  要解释清楚这个问题，我们得从当下最流行的音频视频p2p软件聊起。先来说说较为简单的一个音频p2p软件，酷狗。

    酷狗的原型应该是来自于一个国外的公司，名字我已忘记，那家公司也是通过mp3的p2p下载作为主要业务，不过可惜的是在美国mp3因为版权问题非常严重，所以那家公司的最终结局只有一个，就是关门。

    但是在中国就不一样了，版权问题没有这么严重，或者说相当的不严重，所以酷狗活得很好。这就是国情啊（理论上来讲，酷狗应该也有一些版权问题，可能跟版权商有合作关系，不过大多数的音频应该没有版权问题，这种情况和视频网站是类似的）。

    下面让我们来看看酷狗的技术实现。以下都是ahuaxuan的猜测，供大家讨论，未必完全正确，也未必完全错误，拿出来和大家探讨。

    酷狗应该是采用中心的目录服务器结构来实现p2p的功能，也就是说，所有的音频文件的基本信息都会注册到酷狗的中心目录服务器上，那么酷狗的客户端需要下载某个视频的时候，则从中心目录服务器上查找，找到相信的音频信息，每个音频信息都会对应一堆地址，这些地址是其他的拥有该音频的客户端ip。让我们用一张图来描述一下这个问题：

这样我们就可以虚拟一个流程出来：  
1．1号客户端请求中心目录下载服务器，要求下载“”。  
2. 中心目录服务器通过搜索引擎分词，查询之后，得到一堆id。  
3. 中心目录服务器根据id查找id对应的ip。显然一个id拥有多个ip，是1:n的关系。  
    （很不巧，这首歌2，3号客户机上都有， 当然这里并不是应该返回所有的ip地址，而是应选择最短路径的地址返回。让我们来怀念一下dijikstra。  
4. 中心目录服务器返回音频文件的ip列表。  
5. 1号客户机得到两个ip地址，然后分别2号机请求音频的第一段，从3号机请求音频的第二段。即多地址多线程分段下载。  
6. 1号机下载完成之后通知中心目录服务器，这样中心目录服务器关于这个视频又多了一个ip地址供其他客户端下载。

这个应该是最概要的流程，接着可以在这个流程上细化。

从上图我们可以看到，任何一个客户机既是client，又是server。作为client，它从其他server上下载数据，作为server，它提供数据给其他client下载。

    所以当我们开着酷狗听歌的时候，其实你的机器就变成了下载服务器了，同样，如果你用的是迅雷，而且一直不把迅雷关掉，那么你的机器就成为专职的下载服务器了。

    看到这里，我们有理由相信，如果掌握了下列技术，做一个酷狗不是什么难事，括号后面是ahuaxuan的选型。

1.    搜索技术(lucene)-服务器端  
2.    音频管理系统(java，同时涉及到缓存和数据库系统)-服务器端  
3.    客户端ui编程（最好是mfc之流）-客户端  
4.    下载服务器(c，对windows的io比较熟悉)-客户端

正如前面所说，这个只是非常高层次的设计，而且对于有过大型网站系统经验的人来说，1，2点是没有问题的。然后3，4点需要对c/c++比较熟悉的人来做，当然btcomet据说是python写的。所以我也在思考python+c实现客户端的可行性。

    上面讲到的是基本的整个软件的结构体系，或者称为“架构“，在high level层面还有一个问题，就是协议的问题，客户端之间相互下载应该使用说明协议，以及客户端和服务器端的交互应该使用什么协议，目前ahuaxuan 选择的是bt协议。利用成熟的协议可以减少很多的工作量。或者电驴的协议应该也不错，不过没有深入研究过。

tcp udp的区别  
文章分类:JavaEye   
TCP---传输控制协议,提供的是面向连接、可靠的字节流服务。当客户和服务器彼此交换数据前，必须先在双方之间建立一个TCP连接，之后才能传输数据。TCP提供超时重发，丢弃重复数据，检验数据，流量控制等功能，保证数据能从一端传到另一端。

    UDP---用户数据报协议，是一个简单的面向数据报的运输层协议。UDP不提供可靠性，它只是把应用程序传给IP层的数据报发送出去，但是并不能保证它们能到达目的地。由于UDP在传输数据报前不用在客户和服务器之间建立一个连接，且没有超时重发等机制，故而传输速度很快。

        UDP 与 TCP 的主要区别在于 UDP 不一定提供可靠的数据传输。事实上，该协议不能保证数据准确无误地到达目的地。UDP 在许多方面非常有效。当某个程序的目标是尽快地传输尽可能多的信息时（其中任意给定数据的重要性相对较低），可使用 UDP。ICQ 短消息使用 UDP 协议发送消息。   
许多程序将使用单独的TCP连接和单独的UDP连接。重要的状态信息随可靠的TCP连接发送，而主数据流通过UDP发送。

