

eMule 协议指南

The eMule Protocol Specification

作者:

Yoram Kulbak &
Danny Bickson

译者:

flakever

VISIT ME AT WWW.FLAKEVER.CN

eMule 协议指南

作者: Yoram Kulbak and Danny Bickson

译者: Flakever (flake.sun@gmail.com)

0 译者序

0.1 使用指南

文中出现蓝色字体,如 [2.1](#),则意为见 2.1 节.绿色字体为参考文献编号,原作者提供了 [1-5](#) 个参考文献,其后为译者添加.此处取消了原文中的这两种格式的连接功能,避免打断阅读过程.请自行使用[书签](#).

0.2 版权声明

原文版权由原文作者 Yoram Kulbak and Danny Bickson 所有,译文版权由 flakever 所有.
该译文可在非商业目的下任意传播和复制.
对译文的修改或将其用于任何商业目的的行为需获得译者同意.

0.3 版本

本文为<<eMule 协议指南>> 1e

0.4 鸣谢

感谢父母给予的关怀.
特别感谢女友 Snow 的支持,祝你 GMAT800+6,^_^.

1 前言

1.1 文章简介

eMule 是一款基于 eDonkey 协议的流行文件共享应用程序,本文主要描述 eMule 的网络行为,并解释了理解 eMule 协议所需要了解的相关术语.与此同时,本文给出了完整的 eMule 协议指南,并在附录中表述了具体的报文编码格式.本文讨论基于一款开源的 eMule 客户端[2].前言向读者提供理解文本所需的背景知识.如果需要获取更多关于 eMule 的信息,可以参考[3].

1.2 概览

在 eMule 网络世界中,全球范围内数以百计的 eMule 服务器和成百上千万的 eMule 客户端[1]导演着 eMule 的繁荣.客户端(Clients)为了得到 eMule 网络服务,需要与至少一个服务器(Server)建立连接,这个与服务器的连接会在客户端整个生命周期内长期保持.在这个体系内,服务器提供集中的索引服务(如同 Napster),值得注意的是,服务器之间并不进行通信.

每个 eMule 客户端都中预设置有两个列表,其一包括服务器地址,另一个列表则包含了其本地文件系统中的共享文件.客户端与一个 eMule 服务器建立一个对应的 TCP 连接,从而登录网络并获取关于所需文件的信息,并获取用于找寻其他可联络的客户端的信息.于此相对应,eMule 客户端为了实现与其他客户端间进行文件上传与下载,与这些客户端建立了数以百计的 TCP 连接.每个 eMule 客户端为它的每个共享文件都独立得维护一个上传有限队列.一个待下载的客户端将会从这个队列底部加入,并逐步提升其在队列中的位置,直到这个它达到队列顶部,就开始下载文件.一个客户端可以从多个不同客户端下载同一个文件,从每一个客户端中下载文件的不同部分.同理,上传客户端也可以给待下载客户端上传文件的部分块,即便上传客户端没有完成整个文件的下载.eMule 还扩展了 eDonkey 的性能,允许客户端之间交换服务器信息和共享文件及其他客户端的相关信息.值得注意的是,客户端-客户端通信和客户端-服务器通信都是基于 TCP 的.

服务器通过其数据库保存文件和客户端的信息,它并不保存任何共享文件,它只是为共享文件的位置提供一个集中索引.服务器还有一个已经逐渐被放弃的功能,即为因有防火墙阻碍或其他原因造成的无法接受进入连接的客户端(俗称 LowID 客户端)提供桥接服务.但是这种桥接功能极大地增加了服务器的载荷.eMule 还使用 UDP 来提升客户端-服务器及客户端-客户端连接的性能.但是,对于一个客户端是否能正确的传输和接受 UDP 报文并无强制性要求,也不会影响到常规功能,即便防火墙禁止了 UDP 传输也可以无故障的运行 eMule.

1.2.1 客户端-服务器连接

客户端启动时,首先与一个 eMule 服务器建立 TCP 连接,服务器向其提供一个客户端 ID(详见 1.3 节).客户端 ID 的生命周期等于其客户端-服务器连接的生命周期,对于 HighID 而言,所有的服务器都会根据其 IP 计算一个固定的客户端 ID,这点与 LowID 不同.随着连接建立,客户端向服务器上传它的共享文件列表,服务器将这个列表储存在数据库中,同时,客户端向服务器上

传一个下载文件列表,其中包含了客户端需要下载的文件信息.本文第 2 部分详细分析了 eMule 的客户端-服务器 TCP 连接的报文交换机制.

连接建立完毕后, eMule 服务器向这个客户端发送一个源列表,这个列表包含了拥有这个客户端待下载文件的其他客户端的信息.这种拥有别的客户端所需求的资源或文件的客户端就称作源.从这时起,这个客户端就开始与其他客户端建立连接(见 1.2.2).

这个客户端-服务器 TCP 连接在整个客户端会话过程中一直保持开启.在初始的握手会话后,客户端-服务器会话往往由客户端的某些活动所引起,比如客户端的文件搜索请求会引起服务器的文件搜索结果应答,在搜索会话之后,客户端会发起一个对特定文件的源查询请求,此时,服务器会返回一个源列表(包括 IP 地址和端口号),客户端就可以使用这个列表上的信息下载所查询的文件.

客户端-服务器 UDP 连接是客户端用于和其他未建立 TCP 连接服务器间建立通讯的.这种 UDP 通信的目的在于加强文件搜索和源搜索功能,并更高效的使用客户端所拥有的服务器信息. 本文第 3 部分详细分析了 eMule 的客户端-服务器 UDP 连接的报文交换机制.

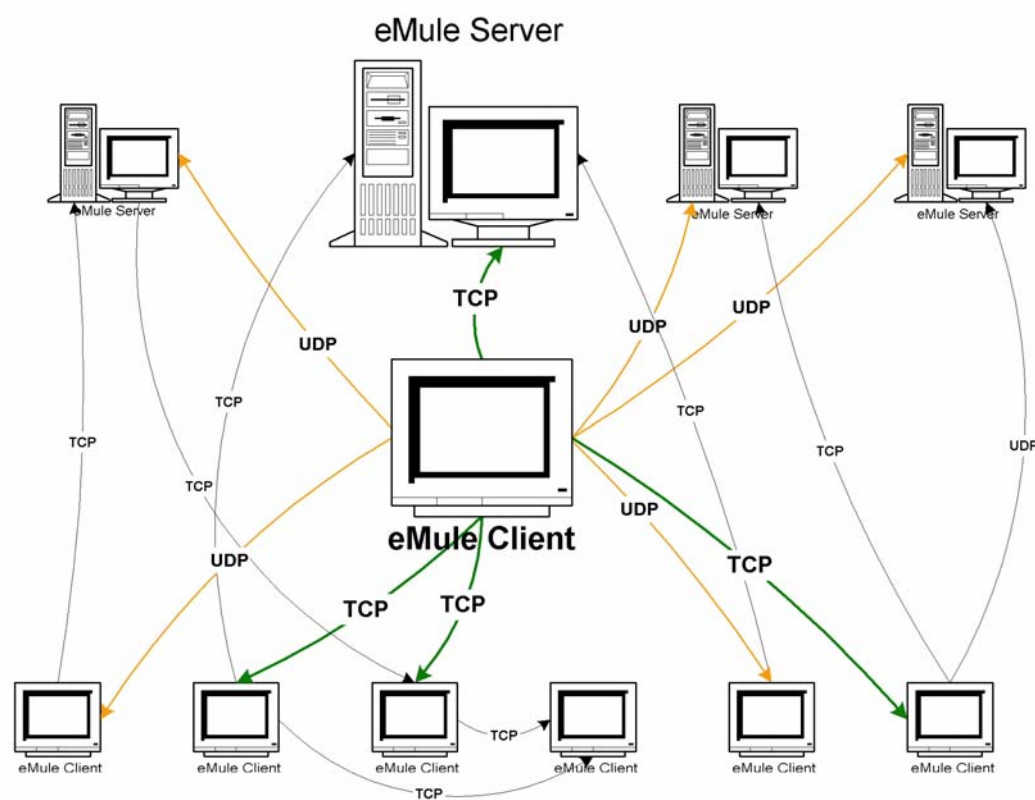


图1.1 : eMule网络概观

1.2.2 客户端-客户端连接

eMule的客户端-客户端连接实现文件的共享,在共享过程中,文件被分为不同部分,每部分并被细分为不同的块,一个客户端可以同时从多个不同的客户端处下载一个文件的不同部分.

当客户端-客户端连接建立后,两个客户端会进行传输性能信息交换,随后对上传或下载的开始时间进行协调.每个客户端都会对发出下载请求的待下载客户端建立一个下载队列.当一个 eMule客户端的下载队列为空时,新的下载请求会立即开始下载会话,除非这个请求客户端被源禁止.当这个下载队列非空时,新的下载请求就将请求客户端添加到下载队列底部.如果将传输带宽最小值定为2.4kbytes/秒,一个客户端可同时进行的上传数通常不超过几个.在下载

队列中,允许一个正在等待的客户端取代正在下载的客户端,这取决于他们的队列优先级别,在一个客户端开始下载会话的十五分钟之内,它在下载队列中的优先级别会提高到最高水平,以防止频繁的下载竞争。

当一个待下载客户端到达下载队列顶部,上传客户端会在这两个客户端之间启动一个用于文件传输的连接.一个eMule客户端可出现在多个源的待下载队列中,等待按部分从每个目标源中下载文件.当客户端从其中一个源下载到文件的某些部分,这个下载客户端并不会告知所有它在下载队列出现的源,并让它们删除这个下载客户端的请求.而是当其到达某个源的下载队列顶部时,拒绝源的上传会话。

eMule使用了一种鼓励上传的信用机制(见1.4节),为了防止针对这种机制的冒名欺诈,eMule使用RSA公钥算法。

客户端-客户端连接会用到一组不同于eDonkey协议中定义的报文,被称为扩展协议.扩展协议提供一些增强服务,如信用机制的具体实现,增加的服务器和源列表更新功能,以及通过压缩传输增强文件块传输的性能。

eMule的客户端-客户端连接中的UDP使用较为局限,主要用于客户端在对列中等待开始下载时,周期性地查询其在队列中的状态。

1.3 客户端 ID

客户端ID是在客户端-服务器连接握手时由服务器提供的一组4字节标识码.对于LowID而言,这个客户端ID只在整个客户端-服务器TCP连接周期中有效;而对于HighID,客户端ID是由其IP地址计算而来,只随IP的改变而改变.因此,客户端ID有HighID和LowID之分.典型的,eMule服务器会给一个无法接收进入连接的客户端分配LowID.当一个客户端被分配LowID,将会在eMule网络受到限制,并可能会因此引起服务器拒绝其连接.对于HighID而言,其ID是经下文中描述的方法由IP计算而得.此处的计算和重要性判断是基于一个eMule具体实现[3]的.被授予HighID的客户端需要具备如下条件,即其他客户端可以自由地连接到这台主机的eMuleTCP端口(默认为4662端口). 被授予HighID的客户端在整个eMule网络中不受限制.然而,只要服务器无法在客户端的eMule的TCP端口建立连接,就会给其赋予一个LowID.所以,一个客户端会因如下情况被赋予一个LowID:

- * 主机防火墙阻止了连入连接.
- * 客户端使用NAT[5]或代理服务器连入网络.
- * 服务器太忙(造成再连接计时器计时结束).

