:653K Release306K	Png	提供对 PNG 文件处理的支持
5	Debug:1432K Release:517K	Resizable Lib	一个界面库,可以根据父窗口的位置和大小动态调整控件窗口的大小.
6	Debug:29284K Release:26305K	Crypto51	密码类库，实现了各种公开密钥算法、对称加密算法、数字签名算法、信息摘要算法。eMule之用主要是实现RSA签名，支持独有的积分机制

2 Src\目录下代码结构

模块序号	模块名称	备注说明
1	CxImage	图像处理库，与Windows、MFC支持极好，支持图像的多种操作（线性滤波、中值滤波、直方图操作、旋转缩放、区域选取、阈值处理、膨胀腐蚀、alpha混合等等）
2	Kademlia	KAD 网络的支持
3	Lang	本地语言化支持
4	Res	eMule 使用的资源包括图标，音乐文件等
5	Wordfilter	搜索关键词语过滤

2。1 主连接的启动和断开：

1。启动 void CemuleDlg::StartConnection()　src/interface/emuleDlg.cpp 2000 行

2。断开连接 void CemuleDlg::CloseConnection()　src/interface/emuleDlg.cpp 2021 行

3. 连接服务器动作的发起点在 sockets.cpp 的 145 行

4。EMSocket.cpp 的 161 行开发发起连接，先检查是否设置了代理。

1。ProcessCommandline。Emue.cpp的331行

通过在注册表里添加一些项目可以让一个程序和某种链接或者某个后缀的文件产生关联。具体办法可以参见OtherFunctions.cpp中的Ask4RegFix，BackupReg，RevertReg三个函数的功能。

启动流程：

3 重要的功能子类

序号	类名称	备注与说明
1	CPreferences	掌握着程序的大部分配置数据，它们的特点都是有很多的成员变量，而且还是静态的，这种方式可以保证它们的唯一性，而且把这些变量统一到一个类管理。但是实际上并不需要了解每个变量的含义。thePrefs和theStats是这两个类的唯一的实例。
2	CStatistics	后者则进行各种统计，thePrefs 和 theStats 是这两个类的唯一的实例。在 CPreferences.cpp 的 485 行 Init 函数中开始创建原始的配置文件目录；
3	SafeFile StringConversion CFileDataIO DataIO	数据操作的行为和数据操作的对象分割开来, 这些类要读取的数据对象通常有这些，各种整型，字符串，以及 Tag 类型。整型读写起来比较简单，从 1 个字节的，2 个字节的到 4 个，8 个或者 16 个字节类型的数据读写方法都比较类似。
4	CKnownFileList CKnownFile(CKnownFile 把读到的文件信息都保存成一个一个的 Tag。它在运行中会尽量得获取更多的文件信息，例如，对于媒体类型的文件，它能够调用 id3lib 库来获取诸如作者，唱片发行年代，风格等 tag 信息。如果是视频媒体文件，它还会去抓图(功能实现：CFrameGrabThread)。CKnownFile 还能够随时掌握目前该文件的下载情况(内部有个 CUpDownClient 的列表)，当然，还会根据要求序列化和反序列化自己，LoadFromFile 和 WriteToFile 都以 CFileDataIO 为参数，这样方便 CKnownFileList 保存和读取它的列表中的所有文件的信息)	CKnownFileList 类使用了 MFC 的 CMap 类来维护内部的 hash 表，它内部维护了一个已知的文件的列表和取消了的文件列表。hash 表的关键字都是文件 hash 值。能够判断出文件名不同而内容相同的文件。 CKnownFile 类是一个专门关注某个特定文件的信息的类，它仍然有其基 CAbstractFile。但是它和 CAbstractFile 类的主要区别就是 CAbstractFile 类只有基本的信息存取的功能，而 CKnownFile 能够主动的生成这些信息，
5	CKnownFile::CreateFromFile CAICHHash CAICHHashTree CAICHHashAlgo	emule中的分块处理和恢复机制,分块处理以及hash计算相关的类都在SHAHashSet.cpp和SHAHashSet.h中。下面介绍其中几个主要的类：CAICHHash类只负责一块hash值，提供两个CAICHHash类之间的直接赋值，比较等基本操作。CAICHHashAlgo是一个hash算法的通用的接口，
6		兼容 CAsyncSocket 类，即把应用程序中所

	CAsyncSocketEx ThrottledControlSocket ThrottledFileSocket (任何其它的网络套接字类如果想实现限速的功能，只需要在其默认的发送函数(如 Send 或 Sendto)中不发送数据而是把数据缓存起来，然后在实现 ThrottledControlSocket 或者 ThrottledFileSocket 接口中的 SendFileAndControlData 或 SendControlData 方法时才真正把数据发送出去)	以的 CAsyncSocket 换成 CAsyncSocketEx，程序仍然能够和原来的功能相同，因此在使用上更加方便。效率更高，主要是在消息分发机制上，即它处理和 SOCKET 相关的消息的效率要比原始的 MFC 的 CAsyncSocket 类更高。另外，CAsyncSocketEx 类支持通过实现 CAsyncSocketExLayer 类的方式，将一个 SOCKET 分成若干个层，从而可以很方便得实现许多网络功能，如设置代理，或者是使用 SSL 进行加密等
7	UploadBandwidthThrottler 保存若干Socket队列，这些队列的处理方式略有不同。在标准队列(m_StandardOrder_list)里面排队的都是实现了ThrottledFileSocket接口的类，通常这些类能够传输文件内容也可以传输控制信息。而其它四个队列都是实现ThrottledControlSocket接口的类的队列,主要以传输控制信息为主。这四个队列为临时高优先级，临时普通优先级，正式高优先级，正式普通优先级。和把套件字直接添加到普通队列(AddToStandardList) 不同，QueueForSendingControlPacket把要添加到队列的套接字全部添加到两个临时队列。根据它们的优先级添加到普通的临时队列。在RunInternal的大循环中，临时队列中的项目先被移到普通队列中，然后再进行处理。 UploadBandwidthThrottler使用了两个临界区，两个事件。pauseEvent是用来暂停整个大循环的动作的。而threadEndedEvent是标志整个线程停止的事件。sendLocker是大循环中使用的主要的临界区，而tempQueueLocker是为两个临时队列额外添加的锁，这样可以一边发送已有队列中的套界字要发送的数据，一边把新的套接字加到队列中。	一个 WinThread 的子类，平时单独运行一个线程。控制全局的上传速度的. UploadBandwidthThrottler 的 RunInternal 中的大循环是该工作线程的日常操作。这个大循环中做了以下事情，计算本次配额，即本次循环中能够发送多少字节，好安排调度，计算本次循环应该睡眠多少时间，然后进行相应的睡眠，从而进行限速。操作控制信息队列，发送该队列中的数据，注意，控制队列中的套接字(m_ControlQueueFirst_list 和 m_ControlQueue_list)只使用一次就离开队列。而标准队列中的套接字不会这样。在一轮循环结束后，如果还有没有用完的发送数据的配额，则会有部分配额保存到下一轮。
8	CEMSocket CEMSocket 是 CAsyncSocketEx 和 ThrottledFileSocket 的子类。它可以分出状态，如当前是否在发送控制信息等。 考察它的 SendControl	CEMSocket 的 SendControlData 和 SendFileAndControlData 方法其实都是调用自己的另一个重载的 Send 方法。这个方法是在 UploadBandwidthThrottler 的工作线程中的大循环中被调用的，而这个 Send 方法的内容本身也是一个大循环，就是在不

