

# eMule 协议规范

本文档翻译自：

Yoram Kulbak and Danny Bickson 《The eMule Protocol Specification》

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。

eMule 协议规范.....	1
1 简介.....	3
1.1 目的和范围.....	3
1.2 概述.....	4
1.2.1 客户端到服务器的连接.....	4
1.2.2 客户端到客户端的连接.....	5
1.3 客户 ID.....	6
1.4 用户 ID.....	7
1.5 文件 ID.....	7
1.5.1 文件哈希.....	7
1.5.2 根哈希.....	7
1.6 eMule 协议扩展.....	8
1.7 软件和硬件限制.....	8
2 客户端服务器的 TCP 交流.....	8
2.1 建立连接.....	8
2.2 连接启动时消息交换.....	10
2.3 文件搜索.....	12
2.4 回调机制.....	12
3 客户端服务器的 UDP 交流.....	13
3.1 服务器保持连接和状态信息.....	13
3.2 增强文件搜索.....	15
3.3 增强文件源搜索.....	15
4 客户端到客户端的 TCP 交流.....	16
4.1 初始的握手.....	16
4.2 安全的用户身份认证.....	17
4.2.1 信用系统.....	17
4.3 请求文件.....	18
4.3.1 基本消息交换.....	18
4.3.2 没找到文件的情景.....	19
4.3.3 加入上传队列.....	19
4.3.4 上传对列管理.....	20
4.3.5 到达上传队列的顶部.....	20
4.4 数据传输.....	21
4.4.1 数据包.....	21

4.4.2 数据传输顺序.....	22
4.4.3 选择块下载.....	23
4.5 浏览共享的文件和文件夹.....	24
4.6 交换片哈希集.....	25
4.7 取得文件预览.....	26
5 客户端到客户端的 UDP 连接.....	26
6 附录详细的消息编码格式.....	28
6.1 一般消息编码要点.....	28
6.1.1 字节序.....	28
6.1.2 消息头.....	28
6.1.3 消息标签.....	28
6.2 客户端服务器 TCP 消息.....	29
6.2.1 登录.....	29
6.2.2 服务器消息.....	30
6.2.3 ID 改变.....	31
6.2.4 文件提供.....	31
6.2.5 获得服务器列表.....	33
6.2.6 服务器状态.....	33
6.2.7 服务器列表.....	34
6.2.8 服务器身份证明.....	34
6.2.9 搜索请求.....	35
6.2.10 搜索结果.....	37
6.2.11 获得源.....	38
6.2.12 已找到的源.....	39
6.2.13 回调请求.....	39
6.2.14 被请求回调.....	40
6.2.15 回调失败.....	40
6.2.16 消息被拒绝.....	41
6.3 客户端服务器 UDP 消息.....	41
6.3.1 获取源.....	41
6.3.2 发现的源.....	42
6.3.3 状态请求.....	42
6.3.4 状态回应.....	43
6.3.5 搜索请求.....	44
6.3.6 搜索回应.....	44
6.3.7 服务器描述请求.....	45
6.3.8 服务器描述回应.....	45
6.4 客户端到客户端 TCP 消息.....	45
6.4.1 Hello.....	46
6.4.2 Hello 回应.....	47
6.4.3 发送文件块.....	47
6.4.4 请求文件块.....	48
6.4.5 下载结束.....	49
6.4.6 改变客户 ID.....	49

6.4.7 聊天消息.....	50
6.4.8 块 hashset 请求.....	51
6.4.9 块 hashset 回应.....	51
6.4.10 开始上传请求.....	52
6.4.11 接受上传请求.....	52
6.4.12 取消传送.....	53
6.4.13 Out of part requests.....	53
6.4.14 文件请求.....	53
6.4.15 文件请求回答.....	54
6.4.16 找不到文件.....	55
6.4.17 被请求的文件 ID.....	55
6.4.18 文件状态.....	56
6.4.19 Change slot.....	56
6.4.20 队列等级.....	57
6.4.21 查看共享文件.....	57
6.4.22 查看共享文件回答.....	58
6.4.23 查看共享文件夹.....	58
6.4.24 查看共享文件夹回答.....	59
6.4.25 查看共享文件夹内容.....	59
6.4.26 查看共享文件夹内容回答.....	60
6.4.27 查看共享文件夹或内容拒绝.....	61
6.5 客户端到客户端 TCP 扩充消息.....	61
6.5.1 eMule 信息.....	61
6.5.2 eMule 信息回答.....	63
6.5.3 发送压缩的文件块.....	63
6.5.4 队列等级.....	64
6.5.5 文件信息.....	65
6.5.6 源请求.....	65
6.5.7 源回答.....	66
6.5.8 安全身份认证.....	67
6.5.9 公匙.....	67
6.5.10 签名.....	68
6.5.11 预览请求.....	69
6.5.12 预览回答.....	69
6.6 客户端到客户端 UDP 消息.....	70
6.6.1 重复询问文件.....	70
6.6.2 重复询问回应.....	70
6.6.3 队列满.....	71

# 1 简介

## 1.1 目的和范围

eMule是流行的文件共享程序，基于eDonkey协议。这份报告描述了eMule的网络行为和解释了理解该协议所需的基本术语。本报告也给出了eMule网络协议的完整规范，包括一个附录，它提供了消息格式。这份文档的信息是基于开源的eMule客户端。接下来的简介目的是提供基本的背景知识，让读者阅读和理解这份文档。关于eMule的更多消息在这里找到。

## 1.2 概述

eMule网络是由上百个eMule服务器和几百万个eMule客户端组成。客户端必须连接到一个服务器来取得网络服务，只要该客户端在系统中，服务器连接保持打开状态。这些服务器主要执行集聚索引服务（好像在Napster），相互间不联系。

每个eMule客户端都预配置了一个服务器列表和当地文件系统的共享文件列表。客户端用单独的TCP连接到一个eMule服务器登录到网络中，获得想得到的文件信息和客户端。eMule客户端也用几百个TCP连接到其他客户端进行上传和下载文件。每个eMule客户端对它的每个共享文件都维护着一个上传队列。要下载的客户端先加入到队列的底部，然后逐渐前进直到到达队列的顶部并开始下载它的文件。一个客户端可以从几个不同的eMule客户端中下载同一个文件的不同的文件块。客户端也可以上传它还没有完成的文件的文件块。最后，eMule扩展了eDonkey的能力，允许客户端之间交换关于服务器、其他客户端和文件的信息。注意，客户端和服务器的交流都是基于TCP的。

服务器使用了一个内部数据库，用来存储关于客户端和文件的信息。一个eMule服务器不存储任何文件，它为关于文件位置的存储信息作集聚索引。服务器的另一个功能，开始变得被抗议，是连接由于通过防火墙连接而无法接收到连接的两个客户端。这个连接功能增加了服务器的负载。相对于服务器和其他客户端，eMule使用UDP来增强客户端的能力。客户端发送和接收UDP信息的能力在日常使用中不是强制使用的，当有防火墙阻止它收发UDP信息时也能无瑕疵的运行。

### 1.2.1 客户端到服务器的连接

在开始启动时，客户端用TCP连接到一个eMule服务器。服务器提供一个客户ID给客户端，在整个客户端-服务器连接的生命周期里，它是有效的（注意，如果客户端有一个高ID，它会从所有的服务器中接收到相同的ID，直到它的IP地址改变）。在连接建立之后，客户端发送它的共享文件列表到服务器中。服务器把这个列表存储到它的内部数据库中，这个数据库

通常包含了成百上千有效的文件和活动的客户端。eMule客户端也发送它的下载列表，包含着它想下载的文件。第二章提供了eMule客户端和服务器的TCP信息交换的详细描述。

建立连接之后，eMule服务器给客户端发送有它想下载的文件的其他客户端列表（这些客户端称作“源”）。从这点起，eMule客户端开始与其他客户端建立连接，如1.2.2所述。

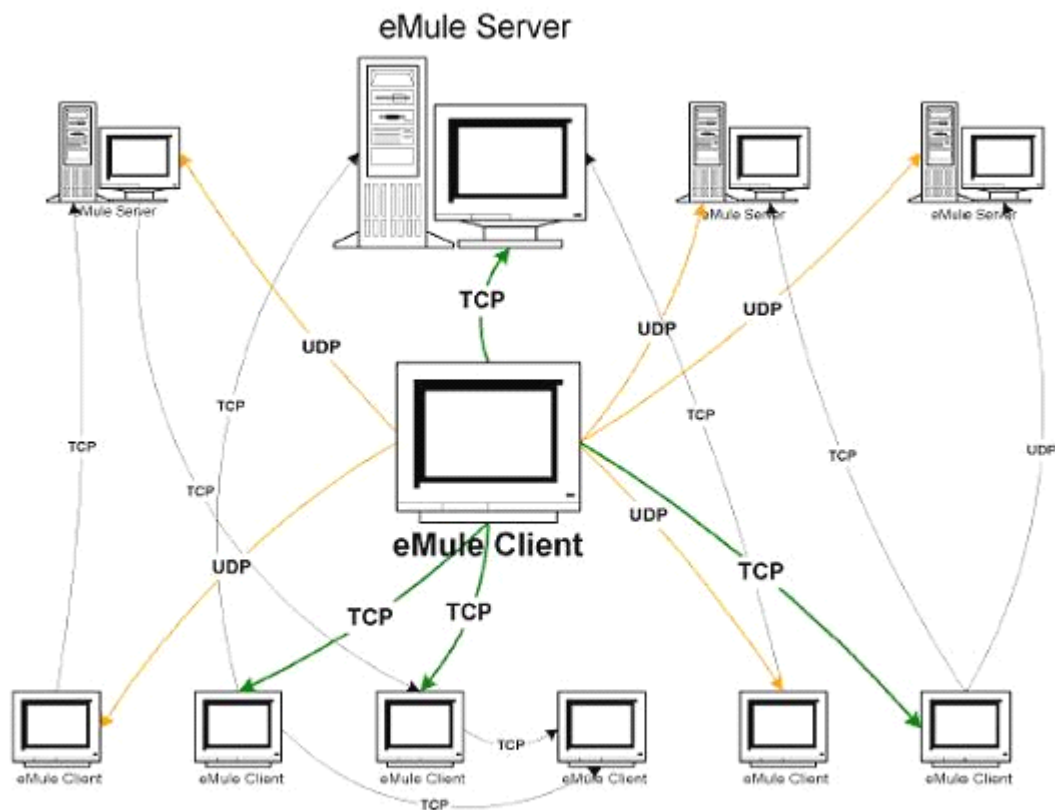


Figure 1.1: eMule high level network diagram

注意，在整个客户端会话期间，客户/服务TCP连接一直保持连接状态。初次握手后主要是用户活动激发事务：有时，客户端发送文件搜索需求，由搜索结果回应，一个搜索事务一般在对源中指定文件查询之后，用源（IP和端口）列表来回答这个查询，查询者可以从这列表中下载文件。

客户端和它没有连接的服务器的交流是用UDP。UDP信息的目的是增强文件搜索，增强源搜索，最后保持连接状态（确保客户端服务器列表中的eMule服务器有效）。在第三章中可找到更多的关于客户-服务UDP信息交换的细节。

### 1.2.2 客户端到客户端的连接

一个eMule客户端连接到另一个eMuel客户端（源）是为了下载文件。一个文件分成很多部分，进一步的碎片。客户端可以从几个（不同的）客户端中下载同一个文件来分别获得不同的文件碎片。

当两个客户端连接时，它们交换容量信息，然后协商一个下载（或者上传，根据看法）的开始。每个客户端有一个下载列表，记住一系列等待下载文件的客户端。当eMule客户端下载队列空的时候，一个下载请求很可能会导致一个下载开始（除非，比如这个请求者被禁止）。当下载队列不是空的时候，就会将这个请求的客户端加入到队列中。在给定的时间内，不能为几个以上客户端各自提供最小带宽2.4k/s。一个下载的客户端可能被一个比它较高队列等级的等待的客户端抢占，在下载会话的最初15分钟内，正在下载的eMule客户端的队列等级会增加直到能防止被击溃。

当下载的客户端到达下载队列的头部时，上传的客户端初始化一个连接来给它发送需要的文件块。eMule客户端可以在几个其他客户端的等待队列中，都注册下载相同文件的块。当一个等待的客户端实际上完成了（从它们中的一个）下载文件块，它不会通知其他客户端在其队列中删除它，当它到达它们的队列头时只是简单的拒绝它们的上传意图。

EMuley用一个信用系统来鼓励上传，为了防止假冒用RSA公匙密码系统来保护信用系统。

客户端连接可能用一套eDonkey协议没有定义的信息，这信息称作扩展协议。扩展协议用来实施信用系统，一般信息的交换（像服务器和源列表的更新），通过收发压缩的文件块来改善性能。