HighID按如下方法计算而得:

假设 IP 地址为 X.Y.Z.W,则客户端 ID 按公式 $X+28*Y+216*Z+24*W$ (Big Endian[6])计算.而LowID 总是一个小于 16777216(0x1000000)的值.作者并没有找到一个确定的 LowID 计算公式,但是值得注意的是,LowID 的分配因服务器而异.

因为 LowID 没有相应的 IP 地址来建立客户端-客户端连接,所以,所有的通信需要 eMule 服务器参与,这增加了服务器的负载同时也导致了服务器接收 LowID 客户端请求延迟.同时,这意味着一个 LowID 客户端无法和在其他服务器上的另一个 LowID 客户端建立连接.因为 eMule 不支持服务器间的通道连接.为了支持 LowID 客户端, eMule 引入了一种回叫机制.通过运用这种机制,HighID 客户端可以通过服务器询问一个 LowID 客户端,从而发起一个连接,并实现文件交换。

1.4 用户 ID

eMule 支持一个鼓励上传的信用体系,当一个用户向其他客户端上传的文件量越大,它就会在其他客户端的等待队列中享有更高的优先级别和更快的上升速度[3].用户 ID 是一组由随机连接数字组成的 128 位(16 字节)GUID[7],其中,第 6 位和第 15 位是固定的,分别为 14 和 111.与客户端 ID 只在和特定服务器的连接会话生命周期中有效不同,用户 ID(也叫做用户 Hash)是独一无二的,它在每次的会话中都保持不变.因为用户 ID 在整个信用体系中扮演重要角色,这也成了驱使一些所谓的“黑客”冒名替代信誉等级高的用户 ID 的原因.因此,eMule 使用了一种加密机制来防止用户欺骗和冒名替代.这个加密机制的实现依赖于一种给予 RSA 公钥加密系统的挑战-应答机制.

1.5 文件 ID

文件 ID 被用于网络中文件识别和文件损坏检测及修复.需要注意的是,eMule 并不是依靠文件名对文件进行唯一性辨认和分类,而是使用一个通过文件内容散列出的全局唯一 ID 进行标识和分类的.文件 ID 分为两种,其一主要用于产生唯一的文件 ID,第二种则用于文件损坏检测与修复.

1.5.1 文件 Hash

GUID 是客户端基于对文件内容进行 MD4 算法散列得出的一个 128 位标识符.计算时,文件被分为以 9.28MB 为大小的块,通过结合每块的独立计算结果,最终得出文件 ID.当一个客户端结束一个文件部分的下载,它会计算其 Hash 值,并与上传者提供的 Hash 值进行比较.一旦发现这两个值出现不同,则检测到文件损坏.此时,客户端会按 180KB 大小逐渐替代这个文件部分的块,直到最终的 Hash 值相符为止.

1.5.2 根 Hash

根 Hash 基于 SHA1 算法,它以 180KB 为大小对每个文件块进行运算.这样,提供了更强的可靠性和错误修复能力.具体内容请访问 eMule 网站.

1.6 eMule 协议扩展

eMule 协议与 eDonkey 协议完全兼容的同时,还加入了一些扩展实现,从而使得 eMule 客户端可以为用户提供一些附加功能.扩展协议主要在于客户端-客户端通信,尤其在于安全方面和 UDP 应用.本文中,所有 eMule 协议扩展部分的信息流都用灰色注明.

1.7 服务器容量限制

eMule 服务器有两种关于活动用户数的容量限制-硬限制和软限制.其中硬限制大于等于软限制数.当服务器活动用户数超过软限制数,它将不再接受来自 LowID 的新连接;当达到硬限制数,服务器饱和,不再接受任何客户端的连接.

2 客户端-服务器 TCP 通信

每个客户端仅与一个服务器建立 TCP 连接.这时,服务器会向客户端分配一个用于识别此客户端的 ID,这个 ID 在整个客户端-服务器会话中有效(HighID 则根据其 IP 地址分配客户端 ID).eMule 的可视化界面客户端需要以建立一个客户端-服务器连接为前提,完成其他操作.客户端不能同时与多个服务器建立 TCP 连接,除非用户干涉,客户端也不会自动地动态改变服务器.

2.1 建立连接

当客户端试图建立客户端-服务器 TCP 连接时,它会并行地向几个服务器发送相关请求,然而,当与其中一个服务器首先完成登录程序之后,将会排他地放弃向其他服务器发出的请求.

连接建立的过程有如下几种情况:

- 1.HighID 连接 – 服务器向申请连接的客户端授予一个 HighID.
- 2.LowID 连接 – 服务器向申请连接的客户端授予一个 LowID.
- 3.拒绝会话 – 服务器拒绝客户端请求.

当然,还存在一些其他的具体情况,如服务器停机或不可连接的情况等.

如下图 2.1 所示为 HighID 建立连接的报文传输过程.在这种情况下,客户端建立一个到服务器的 TCP 连接并向服务器发出登录(login)请求.之后,服务器建立另一个到该客户端的 TCP 连接,同时,服务器会发起一个由此客户端参与的客户端-客户端握手会话,并由此判断该客户端是否具有接受由其他 eMule 客户端发起的连接的能力.再次之后,服务器将关闭这个会话连接,传送一个改变 ID 报文,以此完成客户端-服务器握手.图中的 eMule-info 报文是灰色的,这代表了这个报文是由 eMule 扩展协议(见 1.6 节)定义的.

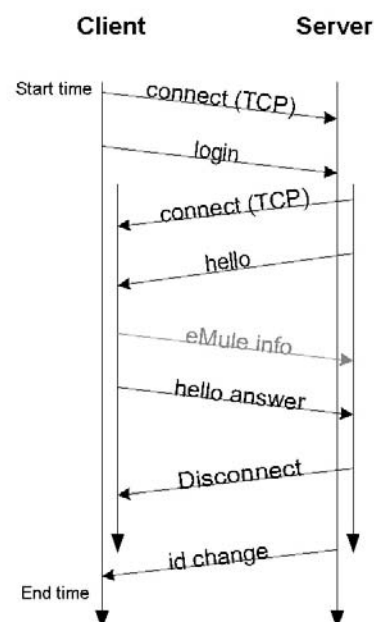


图 2.1 : HighID 建立连接的报文传输过程

如下图 2.2 所示,为 LowID 建立连接的报文传输过程,在这种情况下,服务器无法与请求连接的客户端连接,所以,它将授予此客户端以 LowID.同时,服务器会发送一条诸如此的报文:”注意,现在为 LowID 状态,请检查您的网络或客户端设置”.
无论是 LowID 还是 HighID,初次握手会话都将以接受一个 ID 变更报文为结束,这个变更报文最终定义了客户端的 ID,为下次与服务器的会话做好准备.

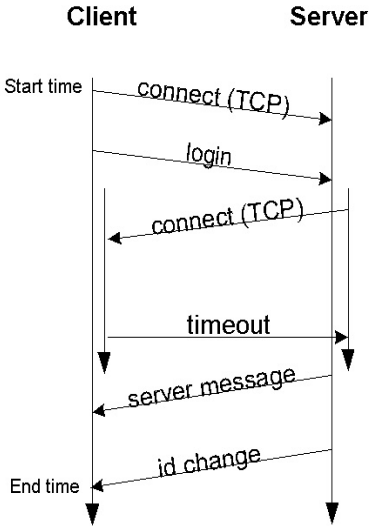


图 2.2 : LowID 建立连接的报文传输过程

下图 2.3 所示是当服务器拒绝客户端的登录请求时的报文会话过程.服务器拒绝此会话可能是因为达到其软限制且客户端为 LowID 或者达到了服务器的硬限制.在服务器返回的报文中,将会有有一个段简短的字符串描述拒绝会话的原因.

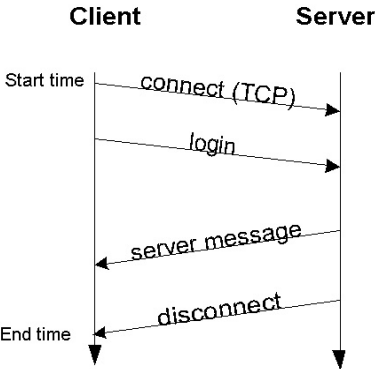


图 2.3 : 拒绝会话报文传输过程

2.2 连接启动的报文交换

当客户端-服务器连接成功建立之后,两者将进行一系列的报文交换,从而完成状态更新.起初,客户端会向服务器提供它自由的共享文件列表(见 6.2.4),随后要求更新客户端所有的服务器信息列表.这时,服务器会向此客户端发送关于其状态和版本的报文(见 6.2.6 和 6.2.2),此后,服

务器还会向此客户端发送更多关于其他 eMule 服务器的信息并提供更详细的自身信息.最终,客户端向服务器发送源请求,服务器接受此报文后,对于该客户端的待下载文件列表中的每个文件分别的发送相关的源列表,直到完成这一系列报文通信为止.整个连接启动报文交换过程见下图 2.4.

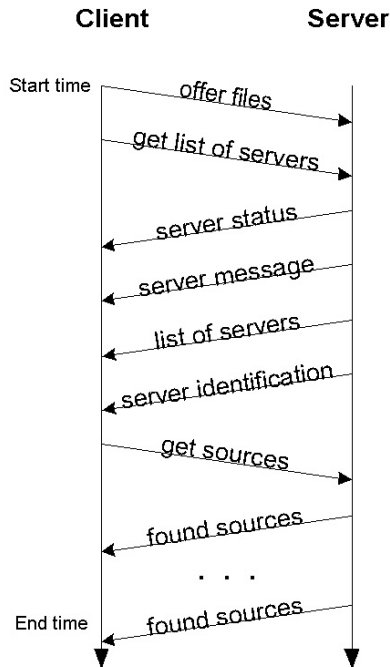


图 2.4：连接启动的报文交换过程

2.3 文件搜索

文件搜索会话总是由用户操作引起的,整个操作过程可以简单地概述为一个文件搜索请求(见 6.2.9)和对此请求的文件搜索请求应答(见 6.2.10).当搜索结果较多时,服务器返回的报文会被压缩传输.随后,用户选择下载其中的一个或多个文件,客户端会为每个选择下载的文件向服务器发起源请求.服务器会为每个请求返回一个源列表(见 6.2.12).在服务器发送源请求回复之前,可能会先向客户端发送一个服务器状态报文.这个服务器状态报文(见 6.2.6)包含了服务器关于目前服务器上用户和文件数量.值得指出的是,此处可能会使用 UDP 报文交换过程从而加强客户端定位源的能力,详情请见第 3 部分.此时,eMule 客户端会确认源是可用的,并按照它们添加到源列表中的顺序,启动与源的连接.图 2.5 描述了文件搜索报文交换过程.虽然 eMule 客户端按照添加顺序与源进行连接,而没有源连接的优先级机制,但是 eMule 客户端设计了一种解决从单一客户端获取超过一个待下载文件的机制.请注意, eMule 协议是只允许客户端-客户端连接间存在一路上传.选择算法基于用户的优先等级,当优先级相当时,则默认按照字母排序进行.解决从一个源客户端下载多个文件的具体描述见 eMule 网站.