	Data 和 SendFileAndControlData 方法,这些方法是用来和 UploadBandwidthThrottler 进行配合，以便完成全局的限速功能的。它的功能应该是按照 UploadBandwidthThrottler 的要求，在本次轮到它发送数据时发送指定数量的字节数。 因此，应用程序的其它部分在使用 CEMSocket 时，如果要达到上传数据限速的目的，不应该直接调用标准的 Send 或者 SendTo 方法，而是调用 SendPacket。这里就有了另外一个结构 Packet，它通常包含一个 emule 协议中完整的包，例如有协议的头部数据等，还内置了 PackPacket 和 UnPackPacket 方法，可以自行进行压缩和解压的功能。SendPacket 把要发送的 Packet 放到自己的队列中，这个队列也有两个，控制信息包队列，和标准信息包队列。如果有必要，把自己加入到 UploadBandwidthThrottler 的队列中。	超过自己本次发送的配额的情况下，把自己的包队列中的包取出来，并且发出去。用到了一个临界区，它是为了保证从包队列中取出包来发送和把包往队列中放的操作是互斥的。把它和 UploadBandwidthThrottler 结合起来，就看到了一个两层的队列，即所有的套接字组成了一个发送队列，在 UploadBandwidthThrottler 的控制下保证了对速度的限制，而每个套接字即将发送的数据包又组成了一个队列，保证了每次进行数据发送的时候都会满足 UploadBandwidthThrottler 的要求。
9	CSearchList CSearchFile 是 CAbstractFile 的另一个子类(CKnownFile 也是)，它保存了某个文件和搜索相关的信息，而不是这个文件本身的信息，就是都在哪些机器上有这个文件，以及哪个服务器上搜到的这个文件。甚至还可以向搜索文件添加预览。在这个类的定义中嵌套定义了两个简单的结构 SServer 和 SClient，表示了该搜索文件的可能来源，服务器或者其它客户端。m_aClients 和 m_aServers 是这两个简单结构的一个数组，CSearchFile 自然也提供了对这个数组的操作的接口，方便 CSearchList 使用。 CSearchList 对外提供了搜索表达的接口，即每当有一个新的搜索提交时 CSearchList::NewSearch 会建立一个新的搜索项，但是此时还没有任何对应的搜索文件，因此只是在文件个数和搜索 ID 的对应表(m_foundFilesCount 和 m_foundSourcesCount)中建立新的项目。	CSearchList 是 emule 中的搜索列表，掌管 emule 中所有的搜索请求。CSearchFile 是这个列表中的元素，代表了一次搜索的相关信息。它们的关系和之前描述的已知文件和已知文件列表有一些类似的地方。 CSearchList 的主要任务就是对其一个叫做 list 的类型为 CSearchFile 列表的内部变量进行维护，提供很方便得往这个列表中添加，删除，查询，变更等操作的接口。另外，每一个搜索都有一个 ID，是一个 32 位的整数。CSearchList 中记录了每个搜索目前搜到的文件个数和源的个数(m_foundFilesCountm_foundSourcesCount) 当有搜索结果返回时 ProcessSearchAnswer 或 ProcessUDPSearchAnswer 能够对返回的包直接做处理，创建相应的搜索文件信息 CSearchFile 对象，并加入到自己的列表中。当然，要把重复的搜索结果去除，发现同一个 hash 的文件的多个源时也会给它们建立一个二级列(CSearchFile::m_list_parent)。 CSearchList 只负责和搜索有关的信息的储存和读取，本身并不进行搜索。
10	CServerList CServerList 除了提供通常的 CServer 信息外，还提供一些统计信息诸如所有的服务器的用户数，共享的文件数等。这些统计信息也是基于每个单独的 CServer 的相关信息计算出来的。	CServerList 是 emule 中负责管理服务器列表的类。CServerList 需要对外提供列表的增加，删除，查找，修改等接口。在 CServerList 中，每个服务器的信息是一个 CServer 类。和搜索信息不一样，但是和已知文件列表一样，服务器的信息列表是需

		要长期保留的，因此 CServerList 和 CKnownFileList 类一样提供了把它所包含的所有信息保存到一个文件中，以及从这个文件中读回其信息的功能。CServer 中的结构比较简单，只需要保留服务器的各种信息即可。它可以通过 IP 地址和端口来创建，也可以通过一个简单的结构 ServerMet_Struct 来创建，其中后者是用来直接从文件中读取的。该结构仅仅包含 IP 地址和端口以及属性的个数，CServer 中其它的属性在保存到文件中时，均采用 Tag 方式保存。
11	Packet 它内部实现了压缩和解压的方法，该方法直接调用 zlib 库中的压缩方法，可以减少数据的传输量。这里要注意一点的就是压缩的时候协议簇代码是不参与压缩的，压缩完毕后会更换协议簇代码，例如代码为标准 edonkey 协议 0xE3 的包在压缩后，协议代码就变成 0xD4 了，这里进行协议代码变化是为了使接受方能够正确识别并且进行相应的解压操作。	它是 emule 的通信协议的最小单位。我们可以看出，它的构造函数有多个版本，这也是为了可以用不同的方式来创建 Packet。例如只包含一个头部信息的缓冲区，或者只是指定协议簇代码等。

4TCP 和 UDP 网络通迅过程详细介绍：

通迅的双方及方式	完成此功能的主要类	说明
Client－Server TCP	CServerConnect （通过分析 CServerSocket::ProcessPacket 就可以直接把emule客户端和服务器之间的通信协议理解清楚，这里是服务器发回的包。TCP连接建立后的第一个包是在 CServerConnect::ConnectionEstablished中发出的，即向服务器发出登陆信息。如果登陆成功，则能够从服务器处获取自己的ID，这是一个32位的长	CServerConnect 本身不是套接字的子类，但是它的成员变量 CServerSocket 类型的 connectedsocket 是。CServerConnect 内部有一列表，可以保存若干 CServerSocket类型的指针。但是这并不说明它平时连接到很多服务器上。它只是可以同时试图连接到若干个服务器上，这只是因为连接到服务器上的行为不一定能成功。CServerSocket类是CEMSock

	整数。如果这个数小于16777216，那么我们称它为LowID。具有LowID的客户端通常情况下其它客户端将不能直接连接它。得到LowID的原因比较多，例如当自己处于NAT的后端的时候。获取自己的ID后将会向服务器发送自己的共享文件列表，这一动作由共享文件列表类CSharedFileList来完成。）	et的子类，它比CEMSocket要多保存一些状态，比如当前的服务器连接状态。它同时还保留它当前所连接的服务器的信息。
Client-Client TCP	CListenSocket CClientReqSocket （CListenSocket和CClientReqSocket类之间的关系和前面分析的列表类和它对应的成员类的关系是相似的，CListenSocket提供对自身的CClientReqSocket列表中的元素的增加，查询，删除等操作。同时也维护关于这些成员的一些统计信息。我们注意到CListenSocket在其构造函数中就把自己添加到CListenSocket类(theApp.listensocket，该类的唯一实际示例)的列表中。 CClientReqSocket类和CUpDownClient类之间存在着对应关系。它们都表示了另外一个客户端的一些信息，但是CClientReqSocket类主要侧重在网络数据方面，即负责两边的互相通信，而CUpDownClient类负责的是从逻辑上对网络另一边的一个客户端进行表达。）	由CListenSocket和CClientReqSocket完成。这也是提供网络服务的应用程序的典型写法。其中CListenSocket只是CAsyncSocketEx的子类，只负责监听某个TCP端口。它只是内部有一个CClientReqSocket类的列表。而CClientReqSocket是CEMSocket的子类，因此它能够自动完成emule的packet识别工作。它有ProcessPacket和ProcessExtPacket来处理客户端和客户端之间的包，其中前者是经典的eDonkey协议的包，后者是emule扩展协议的包。
Client－Server UDP	CUDPSocket	走UDP协议的包，因为UDP本来就是以一个包一个包作为单位在网络上流传的，不需要在包的内容中再包含表示长度的字段。每个UDP包的第
Client-Client UDP	CClientUDPSocket Kademlia::CKademliaUDPListener	