当EMule客户端在等待开始下载文件时，有限地用UDP周期性检查在它对等的客户端上的上传队列客户端状态。

## 1.3 客户 ID

客户ID是服务器在它们连接握手时提供的一个4字节标识符。客户ID只在客户-服务器TCP连接的生命期中有效，尽管万一客户端有一个高ID，所有的服务器都会分配它同样的ID直到IP地址改变。客户端ID分为低ID和高ID。当一个客户端不能接收一个输入连接时，eMule服务器将特有地分配给客户端一个低ID。拥有一个低ID会限制客户端对eMule网络的使用，和可能导致服务器拒绝一个客户端连接。高ID的计算是以客户端IP地址为基础的，如下所述。本节从eMule协议观点描述了客户ID的分配和重要性。允许其它客户端自由地连接到其本机上的eMule的TCP端口（默认端口号是4662）的客户端会分配给一个高ID。有高ID的客户端没有限制使用eMule网络。当服务器无法打开一个TCP连接到客户端的eMule端口时，会分配一个低ID给该客户端。这主要发生在机器上装有防火墙的客户端，阻止了输入连接。当出现下面情况时，客户端也会接收到一个低ID：

1 当客户端通过NAT或代理服务器连接

1 当服务器繁忙（导致服务器重连接计时器超时）

高ID用下面的方法计算：假设主机IP是X.Y.Z.W，ID就是 $X+2^8*Y+2^{16}*Z+2^{24}*W$ 。低ID总是小于16777216（0x1000000），关于它是怎样计算的，我找不到任何线索，在不同的服务器中得到不同的低ID。

低ID客户端没有其他客户端可以连接到的公网IP，这样所有的交流必须通过eMule服务器完成。这增加了服务器计算能力的负担，并且导致服务器勉强接收低ID客户端。这也意味着低ID客户端不能连接到不在同一个服务器上的其他低ID客户端，因为eMule不支持在服务器

间管道连接。

为了支持低ID客户端，引入了回调机制。使用这机制，高ID客户端请求（通过eMule服务器）低ID客户端连接它来交换文件。

## 1.4 用户 ID

eMule支持信用系统来鼓励用户共享文件。用户上传越多的文件给其他客户端，它接收的信用越多，它在它们的等待队列中前进得越快。

用户ID是128位（16字节）、连接随机数字创建的GUID，第6和第15字节不是随机产生的，它们的值分别是14和111。在整个客户端和指定的服务器会话中，客户ID是有效的，然而用户ID（也叫**用户哈希**）是唯一的并且跨越会话时用来识别客户端（用户ID识别工作站）。用户ID在信用系统中扮演重要角色，这为“黑客”假冒其他用户来获得他们信用赋予的优先权提供了动机。Emule提供加密方案设计来阻止欺骗和冒名顶替。这个实施是简单的应答交换，依靠RSA公有/私有钥匙加密。

## 1.5 文件 ID

文件ID用来惟一的标识网络中的文件和文件损坏侦测和修复。注意，eMule不依靠文件名来惟一标识和编目文件，通过哈希文件内容计算出的GUID标识文件。有两种类型文件ID-一种主要用来产生惟一的文件ID，另一种是用来损坏侦测和修复。

### 1.5.1 文件哈希

文件是用由客户端和基于文件内容计算出来的128位GUID哈希来标识的。GUID是应用MD4算法到文件数据中计算而来。当计算文件ID时，文件被分成每段9.28MB长的部分。每部分单独计算出一个GUID，然后所有的哈希组合成一个惟一的文件ID。当下载的客户端完成一个文件部分下载时，它计算这部分哈希，然后和发送过来的这部分哈希对比，如果这部分发现损坏了，客户端尝试通过逐渐替换这部分中的位（每个180kb）来修复损坏部分，直到哈希计算OK。

### 1.5.2 根哈希

用SHA1算法来为每部分计算根哈希，基于每块180kb大小。它提供了更高等级的可靠性和可修复性，更多信息可在eMule官方网站得到。

## 1.6 eMule 协议扩展

尽管eMule完全兼容eDonkey，它还是实行了几种扩展，允许eMule两个客户端为用户提供另外的功能。扩展只要集中在客户端与客户端的交流，特别是在安全和UDP使用领域上。在本文档中，所有信息流图标明的信息，是eMule扩展部分的，用灰色表示。

## 1.7 软件和硬件限制

在活动用户数量的服务器配置中有两种限制-软件和硬件。硬件限制远大于软件限制。当活动用户的数量达到软件限制时，服务器停止接收新的低ID客户连接。当用户数量达到硬件限制时，服务器满了，不再接收任何客户端连接。

# 2 客户端服务器的 TCP 交流

每个客户端用TCP精确地连接到一个服务器。服务器分配给客户端一个ID,在与服务器其余的会话中标识该客户端（高ID客户端总是根据它的IP地址分配）。eMule GUI客户端需要建立一个服务器连接来用于操作。客户端不能同时与几个服务器连接，也不能在没有用户干涉的情况下动态更换服务器。

## 2.1 建立连接

在准备建立与服务器的连接时，客户端会尝试并行地连接到几个服务器，根据成功的登陆顺序放弃其他的。

有下面几个可能的连接建立个案：

- 1、高ID连接-服务器分配一个高ID给正在连接的客户端
- 2、低ID连接-服务器分配一个低ID给正在连接的客户端
- 3、拒绝会话-服务器拒绝客户端



当然，也有不重要的个案-服务器崩溃或者不可连接。

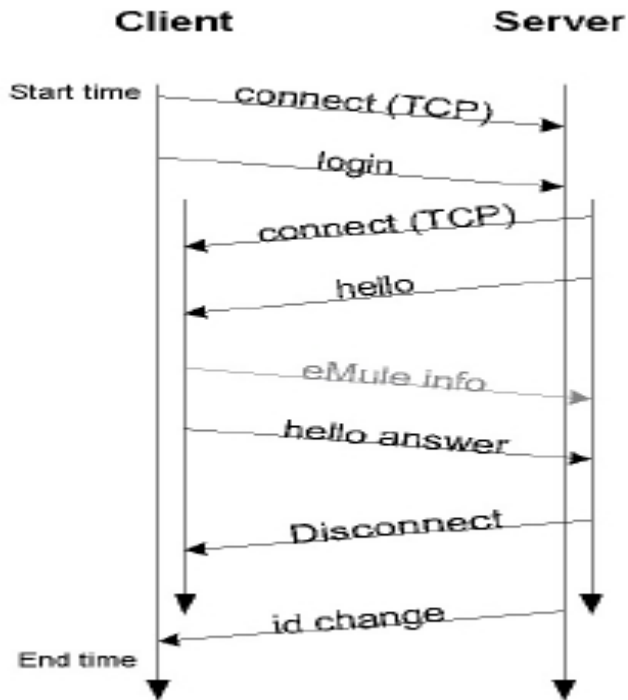


图2.1高ID连接的信息顺序

图2.1描述了导致高ID连接的信息顺序。在这种情况下，客户端建立一个TCP连接到服务器，然后发送一个登录信息到服务器。服务器用另一个TCP连接到客户端，执行一个客户端-客户端的握手来保证连接的客户端有能力接收来自其他eMule客户端的连接。在完成客户端握手后，服务器关闭第二个连接，通过发送ID更改信息来完成客户端-服务器的握手。你可能注意到eMule信息消息是灰色的。这是因为这个消息是eMule协议扩展的一个部分（1.6节）

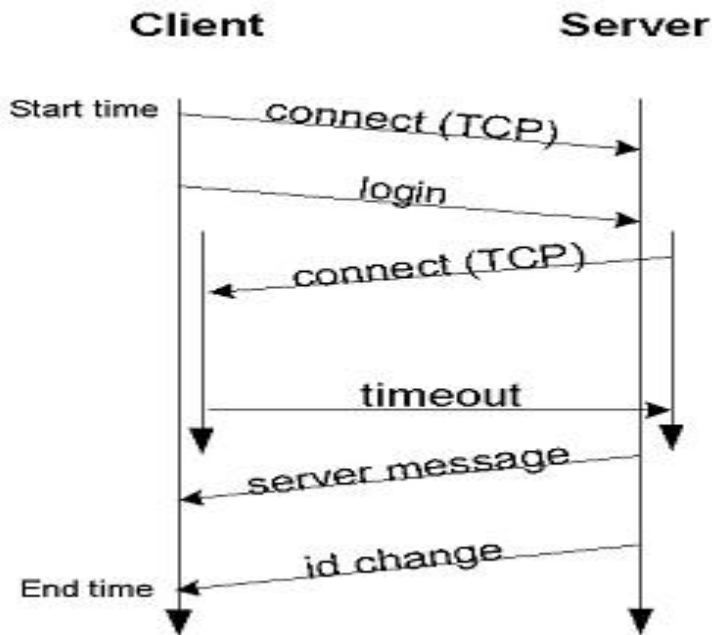


图2.2描述了导致低ID连接的信息顺序

图2.2描述了导致低ID连接的信息顺序。在这种情况下，服务器不能连接到发送请求的客户端，分配一个低ID给客户端。服务器消息一般包含警告信息，就像“警告[服务器细节] - 你是低ID。请察看你的网络配置和/或你的设置”低ID和高ID握手都是通过随着ID更改消息完成的，这个ID更改消息分配客户端一个客户端ID，用在与服务器的下一个会话。

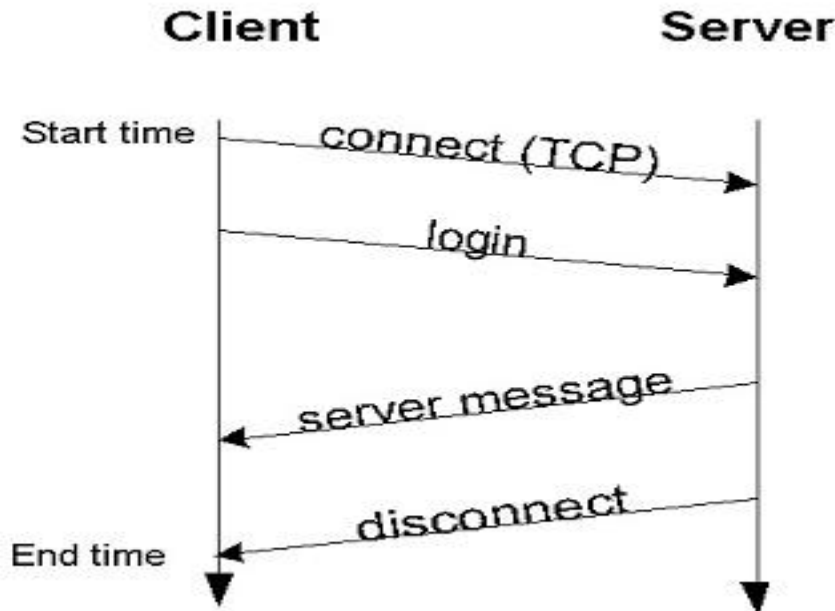
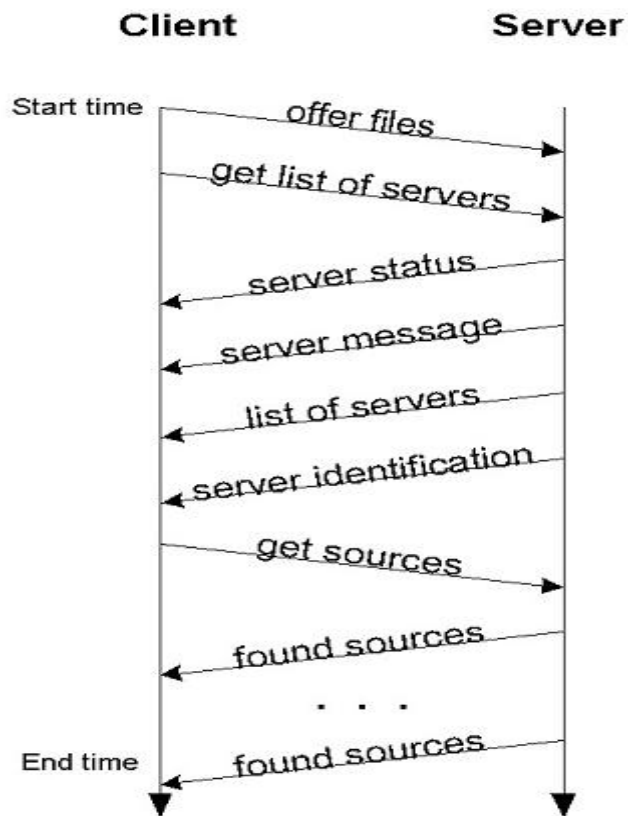