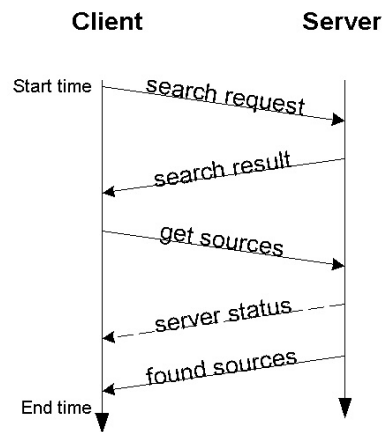


图 2.5： 文件搜索报文交换过程

2.4 回叫机制

回叫机制用于解决 LowID 客户端无法接受进入连接而无法与其他客户端共享文件的问题.回叫机制简述如下:加入有客户端 A 与 B,两者共在同一个 eMule 服务器之下,A 请求下载 B 上的文件,然而 B 为 LowID 客户端.此时,为了能够共享文件,A 向服务器发送一个回叫请求(见 6.2.13),让服务器要求 B 回叫 A.因为服务器与 A 之间建立着一个 TCP 连接,服务器可以直接传送给 B 一个回叫要求报文(见 6.2.14),在这个报文中,提供了 A 的 IP 地址和端口.这样, LowID 客户端 B 就可以直接不经过服务器与客户端 A 建立连接.显而易见,只有 HighID 客户端能要求 LowID 客户端回叫.图 2.6 表述了回叫报文交换过程.两个 LowID 客户端之间交换文件只能依靠服务器当作中继,绝大多数服务器不支持这种方式,因为这样会给服务器带来很大负荷.

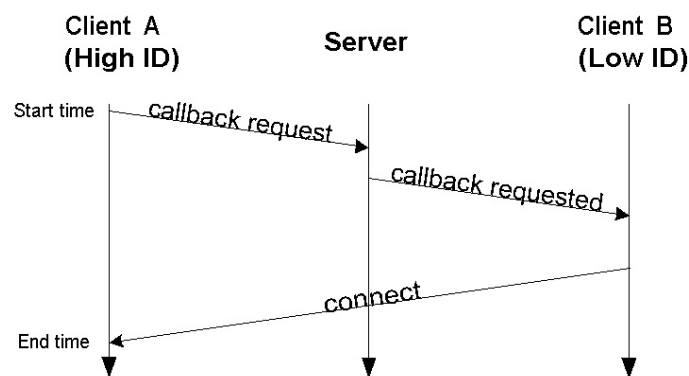


图 2.6： 回叫报文交换过程

3 客户端-服务器 UDP 通信

eMule 客户端-服务器之间使用不可靠的 UDP 连接增强搜索功能并增加更新能力.在 eMule 客户端所生成的数据包中,UDP 数据包占到约 5%的数量.这个百分比值因客户端的服务器列表中可用服务器数,下载文件可用源数,以及用户进行搜索的次数而变化.UDP 发包由一个以 100 毫秒为耗尽周期的计时器所控制,这也表明为何 eMule 客户端的相关专用线程会以最大 10 包/秒的速度发送 UDP 包.

3.1 服务器状态更新

eMule 客户端会周期性的对其服务器列表中的服务器状态进行更新.这个更新过程通过发送 UDP 服务器状态请求(见 6.3.3)和 UDP 服务器描述请求(见 6.3.7)完成.这里描述的服务器更新机制每小时不过生成几十个 UDP 数据包,它的最大限度为 0.2 包/秒.每当客户端查询服务器状态时,先发送一个服务器状态查询报文,服务器对每两次这样的查询报文响应一次,返回服务器状态描述.整个过程见下图 3.1.

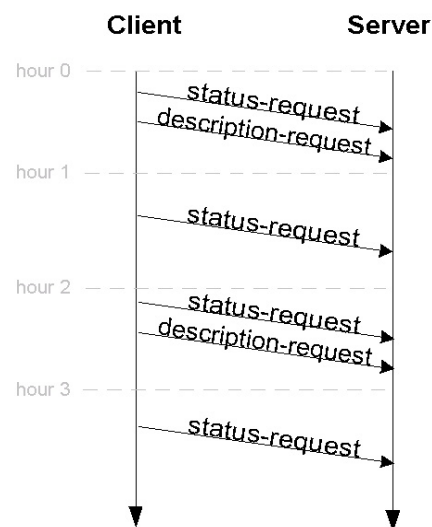


图 3.1 : UDP 服务器更新报文交换循环

在客户端发送的服务器状态更新请求中包含了一个随机生成数,这个数会包含在服务器响应的报文中.如果返回的这个随机数与发送时不同,则抛弃这个服务器响应报文.每发送一条服务器更新请求,客户端都会在一个叫做“尝试计数器”的计数器增加一.服务器的任何响应都会将这个计数器的值置零.一旦这个计数器的值达到一个设置的限度,客户端认为这个服务器无法响应,将其从服务器列表中去除.服务器应答包括如下一些数据项:服务器状态应答(见 6.3.4)包括服务器用户数和文件数,服务器软限制和硬限制数.服务器描述应答(见 6.3.8)包含服务器名称和一个简短的描述字符串.下图 3.2 描述了客户端-服务器间的更新报文交换过程.

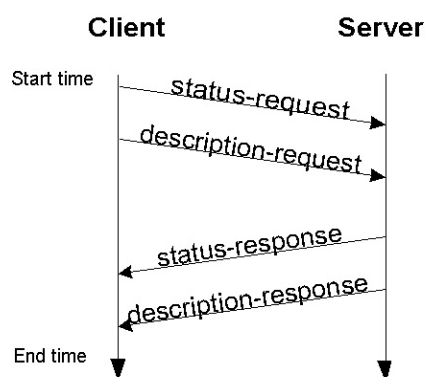


图 3.2 : UDP 服务器更新报文交换过程

3.2 文件搜索增强

eMule 客户端可以通过 UDP 增强文件搜索功能.UDP 搜索请求的报文格式与 TCP 搜索请求的报文格式(见 2.3)基本相同.服务器只有在它有搜索结果的时候才会应答.UDP 文件搜索请求包会发送给所有在客户端的服务器列表中的服务器.详见 6.3.5 和 6.3.6.

3.3 源搜索增强

当客户端的某个待下载文件的源列表中的可用源数量少于某设定值(通常为 100)时,客户端会周期性的发送 UDP 源搜索请求包,这些源搜索请求包发给搜索在这个客户端的服务器列表中的服务器.这些包通常每秒发送一次.从数量上讲,这是客户端 UDP 包发送流量的主要部分.源搜索报文(见 6.3.1)格式与 TCP 的源搜索格式几乎相同.一个需要注意的不同点在于,UDP 源搜索是独立于文件搜索的,它仅取决于某个待下载文件现有的源数量.

4 客户端-客户端 TCP 通信

eMule 客户端完成了在服务器上的初始会话,查询所需的文件和源后,此时,客户端需要和其他客户端建立连接以下载文件.每对文件-客户端都会建立一个 TCP 连接.当对方客户端终止连接或在一定时间内(默认为 40 秒)无活跃网络 Socket 时,TCP 连接会断开.

为了达到一定的下载速度,eMule 直到两个客户端之间的连接速度大于一个速度下限时才允许开始文件传输,这个值目前被设置为 2.4kbytes/秒.

4.1 初始化握手

握手会话是对称的,客户端-客户端连接的两端都向对方发送相同的信息.两个客户端交换诸如识别,版本号和性能等信息.在这个过程中,有两种报文参与.其一,是 Hello 报文(见 6.4.1),它是 eDonkey 协议的一部分并与 eDonkey 客户端兼容;另一个报文是 eMule 信息报文(见 6.5.1),它属于 eMule 扩展协议.下图 4.1 描述了两个 eMule 客户端之间的握手会话过程.在扩展信息中,包括有 UDP 报文交换,安全认证和源交换.

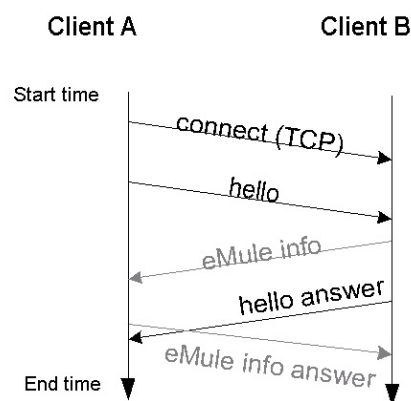


图 4.1 : 客户端初始化握手

4.2 用户身份认证

第 1.4 节简要的介绍了用户 ID 和信用体系,也提到了由此产生的用户 ID 冒名威胁.用户身份认证是 eMule 协议的一个扩展内容,只要客户端支持这种用户身份认证,它会在初始化握手之后立即完成.所以,使用用户身份验证正是为了防止用户 ID 顶替,整个验证过程按如下步骤进行:

1. 在初始化握手过程中,客户端 B 表示它支持用户身份验证并期望使用此功能.
2. 客户端 A 对 B 的报文进行回应,它发送身份验证报文(见 6.5.8),并表明是否需要 B 的公钥.这个报文还包含了一个等待 B 回应的 4 字节挑战.
3. 如果 A 需要 B 的公钥,B 将这个公钥发送给 A(见 6.5.9).
4. 客户端 B 根据 A 发送的挑战和一个附加双字生成签名,并按照签名报文(见 6.5.10)发送给

A.此处的附加双字是根据 B 或 A 的 IP 产生的,当 B 是 LowID 时,这个双字为 A 的 IP 地址,当 B 为 HighID 时,这个双字为 B 的 ID 值.下图 4.2 描述了身份验证过程.

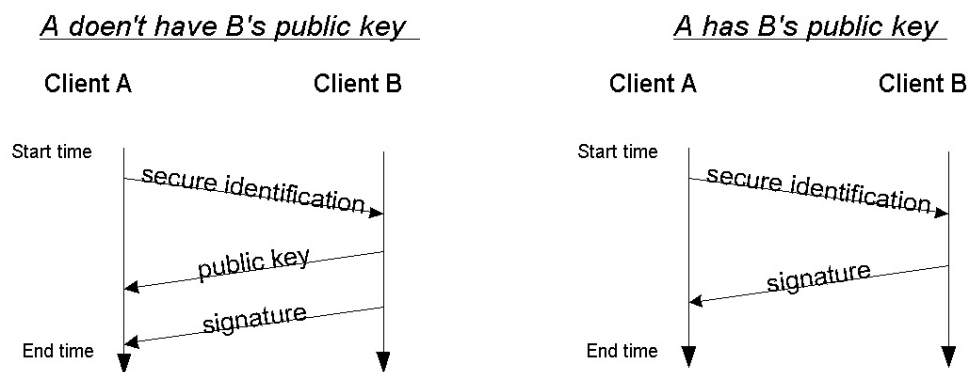


图 4.2：身份验证过程

4.2.1 信用体制

本段简述客户端的信用机制,如前所述,这种信用机制的主要目的在于鼓励上传.当一个上传客户端向一个下载客户端上传文件时,这个下载客户端会根据所收到的数据量增加对上传客户端的信用评价.此处的上传客户端信用并不是全局范围之内的,它的信用的增加只体现在当它下载此时这个下载客户端的某些文件时.也就是说,一个客户端的信用分别建立再各个它曾经上传过的客户端上.信用等级的计算如下:

1. $(\text{总上传量} \times 2) / \text{总下载量}$.

这个计算式的最大值取 10,当总下载量为 0 时,此值默认为 10.

2. $\text{Sqrt}(\text{总上传量} + 2)$.

当总上传量小于 1Mb 时,这个表达式最小值取 1.

总下载/上传量都以 Mb 为单位进行计算,按上边的公式,一个客户端的信用等级为一个取值范围为 1 到 10 的数.