		一个字节是协议簇代码，其它内容就是包的内容。CKademliaUDPListener类，专门处理和Kademlia协议相关的UDP包。

5eMule 中信誉机制的实现

	主要实现的类	说明
1	CClientCreditsList	和信誉相关的信息是需要永久保存的，这样才有意义，因此 CClientCreditsList 提供了 LoadList和SaveList方法。
2	CClientCredits	CClientCredits 类可以使用 CreditStruct结构来创建，而 CreditStruct结构只包含静态信息，主要是上传量和下载量等。
信誉机制的信息需要有一定的可靠性，在emule中采用了数字签名的方式来做到这一点。Crypto++库为emule全程提供和数字签名验证相关的功能。CClientCreditsList在创建时，会装载自己的公钥私钥，如果没有的话，会创建一对。CClientCreditsList中包含的有效的信息都是经过其它人数字签名的，所以更加有信服力。在实际使用中，这些信息和自己的私钥要注意保存。重装emule后应该把配置文件目录先备份，这样能够保留自己辛辛苦苦积攒的信誉。		

6 下载如上传任务及队列的详细说明：

功能说明	类	说明
部分文件表示	CPartFile（CPartFile 它的创建就有几种可能，从搜索文件 CSearchFile 中创建，这种情况发生在用户搜索到他想要的文件后点击下载时发生。从一个包含了 ed2k 链接的字符串中创建，它会提取出该 ed2k 链接中的信息，并用来创建 CPartFile。剩下的一种，就是当 emule 程序重启后，恢复以前的下载任务。这时就是去	CPartFile 类是 emule 中用来表示一个下载任务的类。这就是一个还没有完成的文件。当一个下载任务被创建时，emule 会在下载目录中创建两个文件，以三位数字加后缀 part 的文件，例如 001.part，002.part 等。还有一个以同样的数字加上.part.met 的文件，表示的是对应文件的元信息。part 文件会创建得和原始文

	下载目录中寻找那些.part 和.met 文件了。另外它还需要不断得处理下载到的数据，为了减少磁盘开销，使用了 Requested_Block_Struct 结构来暂存写入的数据。它内部维护一个 CUpDownClient 的列表，如果知道了该文件的一个新的来源信息，就会创建一个对应的 CUpDownClient。后者是 emule 中代码量最大的类。它还要把它的状态用彩色的条装物显示出来提供给 GUI。）	件大小一样，当下载完成后，文件名会修改成它本来的名称。而事实上，诸如这个文件原来叫什么名称，修改日期等等信息都在对应的.part.met 元文件中。.part.met 中还包含了该文件中那些部分已经下载完成的信息。CPartFile 类中 Gap_Struct 来表示文件的下载情况，一个 Gap_Struct 就是一个坑，它表示该文件从多少字节的偏移到多少字节偏移是一个坑。下载的过程就是一个不断填坑的过程。CPartFile 类中有个成员变量 gaplist 就是该文件目前的坑的状况列表。需要主要的是有时填了坑的中间部分后，会把一个坑变成两个坑。坑的列表也会被存进.part.met 中。
下载	CDownloadQueue （CDownloadQueue 中的 Process 方法的主要任务就是把它的列表中的 CPartFile 类中的 Process 方法都调一遍，另外主要的一些关于下载情况的统计信息也是在每一轮的 Process 后进行更新的。从这里我们也可以看出 Process 方法在 emule 中的意义，就是一个需要经常执行的方法，通过经常执行它们来完成日常工作，而且所有的这些 Process 方法肯定是顺序执行，因此可以减少很多多线程的同步之类的问题。emule 中已经尽量减少了多线程的使用，但是在很多地方如果多线程是不可避免的话，也不会排斥。 ）	CDownloadQueue 是下载队列类。这个队列中的项目是 CPartFile 指针。它需要能够提供对这个列表中的元素进行增加，查询，删除的功能。例如查询的时候能够根据该文件的 hashID 或者索引来进行查询。CDownloadQueue 同时还要完成一些统计工作。 和其它的列表类不一样的是，它的所有元素的信息并不是集中存放于一个文件，而是对应于每一个下载任务，单独得存放在一个元信息文件(.part.met)中，因此当该类进行初始化的时候，它需要寻找所有可能的下载路径，从那些路径中找到所有的.part.met 文件，并且试图用这些文件来生成 CPartFile 类，并且将这些通过.part.met 文件正确生成的 CPartFile 类添加到自己的列表中，同样，在退出时，所有的下载任务的元信息也是自行保存，不会合成为一个文件。
上传	CUploadQueue （CUploadQueue 的 Process 方法就相对简单了，那就是向上传队列中的所有客户端依次发送数据，而排队的客户端是不会得到这个机会的。另外它还需要完成关于上传方面的一些统计信息。在 CUploadQueue 的构造函数里面，创建了一个以 100 毫秒为间隔的定时器，这个定时器成为以上所有的 Process 所需要的基础。）	CUploadQueue 是上传队列类。这个列表类中只有以 CUpDownClient 为元素的列表，它和其它列表类还有一个很大的不同就是它所保存的信息都不需要持久化，即不需要在当前的 emule 退出后还记住自己正在给谁上传文件，然后下次上线的时候再继续给他们传，这在大部分情况下是没有意义的。他有两个列表，上传列表和排队列表。当收到一个新的下载请求后，它会把对应的客户端先添

		加到排队列表中，以后再根据情况，把它们不断添加到上传列表中。在这里，信誉机制将会对此产生影响。
上传和下载的辅助类	CUpDownClient BaseClient.cpp 中实现的是该类的一些基本的功能，包括基本的各种状态信息的获取和设置，以及按照要求处理和发送各种请求。在这里，逻辑实现和网络进行了区分，CUpDownClient 类本身不从网络接受或者发送消息，它只是提供各种请求的处理接口，以及在发送请求时，构造好相应的 Packet，并交给自己对应的网络套接字发出去。 DownloadClient.cpp 中实现的是和下载相关的功能，它包括了各种下载请求的发送以及相应的数据的接收。另外还有一个 A4AF 的机制，它是 emule 中的一个机制，因为一个客户端在同一个时间内只能向另外一个客户端请求同一个文件。这样，对于很多个下载任务(CPartFile)，有可能出现它们的源(即有该文件的客户端)有部分重叠的现象，而这时，如果其它下载任务正在从这个源下载，那么当前的下载任务就不能从这个源下载了。但是 emule 允许用户对其手动进行控制，如对下载任务的优先级进行区分，这样他就可以将一个源从另外一个下载任务那里切换过来。A4AF 其实就是 ask for another file 的简称。 UploadClient.cpp 中实现的是上传相关功能，即接受进来的下载请求，并且生成相应的文件块发送出去。	CUpDownClient 类的作用是从逻辑上表示一个其它的客户端的各种信息，它是 emule 中代码量最大的类。我们注意到，定义它的头文件是 UpDownClient.h，但是却没有对应的 CUpDownClient.cpp，而它的实现，都分散到 BaseClient.cpp，DownloadClient.cpp，PeerCacheClient.cpp，UploadClient.cpp 和 URLClient.cpp 中。 PeerCacheClient.cpp 实现的是和 PeerCache 相关的功能，PeerCache 是一个由 Joltid 公司开发的技术，它可以允许你从 ISP 提供的一些快照服务器上快速得上传或者下载一些文件(或者是一部分)，这个技术的好处是可以减少骨干网络的带宽消耗，将部分本来需要在骨干网上走的流量转移到 ISP 的内部。当然这个功能需要 ISP 的配合。如果发现 ISP 提供了这项服务的话，emule 会利用它来减少骨干网的带宽消耗。 URLClient.cpp 实现的功能是利用 http 协议对原有的 emule 协议进行包装，以便使它能够尽可能地穿越更多的网络的防火墙。