图2.3描述了被拒绝的会话顺序。因为客户端拥有一个低ID或者到达了服务器硬件的容量限制，服务器就可能拒绝会话。服务器消息会包含一个短字符串描述拒绝的理由。

## 2.2 连接启动时消息交换

在建立成功的连接后，客户端和服务端交换几个设置消息。这些消息的目的是根据双方状态来双方更新。客户端通过提供它的共享文件列表（见6.2.4节）给服务器来开始，然后要求更新它的服务器列表。服务器发送它的状态和版本（6.2.6节和6.2.2节），然后发送它所知的eMule服务器列表和提供更多一些自我认定的细节。最后客户端要求源（可以访问下载它下载列表中的文件的其它客户端）和服务端回应一系列的消息，客户端下载列表中的每个文件，直到下载所有的源列表到客户端。图2.4图解了这个顺序。



## 2.3 文件搜索

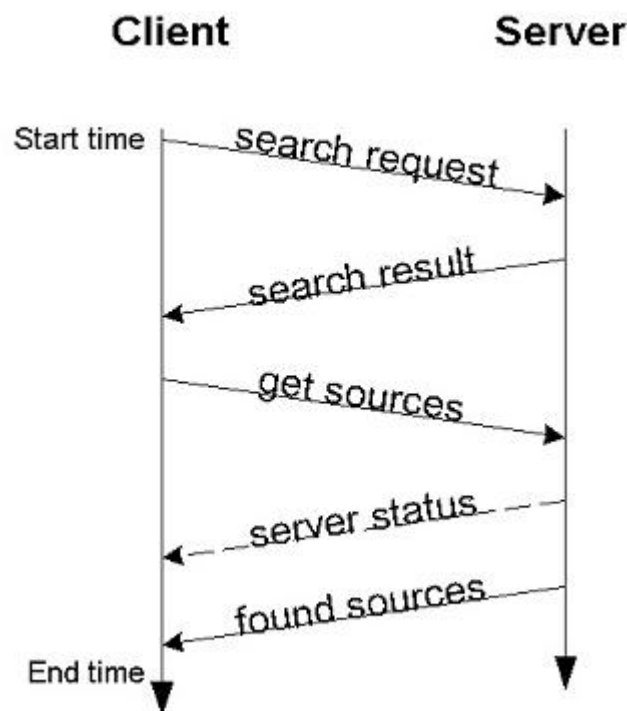


图2.5描述了文件搜索顺序

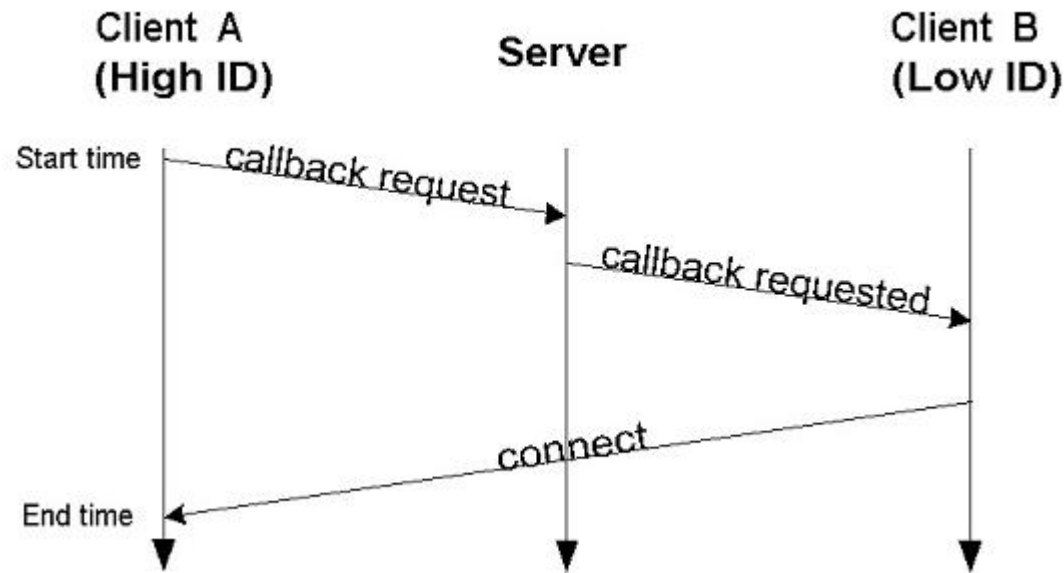
文件搜索是由用户发起的。这个操作简单，一个搜索要求（见6.2.9节）发送到服务器，然后服务器用一个搜索结果回应。当有很多结果时，搜索结果消息就会被压缩。接着，用户选择下载一个或多个文件，客户端就要求源为选中的文件和服务器返回每个要求文件的源队列（见6.2.12节）。就在回应发现的源之前，可以发送一个可选的服务器状态消息。这个状态消息（6.2.6节）包含关于当前用户数量和服务器支持的文件等信息。重要的是，UDP消息有个补充顺序事件，用来增强客户端为它搜索的文件定位源的能力，详细的细节见第3章。在检验出源是新的之后，eMule客户端开始尝试连接和把它们加入到它的源列表。源联系的顺序就是eMule客户端接收到它们的顺序。图2.5描述了文件搜索顺序。

eMule客户端根据源加入到它的列表中的顺序来连接源。没有优先机制来决定连接那个源。当可以要求同一个源来下载客户端下载列表中的几个文件时，有一种复杂的机制来解决这个局面（注意，载客户端之间eMule只允许一个单独的上传连接）。选择算法是基于用户优先规则，当没有指定优先时，默认是字母顺序。关于处理可以上传多于一个文件的源的详细描述，可以在网站中找到。

## 2.4 回调机制

回调机制是设计来克服低ID客户端不能接收输入的连接的，这样客户端之间就能共享它们

的文件。机制很简单：假如客户端A和B都连接到同一个eMule服务器，A需要的文件在B上，但B是低ID的，A可以向服务器发送一个回调请求（见6.2.13节），请求服务器叫B呼叫回它。服务器，已经有一个与B的打开的TCP连接，发送一个回调请求消息（见6.2.14节）到B，为它提供A的IP和端口。B就能连接到A并发送文件，没有给服务器增加负担。很明显，只有高ID客户端可以要求低ID客户端回调（低ID客户端是没有接收输入连接的能力的）。图2.6图解了回调消息交换。



也有允许两个低ID客户端交换文件的能力，通过它们的服务器连接，用服务器接力。大部分服务器不再提供这个选项，因为它招致服务器的负担。

### 3 客户端服务器的 UDP 交流

eMule客户端和服务端用不可靠的UDP服务来保持连接和增强搜索。eMule客户端产生UDP包的总量可以达到它发送包的总数目的5% - 这些根据客户端服务器列表中服务器的数目，客户端下载列表中每个文件的源数目和用户执行的搜索数目而定。UDP包通过计时器触发，计时器每100ms过期，有一个单独的线程负责发送UDP输送结果，以每秒10个UDP的最大速率。

#### 3.1 服务器保持连接和状态信息

客户端周期性验证它服务器列表中的服务器状态。验证是通过发送UDP服务器状态请求（见6.3.3节）和UDP服务器描述请求（见6.3.7节）消息完成的。这里描述的简单保持连接计划每小时产生不超过几打包。任何情况下，包的最大速率是每秒0.2个包（或每5秒一个包）。当检查服务器的状态时，客户端会首先发送一个服务器状态请求消息，接着，每两次试图（发

送一个服务器状态请求)中就发送一次服务器描述请求,见图3.1。

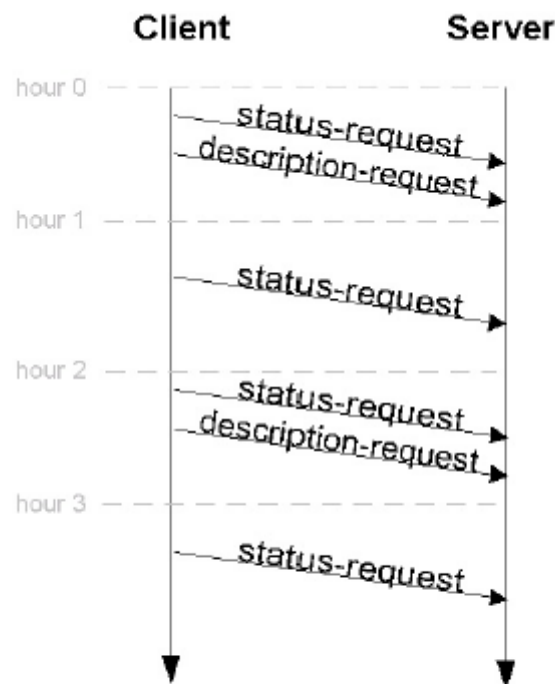


Figure 3.1: UDP Keep alive cycle

客户端发送的服务器状态请求中包括一个随机数字,在服务器回应中返回。在服务器返回的数字与客户端发送的要求中数字不同的情况下,回应的信息就会被丢弃。每次发送到服务器的包是状态请求,客户端就移动尝试计数器。任何来自服务器的消息(包括搜索结果)都重置尝试计数器。当尝试计数器达到一个可配置的限制时,服务器就认为是死机,从客户端的服务器列表中删除。服务器回应包括几个数据项:服务器状态回应(见6.3.4)包括服务器中当前用户数目和文件数目,也包括服务器的软件和硬件限制(见1.7节)。服务器描述回应(见6.3.8节)包括服务器名称和一个短的描述字符串。图3.2演示了客户端和活动服务器中满连接序列的消息流。

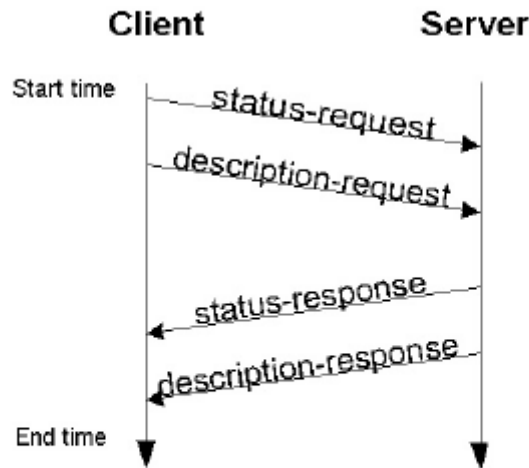


Figure 3.2: UDP Keep alive sequence

## 3.2 增强文件搜索

eMule客户端可以设置用UDP来增强它的文件搜索。UDP搜索结果格式几乎与??节描述的TCP搜索请求一样。服务器只在有了搜索结果才回应。UDP搜索结果消息有两种不同的opcodes, 我也无法说清它们之间的不同。UDP搜索包发送到客户端服务器列表中的服务器上。更多信息见6.3.5节和6.3.6节。

## 3.3 增强文件源搜索

当客户端下载列表中的特定文件的源数目小于配置限制(100)时, 客户端就周期性地发送获取源的UDP包到它的服务器列表中的服务器上为该文件寻找更多的源。可能每秒发送一个包, 使得源搜索在客户端产生的UDP输送中成为可观的部分。消息的格式(6.3.1节描述)非常相似它的TCP计数器部分。注意, 与TCP源搜索相反, UDP源搜索在文件搜索中减弱, 对于指定的文件, 只是依靠客户端拥有的源数目。

## 4 客户端到客户端的 TCP 交流

在eMule客户端注册到服务器和向服务器查询文件和源之后，为了下载文件，eMule客户端需要联系其它客户端。为每对[文件，客户端]创建一个专用的TCP连接。当特定的周期内（默认40秒）没有任何socket活动或者对方已经关闭了这个连接，那么这个连接就会关闭。

为了提供合理的下载速率，直到可能提供给它（和所有其它下载的客户端）至少最小允许速率（当前的硬编码常量设置为2.4k/s），eMule才允许客户端开始下载文件。

### 4.1 初始的握手

初始的握手是对称的 - 双方都相互发送相同的信息给对方。客户端交换双方的信息，信息包括身份认证、版本和容量等。参与的有两种类型消息 - Hello消息（6.4.1节）和eMule信息消息（6.5.1节），第一种是eDonkey的一部分，兼容eDonkey客户端，第二种是eMule独有的扩展客户端协议的一部分。图4.1图解了两个eMule客户端之间的握手。在扩展信息中包含的有UDP消息交换、安全身份证明和源交换能力。