4.3 文件请求

如前所述,按照 eMule 协议,会为每个客户端-文件对建立一个 TCP 连接.当连接建立以后,客户端立即发送一些请求报文,请求下载它期望获得的文件.一个典型的成功的文件请求过程如图 4.3 所示.



图 4.3：文件请求过程

4.3.1 基本报文交换

基本报文交换由四组报文组成,客户端 A 首先发送一个文件请求报文(见 6.4.18),紧接着发送一个请求文件 ID 报文(见 6.4.17).随后,客户端 B 发送文件请求应答(见 6.4.15)和文件状态(见 6.4.18)报文.这个过程可以看作是 A 和 B 两个客户端进行的一问一答的报文交换过程.

eMule 扩展协议在这个过程中还加入了一对源请求(见 6.5.6)和源请求应答(见 6.5.7)报文.这对扩展的报文交换可以将客户端 B 的源信息传给 A.为了加强客户端的协作功能,客户端 B 可以在尚未完成下载的时候就将它已经完成的文件部分传给客户端 A,即便客户端 B 仅仅下载了这个文件的一小部分.

4.3.2 未找到文件报文交换过程

当客户端 A 向客户端 B 请求一个文件时,若 B 的共享文件列表中并没有这个文件,B 将不发送文件请求应答报文,而是在客户端 A 发送文件 ID 请求后,立即发送一个未找到文件报文(见 6.4.16).详见下图.

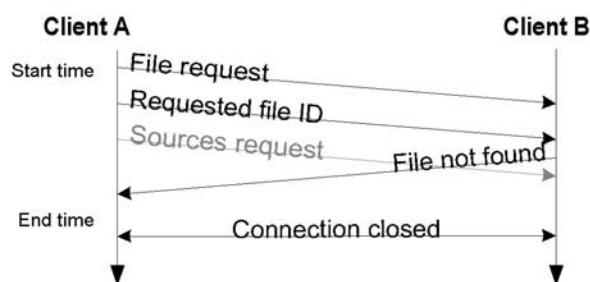


图 4.4：文件请求失败-未找到文件

4.3.3 加入上传队列

当 A 和 B 按照图 4.3 完成文件请求的握手对话后,B 客户端也有 A 客户端所需要的文件,然而此时 B 的上传队列并非为空,亦即是说有其他客户端正在下载 B 的文件.在这种情况下,B 会先将 A 添加在它的上传队列中,并发送给 A 一个队列排名报文(见 6.5.4),这个报文包含了 A 在队列中的位置以及 B 的上传队列的相关信息.详见下图.

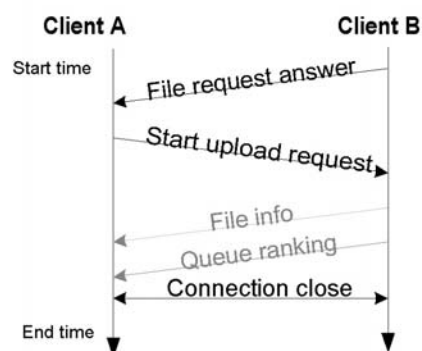


图 4.5：文件请求等待队列

4.3.4 上传队列管理

客户端为每个上传文件维护一个上传优先级队列,在这个上传队列中,待下载客户端的优先级是根据它在这个队列中的时间以及一个优先级参数共同计算的.在队列的顶端是优先级得分最高的客户端.具体的优先级计算公式为:优先级=(等级*在队列中时间)/100.此处的等级值有如下几个情况,若对方客户端为好友,则此值为无穷大,即表明好友的优先级为无穷大,直接到达队列顶端开始下载.若客户端为普通待下载客户端,等级的初始值为 100.若对方为被禁止的客户端,则此值为 0,使其永远无法到达队列顶端,拒绝其下载要求.对于普通用户而言,这个等级值随后根据客户端信用和下载文件本身的优先级变化,下载文件优先级取值范围为 0.2-1.8,上传用户可以对此进行调整.当一个待下载客户端比队列里的其他客户端具有更高的优先级时,它便开始下载.它将保持下载直到发生以下情况:

1. 上传客户端退出.
2. 下载客户端下载完毕.
3. 待下载队列中有其他客户端优先级超过了正在下载的客户端优先级.

为了可以让正在下载客户端完成至少一定量的下载,eMule 上传客户端会在下载开始前十五分钟将下载的等级调到 200 以防止频繁的下下载竞争.

4.3.5 达到队列顶端报文交换过程

当客户端 A 到达客户端 B 的上传队列顶端,B 将向 A 发出连接,进行初始化握手会话.接着,B 向 A 发送一个接受上传请求报文(见 6.4.11).此时,若 A 选择继续并下载文件,它会向 B 发送一个请求文件部分报文;若 A 已经获得了这部分文件,则它会向 B 发送一个取消传输报文(见 6.4.12).下图详细描述了这两种报文传递过程.

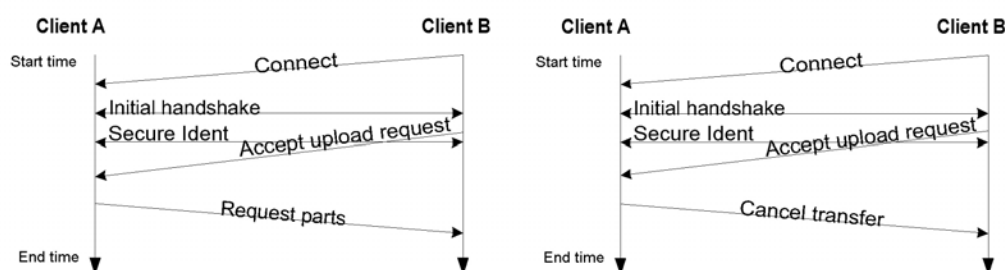


图 4.6：达到队列顶端报文交换过程

4.4 数据传输

4.4.1 数据包

eMule 网络活动主要在于发送和接受文件部分.一个文件部分的大小由 5000 到 15000 字节.为了防止产生文件碎片,一个文件部分报文被分开在一个 TCP 数据包的许多片断内.以 eMule0.30e 客户端为例,其最大片断大小为 1300 字节.这又就是说,尽管控制类的报文都在一个 TCP 数据包中完全包含,而有的时候数据报文是被分在几个 TCP 包中的.其中,第一个包中包含了传送部分的报文头(见 6.4.3),其后的包中就只包含数据.如果当文件被 1300 字节整除而有余数时,不足一个包的数据将和报文头一起放在第一个包中.详情见下图 4.7.

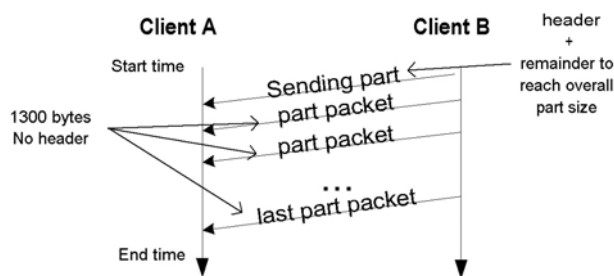


图 4.7 ：文件包传输过程

4.4.2 文件传输报文交换过程

文件传输报文交换在文件请求应答后立即开始,此时待下载客户端 A 发送一个启动上传请求(见 6.4.10),目标客户端会回复一个接受上传请求报文(见 6.4.11).紧接着,客户端 A 开始请求文件部分(见 6.4.4),随后客户端 B 向其发送被请求的文件部分(见 6.4.3).此处的文件部分请求可以请求最多 3 个文件部分,所以,这样的情况下,一个文件部分请求报文最多会导致 3 个文件部分的传输过程.

当上传下载客户端都支持 eMule 扩展协议时,文件部分可以压缩传输(见 6.5.3),同时,扩展协议还支持文件信息报文(见 6.5.5),这个报文会在接受上传请求前发送.文件传输报文交换过程详见下图.

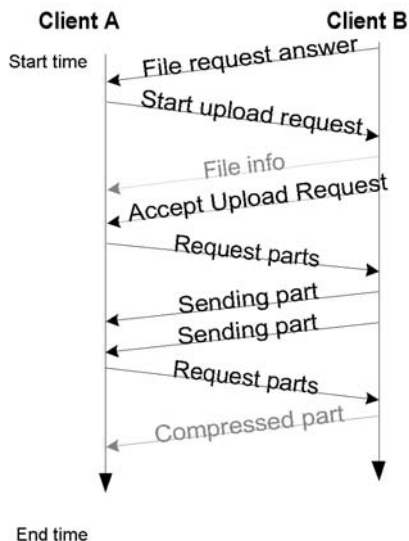


图 4.8 ：文件传输报文交换过程

4.4.3 下载选择

eMule 客户端会按照一定规则选择下载文件各部分的顺序,通过这样的机制,试图达到整个网络共享的优化.如前所述,共享的文件都按照 9.28Mb 为单位分割为文件部分,每个部分又被细分为 180Kb 的文件块.文件部分的下载顺序由下载客户端所决定,它在下载时发送文件部分下

载请求报文(见 6.4.4).下载客户端可以在任意已知的源处下载任意未获得的文件部分,一个部分内的各个文件块则从同一个源处获得.文件部分的下载顺序依照如下原则进行选择:

1. 按可获得文件块选择:极稀缺文件块应以最快速度下载,从而未其他客户端做源.
2. 用于预览和检查的文件块先下载.
3. 按请求要求:从不同源出按请求顺序下载.
4. 按完成速度:即将完成的文件块优先下载.

在衡量文件部分下载顺序时,引入频率参数.它可被划分为极稀缺,稀缺和普通.在这三种情况下,计算文件部分等级时,频率分别有不同的影响加权.文件部分的下载等级值越低则越先被下载.这个值的计算方法如下:

- 0-9999 – 被请求或未被请求的极稀缺文件部分.
- 10000-19999 – 未被请求的稀缺文件部分和预览文件部分.
- 20000-29999 – 未被请求的即将完成文件.
- 30000-39999 – 被请求的稀缺文件部分和预览文件部分.
- 40000-49999 – 被请求的普通文件部分.

这个算法总是选择最稀缺的文件部分进行优先下载,即将下载完成的文件部分也会被优先选择.对于普通文件部分而言,会在许多不同的源处下载.

4.5 查看共享文件

有两种报文流用于查看其他客户端的共享文件和文件夹.其一,查看共享文件报文(见 6.4.21)会在初始化握手后立即发送,随后,对方会发送查看共享文件应答报文(见 6.4.22).如果应答的客户端想要隐藏共享文件列表,则返回的应答中不包含任何文件(并不告知对方请求被拒绝).下图描述了查看共享文件报文交换过程:

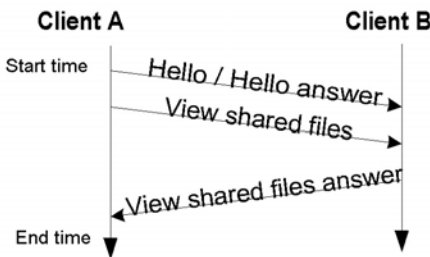


图 4.9：查看共享文件报文交换过程

第二种情况,报文流开始于一个查看共享文件夹列表请求(见 6.4.23),对方客户端将以共享文件夹列表(见 6.4.24)作为应答.随后,客户端发送查看共享文件夹内容报文(见 6.4.25),对方客户端则发送共享文件夹内容列表(见 6.4.26).下图 4.10 描述了这样的报文传递过程.

