即时通讯（Instant Messenger，简称IM）软件多是基于TCP/IP和UDP进行通讯的，TCP/IP和UDP都是建立在更低层的IP协议上的两种通讯传输协议。前 者是以数据流的形式，将传输数据经分割、打包后，通过两台机器之间建立起的虚电路，进行连续的、双向的、严格保证数据正确性的文件传输协议。而后者是以数 据报的形式，对拆分后的数据的先后到达顺序不做要求的文件传输协议。

QQ就是使用UDP协议进行发送和接收消息的。当你的机器安装了OICQ以后，实际上，你既是服务端（Server），又是客户端（Client）。当你登录OICQ时，你的OICQ作为Client连接到腾讯公司的主服务器上，当你看谁在线时，你的OICQ又一次作为Client从QQ Server上读取在线网友名单。当你和你的OICQ伙伴进行聊天时，如果你和对方的连接比较稳定，你和他的聊天内容都是以UDP的形式，在计算机之间传 送。如果你和对方的连接不是很稳定，QQ服务器将为你们的聊天内容进行中转。其他的即时通信软件原理与此大同小异。

**一般的步骤：**

首先，用户A输入自己的用户名和密码登录即时通讯服务器，服务器通过读取用户数据库来验证用户身份，如果用户名、密码都正确，就登记用户A的IP地址、IM客户端软件的版本号及使用的TCP/UDP端口号， 然后返回用户A登录成功的标志，此时用户A在 IM系统中的状态为在线（Online Presence）。

其次，根据用户A存储在IM服务器上的好友列表 （Buddy List），服务器将用户A在线的相关信息发送到也同时在线的即时通讯好友的PC机，这些信息包括在线状态、IP地址、 IM客户端使用的TCP端口（Port）号等，即时通讯好友PC机上的即时通讯软件收到此信息后将在PC桌面上弹出一个小窗口予以提示。

第三步，即时通讯服务器把用户A存储在服务器上的好友列 表及相关信息回送到他的PC机，这些信息包括也在线状态、IP地址、IM客户端使用的TCP端口（Port）号等信息，用户A的PC机上的IM客户端收到 后将显示这些好友列表及其在线状态。

接下来，如果用户A想与他的在线好友用户B聊天，他将直接通过服务器发送过来的用户B的IP地址、TCP端口号等信息，直接向用户B的PC机发出聊天信息，用户B的IM客户端软件收到后显示在屏幕上，然后用户B再直接回复到用户A的PC机，这样双方的即时文字消息就不通过 IM服务器中转，而是通过网络进行点对点的直接通讯，**这称为对等通讯方式（Peer To Peer）** 。在商用即时通讯系统中，如果用户A与用户B的点对点通讯由于防火墙、网络速度等原因难以建立或者速度很慢，IM服务器还提供消息中转服务，即用户A和用户B的即时消息全部先发送到IM服务器，再由服务器转发给对方。早期的IM系统，在IM客户端和IM服务器之间通讯采用采用UDP协议，UDP协议是不可靠的传输协议，而在 IM客户端之间的直接通讯中，采用具备可靠传输能力的TCP协议。随着用户需求和技术环境的发展，目前主流的即时通讯系统倾向于在即时通讯客户端之间、即时通讯客户端和即时通讯服务器之间都采**用TCP协议** 。

第一种：C1每次想和C2通信，先向Server递一个申请，然后Server同意，把信息转交C2,　 以后每次通信都这样 。这种方式对服务器的性能要求比较高，要求服务器可以同时处理很高的连接数，因为所有信息都要通过服务器进行传递，同时，它可以对所传递的信息进行控制。

C1向服务器发出请求

C2

Server

C1

同意C1与C2通信

将C1信息递交给C2

※第二种：C1第一次想和C2通信，向Server递一个申请，Server同意，告诉C1,C2,然后 C1和 C2之间建立了一条连接，可以直接通信，无需经过Server。这种仅仅是用户登陆或下线时与服务器进行连接，平时进行通信时，是用户之间直接进行点对点的连接，这种实现更合理一些。

Server

C2

C1向服务器发出请求

C1

同意C1与C2通信,发送C2的IP和端口

C1与C2通过TCP建立点对点的连接并开始通信

QQ聊天信息是在两个用户间直接通信的,而MSN要经过服务器中转。

QQ当用户登陆时，用户需要首先与qq服务器进行连接进行登录，服务器会向客户返回一些信息，比如你的好有的在线情况以及ip的信息，然后客户可以与你所请求的朋友进行点对点连接，两者互相通信。

**二、IM技术原理和工作方式**

典型的IM工作方式如下：登陆IM通讯中心（IM通讯服务器），获取一个自建立的历史的交流对象列表（好友列表），然后自身标志为在线状态，当好友列表中的某人在任何时候登录上线并试图通过你的计算机联系你 时，IM系统会发一个消息提醒你，然后你能与他建立一个聊天会话通道进行各种消息如键入文字、通过语音等的交流.