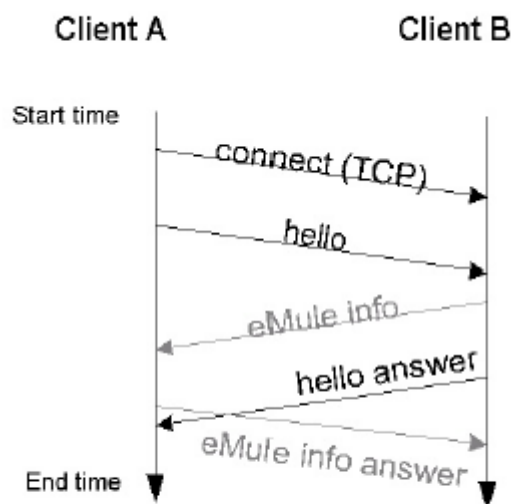


Figure 4.1: eMule client initial handshake



## 4.2 安全的用户身份认证

1.4节简单解释了关于用户ID和用户假冒其它用户的动机。安全用户认证是eMule扩展的一部分。如果客户端支持安全认证，就会在初始化握手之后立即执行。安全认证的目的是防止用户冒名顶替。当实施安全认证时，执行以下步骤：

- 1.在初始化握手中，B客户端指明它支持和希望使用安全认证。
- 2.通过发送安全认证消息（见6.5.8）来回应，指明A是否需要B的公匙，也包含了B发出的4字节的询问。
- 3.如果A指明它需要B的公匙，B就将它的公匙发送给A（6.5.9节）。
- 4.B发送一个签名消息（6.5.10节），签名消息是用发送过来的询问和额外的双字节创建的，双字节要么是A的IP地址如果B是低ID，要么是B的ID如果它有高ID。图4.2演示了这个顺序。

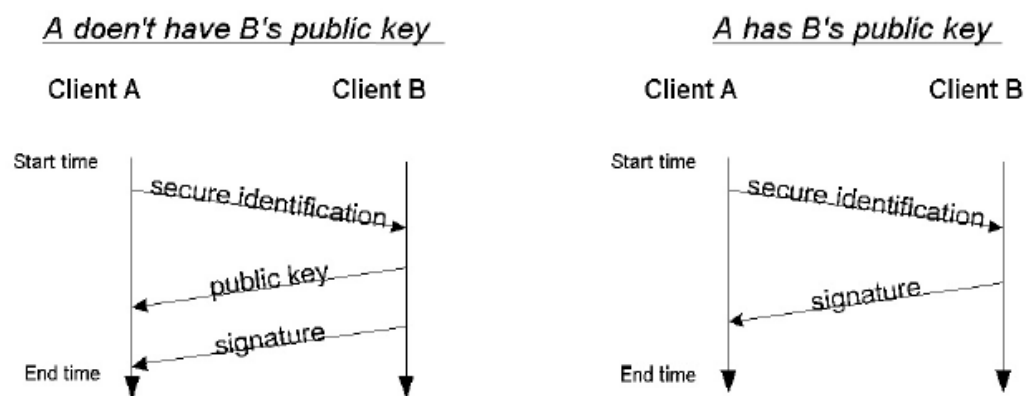


Figure 4.2: Secure identification flow

### 4.2.1 信用系统

本节简要地描述了客户端的信用系统。信用系统的目的是鼓励用户共享文件。当客户端上传文件给它的对方，下载的客户端就根据数据传输的数量来更新它的信用。注意，信用系统不是全局的 - 传输的信用被下载的客户端局部保存，只有当上传的客户端（获得信用的那个）要求从这个特定的客户端下载时，信用才会被考虑。信用是用下面最小值计算的：

1. 上传的总量 \* 2 / 下载的总量

当下载的总量是零时，这个表达式估值是10

2. 上传的总量 + 2 的和开方

当上传的总量小于1MB，这个表达式估值是1

上传/下载数量是以M为单位计算。任何情况下，信用不会高过10或者低于1。

### 4.3 请求文件

正如已经提到的一样，每对[客户端，文件]都创建一个独立的连接。在连接建立之后，客户端立即发送几个关于它希望下载的文件请求消息。4.3节描述了一个典型的、成功的情景。



Figure 4.3: File request

#### 4.3.1 基本消息交换

基本消息交换是由四个消息构成：A发送一个文件请求消息，立即跟着的是请求文件ID消息（6.4.17节）。B用文件请求回答回应文件请求，用文件状态（6.4.18节）来回应请求文件的ID消息。我找不到任何理由来把这些发送过来的消息中的信息分成四个消息，它可以容易地用两个消息（请求和回应）来处理。

扩展协议增加两个消息到这个顺序中，源请求消息（6.5.6节）和源回应消息（6.5.7节）。用这个扩展来传递B的源（假定B当前下载着文件）到A中。详细阐述就是，在它发送文件块给其它客户端之前B完成下载一个文件是没有任何要求的，B可以发送任何它已经完成下载的文件块，甚至当它只是用文件的一小块。

### 4.3.2 没找到文件的情景

当A向B请求一个文件，但是B的共享文件列表中没有这个文件。B忽略这个文件请求回应消息，在请求文件ID消息之后立即发送一个没有文件消息（6.4.16节），如图4.4所演示。

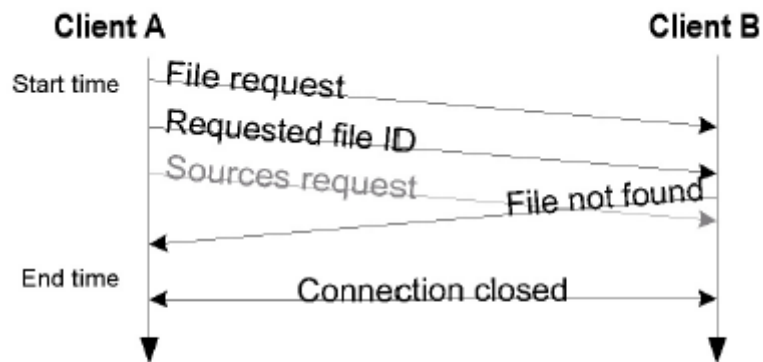


Figure 4.4: File request failure - file not found

### 4.3.3 加入上传队列

当B有被请求的文件但是它上传队列不为空，意味着有正在下载文件的客户端，也可能有客户端在上传列表中，在这种情况下，A和B执行满握手，如图4.3所述，但是当A请求B开始下载文件时，B把A加入到它的上传队列中，回应一个队列等级消息，这个消息包含A在B地上传队列中的位置。图4.5演示了这个顺序。

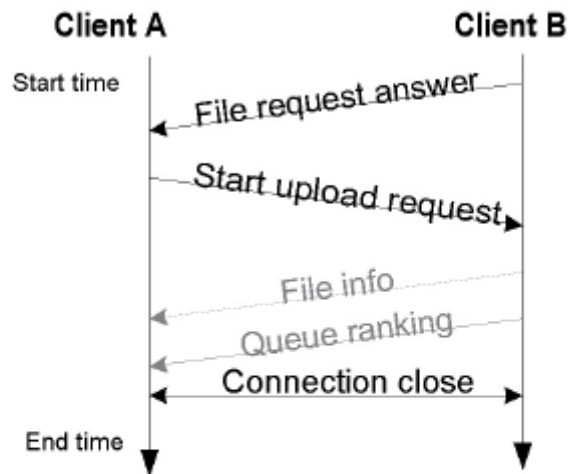


Figure 4.5: File request waiting queue

#### 4.3.4 上传队列管理

对每个上传的文件，客户端维护着一个上传优先级队列。队列中的每个客户端的优先级是以客户端在队列中的时间和优先级修正为基础计算的。位于队列头部的客户端有一个最高级别的分数。分数是用以下的方程式计算的：分数 = (等级 \* 队列中的秒数) / 100 或者无穷大，如果下载的客户端被定义为朋友。初始化的等级值是100，除开禁止的用户是0等级（这样阻止达到队列的前面）外。等级可以被下载客户端的信用（范围1-10）或上传文件优先级（0.2-1.8）修改，上传文件优先级是由上传客户端设置的。当一个客户端的分数比其它的客户端高时，它就开始下载文件。客户端可以继续下载文件直到产生以下任一个条件：

1. 用户关闭了上传客户端。
2. 下载的客户端得到了它所需文件的所有部分。
3. 下载的客户端给其它拥有比它更高优先级的客户端抢占。

为了允许一个刚刚开始客户端在它被抢占之前可以得到几M的数据，eMule在客户端下载的前15分钟内增加初始等级到200。

#### 4.3.5 到达上传队列的顶部

当A到达B上传队列的顶部时，B连接A，执行初始握手，然后发送一个接收上传请求消息（6.4.11节）。A现在可以选择要么发送请求块消息来继续下载文件，要么发送取消传输消息来取消（如果它已经从别的源得到了这一块）。图4.6演示了这些选择。

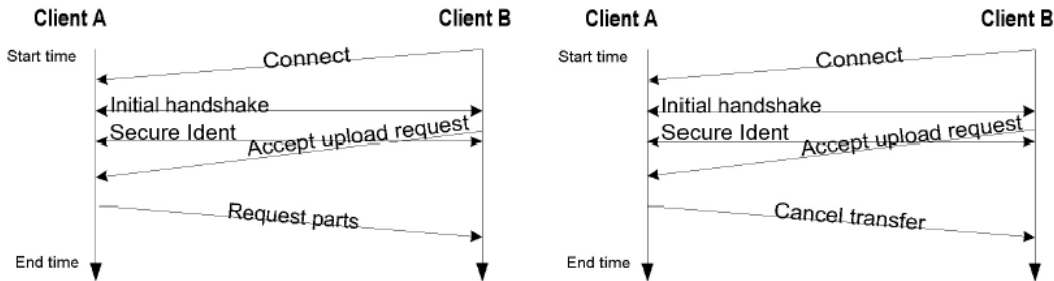


Figure 4.6: File request resume download

## 4.4 数据传输

### 4.4.1 数据包

发送和接收文件块是eMule网络活动的主要部分。用FTP解释eMule可以推论出，当所有其它eMule可以控制，发送的文件块适合数据传输。发送的文件块大小可以是在5000到15000位（也根据压缩）范围内。为了避免出错，文件块消息在碎片中发送，每碎片在一个独立的TCP包中。在eMule 0.30e版中，最大的碎片大小是1300位（注意，这个数字只与TCP有效负载有关）。换句话说，当每个控制消息在单独的TCP包中发送，有时和其它消息共享，数据消息被分成几个TCP包。第一个包包含发送文件块消息头部（6.4.3节）。剩下的包值包含数据。当被分成1300，如果发送块的大小有剩余，和第一个包（这个包带有头部）一起发送。图4.7演示了文件块消息。

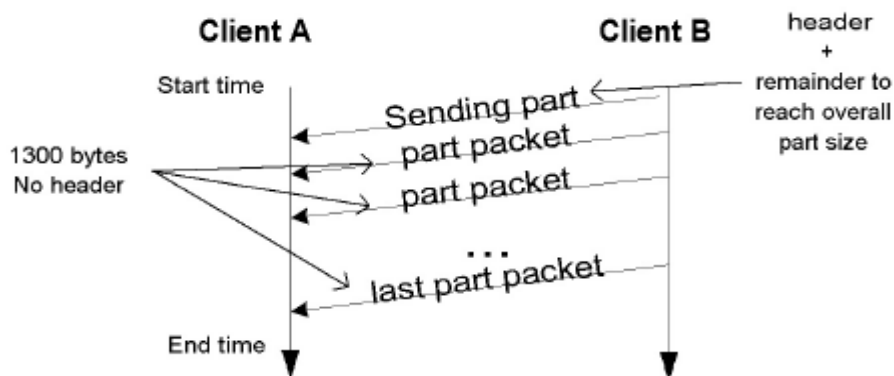


Figure 4.7: File part message details

#### 4.4.2 数据传输顺序

在文件请求回应之后立即开始块传输顺序。下载客户端A发送一个开始上传请求（6.4.10节），然后一个接收上传请求消息（6.4.11节）回应这个请求。A在这之后立即开始请求文件块（6.4.4节），B通过发送被请求块（6.4.3节）来回应。注意，单独的文件块请求可能请求可达3块之多，所以每个文件块请求可能被可达3个发送的块顺序回应。

当两个客户端都支持扩展的客户端协议，文件块可能压缩发送。扩展协议也支持可选的文件信息消息（6.5.5节），该消息就在接收上传请求消息之前发送。图4.8演示了块传输消息顺序。