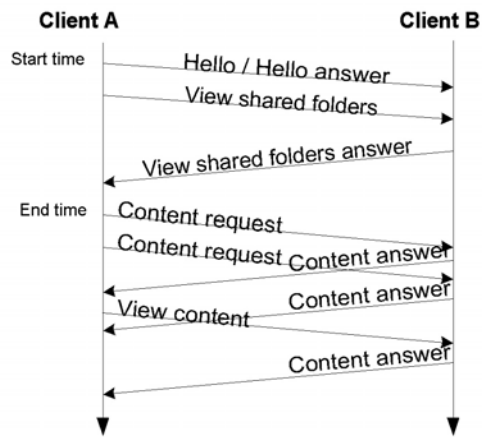


图 4.10：查看共享文件夹报文传递过程

如果对方客户端阻止查看共享文件和文件夹,则将返回一个查询被拒绝报文,见下图 4.11.

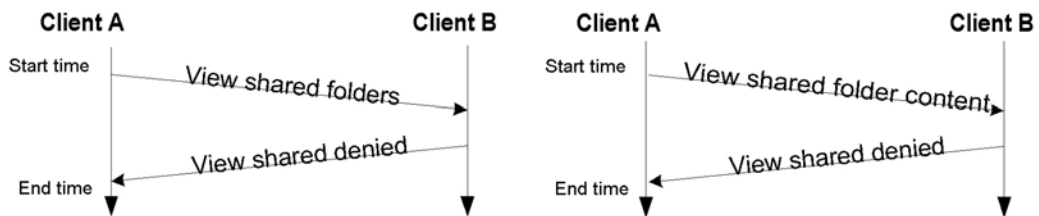


图 4.11：拒绝查看共享文件夹

4.6 交换文件 Hash

为了获得一个已下载文件部分的 Hash 值,客户端会向上传客户端发送一个 Hash 请求报文(见 6.4.8),上传客户端会发送一个 Hash 值回复(见 6.4.9).这个报文中包含了每个文件部分的 Hash 的集合.详见下图.

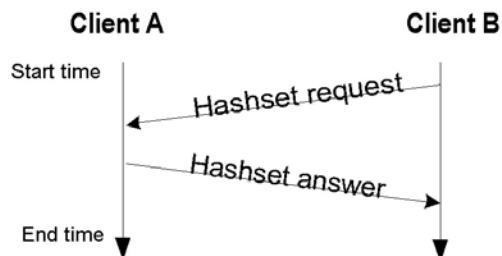


图 4.12：Hash 值请求报文

4.7 预览文件

客户端可以向其他客户端请求预览一个共享文件.总的来说,预览因不同的文件种类的差别和应用程序的不同而各有不同.eMule0.30e 版本仅支持图像预览.正如下图 4.13 所示,整个过程包含了文件预览请求(见 6.5.11)和文件预览应答(见 6.5.12).

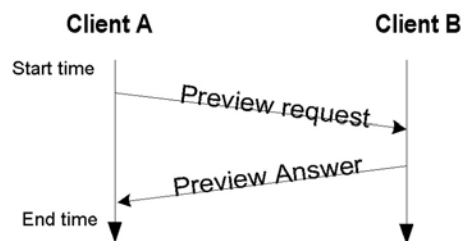


图 4.13：文件预览过程

5 客户端-客户端 UDP 通信

eMule 客户端周期性的使用 UDP 发送报文.在 eMule0.30e 中,UDP 报文仅用来查询客户端在其他上传客户端的队列中的位置.这是一个简单的查询-应答机制,首先由客户端发送一个询问文件报文(见 6.6.1).如下图所示,一共有三种不同的应答:

1. 队列等级 – 客户端在上传队列中的等级.
2. 队列已满 – 上传客户端的队列已满.
3. 未找到文件 – 上传客户端并没有这个文件.

这个询问报文约每 20 分钟向每一个将其加入上传队列的客户端发送一次.

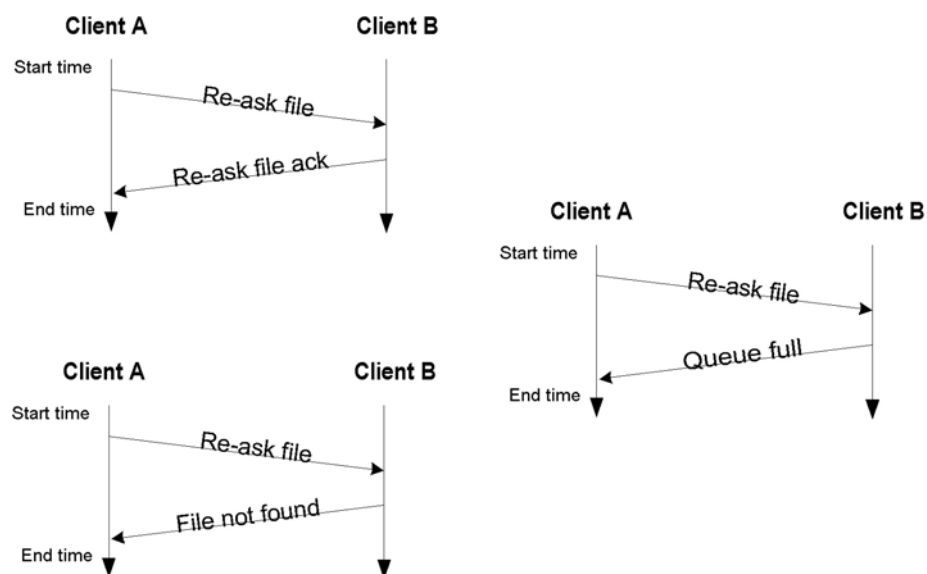


图 5.1：查询文件报文交换过程

参考文献

- [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>
- [5] NAT. <http://baike.baidu.com/view/16102.html?wtp=tt>
- [6] Little Endian & Big Endian. <http://dev.csdn.net/article/39/39864.shtm>
- [7] GUID. <http://baike.baidu.com/view/185358.html?wtp=tt>

6 附录 – 报文编码

6.1 报文编码

这节将讨论关于在 TCP/UDP 连接中报文编码的广义实现.

6.1.1 大尾 or 小尾

在 eMule 协议中用到的所有报文都以小尾(Little-Endian)形式而非大尾(Big-Endian)形式出现,这是网络上字节的习惯顺序.这也是因为客户端-服务器大多是基于使用 Windows 系统和 Intel 处理器的机器.

6.1.2 报文头

所有报文都包含一个 6 字节报文头,其结构由一下几个内容构成:

1. 协议 – 表示客户端使用的协议 - 0xE3 表示使用 eDonkey 协议 ,0xC5 表示使用 eMule 协议.
2. 大小 – 4 字节的报文大小描述 – 报文大小(不包含报文头),例如一个报文(见 6.4.11)除了报文头之外没有内容,则其大小为 0.
3. 类型 – 占 1 字节 – 记录一个独一无二的报文 ID.

6.1.3 报文标志

报文标志是和上边描述的协议,类型,大小结构类似的一种结构,它主要用于将可选数据添加在报文之后.报文标志分为几类,本节将罗列出每一个类型.文中描述报文标志时,只会提及标志类型,读者应将本节视同一个参考,用来查询每种标志类型的结构.

每个报文标志有 4 个域,并不是所有的报文都使用这 4 个域.

1. 类型 – 1 字节整型
2. 名称 – 1 字节整型或可变长度字符串.
3. 值 – 4 字节整型或 4 字节浮点型或可变长度字符串.
4. 特殊标志 – 1 字节整型,用于特殊用途

按照类型,报文标志可以分为字符型标志,整型标志和浮点型标志,分别编号为 2,3,4.当标记进行编码时,按照先类型再名称再值的次序进行编码,类型编码为一个字节,名称按照值方式编码为一个 2 字节的值.例如,一个整型名称 0x15 将会被编码为 0x01 0x00 0x15 的序列.对于值的编码而言,整型和浮点型值会按照他们本身的值编码,而字符型值会按照与名称相同的方式进行编码.

注意:这里的标志名称并不是协议意义上的名称,仅仅是为了方便描述后文协议中的报文而提出的.

6.2 客户端-服务器 TCP 报文编码

本节将具体介绍客户端-服务器之间传输的 TCP 报文。

6.2.1 登录

登录报文是在客户端-服务器之间建立了 TCP 连接后第一个被发送的报文。这个报文的长度依用户的信息而变化,比如用户昵称的长度就会影响这个登录报文的长度。

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		以字节计的报文大小,不计报文头和大小域本身
类型	1	0x01	操作码 OP LOGINREQUEST 的值
用户 Hash	16		详见 1.4 节
客户端 ID	4	0	第一次连接此值通常为 0,详见 1.3 节
TCP 端口	2	4662	客户端默认的 TCP 端口
标签数	4	4	此报文中的标签数
名称标签	可变	NA	用户昵称.此标签为字符串标签,名称为整型值 0x1
版本标签	8	0x3C	客户端版本号,整型标签,名称为整型值 0x11
端口标签	8	4662	客户端使用 TCP 端口号,整型标签,名称为整型值 0x0F
标识标签	8	0x01	整型标签,名称为整型值 0x20

6.2.2 服务器报文

服务器报文是一种变长报文,它在不同情况下由服务器发送给客户端。这个报文第一次发送发生在客户端登录请求之后。一个服务器报文通常由多个报文组成,它们由分行符分隔开。有一些特殊的服务器报文以“服务器版本”,“警告”,“错误”或“emDynIP”开头,这中报文对客户端有着特殊用途。而其他普通的服务器报文则仅仅简单的显示给用户。

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x38	操作码 OP SERVERMESSAGE 的值
大小	2	NA	报文剩余部分的大小
报文	可变	NA	由分行符分隔的服务器报文列表

特殊报文:

- 1.版本 – 通常用于成功的握手连接。
- 2.错误
- 3.警告 – 用于服务器拒绝连接或客户端为 LowID。
4. emDynIP.

6.2.3ID 更改报文

ID 更改报文是服务器发出的客户端登录报文的响应,表明服务器接受了客户端的连接.此报文大小为 14 或 10 字节,具体的大小取决于可选 TCP 连接位图.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x40	操作码 OP IDCHANGE 的值
客户端 ID	4	NA	详见 1.3 节
TCP 连接位图	4	0x00000001	目前为止,只有一个位有具体含义,这位为 1 时,表示服务器支持压缩传输

6.2.4 提供文件报文

这个报文是客户端用来描述本地可供下载的共享文件的报文.如果客户端有可以提供共享的文件,则这个报文在建立连接后会立刻传输.当客户端的共享文件列表发生变化时,也会传输这个报文.这个报文也用作更新,会周期性地发送给服务器.如果服务器支持压缩,这个报文会被压缩传输,但如果是用于更新用途,则不会压缩处理.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x15	操作码 OP OFFERFILES 的值
文件数	4	NA	共享列表中的文件数,这个数不超过 200,服务器也能设置一个更小的上限数
文件	可变	NA	可选的文件列表,单个项的描述见下

单个文件项格式:

下表描述的是单个文件项的格式,多个文件项会集中在一个提供文件报文中.