7 其他辅助功能类的说明：

在 emule 中使用到的其它类及其功能。我们可以看到，如果单纯只是为了能够搜到以及下载到文件的话，有不少类是可以精简的，但是，正是由于它们的存在，使得 emule 的功能更加的完善。CIPFilter，IP 地址过滤器，通过识别各种类型的 IP 地址过滤信息，它能够把不希望连接的网络地址过滤掉，emule 中所有需要连接网络的地方使用的都是统一的过滤数据。CWebServer 能够在本地打开一个 Web 服务器，然后你可以通过浏览器来控制你的 emule。CScheduler 能够实现下载任务的定时下载。CPeerCacheFinder 为前面提到的 PeerCache 技术的主控制类。另外，emule 还内置了一个 IRC 客户端，一个主要成员函数都为静态的

CPartFileConvert 类，能够对其它版本的驴的下载文件进行转换。它甚至还提供了一个自动处理 zip 和 rar 的类 CArchiveRecovery。

8 协议通迅过程的主要约定如下：

emule 的通信协议格式设计成一种便于扩充的格式。对于 TCP 连接来说，连接中的数据流都能够划分成为一个一个的 Packet，CEMSocket 类中就完成了把接收到的数据划分成 Packet 这一工作。但是具体的对于每个 Packet 进行处理的工作被转移到它的子类中进行。
CEMSocket 类的两个子类 CServerSocket 和 CClientReqSocket 所代表的 TCP 连接就分别是客户端和服务器之间的 TCP 连接以及客户端之间的 TCP 连接。在数据流中的第一个字节代表的是通信的协议簇代码，如 0xE3 为标准的 edonkey 协议，0xE4 为 kademlia 协议等等。接下来的四个字节代表包内容的长度，所有的包都用这种方式发送到 TCP 流中，就可以区分出来了。另外每个包内容中的第一个字节为 opcode，即在确定了某个具体协议后，这个 opcode 确定了这个包的具体含义。
对于走 UDP 协议的包，处理起来更加得简单，因为 UDP 本来就是以一个包一个包作为单位在网络上流传的，因此不需要在包的内容中再包含表示长度的字段。每个 UDP 包的第一个字节是协议簇代码，其它内容就是包的内容。CClientUDPSocket 类负责处理客户端和客户端之间的 UDP 包，而 CUDPSocket 类负责处理客户端和服务器之间的 UDP 包。另外还有个 Kademlia::CKademliaUDPListener 类，专门处理和 Kademlia 协议相关的 UDP 包。

附录 一 ED2K 通迅报文处理细节：

（以 0。47 版 VeryCD 为准），Ethereal 截获报文（连接服务器的 TCP 5000 端口）下面的是登陆过程中实际的通迅报文。
1。登陆报文
报文结构如下：

```
0000    00 00 0c 07 ac 47 00 18 8b 88 94 30 08 00 45 00    .....G.....0..E.
0010    00 7b 49 91 40 00 80 06 65 e5 ac 1e 15 7b 55 11    .{I.@...e....{U.
0020    34 5c 08 c7 13 88 fb 92 96 ad 1f e8 22 d4 50 18    4\..........".P.
0030    ff ff 4b 74 00 00 e3 4e 00 00 00 01 eb 28 9a dd    ..Kt...N.....(..
0040    33 0e 64 10 a4 d1 42 9d c2 91 6f 62 dc aa 01 00    3.d...B...ob....
0050    e5 6a 04 00 00 00 02 01 00 01 15 00 5b 43 48 4e    .j..........[CHN
0060    5d 5b 56 65 72 79 43 44 5d 79 6f 75 72 6e 61 6d    ][VeryCD]yournam
0070    65 03 01 00 11 3c 00 00 00 03 01 00 20 19 03 00    e....<..... ...
0080    00 03 01 00 fb 00 bd 00 00                         .........
```

2。登陆回应报文：

```
0000    00 18 8b 88 94 30 00 0e d6 d4 cc 00 08 00 45 00   .....0........E.
0010    00 83 c3 5e 40 00 28 06 3e d9 c1 8a cd 19 ac 1e   ...^@.(.>.......
0020    15 7b 13 88 08 c8 82 96 3b 09 ba aa 2b f4 50 18   .{......;...+.P.
0030    16 d0 7e 90 00 00 e3 56 00 00 00 38 53 00 57 41   ..~....V...8S.WA
0040    52 4e 49 4e 47 20 3a 20 59 6f 75 20 68 61 76 65   RNING : You have
0050    20 61 20 6c 6f 77 69 64 2e 20 50 6c 65 61 73 65    a lowid. Please
0060    20 72 65 76 69 65 77 20 79 6f 75 72 20 6e 65 74    review your net
0070    77 6f 72 6b 20 63 6f 6e 66 69 67 20 61 6e 64 2f   work config and/
0080    6f 72 20 79 6f 75 72 20 73 65 74 74 69 6e 67 73   or your settings
0090    2e                                                .
```

3.ID改变 和服务器消息

服务器发送 ID 改变的消息，作为对登录请求消息的回应和表示服务器已经接收客户端的连接。

```
0000    00 18 8b 88 94 30 00 0e d6 d4 cc 00 08 00 45 00   .....0........E.
0010    02 7b c3 5f 40 00 28 06 3c e0 c1 8a cd 19 ac 1e   .{._@.(.<.......
0020    15 7b 13 88 08 c8 82 96 3b 64 ba aa 2b f4 50 18   .{......;d..+.P.
0030    16 d0 a0 e1 00 00 e3 11 00 00 00 40 50 30 72 00   ...........@P0r.
0040    f9 07 00 00 88 13 00 00 db 85 28 fa e3 09 00 00   ..........(.....
0050    00 34 85 46 01 00 a1 28 37 01 e3 22 00 00 00 38   .4.F...(7.."...8
0060    1f 00 73 65 72 76 65 72 20 76 65 72 73 69 6f 6e   ..server version
0070    20 31 37 2e 31 34 20 28 6c 75 67 64 75 6e 75 6d    17.14 (lugdunum
0080    29 e3 65 00 00 00 38 62 00 ef bb bf e5 8e 8c e7   ).e...8b........
0090    83 a6 e4 ba 86 e7 ad 89 e4 b8 8a e5 a5 bd e5 87   ................
00a0    a0 e5 a4 a9 e6 89 8d e8 83 bd e8 8e b7 e5 8f 96   ................
00b0    e6 96 87 e4 bb b6 ef bc 9f e4 b8 8d e6 83 b3 e8   ................
00c0    ae a9 e5 85 b6 e4 bb 96 e4 ba ba e5 91 98 e8 ae   ................
00d0    bf e9 97 ae e6 82 a8 e7 9a 84 20 49 50 20 e5 9c   .......... IP ..
00e0    b0 e5 9d 80 ef bc 88 70 32 70 20 e3 21 00 00 00   .......p2p .!...
00f0    38 1e 00 e5 85 b7 e6 9c 89 e5 9c b0 e5 9d 80 e4   8...............
0100    ba a4 e6 8d a2 e5 8a 9f e8 83 bd ef bc 89 ef bc   ................
0110    9f e3 6a 00 00 00 38 67 00 e4 bd bf e7 94 a8 20   ..j...8g........
0120    68 74 74 70 3a 2f 2f 63 6e 2e 72 61 7a 6f 72 62   http://cn.razorb
0130    61 63 6b 33 2e 63 6f 6d ef bc 8c e6 82 a8 e5 8f   ack3.com........
0140    af e4 bb a5 e4 bd bf e5 ae bd e5 b8 a6 e8 bf 9e   ................
0150    e6 8e a5 e9 80 9f e5 ba a6 e8 be be e5 88 b0 e6   ................
0160    9e 81 e8 87 b4 ef bc 8c e5 b9 b6 e5 8f af e5 8c   ................
0170    bf e5 90 8d e4 b8 8b e8 bd bd 20 31 30 30 25 20   .......... 100%
0180    e3 0c 00 00 00 38 09 00 e5 86 85 e5 ae b9 e3 80   .....8..........
0190    82 e3 21 00 00 00 38 1e 00 e8 8e b7 e5 8f 96 e4   ..!...8.........
01a0    b8 a4 e5 91 a8 e7 9a 84 e5 85 8d e8 b4 b9 e8 af   ................
```