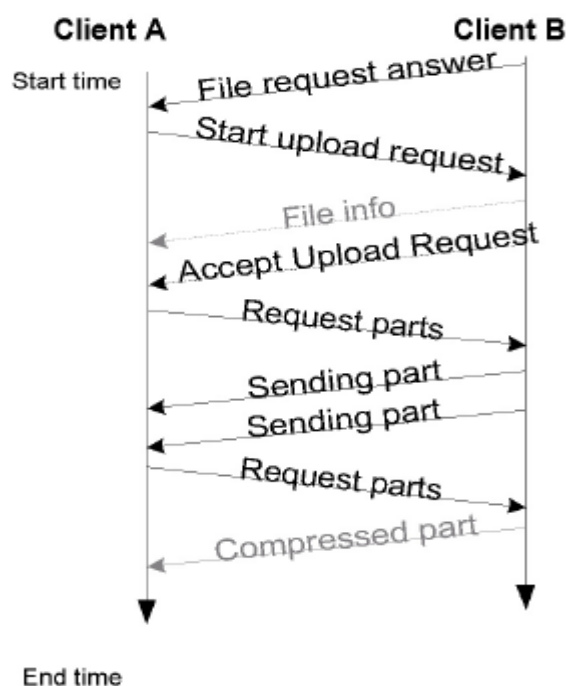


Figure 4.8: File part exchange

#### 4.4.3 选择块下载

为了最大化整个网络的吞吐量和共享，eMule仔细挑选选择块的下载顺序。每个文件被分成9.28M的块，每部分分成180KB的片。

块下载的顺序是由发送请求文件块消息（6.4.4节）的下载客户端决定。下载客户端可以在任何给定时刻从各个源中下载一个单独的文件块，所有从相同源中请求的片都在同一个块中。下面的原理（以这个顺序）应用于下载块等级：

- 1.（可获得的）大片的频率，尽可能快的下载非常稀少的大片来形成一个新的源。
- 2.用来预览的块（最初+最后的大片），预览或检查文件（比如，电影、mp3）
- 3.请求状态（过程中下载），尝试向每个源询问其它的大片。在所有源之间扩散请求。
- 4.完成（未到某种程度的完成），在开始下载另一个时应该完成获得部分的大片

频率标准定义了三个区域：非常稀少、稀少和一般。在每个区域里，标准有特定的权重，用来计算块等级。较低等级的块先下载。下面的列表根据上面的原理指定文件等级范围：

- 1 0-9999 - 不请求和请求非常稀少的块

- 1 10000-19999 - 不请求稀少和预览块
- 1 20000-29999 - 不请求大部分完成的一般的块
- 1 30000-39999 - 请求的稀少和预览的块
- 1 40000-49999 - 请求的没有完成的一般的块

这个算法通常选择第一个最稀少的块。然而，部分完成的块，接近完成的，也可能被选中。对于一般的块，在不同的源之间扩散下载。

## 4.5 浏览共享的文件和文件夹

有两个消息流程来处理每对客户端之间的共享文件和文件夹的浏览。第一个是浏览共享文件消息（6.4.21节），该消息在初始握手后立即发送。通常由一个浏览共享文件回应消息（6.4.22）来回应这个消息。当回应的客户端想隐藏它的共享文件列表时，回应就包含零个文件（而不是发送一个指示访问拒绝的消息）。图4.9演示了这个消息顺序。

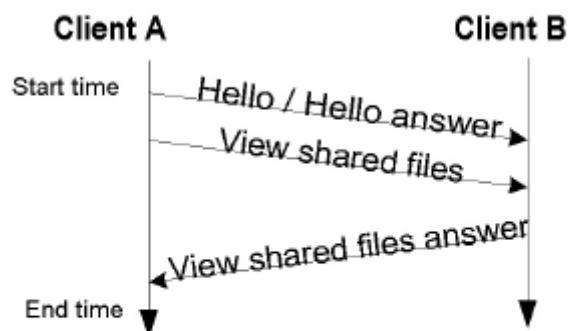


Figure 4.9: View shared files

第二个消息流程随着一个浏览共享的文件夹列（6.4.23节）表请求开始，通过共享文件夹列表来回应这个消息，然后，对于回应中的每个文件夹，发送浏览共享文件夹内容的消息。当消息到达时，回应的每个消息带有内容列表。图4.10演示了这个消息顺序。



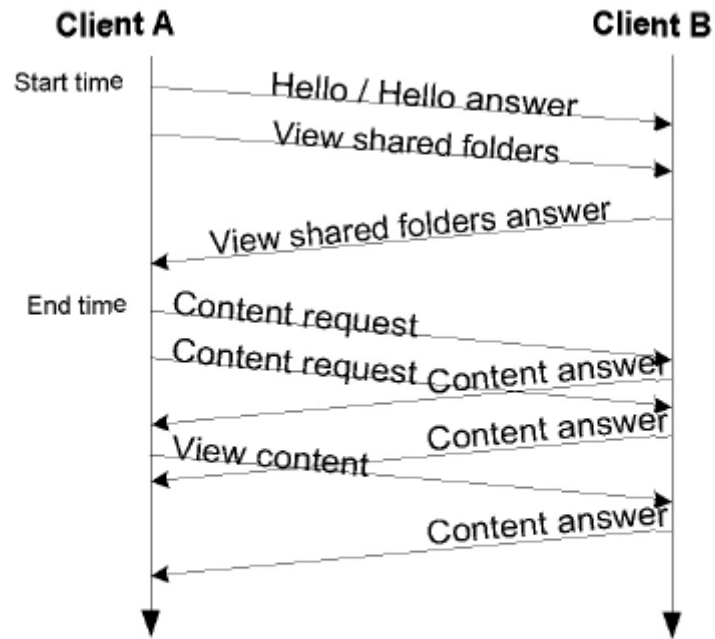


Figure 4.10: View shared file and folders

如果接收的客户端设置了阻塞共享文件/文件夹请求，它用请求共享拒绝消息回应，如图4.11所示。

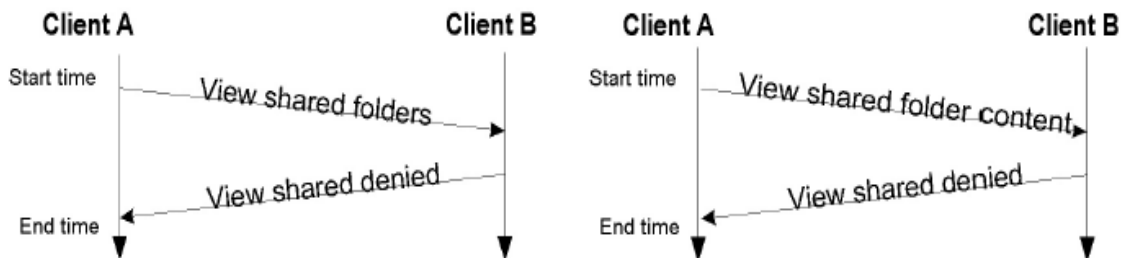


Figure 4.11: View shared denied

## 4.6 交换片哈希集

为了取得片的哈希，发送一个哈希集请求，这个请求通过一个包含文件中每块的哈希集回应来回答。图4.12演示了这个。

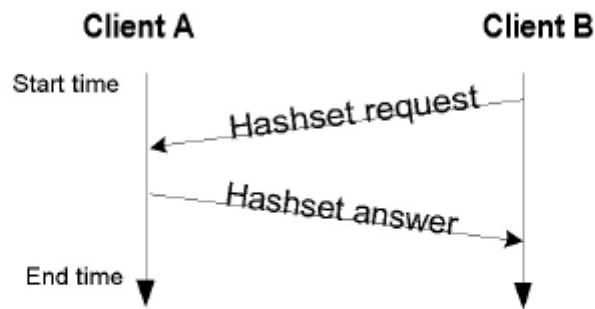


Figure 4.12: Hashset request

## 4.7 取得文件预览

客户端可以请求对方来获得下载文件的预览。预览是种独立的、随着文件类型不同而不同的应用。eMule 0.30e只支持图像预览。这个消息交换如图4.13描述，只包含两种消息：预览请求（6.5.11节）和预览回应（6.5.12节）。

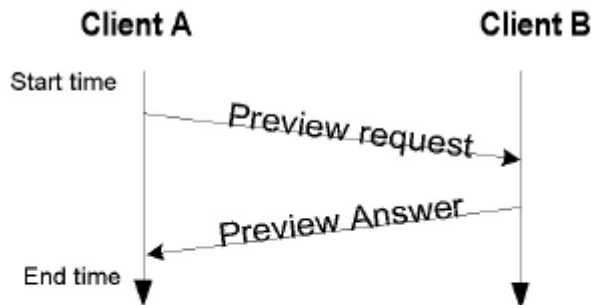


Figure 4.13: Get file preview

## 5 客户端到客户端的 UDP 连接

eMule客户端周期性地用UDP协议发送消息。在eMule0.30e中，使用的UDP消息只是询问客户端在对方下载队列中的位置。这个简单的请求-应答方案是随着重复要求文件消息(6.6.1节)而开始的。对于这个请求，有三种可能的回应，如图5.1所示。

1. 队列等级 - 客户端在发送者队列中的等级
2. 队列满 - 发送者队列已满
3. 找不到文件 - 发送者在它的列表中没有被请求的文件

重复要求文件消息大约每20分钟的间歇发送到每个客户端，这些客户端把发送者加入到它的下载队列中。

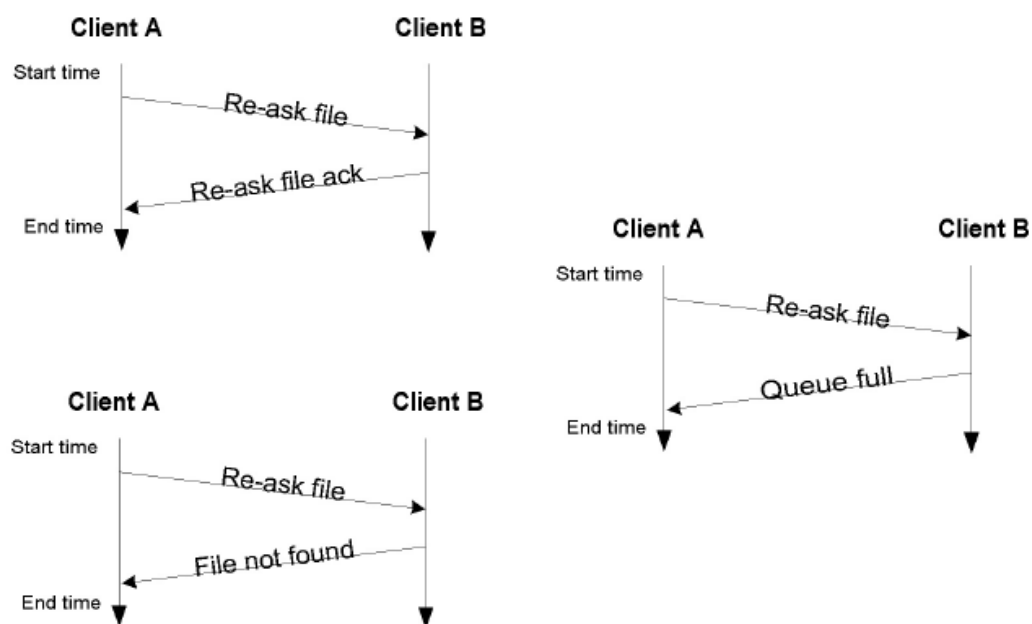


Figure 5.1: Re-ask file message

## Bibliography

- [1] D. Bickson and D. Malkhi. A study of privacy in file sharing networks. In *Technical Report TR-2003-67 Leibniz Research Center, the Hebrew University of Jerusalem, Israel*, 2003.
- [2] eMule at source force. <http://sourceforge.net/projects/emule/>.
- [3] eMule project. <http://www.emule-project.net/>.
- [4] MD4 hash function specification. <http://www.ietf.org/rfc/rfc1320.txt>.