从技术上来说，IM的基本技术原理如下：

1.通过IM服务器登陆或注销；

2.用户A通过列表找到B，用户B获得消息并与之交谈；

3.通过IM服务器指引建立与B单独的通讯通道。

第一步，用户A输入自己的用户名和密码登录IM服务器， 服务器通过读取用户数据库来验证用户身份，如果验证通过，登记用户A的IP地址、IM客户端软件的版本号及使用的TCP/UDP端口号，然后返回用户A登录成功的标志，此时用户A在IM系统中的状态为在线（Online Presence）。

第二步，根据用户A存储在IM服务器上的好友列表 （Buddy List），服务器将用户A在线的相关信息发送给也同时在线的IM好友的PC机，这些信息包括在线状态、IP地址、IM客户端使用的TCP端口 （Port）号等，IM好友的客户端收到此信息后将在予以提示。

第三步是IM服务器把用户A存储在服务器上的好友列表及 相关信息回送到他的客户端机，这些信息包括也在线状态、IP地址、IM客户端使用的TCP端口（Port）号等信息，用户A的IM客户端收到后将显示这些好友列表及其在线状态。

返回登录成功标志（在线）

获取交流对象列表信息和在线状态

将Client1登记信息发送给Client2，以提示在线

向服务器发出登录请求

登记Client1的IP地址、客户端版本、TCP/UDP端口号

C1与C2通过TCP建立连接并开始通信

Client2

Server

Client1

**三、IM通讯方式**

1.在线直接通讯

如果用户A想与他的在线好友用户B聊天，他将直接通过服务器发送过来的用户B的IP地址、TCP端口号等信息，直接向用户B的PC机发出聊天信息，用户B 的IM客户端软件收到后显示在屏幕上，然后用户B再直接回复到用户A的PC机，这样双方的即时文字消息就不再IM服务器中转，而是直接通过网络进行点对点 的通讯，即对等通讯方式（Peer To Peer）。

A与B通过TCP建立连接并开始通信(P2P)

A向服务器发出请求

Client B

Client A

Server

取得在线客户端B的IP、端口

2.在线代理通讯

用户A与用户B的点对点通讯由于防火墙、网络速度等原因难以建立或者速度很慢，IM服务器将会主动提供消息中转服务，即用户A和用户B的即时消息全部先发送到IM服务器，再由服务器转发给对方。

A向服务器发送消息

Client A

Server

中转A和B的消息

接收从服务器发来的B消息

接收从服务器发来的A消息

B向服务器发送消息

Client B

3.离线代理通讯

用户A与用户B由于各种原因不能同时在线的时候，如此时A向B发送消息，IM服务器可以主动寄存A用户的消息，到B用户下一次登陆的时候，自动将消息转发 给B。

接收从服务器发来的B消息

A向服务器发送消息

Client B

Client A

Server

暂存A消息

接收从服务器发来的转存A消息

B上线后

4.扩展方式通讯

用户A可以通过IM服务器将信息以扩展的方式传递给B，如短信发送方式发送到B的手机，传真发送方式传递给B的电话机，以email的方式传递给B的电子邮箱等。

早期的IM系统，在IM客户端和IM服务器之间通讯采用UDP协议，UDP协议是不可靠的传输协议，而在IM客户端之间的直接通讯中，采用具备可靠传输能力的TCP协议。随着用户需求和技术环境的发展，目前主流的IM系统倾向于在IM客户端之间、IM客户端和IM服务器之间都采用TCP协议。

即时通讯相对于其他通讯方式如电话、传真、email等的最大优势就是消息传达的即时性和精确性，只要消息传递双方均在网络上可以互通，使用即时通讯软件传递消息，传递延时仅为1秒种.