01b0 95 e7 94 a8 ef bc 81 e3 3d 00 00 00 38 3a 00 e5 .....=...8:..
01c0 9c a8 20 68 74 74 70 3a 2f 2f 63 6e 2e 72 61 7a .. http://cn.raz
01d0 6f 72 62 61 63 6b 33 2e 63 6f 6d 20 e4 b8 8a ef orback3.com ....
01e0 bc 8c e6 82 a8 e5 8f af e4 bb a5 e8 ae bf e9 97 ....
01f0 ae 20 33 30 30 20 54 42 20 e3 56 00 00 00 38 53 . 300 TB .V...8S
0200 00 e5 ae b9 e9 87 8f e7 9a 84 e5 ba 94 e7 94 a8 ....
0210 e7 a8 8b e5 ba 8f 5b e8 a8 bb 3a e8 bd af e4 bb ......[...:.....
0220 b6 5d e3 80 81 e7 94 b5 e5 bd b1 e3 80 81 e6 b8 .]..............
0230 b8 e6 88 8f e3 80 81 4d 50 33 20 e6 88 96 e4 b8 ....MP3 .....
0240 8d e5 8f 97 e9 99 90 e5 88 b6 e7 9a 84 20 58 58 XX
0250 58 e3 80 82 e3 30 00 00 00 38 2d 00 e6 af 8f e5 X....0...8-.....
0260 a4 a9 e5 a2 9e e5 8a a0 20 31 35 30 30 20 47 42 .... 1500 GB
0270 20 e4 bb a5 e4 b8 8a e5 ae b9 e9 87 8f e7 9a 84 ....
0280 e6 96 87 e4 bb b6 ef bc 81 ....

4。**文件提供**

该消息被客户端用来描述可获得的当地文件，让其它客户端下载。如果客户端有文件提供，文件提供消息立即在连接建立之后发送。当客户端共享文件列表发生改变时，该消息也会发送。该消息的另一用法是周期地发送到服务器，保持活动状态。如果服务器支持压缩，文件提供（消息）会被压缩。当用作保持活动状态（没有文件压缩和消息大小很小）时，该消息不被压缩。

0000 00 00 0c 07 ac 47 00 18 8b 88 94 30 08 00 45 00 .....G.....0..E.
0010 00 61 49 99 40 00 80 06 60 c0 ac 1e 15 7b c1 8a .aI.@...`....{..
0020 cd 19 08 c8 13 88 ba aa 2b f4 82 96 3d b7 50 18 ....+...=.P.
0030 fd 51 50 91 00 00 e3 34 00 00 00 15 01 00 00 00 .QP....4........
0040 0d 6f 41 12 a1 b5 61 d8 d0 de be 75 8b 28 b8 06 .oA...a....u.(..
0050 fb fb fb fb fb fb 03 00 00 00 9a 01 55 70 64 61 ....Upda
0060 74 65 2e 65 78 65 83 02 2b 7f 21 00 89 03 04 te.exe..+.!....
0000 00 00 0c 07 ac 47 00 18 8b 88 94 30 08 00 45 00 .....G.....0..E.
0010 00 28 49 9d 40 00 80 06 60 f5 ac 1e 15 7b c1 8a .(I.@...`....{..
0020 cd 19 08 c8 13 88 ba aa 2c 33 82 96 40 cd 50 10 ....,3..@.P.
0030 ff ff 50 58 00 00 ..PX..

5.获得服务器列表

e3 01 00 00 00 14 cd cd cd cd fd fd fd fd ab ab ab ab ab ab ab ab ee fe 00 00 00 00 00 00 00 00 ã.....ÍÍÍÍýýýý«««««««««îþ........

0x032E8798 0b 00 09 00 c5 07 23 04 78 74 02 03 58 87 2e 03 14 c8 58 01 8d 01 00 00 11 00 00 00 01 00 00 00 ....Å.#.xt..X ...ÈX. ....

0x032E87B8 a6 5d 07 00 fd fd fd fd 85 c5 19 00 f9 07 00 00 88 13 00 00 db 85 28 fa cd fd fd fd fd ab ab ab ¦]..ýýýý Å..ù... ...Û (úÍýýýý«««