## 6 附录详细的消息编码格式

### 6.1 一般消息编码要点

本章描述了在TCP/UDP有效负载中消息编码的一般方法。

#### 6.1.1 字节序

所有消息都是用 little-endian 编码，而不是 big-endian, big-endian 是约定的网络字节序。这个很容易解释，事实上客户端/服务器都是基于微软窗口的应用程式，运行在Intel处理器上。

#### 6.1.2 消息头

所有的消息有一个6字节的头，头有着下面的结构：

1. 协议 - 一个字节，协议ID - 0xE3是eDonkey, 0xC5是eMule
2. 大小 - 4字节消息大小 - 消息大小，以字节为单位，不包含头，例如，如果消息不包含任何有效负载，如6.4.11节中，则消息长度是零。
3. 类型 - 一个字节，类型 - 独一无二的消息ID

#### 6.1.3 消息标签

标签类似TLV（类型，长度，值）结构，用来增加可选的数据到eMule消息中。有几种类型的标签，本章中列出了所有的标签。当提及到协议消息里特定的标签时，只有标签类型是指定的，读者应该把本章作为一个参考来确定协议消息的准确的结构。每个标签拥有4个域，在消息中它们并不都是连续的：

1. 类型 - 1字节整型
2. 名称 - 可以是以下之一
  - 1 可变长度字符串
  - 1 1字节整型
3. 值 - 可以是以下之一
  - 1 4字节整型
  - 1 4字节浮点数
  - 1 可变长度字符串
4. 专用的 - 1字节整型，专用的标签指定者

带有整型值的标签叫做整型标签，类似的我们有字符串标签和浮点数标签。字符串标签的类型值是2，整型标签的类型值是3和浮点数标签的类型值是4。当标签被编码来发送时，是按照上面的次序编码，例如，类型，然后名称，最后值。类型被编码成一个字节。名称被编码成2个字节长度值，可以是字符串名称和整型名称。例如，整型名称0x15编码顺序是0x01 0x00 0x15。

固定的值域（好像整型和浮点数字）正如它们这样写入，字符串值就以相同的长度值方式编码。

注意：标签给出的名称是没有特定的协议意义的，只是易于在以后的协议消息描述中引用。

## 6.2 客户端服务器 TCP 消息

本章描述了在服务器和客户端之间用TCP传送的消息。

### 6.2.1 登录

在TCP连接建立之后，客户端发送到服务器的第一个消息是登录消息。这个消息的长度可变的，因为它是依据用户的配置的，例如用户昵称。为什么客户端的TCP端口出现在TCP的头部，还是再发送（2次），这不是很清楚。

名称	字节大小	默认值	注释
Protocol（协议）	1	0xE3	
Size （大小）	4		消息大小是以字节为单位，不包括头部和大小的域
Type（类型）	1	0x01	OP_LOGINREQUEST操作码的值
User Hash （用户哈希）	16		关于用户哈希的细节可以在1.4节找到
Client ID （用户ID）	4	0	在第一次连接中发送的用户ID通常是零。关于用户ID的细节可以在1.3节找到

TCP Port	2	4662	客户端使用的TCP端口，可配置的
Tag Count	4	4	消息中跟随的标签数目
Name Tag	可变的	NA	用户的昵称（在软件中可配置）。这个标签是字符串标签，标签名称是值为0x1的整数
Version Tag	8	0x3C	客户端支持的eDonkey版本。这个标签是整型标签，标签名称是值为0x11的整数
Port Tag	8	4662	客户端使用的TCP端口。这个标签是整型标签，标签名称是值为0x0F的整数
Flags Tag	8	0x01	这个标签是整型标签，标签名称是值为0x20的整数

### 6.2.2 服务器消息

服务器消息是可变长度的消息，在不同的场合中，第一次客户端登录请求之后立即由服务器发送到客户端。一个服务器消息可以包含几个的消息，用换行操作符（'\r'，'\n'或两者）分隔。用"server version"，"warning"，"error"和"[emDynIP:"开头的消息对客户端有特定的意义。其它的消息简单地显示给用户。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x38	OP_SERVERMESSAGE操作码的值
Size	2	NA	消息剩余字节的数目，不包括目前描述的域
Message	可变的	NA	一系列服务器消息，用换行分隔

特别的消息

- 1、 版本 - 通常在成功的连接我手中发送
- 2、 错误 -
- 3、 警告 - 通常在服务器拒绝连接或者客户端是低ID时发送
- 4、 emDynIP -

### 6.2.3 ID 改变

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。消息大小为14或10字节，依赖于发送的可选TCP连接位图（TCP connection bitmap）。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x40	OP_IDCHANGE操作码的值
Client ID	4	NA	关于客户端ID的描述可在1.3节找到
TCP connection bitmap	4	0x000000001	当前只有1位（LSB）有意义，设置它为1来标识服务器支持压缩

### 6.2.4 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。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x15	OP_OFFERFILES操作码的值
File Count	4	NA	里面描述的文件数目。无论如何不超过200。服务器也可以为这个数目设置一个相对较低的限制
Files	可变的	NA	可选的文件列表，单个条目格式如下所述

### 单个文件条目格式

下面的表格描述了单个文件条目。在一个文件提供消息中可以有一系列的多个条目。

名称	字节大小	默认值	注释
Hash Code	16	NA	执行在文件内容上的哈希结果（TBD规范）。这个哈希用来唯一的标识文件，忽略不同客户端之间的名称差别
Client ID	4	NA	客户端ID，如果客户端有高ID，否则为零
Client Port	2	0x15	客户端TCP端口号，如果客户端是低ID则为零
Tag Count	4	NA	该域以下的标签的数目
File Name Tag	可变的	NA	（必选的）文件名称。这个标签是字符串标签，标签名称是值为0x1的整数
File Size Tag	8	NA	（必选的）文件大小，字节单位。这个标签是整型标签，标签名称是值为0x2的整数
File Type Tag	可变的	NA	（可选的）文件类型。以下之一：“Audio”、“Video”、“Image”、“Pro”或“Doc”。



g	这个标签是字符串标签，标签名称是值为0x3的整数			
File Format Tag	可变的	NA	(可选的) 文件扩展名，小写。例如，“zip”、“exe”。这个标签是字符串标签，标签名称是值为0x4的整数	
Media Length Tag	可变的	NA	(可选的) 如果文件是mp3，歌曲播放时间。该标签是字符串标签，标签名称是“length”的字符串	
Media Bitrate Tag	TBD	NA	(可选的) 如果文件是mp3，编码位率。该标签是整型标签，标签名称是“bitrate”字符串	
Media Codec Tag	可变的	NA	(可选的，从不发送) 如果文件是影片，编码解码器。该标签是字符串标签，标签名称是“codec”的字符串	

### 6.2.5 获得服务器列表

该消息在成功握手完成之后立即从客户端发送到服务器。当客户端被配置通过请求它当前服务器来扩大它的eMule服务器列表时，发送该消息。消息大小是6字节。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x14	OP_SERVERSTATUS操作码的值

### 6.2.6 服务器状态

从服务器发送到客户端。该消息包含关于服务器上用户和文件的当前数目的信息。消息中的信息不但存储在客户端也表现出给用户看。该消息大小14字节。

名称	字节大小	默认值	注释
Protocol	1	0xE3	

Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x34	
User Count	4	NA	当前登录到服务器的用户数目
File Count	4	NA	服务器知道的文件数目

### 6.2.7 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）。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x32	OP_SERVERLIST操作码的值
Entry Count	1	NA	该消息中描述的服务器数目
Server entries	(Entry Count) * 6	NA	服务器描述符条目，每个条目大小是6字节，包含4字节IP地址和2字节TCP端口

### 6.2.8 服务器身份证明

从服务器发送到客户端。包含服务器哈希（TBD），服务器IP地址和TCP端口（当通过代理连接时很有用处），还有服务器描述信息。消息大小可变的。

名称	字节大小	默认值	注释
----	------	-----	----

Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x41	OP_SERVERIDENT操作码的值
Hash	16	NA	服务器的GUID（好像用来调试用）
Server IP	4	NA	服务器IP地址
Server Port	4	NA	服务器监听的TCP端口
Tag Count	4	NA	在消息末处标签的数目
Server Name Tag	可变的	NA	服务器名称。该标签是字符串标签，标签名称是值为0x1的整数
Server Description Tag	可变的	NA	服务器描述字符串。该标签是字符串标签，标签名称是值为0xB的整数

### 6.2.9 搜索请求

从客户端发送到服务器。该消息用来通过用户的搜索字符串搜索文件。消息大小是可变的。搜索字符串可以包含布尔条件” AND”、” OR”、” NOT”。用户可以指定需要的文件的类型和大小，也可以设置一个有效的阈值（例如显示至少来自5个其它客户端的有效结果）。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x16	OP_SEARCHREQUEST操作码的值
Parsed search string	可变的	NA	解析搜索字符串格式，下面描述

File Type Constraint	可变的	NA	可选的。字符串条件。字符串值是“Audio”、“Video”、“Pro”或“Image”其中之一。类型域是3字节：0x1 0x0 0x3
Min Size Constraint	可变的	NA	可选的。整数条件。提供的文件大小单位是M。类型域是4字节：0x1 0x1 0x0 0x2
Max Size Constraint	可变的	NA	可选的。整数条件。提供的文件大小单位是M。类型域是4字节：0x2 0x1 0x0 0x2
Availability Constraint	可变的	NA	可选的。整数条件。设置一个较低的限制在提供搜索文件的客户端数目上。类型域是4字节：0x1 0x1 0x0 0x15
Filename Extension constrain	可变的	NA	可选的。字符串条件。类型域是3字节：0x1 0x0 0x3

### 解析的搜索字符串格式

解析的搜索字符串用布尔操作符‘AND’、‘OR’和‘NOT’与字符串操作数来编码一个表达式的二叉树。树是用预先顺序编码的。操作符编码成2字节整数，0x0对应‘AND’，

0x100对应‘OR’，0x200对应‘NOT’。字符串用TLV格式编码，这里，类型的域是一个字节值为0x1，长度的域是2字节整数。注意，当搜索字符串是一个单词，它就作为单个字符串操作数（没有操作符）编码。以后的eMule版本编码一个只有”AND”操作符的字符串搜索表达式，用空格符代替“AND”，这样适合服务器的搜索字符串解析，把一个语句分成一系列的用“AND”操作符分隔的单词。

### 可选的条件格式

条件是一序列的条目。每个条目以“AND”描述符（2字节0x00）开始，紧接着是已编码的条件。这样，整个搜索行格式是<“搜索字符串” AND 条件1 AND 条件2 等>，如下面的例子示意图所示。编码的条件被分成3个域：

1. 种类 - 单字节，描述是字符串条件（0x2）还是整数条件（0x3）
2. 值 - 要么是类型长度编码的字符串，要么是4字节整型值
3. 类型 - 3或4字节，描述条件的种类（查看上面的主表）

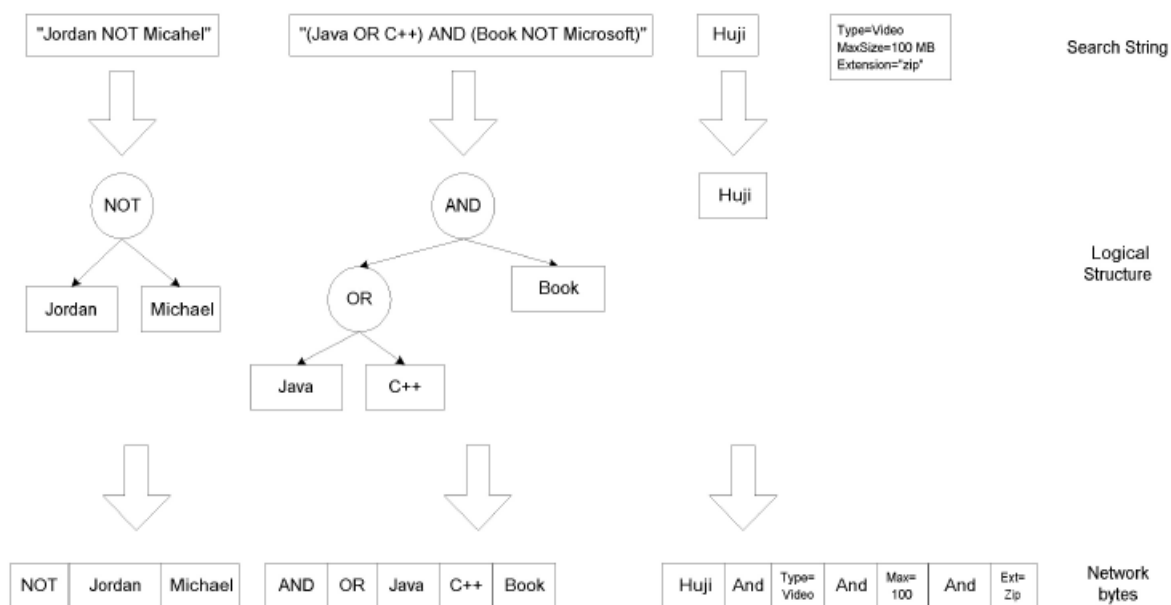


Figure 6.1: Search string encoding example

### 6.2.10 搜索结果

从服务器发送到客户端的消息，作为对搜索请求的回应。这个消息通常被压缩。消息大小可变的。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x16	OP_SEARCHRESULT操作码的值
Result Count	4	NA	消息中搜索结果数目
Result list	可变的	NA	搜索结果列表

### 搜索结果列表项格式

下面的表格描述了一个搜索结果列表项的格式。每个搜索结果包含了一个哈希，这个哈希唯一地标识文件，连同拥有这个文件的其它eMule客户端的细节。也有几个标签描述了文件属性。标签列表如下所述。注意，大部分的标签是可选的，它们的次序是不保证的。标签编码规则在本章开头详细描述。

名称	字节大小	默认值	注释
File Hash	16	NA	哈希值，用来文件的唯一的身份证明
Client ID	4	NA	拥有文件的eMule客户端的ID
Client Port	2	NA	拥有文件的客户端的TCP端口
Tag Count	4	NA	下面描述符标签的个数
Tag list	可变的	NA	描述符标签列表