名称	大小(字节)	默认值	注释
Hash 码	16	NA	对文件内容的 Hash 结果,用于唯一的标志文件而独立于不同客户端的不同命名
客户端 ID	4	NA	客户端 ID 为 HighID 时的值
客户端端口	2	0x15	客户端的 TCP 端口,为 LowID 时此值为 0
标志数	4	NA	此域后的标志数
文件名标志	可变	NA	强制标志,整型,标志名为 0x1 的整型值
文件大小标志	8	NA	强制标志,整型,标志名为 0x2 的整型值
文件类型标志	可变	NA	可选,类型可为"Audio", "Video", "Image", "Pro" 或"Doc". 字符串型,标志名为 0x3 的整型值
文件格式标	可变	NA	可选,扩展类型,可为"zip","exe"等类型, 字符串型,

标志			标志名为 0x4 的整型值
媒体长度标志	可变	NA	当文件类型为 mp3 时,记录声音长度. 字符串型,标志名为"长度"
媒体比特率标志	TBD	NA	当文件类型为 mp3 时,记录比特率. 字符串型,标志名为"比特率"
媒体解码标志	可变	NA	当文件类型为电影相关类型时,记录解码信息. 字符串型,标志名为"解码"

6.2.5 获取服务器列表报文

此报文会在服务器-客户端初始化握手后由客户端发送,或者在客户端需要更多的服务器时由其发送.报文长度为 6 字节.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x14	操作码 OP SERVERSTATUS 的值

6.2.6 服务器状态报文

此报文有服务器发送给客户端,包含了服务器上的用户和文件数量.这些信息由客户端保存,并显示给用户.报文大小为 14 字节.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x34	操作码 OP OFFERFILES 的值
用户数	4	NA	目前登录此服务器的用户数量
文件数	4	NA	服务器获取信息的文件数量

6.2.7 服务器列表报文

此报文由服务器发送给客户端,告知客户端其他可用于扩展功能的服务器.报文大小因服务器列表中服务器数量而变化.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x32	操作码 OP SERVERLIST 的值
服务器数量	1	NA	服务器列表中服务器数量
服务器信息	服务器数量*6	NA	服务器的地址信息,每个服务器为 6 字节,4 字节 IP 地址,2 字节为 TCP 端口信息

6.2.8 服务器识别报文

此报文由服务器发送给客户端,包含了服务器的 hash(TBD),服务器 IP 地址,TCP 端口(这个端口号在实用协议连接时有用)以及服务器描述信息.报文的大小可变.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4	0x41	不包含报文头和大小域的报文大小
类型	1	NA	操作码 OP SERVERIDENT 的值
Hash	16	NA	服务器的 GUID,用于调试
服务器 IP	4	NA	服务器的 IP 地址
服务器端口	4	NA	服务器监听的 TCP 端口
标志数	4	NA	报文之后的标志数
服务器名称 标志	可变	NA	服务器名称, 字符串型, 标志名为 0x1 的整型值
服务器描述 标志	可变	NA	服务器描述字符串, 字符串型, 标志名为 0xB 的整型值

6.2.9 搜索请求报文

此报文由客户端发送给服务器,用于搜索由用户提供的的一个字符串所对应的文件.报文大小可变.在搜索字符串中可能会包含'AND' 'OR'或 'NOT'等布尔型字符.用户也许会为搜索进行定制,限定文件的类型,大小或者为文件搜索提供其他前提.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4	0x16	不包含报文头和大小域的报文大小
类型	1	NA	操作码 OP SEARCHREQUEST 的值
搜索字符串 文法	可变	NA	搜索字符串的文法在此声明
文件类型限制	可变	NA	可选,字符串类型的限制,可选的格式为"Audio", "Video", "Pro" 或"Image" 共包含 3 字节: 0x1 0x0 0x3
最小限制	可变	NA	可选,整型限制,以 Mb 描述,包含 4 字节: 0x1 0x1 0x0 0x2
最大限制	可变	NA	可选,整型限制,以 Mb 描述,包含 4 字节: 0x2 0x1 0x0 0x2
附加限制	可变	NA	可选,整型限制,设定一个拥有此文件的客户端数最小值,包含 4 字节: 0x1 0x1 0x0 0x15
文件名称扩展限制	可变	NA	可选,字符串类型的限制,共包含 3 字节: 0x1 0x0 0x3

文法化的搜索文件字符串格式:

语法化的搜索字符串由一个二进制表示树和布尔操作符'AND','OR','NOT'以及操作数共同组成.二进制表示树以前序表示,布尔操作符编码为2字节的整型,'AND','OR','NOT'的值分别为 0x0,0x100,0x200.字符串以 TLV 格式编码,类型域占一字节,其值为 0x1,长度域为 2 字节整型.如果搜索字符串是单字,则会被编码为一个单一的操作数而非操作符.在后期的 eMule 版本中,将仅包含有'AND'操作符的搜索字符串编码为一个单一的字符串,将'AND'用空格替代.这符合服务器的搜索文法,将不同的文字串默认用'AND'连接起来.

可选限制格式:

限制实际上是一系列的输入条件,以一个'AND'(2 字节,值为 0x00)描述符开始,接着是编码的限制内容.因此,整个搜索的格式为<'搜索字符串'AND 限制 1 AND 限制 2 ...>,编码的限制有如下几个域:

- 1.类型 – 一个字节描述了这个限制是字符串型(0x2)的还是整型的(0x3).
- 2.值 – 一字节长度的编码字符串或 4 字节长度的整型值.
- 3.限制类型 – 3 或 4 字节描述行限制的类型.

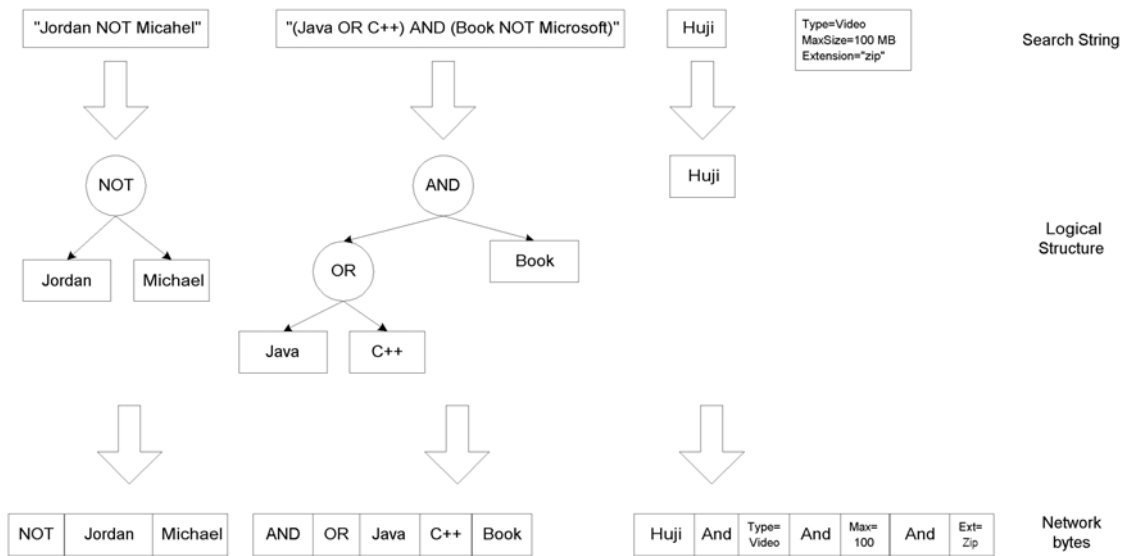


图 6.1：搜索字符串编码示例

6.2.10 搜索结果报文

此报文是由服务器作为对客户端文件搜索请求的回应,其大小可变,且经常以压缩方式传输.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x16	操作码 OP SEARCHRESULT 的值
结果数	4	NA	此报文中包含的结果数目
结果列表	可变	NA	一个结果的列表

搜索结果列表项格式:

下表将描述单一搜索结果的列表项格式,每个搜索结构都包括一个 Hash 值用于辨识文件并识别在其他客户端上共享的与其相同的文件.

名称	大小(字节)	默认值	注释
文件 Hash	16	NA	用于识别文件的 Hash 值
客户端 ID	4	NA	eMule 客户端 ID
客户端端口	2	NA	客户端的 TCP 端口号
标志数	4	NA	其后的属性标志个数
标志列表	可变	NA	标志列表

还有一些用于记录文件属性的标志,详见标志列表.请注意,这里的多数标志为可选标志,并且并不需要按照固定顺序排序.标志编码的详细信息见本章起始节.

名称	标志名	标志类型	注释
文件名	整型,0x01	字符串型	
文件大小	整型,0x02	整型	
文件类型	整型,0x03	字符串型	
文件格式	整型,0x04	字符串型	
源	整型,0x15	整型	此文件的可用源数
艺术家	字符串,"艺术家"	字符串型	
专辑名	字符串,"专辑名"	字符串型	
名称	字符串,"名称"	字符串型	
长度	字符串,"长度"	整型	
比特率	字符串,"比特率"	整型	
解码方式	字符串,"解码方式"	整型	

6.2.11 源请求报文

此报文有客户端发送给服务器,用于请求获得下载特定文件的源.报文的大小为 22 字节.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x19	操作码 OP GETSOURCES 的值
文件 Hash	16	NA	查询文件的 Hash 值

6.2.12 源查找结果报文

由服务器发送给客户端的源查找结果,报文大小可变.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x42	操作码 OP FOUNDSOURCES 的值
文件 Hash	16	NA	相关文件的 Hash 值
源数量	1	NA	此报文中包含的源数
源列表	可变	NA	源列表内容

源结果列表项格式:

下边的表格显示了源列表项的格式,每一个源列表项都包含了关于客户端的详细信息.

名称	大小(字节)	默认值	注释
客户端 ID	4	NA	共享该文件的客户端的 ID
客户端端口	2	NA	共享该文件的客户端端口

6.2.13 回叫请求报文

此报文由一个客户端发送给服务器,请求服务器通知另一个客户端回叫.这个报文是由一个 HighID 客户端发送的,它希望借此连接一个 LowID 客户端(详见 2.4).此报文大小为 10 字节.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x1C	操作码 OP CALLBACKREQUEST 的值
客户端 ID	4	NA	希望请求其回叫的客户端 ID

6.2.14 回叫成功报文

此报文有服务器发送给请求回叫的客户端.这条报文将会在目标客户端为 LowID 时发送,其大小为 12 字节.收到报文的客户端将会试图去连接报文中包含的 IP 和端口号.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x35	操作码 OP CALLBACKREQUESTED 的值
客户端 IP	4	NA	发起回叫的客户端 IP
客户端 TCP 端口	2	NA	发起回叫的客户端 TCP 端口

6.2.15 回叫失败报文

由服务器发送给要求回叫的客户端,通知申请回叫失败,此报文为 6 字节.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x36	操作码 OP CALLBACK FAILED 的值

6.2.16 拒绝报文

由服务器发送给客户端,通知其上次发送的报文被拒绝,此报文大小为 6 字节.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x05	操作码 OP REJECT 的值

6.3 客户端-服务器 UDP 报文编码

本节将讨论客户端-服务器之间进行的 UDP 通信,此处所指代的 UDP 报文头并不包含在 UDP L3 的报文头.其中的多数 UDP 报文都是较小的定长报文,由客户端周期性的发送给其服务器列表中的服务器.