**四、兴起的嵌入式IM工具。**

传统的IM在统治了互联网即时通讯领域长达十年之久，以其日趋稳定的定能，与较强的用户黏着度，至今仍统治着这个巨大的市场。然而，软件行业的技术精英们，并不满足于此。他们厚积薄发，一直致力于开发出性能更 为优越的即时通讯工具。当然，在功能上的不断完善，自然是一个必然的发展方向，在Web2.0时代，如何大力增强用户对网站的黏着度，而不仅仅是对于IM 的拥附，已经成为他们的主攻方向了。于是，嵌入式IM工具，应运而生了。

相对以往的传统的即使沟通工具，它们需要用户下载软件包，需要用户进行安装。对于拥有IM产品的网站而言，用户在登陆网站后，不能直接使用其IM工具，对于流量与用户的黏着度，都是有一定影响的。因此在IM 与网站相互依存的今天，没有哪家网络公司，愿意将IM工具孤立开来。

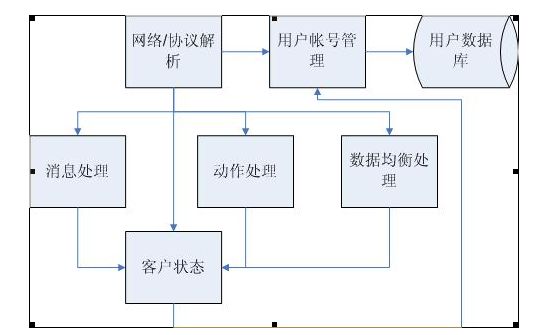
于是，目前，一种新型的嵌入式IM工具就应运而生了。这种IM工具，不需要下载安装，当用户登陆网页后，该IM直接嵌套在网页中，可以直接使用。

而在功能上，则一点也不输于传统的IM，无论是传统的文字沟通的速度与效率，还是近年来越来越成为IM工具必备的音频/视频功能，这种嵌入式IM都能提供非常稳定的传输。更值得一提的是，因为嵌入式IM是嵌套 在网页上的，软件供应商，可以根据网站需求，设计出适合网站风格的IM产品。而不是像传统的IM工具，千篇一律，毫无个性可言。

目前，这类嵌入式IM在社区、交友、社团及协作等类型的 网站上，应用已经较为广泛。在Web2.0时代，将发挥越来越重要的作用。

IM服务器的架构

1. **总的构架结构示意图：**



如上图所示，目前系统总的分成六个模块，分别为网络/协议解析模块，用户帐号管理模块，消息处理模块，动作处理模块，数据均衡处理模块，客户状态处理模块 。

正常流程应该这么实现，以一个或者几个线程运行网络/协议解析模块，然后他根据具体的包类型分发给具体的命令处理模块，每个具体的命令处理模块 至少应该分别运行于不同的线程。

从上面的结构图可以看出，其中客户状态模块和网络/协议解析模块都是公用模块，其他的模块几乎都依赖于这两个模块。目前因为很多功能不予以实现，例如不实现离线消息，所以只有用户帐号管理模块跟数据库相关。

**二、 每个模块的具体功能描述**

1． 用户帐号管理模块：

该模块管理基于用户帐号的所有操作：具体由如下：创建用户，删除用户，修改用户资料，修改用户密码，得到用户个人信息，得到好友列表，登陆，登出，请求认证串，创建群组，加入群组，离开群组。

以上所有操作，除了创建用户不需要登陆外，其他的任何操作都必须是已经登陆的用户，所以必须先有登陆操作，才能进行下一步的操作，这条规则针对所有的命令都是。

描述一下登陆过程：

客户端发送命令请求得到认证串——〉网络/协议解析模块把客户端请求以及它的用户标志转发给用户帐号管理模块 ——〉用户帐号管理模块自动生成一个随机认证串，同时把该客户端的用户标志一起保存在列表中（在列表中该数据只能保存一分钟，超过则删除，因此客户端必须 在一分钟内利用此认证串登陆），同时把认证串由网络/协议解析模块发送给客户端——〉客户端把用户名和认证串、用户名、密码组成的新串的MD5一并发给服 务器请求登陆——〉网络/协议解析模块把客户端请求以及它的用户标志转发给用户帐号管理模块——〉用户帐号管理模块首先检查列表中是否有该用户标志的数 据，如果有则继续——〉从数据库中取出用户名和密码，然后从列表中取出认证串，生成MD5，看是否一致，如果一致，则表示登陆成功，服务器生成会话加密密 钥，以后的所有与服务器的会话信息都采用此密钥加密和解密——〉网络/协议解析模块把结果发送给客户端，如果成功——〉网络/协议解析模块把该用户的用户 标志和会话加密密钥保存在客户状态模块中。

2． 网络/协议解析模块：

网络解析模块应该保存一个最近的登陆用户列表的缓存，该列表保存的信息应该有IP和端口，还有加密密钥， 最后一次访问的时间。在该列表的用户，如果超过一定时间（例如一分钟）没有访问，就要从该列表中删除。该列表的主要用途用于快速的查找登陆用户（这样频繁 访问服务器的用户登录检查就不需要丛客户状态模块中查询了，提高效率），如果在此列表中查找不到，则向客户状态模块查找，如果找到了，则把用户添加到此列 表中。如果找不到，则表示不在线。