### 6.2.11 获得源

该消息从客户端发送到服务器，为文件请求源（其它客户端）。消息大小是22字节。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x19	OP_GETSOURCERS操作码的值
File hash	16	NA	请求文件的哈希

### 6.2.12 已找到的源

该消息从服务器发送到客户端，带有请求文件的源（其它客户端）。消息大小可变的。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x42	OP_FOUNDSOURCERS操作码的值
File hash	16	NA	请求文件的哈希
Sources Count	1	NA	消息中源的个数
List of sources	可变的	NA	源列表

### 源列表项格式

下面表格描述了源列表项的格式。每个源包含了拥有该文件的eMule客户端的细节。

名称	字节大小	默认值	注释
Client ID	4	NA	拥有该文件的eMule客户端的客户ID
Client Port	2	NA	拥有该文件的eMule客户端的TCP端口

### 6.2.13 回调请求

该消息从客户端发送到服务器，请求其它客户端回调 - 例如，连接到请求中的客户端。有高ID的、想连接到一个低ID客户端（见2.4节）发送该消息。消息大小是10字节。

名称	字节大小	默认值	注释
Protocol	1	0xE3	

Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x1C	OP_CALLBACKREQUEST操作码的值
Client ID	4	NA	被请求回调客户ID

#### 6.2.14 被请求回调

该消息从服务器发送到客户端，指出另一个客户端要求接收的客户端连接它。当接收的客户端是低ID（见2.4节），发送该消息。消息大小是12字节。接收的客户端尝试连接这个由回调请求包指定的IP和端口。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x35	OP_CALLBACKREQUESTED操作码的值
Client IP	4	NA	请求呼叫自己的客户端IP
Client TCP Port	2	NA	客户端监听的TCP端口

#### 6.2.15 回调失败

该消息从服务器发送到客户端，指出客户端的回调请求失败。消息大小是6字节。接收的客户端把这个消息记日志，然后丢弃它。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域



Type	1	0x36	OP_CALLBACK_FAILED操作码的值
------	---	------	-------------------------

### 6.2.16 消息被拒绝

该消息从服务器发送到客户端，指出服务器拒绝客户端发出的最后一条命令。消息是6字节。接收的客户端把这个消息记日志，然后丢弃它。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x05	OP_REJECT操作码的值

## 6.3 客户端服务器 UDP 消息

本章描述了在服务器和客户端之间用UDP传送的消息。UDP消息头部不包含消息的大小，因为它可以从UDPL3头部推论出来。大部分的消息都是很小、固定大小的，周期性地发送到客户端服务器列表中的服务器上。

### 6.3.1 获取源

从客户端发送到服务器，为文件（其它客户端拥有这文件）请求源。这个消息会每秒周期发送，为那些只有少数源的文件。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Type	1	0x9A	OP_GLOBGETSOURCES操作码的值
File ID List	NA	NA	文件ID（哈希）（每个16字节长）列表，这些ID一个个排序，没有优先的。

### 6.3.2 发现的源

从服务器发送到客户端，作为对UDP获取源消息的回应。该消息只在当服务器有请求文件的源时才发送。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Type	1	0x9B	OP_GLOBFOUND SOURCES操作码的值
File source list	NA	NA	下面描述的文件源的列表。

#### 源列表项格式

下表包含了一个文件的源。

名称	字节大小	默认值	注释
File ID	16	NA	找到源的文件的ID
Sources Count	1	NA	报告的源的个数
List of sources	NA	NA	源的列表，和6.2.11节的TCP获得源消息的格式一样

### 6.3.3 状态请求

该消息是每几秒发送到服务器的状态请求。这个消息包含随机4字节询问，这个询问应该由服务器反应。消息长度是6字节。这个消息是客户端服务器UDP保持连接方案（见3.1节）的一部分。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Type	1	0x96	OP_GLOBSERVSTATREQ操作码的值
Challenge	4	NA	发送到服务器的无符号整数的询问，用来回应验证（在客户端相对应的变量叫做”time”）

#### 6.3.4 状态回应

服务器回应消息，针对客户端UDP状态请求消息。包含几个服务器信息项，注意，大部分项是可选的。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Type	1	0x97	OP_GLOBSERVSTATRES操作码的值
Challenge	4	NA	无符号整数。可以作为客户端发送的询问的回应，或者其它不同的数值
User Count	4	NA	可选的。当前登录到服务器中的用户数目
Files count	4	NA	可选的。服务器数据库中的文件数目
Soft files limit	4	NA	可选的。无符号整数，服务器软件文件限制
Hard files limit	4	NA	可选的。无符号整数，服务器硬件文件限制
UDP flags	4	NA	可选的。服务器UDP标识。定义了两种不同的标识：0x01指出服务器支持获得源消息。0x02指出服务器支持扩展的获得文件消息

### 6.3.5 搜索请求

当客户端配置成用UDP搜索时，发送该消息到客户端列表中的服务器上。该消息由两个可选的操作码（0x98或0x92），后者指出服务器的以后（加强）版本。客户端根据服务器发送过来的UDP状态回应消息（6.3.4节）中的UDP标识来决定用哪个操作码。为了发送加强的操作码，服务器必须打开0x02位。UDP标识也可以从配置文件中载入。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Type	1	0x98或0x92	分别是OP_GLOBSEARCHREQ或OP_GLOBSEARCHREQ2操作码的值
Search request parameters	可变的	NA	与在客户端服务器TCP交流一章中的搜索请求消息参数，见6.2.9，是一样的。

### 6.3.6 搜索回应

搜索回应消息从服务器发送到客户端。该消息发送来回应搜索消息中的两个操作码。该消息的格式与TCP搜索结果消息非常相似，虽然结果被折叠没有结果个数。请阅读描述TCP搜索结果消息的章节，获得更多的关于该消息不同的字段（6.2.10节）。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Type	1	0x99	OP_GLOBSEARCHRES操作码的值
Result lists	NA	NA	结果列表（没有个数），如6.2.10所述

### 6.3.7 服务器描述请求

每隔几秒发送到服务器。没有包含进有效负载。是UDP ping 策略一部分。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Type	1	0x99	OP_SERVER_DESC_REQ操作码的值

### 6.3.8 服务器描述回应

从服务器发送到客户端，作为对客户端的服务器描述请求的回应。包含名称和回应的服务器的描述。该消息是可变长度的。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Type	1	0x99	OP_SERVER_DESC_RES操作码的值
Name	NA	NA	服务器的名称，以2字节长度编码的字符串，字符数组值格式
Description	NA	NA	服务器的描述，以2字节长度编码的字符串，字符数组值格式

## 6.4 客户端到客户端 TCP 消息

本章描述了在客户端之间用TCP传递的消息。客户端到客户端消息分成eMule和eDonkey两种类型 - 注意，下面描述了消息中的头部字段。也有私有的类型用来传递文件的实际传递中。

### 6.4.1 Hello

该消息是两个emule客户端之间握手的第一个消息。这个消息非常类似服务器登录消息（见6.2.1节）。这两个消息都有相同的类型代码（0x01），它们是协议中唯一拥有重叠类型代码的消息。这两个消息都提供了相同的数据，甚至以相同的顺序。有两个主要不同之处：客户端hello消息以用户哈希大小字段开始，而服务器登录消息立即以用户哈希值开始的，客户端hello消息以额外的服务器IP和端口信息结束，服务器登录信息并不以之相应。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x01	OP_HELLO操作码的值
User Hash size	1	16	用户哈希字段大小
User Hash	16		TBD
Client ID	4	0	TBD
TCP Port	2	4662	客户端使用的TCP端口，可配置的
Tag Count	4	4	消息中紧接着的标签的数目
Tag list	可变	NA	标签列表，指定远程客户端的属性
Server IP	4	NA	客户端连接到服务器的IP
Server TCP Port	2	NA	服务器上监听的端口

标签列表中有三种类型的标签出现。端口标签是可选的，通常不提供。标签编码规则在本章开头详细描述。

名称	标签名称	标签类型	注释
Username	整数, 0x01	字符串	
Version	整数, 0x11	字符串	
Port	整数, 0x0F	整数	

### 6.4.2 Hello 回应

作为Hello消息的回应发送。除了消息类型外，包含与hello消息完全一样的字段。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x4C	OP_HELLOANSWER操作码的值
Hello files			与hello消息中 user hash开始的字段一样

### 6.4.3 发送文件块

这个消息包含一个下载文件的一块。消息大小（在头部）指定了整个块的大小，不只是要发送的消息的包的大小。这消息被分成几个数据包，每个的有效负载小于最大的TCP MTU – 在eMule 0.30e中，有效负载大小是1300字节。第??章详细描述了发送的文件块。关于发送压缩文件块的细节也可见6.5.3节。

名称	字节大小	默认值	注释
----	------	-----	----

Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x46	OP_SENDINGPART操作码的值
File ID	16	NA	通过哈希文件数据得到的唯一文件ID
Start Pos	4	NA	下载数据的开始位置
End Pos	4	NA	下载数据的结束位置
Data	NA	NA	实际下载的数据。数据可能被压缩

#### 6.4.4 请求文件块

发送到对等的客户端请求文件块。这个消息可以请求最大数目为3个文件块下载。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x46	OP_SENDINGPART操作码的值
File ID	16	NA	通过哈希文件数据得到的唯一文件ID
Part 1 Start offset	4	NA	文件中块1的开始位移
Part 2 Start offset	4	NA	



Part 3 Start offset	4	NA	
Part 1 End offset	4	NA	文件中块1的结束位移
Part 2 End offset	4	NA	
Part 3 End offset	4	NA	

#### 6.4.5 下载结束

下载的客户端用来指示一个文件下载完成。EMule 0.30从不发送这个消息，但是客户端处理这个消息的接收（通过释放相关的资源）。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x49	OP_END_OF_DOWNLOAD操作码的值
File ID	16	NA	文件ID

#### 6.4.6 改变客户 ID

当客户端已经更改服务器（例如，中断当前服务器连接，连接到新的服务器）发送该消息，结果是取得新的客户端ID。

名称	字节大小	默认值	注释
----	------	-----	----

Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x4D	OP_CHANGE_CLIENT_ID操作码的值
Client ID	4	0	新的客户端ID
Server IP	4	NA	客户端连接的服务器的IP
Server TCP Port	2	NA	服务器监听的TCP端口

#### 6.4.7 聊天消息

用来执行eMule提供的聊天服务。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x4E	OP_MESSAGE操作码的值
Length	2	NA	消息的长度
Message	可变	NA	实际的消息

#### 6.4.8 块 hashset 请求

发送该消息来请求所有被请求文件的哈希和文件中的每一部分哈希。唯一的文件ID和哈希在1.5节中已经解释。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x51	OP_HASHSETREQUEST操作码的值
File ID	16	NA	

#### 6.4.9 块 hashset 回应

包含一个全局哈希（对于所有文件）和文件中每块的哈希的文件哈希集的回应消息。当这个消息处理完成，客户端发送一个开始上传请求。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x52	OP_HASHSETANSWER操作码的值
File Hash	16	NA	整个文件中抽出的哈希
Part Count	2	NA	文件中块的数量
Part Hashes	可变	NA	每个文件块的哈希 – 每个哈希的大小是16字节

#### 6.4.10 开始上传请求

开始上传请求消息。这个消息开始文件下载队列，4.3节已经讨论。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x54	OP_STARTUPLOADREQ操作码的值
File ID	16	NA	请求文件的ID

#### 6.4.11 接受上传请求

一个正确应答，指示上传请求被接受，上传的客户端现在等待块请求。这个消息除了标准的eMule消息头部，没有其他字段。这个消息是块传输协议的一部分。更多细节参看??节。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x55	OP_ACCEPTUPLOADREQ操作码的值

#### 6.4.12 取消传送

取消文件传输请求。这个请求不包含任何字段。发送这个消息的典型情景是当一个下载已经到达队列的顶部，但是它已经从其它源中下载了这个文件。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x56	OP_CANCELTRANSFER操作码的值

#### 6.4.13 Out of part requests

eMule 0.30e中没有使用。指示下载的客户端完成下载所有上传客户端发送过来的块，正在等待更多的块。注意，接收的客户端没有采取任何实际的动作。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x57	OP_OUTTOPARTREQS操作码的值

#### 6.4.14 文件请求

从向其他客户端请求文件的客户端发送该消息。这个消息包含请求文件的ID和状态字段描述了，哪个块准备下载。

## 块状态字段

eMule允许客户端从其它客户端中下载文件块，甚至当提供的客户端还没有完全下载这个请求的文件。块状态字段帮助区分一个只是不在被请求的客户端上的文件和一个只有部分被下载的文件。如果，文件不存在，则块字段值为0。如果文件是部分的下载，则头2字节是整数，给出已经下载的块的数量，而最后字节是位向量，指示哪个文件是完全下载的（通过设置机器位为1）。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x58	OP_FILEREQUSET操作码的值
File ID	16	NA	惟一的文件ID
Part Status	3	NA	可选的，如果eMule信息中的扩展请求版本指示消息大于0，则发送。文件签名在本章中介绍
Source count	2	NA	可选的，如果eMule信息中的扩展请求版本指示消息大于1，则发送。指示这个文件的当前源的数量