        TCP的目的是提供可靠的数据传输，并在相互进行通信的设备或服务之间保持一个虚拟连接。TCP在数据包接收无序、丢失或在交付期间被破坏时，负责数据恢复。它通过为其发送的每个数据包提供一个序号来完成此恢复。记住，较低的网络层会将每个数据包视为一个独立的单元，因此，数据包可以沿完全不同的路径发送，即使它们都是同一消息的组成部分。这种路由与网络层处理分段和重新组装数据包的方式非常相似，只是级别更高而已。   
为确保正确地接收数据，TCP要求在目标计算机成功收到数据时发回一个确认（即 ACK）。如果在某个时限内未收到相应的 ACK，将重新传送数据包。如果网络拥塞，这种重新传送将导致发送的数据包重复。但是，接收计算机可使用数据包的序号来确定它是否为重复数据包，并在必要时丢弃它。

TCP与UDP的选择

        如果比较UDP包和TCP包的结构，很明显UDP包不具备TCP包复杂的可靠性与控制机制。与TCP协议相同，UDP的源端口数和目的端口数也都支持一台主机上的多个应用。一个16位的UDP包包含了一个字节长的头部和数据的长度，校验码域使其可以进行整体校验。（许多应用只支持UDP，如：多媒体数据流，不产生任何额外的数据，即使知道有破坏的包也不进行重发。）   
        很明显，当数据传输的性能必须让位于数据传输的完整性、可控制性和可靠性时，TCP协议是当然的选择。当强调传输性能而不是传输的完整性时，如：音频和多媒体应用，UDP是最好的选择。在数据传输时间很短，以至于此前的连接过程成为整个流量主体的情况下，UDP也是一个好的选择，如：DNS交换。把SNMP建立在UDP上的部分原因是设计者认为当发生网络阻塞时，UDP较低的开销使其有更好的机会去传送管理数据。TCP丰富的功能有时会导致不可预料的性能低下，但是我们相信在不远的将来，TCP可靠的点对点连接将会用于绝大多数的网络应用。

QQ通信原理--  
我研究的是 QQ2007beta4 版本的通信原理，相信之后的版本暂时也不会有改动。。   
1 、登陆。不管 UDP 还是 TCP ，最终登陆成功之后， QQ 都会有一个 TCP 连接来保持在线状态。这个 TCP 连接的远程端口一般是 80 ，采用 UDP 方式登陆的时候，端口是 8000 。因此，假如你所在的网络开放了 80 端口（ 80 端口是最常用端口。。就是通常访问 Web 的端口，禁掉它的话，你的网络对你来说价值已经不大了），但没有屏蔽腾讯的服务器 IP ，恭喜你，你是可以登陆成功 QQ 的。   
2 、聊天消息通信。采用 UDP 协议，通过服务器中转方式。因此，现在的 IP 侦探在你仅仅跟对方发送聊天消息的时候是无法获取到 IP 的。大家都知道， UDP 协议是不可靠协议，它只管发送，不管对方是否收到的，但它的传输很高效。但是，作为聊天软件，怎么可以采用这样的不可靠方式来传输消息呢？于是，腾讯采用了上层协议来保证可靠传输：如果客户端使用 UDP 协议发出消息后，服务器收到该包，需要使用 UDP 协议发回一个应答包。如此来保证消息可以无遗漏传输。之所以会发生在客户端明明看到 “ 消息发送失败 ” 但对方又收到了这个消息 的情况，就是因为客户端发出的消息服务器已经收到并转发成功，但客户端由于网络原因没有收到服务器的应答包引起的。   
3 、文件 / 自定义表情传送。大家都知道， QQ 可以传送文件，可以发送自定义表情。先说官方表情。官方表情实际发送的是命令字，而没有发送表情。客户端收到命令字后，会自动解释为对应的表情。因此， QQ2007 正式版的客户端发出的新版表情，在 2007beta4 及以前的版本无法找到相对应的表情，就无法解释，看到的就会是空白信息，但查聊天记录就会有 [ 表情 ] 字样。自定义表情的传送是以文件传输方式进行的。下面说文件传输方式： A 要向 B 发送一个文件，于是发出一个文件传送请求。服务器收到这个文件传送请求后，转发给 B ，同时或者在 B 应答后，将 A 的 IP 地址同时发送给 B 。 B 这个时候就得到了 A 的真实 IP 。这里的 IP 是你的本机 IP 。也就是说，如果 A 处在内网， B 得到的地址就是一个内网地址。 B 得到了 A 的地址之后，就会尝试去连接 A 。如果 B 也处于内网，那么，显然 A 跟 B 之间的连接是无法建立的。这个时候，客户端就会请求服务器进行文件中转。因为服务器具有公网 IP ，处在内网的 A 跟 B 都是可以连接到服务器的，于是， A 跟 B 的文件传送就通过服务器中转的方式，顺利进行。（注：服务器文件中转使用 443 端口）   
“ 穿透内网 ” 的原理就是这样的。至于其中的程序细节，暂时还没研究到。。谁有空就研究研究吧，呵呵   
4 、小结。结合 1 跟 2 ，可以知道，如果一个局域网只开放 80 端口， QQ 是可以登陆成功的，也可以进行聊天。但传送文件也是不可以的，除非你们都在同一个内网。如果局域网还同时开放 443 端口，那么，恭喜你， QQ 的功能你都可以正常使用