任何时候收到用户的消息，则要更新登陆用户列表的缓存中的最后访问时间数据，如果该用户不在列表中，但是最后检查已经登陆，则要添加。还有心跳 命令不应该影响该列表的用户数据。但是心跳命令会影响客户状态列表。

网络解析模块应该保存一个用户标志映射列表（也可以直接组合他们的数据地址），把客户端的IP地址和端口映射到一个具体的唯一客户端标志值，然 后这个值以后作为这个客户端的标志，各个具体模块对客户端的任何操作都以这个标志为基础。

有四个功能

第一，根据不同的类型把数据报解密后（如果需要解密）分发给各个具体的处理模块。在分发前，首先检查该用户是否登陆，如果 没有，则直接告诉客户端一个错误。对于新建用户，登陆命令，获取认证字符串的命令不需要检查是否已经登陆。

第二，自己处理相关的消息，目前自己 处理的消息包括心跳消息和客户端请求建立直连的命令，还有任何数据包的格式不对或者版本不对的情况处理，则该模块自己就处理了。

第三，把用户发 送消息时的状态提交给客户状态模块，其中包括用户唯一标志。如果用户在发送消息时，地址已经修改了，那么唯一标志必须保持不变，但是用户标志列表中的用户 地址数据必须跟着修改。

第四，包各个模块发送给客户端的消息加密后（如果需要）处理发送给客户端。

3． 客户状态模块：

该模块的功能如下：

第一、 保存并操作所有当前在线的用户信息，包括用户的登陆时间，用户标志，用户状态（隐身等），用户帐号。提供给其他模块查询某个用户是否登陆。

第 二、 处理获取在线好友的消息，如果收到此命令，则通过用户帐号管理模块查找该用的好友列表，同时查找当前的在线用户列表，然后就得到在线好友列表。

第 三、 处理用户状态改变的命令，如果收到用户的状态改变消息（如登录、登出、隐身等），则查找该用户的在线好友列表，然后一一通知。

第四、 如果长时间没有收到用户的消息，则认为该用户因为意外的原因而断线，则自动认为登出。客户端在没有任何消息需要发送时，会定时的向服务器发送心跳命令，告 诉服务器客户端在线，如果连续一段时间没有收到心跳信息，则认为用户已经断线。

第五、 保存并操作在线群组信息。给其他模块提供获取特定群组或者场景中的用户列表功能。

4． 消息处理模块：

处理所有的消息转发，包括群消息，好友消息。

对于好友消息，该消息处理模块首先通过客户状态模块检查 好友是否在线，如果在线，则把好友消息发送过去，如果不在线，则不发送。

对于群消息，如果是针对普通的群组消息，则通过客户状态模块得到该群组 的在线用户列表，然后给每个人发送该消息，除了自己。

如果是场景中的公有消息，则先检查该用户所在场景的位置，然后计算出能够听到该用户的场景 中的其他用户的用户列表，然后给每个人发送消息。计算的算法参考每个场景用户的坐标，其中在特定场景中的用户不参与运算，除非在同一个的特定场景中的用户 才参与运算。

注意：对于场景消息，如果客户端不是以场景方式登录的，那么就不能发送和接收场景消息。

除了以上的消息，还有广告消息，系统消息之外客户端可以接受外，陌生人不能互发消息，如果服务器接受到此列消息，必须忽略，不发送任何应答。

5． 动作处理模块：

处理所有场景动作，该消息只有场景用户方式登录的才能发送和接收动作命令。服务器接收用户的动作命令，计算出用 户所在的场景中能够看到的人的列表，然后一一发送命令给该用户。计算的算法参考每个场景用户的坐标，其中在特定场景中的用户不参与运算，除非在同一个的特 定场景中的用户才参与运算。

该模块中保存有一个用户的位置和动作状态表（包括是否在某个特定的场景里面信息），保存场景用户的所在场景用户的坐标和最后一次的动作状态信 息。

场景用户登录后，位置每更新一次，都必须把该用户所在的场景位置坐标发送给服务器，对于心跳信息和动作信息不需要携带位置坐标数据。

6． 数据均衡处理模块：

该模块主要为了分布式处理，服务器负载平衡而设计，主要发出网络服务器重定向的命令，对于这些命令，所有客 户端都必须处理，因为一旦服务器发出了这个命令后，就在此服务器不再保存该用户信息了。

目前该模块不实现。