6.3.1 请求源报文

当客户端某个文件源数量较少时,此报文由客户端以每秒为周期发送给服务器.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
类型	1	0x9A	操作码 OP GLOBGETSOURCES 的值
文件 ID 列表	NA	NA	文件 ID(Hash 值)列表

6.3.2 请求源回应报文

当服务器有客户端所请求的源时,服务器会向客户端回复一个请求源的回应报文.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
类型	1	0x9B	操作码 OP GLOBFOUND SOURCES 的值
文件源列表	NA	NA	文件源列表,见下表

源列表项格式:

一个列表包含了对于一个单一文件的所有源的信息.

名称	大小(字节)	默认值	注释
文件 ID	16	NA	所查询源的目标文件 ID
源数量	1	NA	查询到的源数量
源列表	NA	NA	一个源的列表,与 6.2.11 节中 TCP 列表一致

6.3.3 服务器状态请求报文

此报文由客户端每隔几秒发送给服务器,其中包含一个4字节的挑战,在此之后,服务器会对这个挑战进行回复.此报文一共6个字节,是UDP状态更新机制的一部分(见3.1节).

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
类型	1	0x96	操作码 OP GLOBSERVSTATREQ 的值
挑战	4	NA	发送给客户端的无符号整型挑战,用于服务器-客户端之间的“挑战-应答”机制,从而实现回复识别

6.3.4 服务器状态应答报文

服务器对于客户端发送的服务器状态请求 UDP 报文的应答,包含了一些相关的服务器信息.其中多数信息项是可选的.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
类型	1	0x97	操作码 OP GLOBSERVSTATRES 的值
挑战	4	NA	无符号整型,为客户端发送的挑战回复
用户数	4	NA	可选,登录在服务器上的客户端数量
文件数	4	NA	可选,在服务器数据库中包含的文件数量
软文件限制	4	NA	可选,无符号整型,服务器文件限制的软限制
硬文件限制	4	NA	可选,无符号整型,服务器文件限制的硬限制
UDP 标志位	4	NA	客户端,服务器 UDP 标志位,共定义有两个标签:0x01,表明服务器支持获取源报文,0x02,表明服务器支持扩展的文件报文

6.3.5 搜索请求报文

当客户端使用 UDP 进行搜索时,此报文将会由客户端发送給其服务器列表中的服务器.这个报文有两个不同的操作码,0x98 和 0x92.0x92 代表一个加强版的新的服务器版本.客户端将会根据文中 6.3.4 节所述的服务器状态应答中的 UDP 标志位的值来选择搜索请求的操作码.为了发送加强版的操作码,服务器必须打开 0x02 位,同时,也许需要从一个配置文件重载 UDP 标志位.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	

类型	1	0x98 或 0x92	操作码 OP GLOBSEARCHREQ 或 OP GLOBSEARCHREQ2 的值
搜索请求参 数	可变	NA	与 6.2.9 中的搜索请求参数相同

6.3.6 搜索请求回复报文

有服务器发送给客户端的搜索请求回复报文,它作为搜索请求的两个操作码的回复,它的格式与 TCP 连接中的搜索结果报文非常类似,只是不包含结果数量这一项.详见 6.2.10.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
类型	1	0x99	操作码 OP GLOBSEARCHRES 的值
结果列表	NA	NA	如 6.2.10 中描述的结果列表,但不包含结果数

6.3.7 服务器信息请求报文

由客户端每隔几秒发送给服务器一次,UDP ping 机制可见 3.1.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
类型	1	0xA2	操作码 OP SERVER DESC REQ 的值

6.3.8 服务器信息请求回复报文

由服务器发送给客户端作为服务器信息请求报文的回应,包含响应服务器的名称和描述,报文长度可变.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
类型	1	0xA3	操作码 OP SERVER DESC RES 的值
服务器名称	NA	NA	服务器的名称,字符串类型,2 字节
服务器描述	NA	NA	服务器的描述,字符串类型,2 字节

6.4 客户端-客户端 TCP 报文编码

本节描述客户端之间通过 TCP 连接发送的报文,这类报文可以分为 eMule 和 eDonkey 两类.

6.4.1Hello 报文

两个 eMule 客户端之间的握手过程中,该报文为第一个交换的报文,与 6.2.1 中的服务器登录报文非常类似,两者有相同的类型码(0x01),也都是整个协议中唯一拥有叠加类型码的报文.这两个报文以相同的顺序提供相同的数据.两者间有两个主要的区别:第一,客户端 Hello 报文以

用户 Hash 值大小域为起始,而服务器登录报文直接以用户 Hash 的值为起始;第二,客户端间的 Hello 报文以附加服务器 IP 和端口信息为结束而服务器登录报文没有此项内容.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x01	操作码 OP HELLO 的值
用户 Hash 值 大小	1	16	用户 Hash 值域的大小
用户 Hash 值	16		TBD
客户端 ID	4	0	TBD
TCP 端口	2	4662	客户端的 TCP 端口号,可重新配置
标志数	4	4	报文中的标志数量
标志列表	可变	NA	去除客户端属性后的标志列表
服务器 IP	4	NA	客户端连接上的服务器的 IP 地址
服务器 TCP 端口	2	NA	服务器监听的 TCP 端口

在标志列表中可能会有三种标志出现,其中端口标志为可选且通常不提供,标志的编码信息请见本章起始的内容.

名称	标志名称	标志类型	注释
用户名	整型,0x01	字符串	
版本	整型,0x11	字符串	
端口	整型,0x0F	整型	

6.4.2Hello 应答报文

此报文为 Hello 报文的应答,与 Hello 报文的报文域相同,仅是报文类型不同.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x4C	操作码 OP HELLOANSWER 的值
Hello 报文域			与 hello 报文域内容相同,以用户 Hash 开始

6.4.3 发送文件部分报文

此报文包含已下载文件的一部分.在报文头中的报文大小不仅表示报文本身的大小,还间接表示了文件的总大小.此报文被分为几个小于 TCP MTU 值(在 eMule0.30e 协议中,此值为 1300 字节)的数据包,关于发送压缩的数据包,详见 [6.5.3](#).

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x46	操作码 OP SENDINGPART 的值
文件 ID	16	NA	文件的 Hash 值
开始位置	4	NA	下载数据的开始位置
结束位置	4	NA	下载数据的结束位置
数据	NA	NA	实际的下载数据,可能会压缩传输

6.4.4 请求文件部分报文

由一个 Peer 客户端发送给另一个 Peer 客户端,用于请求下载文件部分.此报文最多能请求下载三个部分.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x16	操作码 OP REQUESTPARTS 的值
文件 ID	16	NA	文件的 Hash 值
部分 1 起始偏移量	4	NA	第一个文件开始的偏移量
部分 2 起始偏移量	4	NA	第二个文件开始的偏移量
部分 3 起始偏移量	4	NA	第三个文件开始的偏移量
部分 1 终止偏移量	4	NA	第一个文件结束的偏移量
部分 2 终止偏移量	4	NA	第二个文件结束的偏移量
部分 3 终止偏移量	4	NA	第三个文件结束的偏移量

6.4.5 下载结束报文

由下载客户端发送给上传客户端,通知一个文件的下载已经结束.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x49	操作码 OP END OF DOWNLOAD 的值
文件 ID	16	NA	文件的 Hash 值

6.4.6 改变客户端 ID 报文

当一个客户端更改服务器并因此获得一个新的客户端 ID 时发送。

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x4D	操作码 OP CHANGE CLIENT ID 的值
客户端 ID	4	0	新的客户端 ID
服务器 IP	4	NA	客户端连接的服务器 IP
服务器 TCP 端口	2	NA	服务器监听的端口

6.4.7 聊天报文

用于实现由 eMule 软件提供的聊天功能。

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x4E	操作码 OP MESSAGE 的值
长度	2	NA	报文的长度
报文内容	可变	NA	聊天报文内容

6.4.8 部分文件 Hash 请求报文

用于请求相关文件 Hash 以及文件中各部分的 Hash 值.关于文件 ID 和 Hash 值的内容详见 [1.5](#).

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x51	操作码 OP HASHSETREQUEST 的值
文件 ID	16	NA	被请求文件的 Hash 值

6.4.9 部分文件 Hash 请求应答报文

对于文件 Hash 请求的应答,包括全局 Hash 值(对所有部分)以及每个单独文件部分的 Hash 值.当这个过程结束后,客户端将会发送一个开始上传请求。

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x52	操作码 OP HASHSETANSWER 的值

文件全局 Hash	16	NA	一个由文件全局计算得到的 Hash 值
部分计数	2	NA	文件中部分的数量
部分文件 Hash 值	可变	NA	每个文件部分各自的 Hash 值,每个 16 字节

6.4.10 开始上传请求报文

文件开始上传请求,此报文将会初始化一个文件下载优先队列,关于文件下载优先队列的内容详见 [4.3](#).

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x54	操作码 OP STARTUPLOADREQ 的值
文件 ID	16	NA	请求上传文件的 ID

6.4.11 接受上传应答报文

此报文作为一个确认信号,表明开始上传请求报文被接受.此报文出了 eMule 的标准报文头以外没有其它域.它属于文件部分传输协议.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x55	操作码 OP ACCEPTUPLOADREQ 的值

6.4.12 取消传输报文

此报文用于请求取消传输,不包含任何特殊域.此报文发生的一个典型情况是:当一个客户端到达另一个客户端等待队列的最顶端时,下载客户端已经在其它客户端下载了这个文件部分,此时,将发送这个取消传输报文.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x56	操作码 OP CANCELTRANSFER 的值

6.4.13 无其它可用部分报文

此报文不用在 eMule0.30e 中,当一个客户端从一个上传客户端上完成所有相关可用文件部分的下载后,将会发送此报文给上传客户端并等待更多的文件部分.注意,上传客户端并不对此报文进行响应.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x57	操作码 OP OUTOFPARTREQS 的值

6.4.14 文件请求报文

此报文由一个客户端发送给另一个客户端用于请求下载一个文件.这个报文包含待下载文件 ID 以及一个用于指明已下载文件部分的状态域.

文件状态域:

eMule 协议允许下载客户端从一个客户端下载文件部分,即使这个客户端还没有完成整个文件的下载.文件部分状态域用于区别在上传客户端上不存在的文件部分和未完成的文件部分.如果此文件部分并不存在,则此域内容为 0;如果文件部分被下载,此域的前两个字节用于表示文件中已经完成下载的部分.最后一个字节是一个比特位图,它可以表示那些文件部分已经完成下载.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x58	操作码 OP FILEREQUEST 的值
文件 ID	16	NA	特定文件 ID
文件部分状态	3	NA	可选,如果 eMule 信息报文中的扩展请求版本大于零则此域可用,本段解释文件优先级.
源计数	2	NA	可选, 如果 eMule 信息报文中的扩展请求版本大于零则此域可用,表示目前可用的源数.

6.4.15 文件请求应答报文

此报文作为文件请求的应答,此报文仅是文件部分请求的一种可能的应答,详见 4.3 节.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x59	操作码 OP FILEREQANSWER 的值
文件 ID	16	NA	文件 ID(Hash 值)
名称长度	2	NA	文件名的长度
文件名	可变	NA	文件名

6.4.16 未找到文件报文

此报文作为文件请求的另一种应答,表明请求的文件或者文件部分不存在.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x48	操作码 OP FILEREQANSNOFIL 的值
文件 ID	16	NA	不存在的文件的 ID 值
文件部分状态	3	NA	详见 6.4.14

6.4.17 请求文件 ID 报文

文件下载请求报文,此报文仅由文件 ID 域.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x4E	操作码 OP SETREQFILEID 的值
文件 ID	16	NA	被请求的文件的 ID 值

6.4.18 文件状态报文

当客户端有被查询的文件时,本报文会被发送为请求文件 ID 报文的回复.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x50	操作码 OP FILESTATUS 的值
文件 ID	16	NA	被请求文件的 ID 值
文件部分状态	3	NA	详见 6.4.14

6.4.19 变更 Slot 报文

0.30 版本 eMule 并不发送本报文,收到后会自动丢弃.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x5B	操作码 OP CHANGE SLOT 的值

6.4.20 队列排名报文

eMule0.30e 版本并不发送此报文,收到后会自动丢弃.扩展的队列排序报文请见 [6.5.4](#).