## 6.4.15 文件请求回答

文件请求回答，作为回应文件请求消息发送。这个消息是可能回应文件请求中的唯一一个，更多细节见4.3节。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域

Type	1	0x59	OP_FILEREQANSWER操作码的值
File ID	16	NA	惟一的文件ID
Name length	2	NA	文件名长度
Filename	可变	NA	文件名（长度指定）

#### 6.4.16 找不到文件

这个消息回答文件请求，指出被请求的文件或文件的块在客户端上没有找到。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x48	OP_FILEREQANSOFIL操作码的值
File ID	16	NA	没有存在的文件ID
Part Status	3	NA	在文件请求（6.4.14节）已解释

#### 6.4.17 被请求的文件 ID

文件下载请求消息。这个消息只指出文件ID。

名称	字节大小	默认值	注释
----	------	-----	----

Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x4E	OP_SETREQFILEID操作码的值
File ID	16	NA	被请求的文件ID

#### 6.4.18 文件状态

在客户端拥有被请求的文件时，发送来作为被请求的文件ID消息的回答。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x50	OP_FILESTATUS操作码的值
File ID	16	NA	报告状态的文件ID
Part Status	3	NA	在文件请求（6.4.14节）已解释

#### 6.4.19 Change slot

eMule 0.30e不发送。当收到时被忽略。

名称	字节大小	默认值	注释
----	------	-----	----



Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x5B	OP_CHANGE_SLOT操作码的值

#### 6.4.20 队列等级

eMule 0.30e不发送。当收到时被忽略。对于扩展的协议队列等级消息，参看6.5.4节。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x5C	OP_QUEUERANK操作码的值

#### 6.4.21 查看共享文件

请求共享的文件列表。如果请求的客户端没有被禁止和没有设置为忽略这样的请求，就回答这个请求。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x4A	OP_ASKSHAREDFILES操作码的值

### 6.4.22 查看共享文件回答

这个消息编码共享文件给请求的客户端。这个消息包含共享文件列表的细节。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x4B	OP_ASKSHAREDFILESANSWER操作码的值
Result Count	4	NA	编码进消息的文件名称的数量
Result List	可变	NA	与在服务器搜索结果（6.2.10节）中的一样的编码

### 6.4.23 查看共享文件夹

请求共享文件夹队列。如果请求的客户端没有被禁止和没有设置为忽略这样的请求，就回答这个请求。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x5D	OP_ASKSHARED_DIRS操作码的值

#### 6.4.24 查看共享文件夹回答

回答查看共享文件夹请求。这个回答包含共享文件夹队列。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x5F	OP_ASKSHARED_DIRSANS操作码的值
Folder Count	4	NA	消息描述的文件夹数量
Folder List	可变	NA	文件夹描述队列，以下面描述的长度值格式

#### 文件夹列表格式

文件夹列表格式是一系列的项，每个项包含一个短整数（16单元）长度和字符串值。

#### 6.4.25 查看共享文件夹内容

请求列出客户端共享目录之一的内容。这个请求是期望的，如果目录是共享的和请求的客户端是受信任的客户端。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域

Type	1	0x5E	OP_ASKSHAREDFILES DIR操作码的值
Name length	2	NA	紧接着的目录名的长度
Directory name	可变	NA	共享的文件夹名称

#### 6.4.26 查看共享文件夹内容回答

对共享文件目录请求（OP\_ASKSHAREDFILES DIR）的回答。这个回答包含文件名称（共享目录中存在的所有文件名）列表。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x60	OP_ASKSHAREDFILES DIRANS操作码的值
Name length	2	NA	紧接着的目录名的长度
Directory name	可变	NA	共享文件夹名（被下列文件列表引用）
Files Count	4	NA	紧跟着该字段的文件名称数量
Files list	可变	NA	文件名列表，以（16单元长度，字符串值）格式

### 6.4.27 查看共享文件夹或内容拒绝

发送用来指出共享文件或目录请求被拒绝。发送这个消息来拒绝OP\_ASKSHARED\_DIRS和OP\_ASKSHARED\_FILES\_DIR请求。

名称	字节大小	默认值	注释
Protocol	1	0xE3	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x61	OP_ASKSHARED_DENIED操作码的值

## 6.5 客户端到客户端 TCP 扩充消息

下面的消息是eMule的一部分，而不是eDonkey的一部分，它们叫做扩充协议消息。

### 6.5.1 eMule 信息

携带关于eMule客户端的一般信息。下面的标签列表与eMule 0.30e一致，可能与后续或先前的版本协议不同。该消息通过eMule信息回答消息（6.5.2节）立即返回。是客户端初始化握手的一部分（4.1节）。eMule信息包含了几十个标签，指出eMule客户端的特性，标签的重要性和值范围下面简单解释。

#### UDP 版本标签

提供客户端的UDP协议版本 – 例如，支持哪个客户端到客户端的UDP消息。这标签是整型标签，它的名称是0x22。在0.30e中，标签值为0x03。

#### 来源交换标签

这个标签指出客户端是否支持与其它客户端交换源，标签值被看作是源交换支持版本。标签是整型标签，它的名称是0x23。在0.30e中，标签值为0x02，允许的值要么是0x01或0x02。

## 注释支持标签

指出是否支持文件注释。这标签是整型标签，它的名称是0x24。在0.30e中，标签值为0x01，指出客户端是否支持注释。

## 扩充请求标签

这个标签指出客户端发送或接收扩充文件请求的能力。这标签是整型标签，它的名称是0x25。在0.30e中，标签值为0x02。

## 扩充选项标签

在0.30e中，这标签提供关于两个特性的信息 – 低6位指出客户端是否支持源身份确认，下一位（第7位）指出客户端是否支持文件预览。这标签是整型标签，它的名称是0x27。？ ？ ？

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x01	OP_EMULEINFO操作码的值
Client Version	1	NA	EMule客户端版本
Protocol Version	1	0x01	EMule协议版本
Tag Count	4	7	紧跟着的标签的数量
Compression tag	8	NA	指出这个客户端支持压缩。这标签是整型标签，它的名称是0x20。在0.30e中，标签值为0x01
UDP Version tag	8	NA	上文已解释
UDP Port tag	8	NA	提供客户端监听的UDP端口。这标签是整型标签，它的名称是0x21。默认值是4672
Source exchange tag	8	NA	上文已解释

Comments tag	8	NA	上文已解释
Extend requests tag	8	NA	上文已解释
Extended options tag	8	NA	上文已解释

### 6.5.2 eMule 信息回答

作为回答eMue信息消息发送。包含相同的信息。这消息是客户端初始化握手的一部分（4.1节）。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x02	OP_EMULEINFOANSWER操作码的值
EMule Info fields			这个消息与eMule信息消息有相同的字段

### 6.5.3 发送压缩的文件块

这个消息用来发送压缩的文件块。一个块可以被压缩，压缩的内容以几个顺序包发送，就好像普通的发送块消息（6.4.3节）。

名称	字节大小	默认值	注释
----	------	-----	----

Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x40	OP_COMPRESSED PART操作码的值
File ID	16	NA	惟一的文件ID
Part start offset	4	NA	发送块的开始位移
Compressed content size	4	NA	压缩内容的大小
Compressed content	可变	NA	压缩块

#### 6.5.4 队列等级

报告对等客户端它的队列等级 – 它在下载队列中的等待位置。当一个客户端为某文件加到一个等待队列中时，通常发送该消息。在大部分情况下，该消息跟在关闭客户端TCP连接之后。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x60	OP_COMPRESSED PART操作码的值
Queue position	2	NA	客户端在队列中的位置
Buffer	10	0	10个零位，目的不明



### 6.5.5 文件信息

该消息包含对特定文件的文件描述。发送来回答文件请求或文件上传请求消息。该消息没有包含文件ID，在假设一个TCP连接工作在一个一个文件上的情况下，文件ID隐含地通过接收客户端抽取出来。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x61	OP_FILEDESC操作码的值
File rate	1	NA	文件上的某种字节提供率
Comment length	2	NA	紧接着的注释的长度（不超过128）
Comment	可变	NA	注释本身

### 6.5.6 源请求

发送请求源（有被请求文件的其他客户端）。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域

Type	1	0x81	OP_REQUESTSOURCES操作码的值
File ID	16	NA	被请求文件的文件ID

### 6.5.7 源回答

发送来回答请求源消息。这个消息可以被压缩（当它不压缩的大小大于394字节）。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x82	OP_ANSWERSOURCES操作码的值
File ID	16	NA	发送回答上的文件ID
Source Count	2	30	在eMule 0.30e中，总是0
List of sources	可变	NA	被请求文件的源的列表，如下所述

### 源列表项格式

下表描述了源列表项的格式。每个源包括eMule客户端的细节，eMule客户端持有被请求的文件和最新连接到的服务器。

名称	字节大小	默认值	注释
Client ID	4	NA	有此文件的eMule对等的客户端ID
Client Port	2	NA	有此文件的客户端的TCP端口

Server IP	4	NA	服务器IP地址，从服务器中发现这个用户
Server Port	2	NA	服务器监听客户端连接的TCP端口
Client hash	16	NA	可选的 – 只在两个客户端都有源交换版本（见OP_EMULEINFO）大于1 时发送

### 6.5.8 安全身份认证

通常在初始化握手后发送（4.1节）。该消息只在对等的客户端支持安全身份认证（如在eMule信息消息中指出）时发送。这消息指出发送客户端是否有对等客户端的公匙和是否也提供一个随机值作为对等客户端签署的口令。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x87	OP_SECIDENTSTATE操作码的值
Operation	1	2	指出是否需要一个公匙和签名（2）或者只需要一个签名
Challenge	4	NA	对等客户端签署的随机字

### 6.5.9 公匙

包含客户端的公匙，当两者支持加密时使用。

名称	字节大小	默认值	注释
----	------	-----	----

Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x85	OP_PUBKEY操作码的值
Public key length	1	76	公匙长度
Public key	可变	NA	通常76字节长

#### 6.5.10 签名

客户端使用它的公匙签署4字节的口令。这个口令是由远程客户端发送过来的随机字和其他的字 – 要么是远程客户端IP地址（如果签署的客户端是低ID）或者如果签署的客户端是高ID，签署的客户端的高ID组成。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x86	OP_SIGNATURE操作码的值
Signature length	1	48	签名长度
Signature	可变	NA	通常48字节长

### 6.5.11 预览请求

对特定的图像文件请求图像预览。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x90	OP_REQUESTPREVIEW操作码的值
File ID	16	NA	文件ID

### 6.5.12 预览回答

图像预览回答消息。包含被请求的图像文件的预览或者如果被请求的文件不是图像，只是文件ID。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x91	OP_PREVIEWANSWER操作码的值
File ID	16	NA	文件ID或0，如果只是发送图像
Frame count	1	NA	图像发送的帧数量
Frame	可变	NA	图像帧以（4字节）长度，值编码

## 6.6 客户端到客户端 UDP 消息

本章描述了在客户端与服务器之间使用UDP发送的消息。所有使用UDP发送的消息都是eMule扩充的一部分，这样协议ID是eMule协议，不是eDonkey协议。

### 6.6.1 重复询问文件

重复向对等的客户端询问关于被请求文件的状态。当询问客户端不在请求客户端队列中，并且想周期性了解它是否能开始下载文件时，发送该消息。这个消息有三个可能的回应，如下所述。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x90	OP_REASKFILEPING操作码的值
File ID	16	NA	重复询问的文件ID
Source count	2	NA	可选的。无符号整型。被请求文件的源的当前数量

### 6.6.2 重复询问回应

重复询问文件请求的其中一个回应指出，对等的客户端已经接收到请求，在它的下载队列中有空位，请求的客户端以特定的等级排在队列中。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域

Type	1	0x91	OP_REASKACK操作码的值
Rank	2	NA	无符号整型。请求客户端在请求队列中的位置

重复询问文件请求的另一个回应指出，找不到文件，这个消息不指出哪个文件找不到。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x92	OP_FILENOTFOUND操作码的值

### 6.6.3 队列满

重复询问文件请求的另一个回应，这个消息不指出哪个队列已满。

名称	字节大小	默认值	注释
Protocol	1	0xC5	
Size	4		消息大小是以字节为单位，不包括头部和大小的域
Type	1	0x93	OP_QUEUEFULL操作码的值

(全文翻译完)