6。 **服务器列表**

从服务器发送到客户端。该消息包含关于用另外的 eMule 服务器来扩展客户端服务器列表的信息。消息大小可变的（根据发送的服务器数目）。

```
0000    00 18 8b 88 94 30 00 0e d6 d4 cc 00 08 00 45 00    .....0........E.
0010    04 5e d6 7b 40 00 2e 06 27 18 55 11 34 5c ac 1e    .^.{@...'.U.4\..
0020    15 7b 13 88 05 86 93 99 ab b8 f0 e1 1a c3 50 18    .{............P.
0030    16 d0 73 da 00 00 e3 d4 03 00 00 32 a3 40 22 c2    ..s........2.@".
0040    dd 86 0d d0 35 93 1b c6 1c c2 92 e3 08 22 12 57    ....5........".W
0050    96 0f 0e bd 25 43 9f 2c ae 88 10 55 11 23 35 e1    ....%C.,...U.#5.
0060    10 53 95 65 5b e1 10 43 9f 2c 66 88 10 40 22 c2    .S.e[..C.,f..@".
0070    b3 7f 0d 43 9f 2c af 88 10 40 22 c1 3d 04 12 c1    ...C.,...@".=...
0080    8a dd d5 92 10 43 9f 2c ba 88 10 d4 b3 85 da 88    .....C.,........
0090    10 43 9f 2c a4 88 10 53 95 74 83 88 10 55 11 38    .C.,...S.t...U.8
00a0    15 5c 11 c1 8a e7 d2 92 10 43 9f 2c 78 88 10 40    .\.......C.,x..@
00b0    22 c1 51 a8 09 42 87 20 61 1c 16 43 9f 2c a3 88    ".Q..B. a..C.,..
00c0    10 55 11 38 08 a0 0f 43 9f 2c b5 88 10 43 9f 2c    .U.8...C.,...C.,
00d0    b3 88 10 43 9f 2c 73 88 10 43 9f 2c 6d 88 10 43    ...C.,s..C.,m..C
00e0    9f 2c a5 88 10 43 9f 2c be 88 10 40 22 c2 b4 70    .,...C.,...@"..p
00f0    0a 43 9f 2c 03 88 10 43 9f 2c bd 88 10 43 9f 2c    .C.,...C.,...C.,
0100    7c 88 10 43 9f 2c 6b 88 10 43 9f 2c ac 88 10 42    |..C.,k..C.,...B
0110    87 21 24 0c 21 d4 b3 85 db 88 10 43 9f 2c 80 88    .!$.!......C.,..
0120    10 43 9f 2c aa 88 10 53 95 7b bc e1 10 53 95 62    .C.,...S.{...S.b
0130    10 92 10 50 ef c8 6c b8 0b 43 9f 2c b0 88 10 43    ...P..l..C.,...C
0140    9f 2c ad 88 10 d4 e3 41 43 92 10 3e f1 35 03 92    .,.....AC..>.5..
0150    10 43 9f 2c 67 88 10 43 9f 2c 7a 88 10 53 95 75    .C.,g..C.,z..S.u
0160    38 e1 10 43 9f 2c 6a 88 10 43 9f 2c bc 88 10 40    8..C.,j..C.,...@
0170    1b 12 a9 1c 3d 43 9f 2c ab 88 10 3e f1 35 0f 92    ....=C.,...>.5..
0180    10 50 ef c8 67 b8 0b 42 5a 49 fd c3 22 43 9f 2c    .P..g..BZI.."C.,
0190    a2 88 10 43 9f 2c b9 88 10 43 9f 2c a9 88 10 53    ...C.,...C.,...S
01a0    95 7b bd e1 10 40 22 c1 da 12 13 50 ef c8 6a b8    .{...@"....P..j.
01b0    0b c2 d5 00 1e ea 0c c1 8a dd d6 92 10 43 9f 2c    ............C.,
01c0    7b 88 10 43 9f 2c bb 88 10 50 ef c8 65 b8 0b 50    {..C.,...P..e..P
01d0    ef c8 69 b8 0b 43 9f 2c a6 88 10 43 9f 2c 77 88    ..i..C.,...C.,w.
01e0    10 3e f1 35 04 92 10 50 ef c8 68 b8 0b 50 ef c8    .>.5...P..h..P..
01f0    6b b8 0b 43 9f 2c b1 88 10 50 fc 6e 92 35 12 c2    k..C.,...P.n.5..
0200    d5 00 0a ea 0c c1 8a e6 fb 92 10 40 22 c1 da 18    ...........@"...
0210    12 3e f1 35 11 92 10 3e 5a af 92 88 10 3e f1 35    .>.5...>Z....>.5
0220    02 92 10 c2 1e a0 29 35 12 50 ef c8 6e b8 0b 50    ......)5.P..n..P
0230    ef c8 6d b8 0b 50 ef c8 63 b8 0b 3e f1 35 10 92    ..m..P..c..>.5..
0240    10 43 9f 2c a0 88 10 43 9f 2c 74 88 10 43 9f 2c    .C.,...C.,t..C.,
0250    b2 88 10 43 9f 2c 76 88 10 43 9f 2c 7d 88 10 43    ...C.,v..C.,}..C
0260    9f 2c 81 88 10 43 9f 2c 75 88 10 43 9f 2c 7f 88    .,...C.,u..C.,..
0270    10 43 9f 2c 72 88 10 55 11 a8 83 e1 10 40 22 c1    .C.,r..U.....@".
0280    51 83 21 43 9f 2c 79 88 10 40 22 c4 46 d6 10 43    Q.!C.,y..@".F..C
```

```
0290    9f 2c a7 88 10 43 9f 2c 6f 88 10 43 9f 2c 71 88   .,...C.,o..C.,q.
02a0    10 43 9f 2c 70 88 10 43 9f 2c b7 88 10 c1 8a cd   .C.,p..C.,......
02b0    19 88 13 40 22 c2 dd 3d 09 50 ef c8 70 b8 0b 43   ...@"..=.P..p..C
02c0    9f 2c 6c 88 10 59 36 2b 8e 05 0d d0 7a 11 76 bb   .,l..Y6+....z.v.
02d0    01 53 8f bb 61 92 10 c1 a4 85 37 c3 22 53 95 66   .S..a.....7."S.f
02e0    01 92 10 50 ef c8 6f b8 0b 3e 9e 3f af 73 1a 55   ...P..o..>.?.s.U
02f0    11 34 5c 88 13 43 9f 2c 68 88 10 54 a0 75 8e d7   .4\..C.,h..T.u..
0300    c3 43 9f 2c 6e 88 10 42 87 22 c6 4e 20 43 9f 2c   .C.,n..B.".N C.,
0310    65 88 10 50 ef c8 66 b8 0b 43 9f 2c b8 88 10 57   e..P..f..C.,...W
0320    96 0f 0d bd 25 d5 96 3e b4 35 12 40 5d 5a e6 46   ....%..>.5.@]Z.F
0330    12 40 5d 5a d7 d2 1e 40 5d 5a e4 d5 1a 40 5d 5a   .@]Z...@]Z...@]Z
0340    dc 35 12 d0 44 eb 65 79 b7 40 5d 5a df d7 11 40   .5..D.ey.@]Z...@
0350    5d 5a e5 b3 15 40 5d 5a f2 0a 1a 40 5d 5a f0 94   ]Z...@]Z...@]Z..
0360    22 40 5d 5a f1 10 27 40 5d 5a e9 70 17 40 5d 5a   "@]Z..'@]Z.p.@]Z
0370    ec 35 12 40 5d 5a cc 35 12 40 5d 5a e3 88 13 40   .5.@]Z.5.@]Z...@
0380    5d 5a e0 0a 1a 40 5d 5a ef 05 1a 40 5d 5a da 35   ]Z...@]Z...@]Z.5
0390    12 40 5d 5a d5 35 12 40 5d 5a de 35 12 40 5d 5a   .@]Z.5.@]Z.5.@]Z
03a0    c8 35 12 40 5d 5a ca 35 12 40 5d 5a cd 35 12 40   .5.@]Z.5.@]Z.5.@
03b0    5d 5a c9 0a 1a 40 5d 5a d6 35 12 40 5d 5a d0 35   ]Z...@]Z.5.@]Z.5
03c0    12 40 5d 5a d1 35 12 40 5d 5a ce 1d 16 40 5d 5a   .@]Z.5.@]Z...@]Z
03d0    cf 35 12 40 5d 5a ea 94 26 40 5d 5a ee a0 0f 40   .5.@]Z..&@]Z...@
03e0    5d 5a e7 35 12 40 5d 5a ed 88 13 40 5d 5a cb 0a   ]Z.5.@]Z...@]Z..
03f0    1a 40 5d 5a db e4 17 40 5d 5a dd 35 12 40 5d 5a   .@]Z...@]Z.5.@]Z
0400    d2 35 12 d9 5b 3a 58 68 11 50 be f0 7d 35 12 e3   .5..[:Xh.P..}5..
0410    58 00 00 00 41 32 a2 b0 0a 26 d2 f5 21 4c 17 bc   X...A2...&..!L..
0420    24 29 4c 9f 5b 55 11 34 5c 88 13 02 00 00 00 02   $)L.[U.4\.......
0430    01 00 01 0d 00 52 61 7a 6f 72 62 61 63 6b 20 33   .....Razorback 3
0440    2e 30 02 01 00 0b 24 00 77 77 77 2e 72 61 7a 6f   .0....$.www.razo
0450    72 62 61 63 6b 33 2e 63 6f 6d 20 2d 20 66 65 65   rback3.com - fee
0460    6c 20 74 68 65 20 70 6f 77 65 72 21                l the power!
```