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x5C	操作码 OP QUEUERANK 的值

6.4.21 查看共享文件报文

请求一个共享文件列表.仅当被查询客户端未禁止查询功能并不会在接收到此报文时丢弃该报文时,被查询客户端才会响应此请求.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x4A	操作码 OP ASKSHAREDFILES 的值

6.4.22 查看共享文件应答报文

此报文包含一个共享文件的详细列表,发送给查询的客户端.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x4B	操作码 OP ASKSHAREDFILESANSWER 的值
结果数	4	NA	此报文中包含的结果数目
结果列表	可变	NA	结果将和服务端上文件编码一致,详见 6.2.10

6.4.23 查看共享文件夹报文

此报文用于请求一个共享文件夹目录的列表.仅当被查询客户端未禁止查询功能并不会在接收到此报文时丢弃该报文时,被查询客户端才会响应此请求.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x5D	操作码 OP ASKSHARED_DIRS 的值

6.4.24 查看共享文件夹应答报文

此报文为查看共享文件夹报文的应答,包含一个共享文件夹列表.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小

类型	1	0x5F	操作码 OP ASKSHARED_DIRSANS 的值
结果数	4	NA	此报文中包含的结果数目
结果列表	可变	NA	一个文件夹列表,以下述的一种长度值格式表述

文件夹列表格式:

文件夹列表包含很多相同的项,每项包括一个整形(16 位)长度值和一个字符串值.

6.4.25 查看共享文件夹内容报文

此报文请求在被查询客户端上的某个共享文件夹的内容,此报文仅当该目录为共享目录并且查询客户端为一个可信客户端.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x5E	操作码 OP ASKSHARED_FILESDIR 的值
名称长度	2	NA	下一个域中目录名称的长度值
目录名称	可变	NA	共享文件夹的名称

6.4.26 查看共享文件夹内容应答报文

此报文作为查看共享文件夹内容报文的应答,包含一个文件名列表,此列表中表明所有此文件夹内的文件.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x60	操作码 OP ASKSHARED_FILESDIRANS 的值
名称长度	2	NA	下一个域中目录名称的长度值
目录名称	可变	NA	共享文件夹的名称
文件计数	4	NA	下一个域中文件名称的数量
文件列表	可变	NA	文件名列表

6.4.27 拒绝查看共享文件夹或内容报文

此报文用于拒绝查看共享文件夹请求或查看共享文件夹内容请求,它可以用来拒绝 OP ASKSHARED_DIRS 和 OP ASKSHARED_FILESDIR.

名称	大小(字节)	默认值	注释
协议类型	1	0xE3	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x61	操作码 OP ASKSHARED_DENIEDANS 的值

6.5 客户端-客户端 TCP 扩展报文编码

本节中所涉及的报文都非 eDonkey 的内容,而是 eMule 特有的.它们被称作扩展协议报文.

6.5.1eMule 信息报文

此报文传递了 eMule 客户端的基本信息,下边的标签列表介绍了 eMule0.30e 客户端的信息标签,其中的内容也许会和其后的版本有所不同.在一个客户端接收到此报文之后将会立刻回复以一个 eMule 信息应答报文(见 6.5.2).本报文作为客户端初始化握手过程(见 4.1)的一部分,包含了一些用以描述 eMule 客户端信息的标签,部分标签详解如下:

UDP 版本标签:

UDP 版本标签提供了客户端 UDP 版本信息,如提供关于客户端支持何种客户端-客户端 UDP 报文的信息.

标签描述:整形标签,名称为 0x22,eMule0.30e 版本中其值为 0x03.

源交换标签:

源交换标签表明客户端是否支持和其他客户端进行源信息交换.

标签描述:整形标签,名称为 0x23,eMule0.30e 版本中其值为 0x02 或 0x01.

评价功能支持标签:

评价功能支持标签表明客户端是否支持对文件进行评价.

标签描述:整形标签,名称为 0x24,eMule0.30e 版本中其值为 0x01(表明支持评价功能).

扩展请求标签:

扩展的请求标签表明客户端是否支持接收扩展协议中的文件请求报文.

标签描述:整形标签,名称为 0x25,eMule0.30e 版本中其值为 0x02.

扩展可选项标签:

在 eMule0.30e 版本中,扩展可选项标签分为两种用途,其一,低 6 位表明客户端是否支持安全辨识功能,第七位表明客户端是否支持文件预览功能.

标签描述:整形标签,名称为 0x27.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x01	操作码 OP EMULEINFO 的值
客户端版本	1	NA	eMule 客户端的版本

协议版本	1	0x01	eMule 协议的版本
标签计数	4	7	此域之后标签的数量
压缩标签	8	NA	表明客户端是否支持压缩功能 标签描述:整形标签,名称为 0x20,eMule0.30e 版本中其值为 0x01.
UDP 版本标签	8	NA	见上文
UDP 端口标签	8	NA	提供客户端使用 UDP 端口的信息 标签描述:整形标签,名称为 0x21,默认值为 4672.
源交换标签	8	NA	见上文
评价功能标签	8	NA	见上文
扩展请求标签	8	NA	见上文
扩展可选标签	8	NA	见上文

6.5.2eMule 信息应答报文

作为 eMule 信息报文的应答,与 eMule 信息报文包含相同的内容,也是客户端初始化握手过程的一部分.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x02	操作码 OP EMULEINFOANSWER 的值
eMule 信息域			与 eMule 信息报文中此域内容完全相同

6.5.3 发送压缩文件部分报文

用于发送压缩的文件部分,当一个文件部分以压缩的形式传输,此机制可以保证其与正常的文件传输过程一样进行.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x40	操作码 OP COMPRESSEDPART 的值
文件 ID	16	NA	文件的 ID 值
文件部分起始偏移量	4	NA	发送文件部分的偏移量
压缩部分大小	4	NA	压缩部分的大小

压缩部分内 容	可变	NA	压缩部分的实际内容
------------	----	----	-----------

6.5.4 队列排名报文

队列排名报文作为上传客户端发送给下载客户端告知其在下载队列中的排名.此报文通常在一个客户端加入某个文件的下载等待队列时发送,此报文通常都跟随着关闭 TCP 连接的操作.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x60	操作码 OP COMPRESSED PART 的值
队列位置	2	NA	客户端在此队列中的位置
缓冲	10	0	10 字节缓冲,用途未知

6.5.5 文件信息报文

一个包含文件描述符的报文,作为文件请求报文或文件上传请求报文的应答,此报文并不包括文件 ID 域.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x61	操作码 OP FILEDESC 的值
文件排名	1	NA	一字节,提供一些文件排名的方法
评论长度	2	NA	下一个域(评论的内容)的长度
评论	可变	NA	评论内容本身

6.5.6 源请求报文

源请求报文用于向其它客户端发起源请求.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x81	操作码 OP REQUESTSOURCES 的值
文件 ID	16	NA	请求的目标文件 ID

6.5.7 源请求应答报文

作为源请求报文的应答,本报文可以被压缩,未压缩时也许会大于 394 字节.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x82	操作码 OP ANSWERSOURCES 的值
文件 ID	16	NA	源请求应答对应的文件 ID 值
源计数	2	30	在 eMule0.30e 版本中,此值为 0
源列表	可变	NA	源列表,格式见下

源列表格式:

下表描述了源列表项的格式,每个源项都包括了关于某个拥有所请求文件的 eMule 客户端的详细信息.并且还包括了关于最后一次建立此连接的服务器的详细信息.

名称	大小(字节)	默认值	注释
客户端 ID	4	NA	拥有所请求文件的 eMule 客户端 ID
客户端端口	2	NA	拥有所请求文件的 eMule 客户端 TCP 端口
服务器 IP	4	NA	发现拥有所请求文件的客户端的服务器 IP 地址
服务器端口	2	NA	发现拥有所请求文件的客户端的服务器 TCP 端口
客户端 Hash	16	NA	可选 -仅当客户端-客户端都是支持源交换的版本才会启用此域

6.5.8 安全认证报文

常常在初始化握手之后发送(见 4.1),仅当两个客户端都支持安全认证功能时发送此报文.此保表明客户端是否有另一个客户端的公钥,并会随机生成一个值用于挑战-应答功能.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x87	操作码 OP SECIDENTSTATE 的值
操作	1	2	(2)表示有公钥信息并需要签名,(1)表示仅需要签名
挑战	4	NA	一个结果的列表

6.5.9 公钥报文

当双方客户端都支持加密时,用于发送客户端的公钥.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x85	操作码 OP PUBLICKEY 的值
公钥长度	1	76	公钥的长度
公钥	可变	NA	通常为 76 字节

6.5.10 签名报文

客户端用自己的公钥加密一个4字节的挑战,挑战由两部分组成,一个是随即生成字,另一部分为远程客户端的 IP(当本客户端为 LowID 时)或本客户端的 ID 值(当其为 HighID 时).

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x86	操作码 OP SIGNATURE 的值
签名长度	1	48	签名的长度
签名	可变	NA	通常为 48 字节

6.5.11 预览请求报文

请求预览某一种格式的图片.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x90	操作码 OP REQUESTPREVIEW 的值
文件 ID	16	NA	文件的 ID 值

6.5.12 预览请求应答报文

图片预览请求的应答,为文件预览或当请求的文件不为图片将返回文件 ID.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x91	操作码 OP PREVIEWANSWER 的值
文件 ID	16	NA	文件 ID 或为空
帧计数	1	NA	本图片的帧数
帧	可变	NA	图片以 4 字节编码的帧

6.6 客户端-客户端 UDP 报文编码

本节描述了客户端-客户端之间传递的 UDP 报文.有所使用 UDP 的报文都属于 eMule 扩展协议.所以协议 ID 是 eMule 而非 eDonkey 协议.

6.6.1 重请求文件报文

向一个客户端查询某个文件的状态.此报文当请求客户端不在被请求客户端的等待列表中而需要周期性的查询是否能够下载时使用.它有三种不同的回复.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x90	操作码 OP REASKFILEPING 的值
文件 ID	16	NA	被请求的文件的 ID
源计数	2	NA	可选,无符号整形,表达被请求文件目前所有的数量

6.6.2 重请求文件确认报文

对于重请求文件报文的一种回复,表示被请求客户端接受了重请求报文,并且在其下载队列中有空位同时赋予查询客户端一个特殊的优先级.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x91	操作码 OP REASKACK 的值
排名	2	NA	无符号整形,赋予查询客户端的队列中的位置

另一种对重请求文件报文的回复表示该文件未找到,但并不表明确切的文件信息.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x92	操作码 OP FILENOTFOUND 的值

6.6.3 队列已满报文

第三种对重请求文件报文的回复 ,不包含文件的详细信息.

名称	大小(字节)	默认值	注释
协议类型	1	0xC5	
大小	4		不包含报文头和大小域的报文大小
类型	1	0x93	操作码 OP QUEUEFULL 的值