6.请求服务器，获得源

```
0000    00 00 0c 07 ac 47 00 18 8b 88 94 30 08 00 45 00   .....G.....0..E.
0010    00 42 20 62 40 00 80 06 8f 4d ac 1e 15 7b 55 11   .B b@....M...{U.
0020    34 5c 05 f9 13 88 14 ff 80 5b 0c 99 98 14 50 18   4\.......[....P.
0030    ff ff 4b 3b 00 00 e3 15 00 00 00 23 35 8b 5c 80   ..K;.......#5.\.
0040    82 1a c0 6e 5d b3 48 8a ae 2b 7b a4 ad c6 a1 12   ...n].H..+{.....
0050
```

下载过程中文件分片 9.28M 然后是 180KB 的块，分块校验和分析。

附录二 eMule 中 KAD 网络的说明

当 emule 中开始使用 Kademlia 网络后，便不再会有中心服务器失效这样的问题了，因为在这个网络中，没有中心服务器，或者说，所有的用户都是服务器，所有的用户也是客户端，从而完完全全得实现了 P2P。

模块	类名	详细说明
1。Kademlia 网络的主控类	CKademlia	Kademlia 的主控类是 CKademlia，它负责启动和关闭整个 Kademlia 网的相关代码。在它的 Process 函数中，会处理和 Kademlia 网相关的事务，例如隔一段时间检查某个区间的节点数是否过少，如果是则寻找一些新的节点。另外经常对自己的邻居进行检查等，这些都是属于需要进行日常安排的工作。所有搜索任务的日常处理也需要它来调度。它还作为 Kademlia 网的代表，向 emule 其它部分的代码返回 Kademlia 网的一些统计信息。
2。处理自身的 Kademlia 相关信息	CPrefs	CPrefs，它和 emule 普通代码中的 CPreferences 作用类似，但是 CPrefs 只保留和 Kademlia 网相关的，需要长期保存的本地信息。具体到这个版本来说，主要就是本地的 ID。
3。组成了每个节点所了解的联系信息以及由这些联系信息所组成的数据结构	CRoutingZone CRoutingBin CContact （离自身 ID 逻辑距离越近(也就是共同前缀越长)的联系人信息越有可能被加入，它所对应的节点越有可能因为分裂而获得更多的子节点，也就对应了更多的容量。在 KAD 网中，每一个参与者知道的其它参与者信息中，离自己逻辑距离越近的参与者比例越高。在搜索的时候也只需要不断得寻找更近的 ID，而且每一步都一定会有进展，所以寻找到目标 ID 所需要的时间上的代价是 O(logn)，从这个二叉树的	CRoutingZone 类处于联系人数据结构的最上层，直接为 Kademlia 网提供操作接口。该类的结构为一个二叉树，内含两个 CRoutingZone 指向它的左子树和右子树，另外也包含一个 CRoutingBin 类型的指针。但是只有在当前的 CRoutingZone 类为整个二叉树的叶节点时，这个指向 CRoutingBin 类型的指针才有意义。这个二叉树的特点是，每个节点以下的所有联系人的 ID 都包含一个共同前缀，节点的层数越深，这个共同前缀越长。例如，根节点的左子树的所有的节点的 ID 一定有一个前缀"0"，而右子树的所有节点一定有前缀"1"。同样，根节点的左子树的右子树下的所有节点的 ID 一定有前缀"01"，等等，依此类推。我们设想一下节点不断得往这个二叉树添加的过程。刚开始只有一个根节点，它也就是叶节点，这时它内部的 CRoutingBin 是有意义的，当联系人信息不断得被添加进去以后，这个 CRoutingBin 的容量满了，这时要进行的就是一个分裂的操作。这时，会添加两个左子节点和右子节点，然后把自身的 CRoutingBin 中的联系人信息按照它们的前缀特点分别复制往左节点和右节点，最后把自身的 CRoutingBin 废除掉，这样这个分裂过程就完了。当分裂完成后，就会再次试图添加该联系人信息，此时会试图按照它的 ID，把它添加到对应的子树中。但是并不是所有的这种情况节点都会发生分裂，因为如果允许任意分裂的话，本地所需存储的节点信息数量就会急剧上升。这里，自身 ID 的作用就体现了。只有当自身 ID 和当前准备分裂的节点有共同前缀时，这个节点才会分裂，而如果判断到一个节点不能分裂，而它的 CRoutingBin 又满掉了，那么就会拒绝添加联系人信息。 CRoutingBin 类包含一个 CContact 的列表。是要访问联系人的信息必须通过某个 CRoutingBin，CRoutingZone 内部是不直接包含联系人信息的。可以把新的联系人信息往一个特定的 CRoutingBin 中加，当然也可以进行联系人查找。它也提供方法能够寻找出离某个 ID 距离最近的联系人，并给出这样的一个列表。一个 CRoutingBin 类中能够包含的 CContact 的数量也是有限制的。 CContact 类包含的是一个联系人的信息，主要包括对方的 IP 地址，ID，TCP

	结构来看，由于只有部分节点会分裂，所以实质上存储所需要的空间代价也是O(logn)。实际上CRoutingZone在实现时和理论上的Kad有区别，如从根节点开始，有一个最低分裂层数，如果层数过低的话，是永远允许分裂的，这样它知道的其它地区的联系人信息就能够稍微多一些。）	端口，UDP端口，kad版本号和其健康程度(m_byType)。其中健康程度有0-4五个等级。刚刚加入的联系人，这个数值设置为3。系统会经常通过与各个联系人进行联系的方式对其进行健康状况检查，经常能够联系上的联系人，这个数值会慢慢减少到0。而很久没有联系的，这个数值会慢慢增加，如果增加到4后再过一段时间未能成功联系上的，则将会被从联系人列表中删除。
4。负责处理UDP网络信息	CKademliaUDPListener	CKademliaUDPListener负责处理所有和Kademlia网相关的消息。分拣工作已经在emule的普通UDP客户端处理代码那里处理好了。具体的消息分类：首先是健康检查方面的消息，这样的消息就是一般的ping-pong机制。对应的消息有KADEMLIA_HELLO_REQ和KADEMLIA_HELLO_RES。当对本地联系人信息列表进行检查时，会对它们发出KADEMLIA_HELLO_REQ消息，然后处理收到的KADEMLIA_HELLO_RES消息。 最常用的消息是节点搜索消息，在Kademlia网络中，进行节点搜索是日常应用所需要传输的主要消息，它的实现方式是迭代式的搜索。这种方式就是说当开始搜索某个ID时，在本地联系人信息列表中查找到距离最近的联系人，然后向它们发出搜索请求，这样通常都能够得到一些距离更近的联系人信息，然后再向它们发送搜索请求，通过不断得进行这样的搜索查询，就能够得到距离目标ID最近的那些联系人信息。这里对应的消息代码是KADEMLIA_REQ和KADEMLIA_RES。
5。处理本地存储的索引信	CIndexed	CIndexed中利用了一系列的Map来存储这些对应信息，CMap是MFC中实现标准STL中的map的模板类，CIndexed中包含了四个这样的类，分别用来存储文件源信息，关键词信息，文件评论信息以及负载信息。其中文件评论信息是不长久保存的，而其它的信息都会在退出的时候写到文件中，下次重新启动emule时再重新调入。另外负载信息不是等其它联系人来发布的，而是根据文件源信息和关键词信息的发布情况自行进行动态调整的。每一次收到发布信息时，对应的ID的负载会增大，这一事实会在回应消息(KADEMLIA_PUBLISH_RES)中体现。 CIndexed中的信息会经常进行检查，每隔三十分钟它会把自己存储的所有信息中太老的信息清除掉。其中文件源信息的保存时间为五小时，关键词信息为二十四小时，文件评论的信息保存时间也为二十四小时。因此文件的发布和关键词也要周期性得反复进行。其实这对于整个Kademlia网络的

		稳定性也是有好处的，因为每一次联系都会试图把对方添加到自己的联系人列表中，或者在联系人列表中标注上一次见到对方的时间。 CIndexed 为其它部分的代码提供了它们所需要的增加信息和搜索信息的接口，这样在从网络中获取到相关的搜索或者发布请求，并且 CKademliaUDPListener 完成消息的解释后，就可以交给 CIndexed 来进行处理了。 一个文件源信息是一个文件内容的 hash 值和拥有这个文件的客户端的 IP 地址，各种端口号以及其它信息之间的对应关系。而一个关键词信息则是该关键词和它对应的文件之间的关系。在关键词信息中，它对应的文件信息要更加详细，通常包括这个文件的文件名，文件大小，文件内容的 hash 值，如果是 MP3 或者其它媒体文件，还会包含包括作者，生产时间，文件长度(这个长度是用时间来衡量的媒体文件的播放长度),流派等等 tag 信息。其中文件内容的 hash 值用来区分该关键词对应的不同文件。
6。处理和搜索有关的操作	CSearch CSearchManager	CSearch 和 CSearchManager 是完成具体搜索任务的。CSearch 对应的是一个具体的搜索任务，它包括了一个搜索任务从发起到结束的全部过程，要注意的是搜索任务并不只是指搜索文件源或者关键词的任务，一次发布任务它也需要创建一个 CSearch 对象，并且让它开始执行。CSearchManager 则掌握所有的搜索任务，它包含了一个包含所有 CSearch 指针对象的 CMap，使用 CMap 的原因是因为所有的 CSearch 都一定对应一个 ID，那个 ID 就是该 CSearch 所对应的目标，不管是要查找节点，还是要搜索或者发布信息，一定都要找到和目标 ID 相近的联系人。因此 CSearchManager 可以使用 CMap 来表示所有的搜索任务。 CSearch 在创建的时候就把自己加入到 CSearchManager 当中。另外 CSearch 在创建的时候需要说明它的类型，例如是只是为了搜索节点还是要搜索关键词信息或者文件源信息，当然也有可能是发布文件源信息或者关键词信息。知道 CSearch 的几个方法的作用就可以大概了解 CSearch 的工作过程。Go 是它的启动过程，它会开始第一次从本地的联系人列表中寻找候选的联系人，然后开始发动搜索。SendFindValue 的功能就是向某个联系人发送一个搜索某 ID 的联系人信息这样一个请求。JumpStart 则是在搜索进行到一定地步的时候，如得到了一些中间结果，开始进行下一步的行动，下一步的行动仍然可能是 SendFindValue，也有可能认为搜索到的联系人离目标已经足够近了，于是就可以开始实质性的请求。StorePacket 就是这样一个实质性的请求，例如在一个以发布文件源为任务的 CSearch 中，StorePacket 会向目标联系人发送 KADEMLIA2_PUBLISH_SOURCE_REQ(如果不支持 Kademlia2，那么是 KADEMLIA_PUBLISH_REQ)。最后，CSearch 能够处理各种搜索结果，然后向调用它的代码返回处理好的结果。 CSearchManager 直接和 Kademlia 网的其它部分代码接触，例如，如果 CKademliaUDPListener 搜索到了一些结果，它会把这些结果交给 CSearchManager，然后 CSearchManager 再去寻找这个结果是属于那个搜索任务的，并且进行转交。另外 CSearchManager 对外提供创建各种新的搜索任务的接口，作用类似于设计模式中的 Factory，其它部分的代码只需要说明需要开始一个什么样的搜索任务即可，CSearchManager 来完成相应的创

		建 CSearch 的任务。
7。处理一个 128 位的长整数	CUInt128	CUInt128，实现对 128 位的 ID 的各种处理，并且内置其各种运算

附录: eMule 中内容发布或者搜索

emule 中存储在 Kademlia 网中的信息主要有三类：文件源，关键字信息和文件的评论。文件源对应的是每一个具体的文件，每个文件都用它的内容的 hash 值作为该文件的唯一标示，一条文件源信息就是一条关于某人拥有某个特定的文件的这样一个事实。一条关键字信息则是该关键字对应了某个文件这样一个事实。很显然，一个关键字可能会对应多个文件，而一个特定的文件的文件源也很有可能不止一个。但是它们的索引都以固定的 hash 算法作为依据，这样使得搜索和发布都变得很简单。

我们来看发布过程。每个 emule 客户端把自己的共享文件的底细已经摸清楚了，在传统的有中心索引服务器的场景里，它把自己的所有文件的信息都上传到中心索引服务器里。但是在 Kademlia 网里，它就需要分散传播了，它首先做的事情是把文件名进行切词，即从文件名中分解出一个一个的关键词出来，它切词的方法非常简单，就是在文件名中寻找那些有分割符含义的字符，如下划线等，然后把文件名切开。计算出这些关键字的 hash 值后，它把这些关键字信息发布到对应的联系人那里。并且把文件信息也发布到和文件内容 hash 值接近的联系人那里。对应的消息是 KADEMLIA_PUBLISH_REQ 和 KADEMLIA_PUBLISH_RES。另外 emule 允许用户对某个文件发表评论，评论的信息单独保存，但是原理也是一样的。

当用户使用 Kademlia 网络来进行搜索并且下载文件的时候，首先是对一个关键词进行搜索，由于使用的是同样的 hash 算法，这样它只要找到 ID 值和计算出来的 hash 值结果相近的联系人信息后，它就可以直接向它们发送搜索特定关键词的请求了。如果得到了返回信息，那么搜索者就知道了这个关键词对应了多少文件，然后把这些文件的信息都列出来。当用户决定下载某个文件的时候，针对这一特定文件的搜索过程就开始了，这一次如果搜索成功，那么返回的就是这个文件的文件源信息。这样 emule 接下来就只需要按照这些信息去连接相应的地址，并且使用传统的 emule 协议去和它们协商下载文件了。这里对应的消息是 KADEMLIA_SEARCH_REQ 和 KADEMLIA_SEARCH_RES。

实际的实现中有 Kademlia2 这种协议，它的原理是一样的，只有协议代码和具体的消息格式不一样，例如 KADEMLIA_REQ 和 KADEMLIA_RES 对应了 KADEMLIA2_HELLO_REQ 和 KADEMLIA2_HELLO_RES，但是后者在具体的消息中包含了比前者丰富一些的信息。在实现的时候 0.47c 更加倾向于使用 Kademlia2，而 0.47a 更加倾向于使用 Kademlia。当然，它们两种协议都能够处理。另外，0.47c 增加了一个对于已发出的请求的追踪的特性，就是一个包含 TrackPackets_Struct 类型的列表，这里面详细纪录了什么时间曾经对哪个 IP 发出过那种 opcode 对应的请求。为什么要这样呢？这是为了防止针对 DHT 的一种路由污染攻击，因为在搜索联系人的时候，如果搜索到了一些联系人信息，也会试图把它先加入到本地的联

系人信息列表中。这样如果有人想恶意攻击的话，它只要不断得往它想攻击的 emule 客户端发送 KADEMLIA_RES，并且在消息的内容中包含大量的虚假联系人信息，就可以使对方的联系人信息列表中充满垃圾。这样，由于缺少正确有效的联系人信息，它的 Kademlia 网功能基本上就废了。而在 0.47c 里面增加的这个特性，就会对那种还没有发出请求就收到回应的情况直接无视，从而避免被愚弄。