# 1概述

## 1.1 IM技术概念

IM技术全称Instant Messaging，中文翻译“即时通讯”，它是一种使人们能在网上识别在线用户并与他们实时交换消息的技术，是电子邮件发明以来迅速崛起的在线通讯方式。IM完全基于TCP/IP网络协议族实现。最早出现的即时通讯协议是IRC（Internet RelayChat），但是可惜的是它仅能单纯的使用文字、符号的方式通过互联网进行交谈和沟通。随着互连网变得高度发达，即时通讯也变得远不止聊天这么简单，自1996年第一个IM产品ICQ发明后，IM的技术和功能也开始基本成型，语音、视频、文件共享、短信发送等高级信息交换功能都可以在IM工具上实现，于是功能强大的IM软件便足以搭建一个完整的通信交流平台。目前最具代表性的几款的IM通讯软件有MSN、Google Talk、Yahoo Messenger、腾讯QQ等。

## 1.2 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．IM服务器

2．登陆或注销

3．用户A通过列表找到B，用户B获得的消息并与之交谈

4．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。

## 1.3 IM通讯方式

### 1.3.1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）。

### 1.3.2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。

### 1.3.3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。

### 1.3.4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

## 1.4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，以其日趋稳定的性能，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。

# 2即时通讯协议

IM(Instant Messaging)正在被广泛地采用，特别是在公司与它们的客户互动联接技术方案上。为了解决即时通讯的规范问题，IETF成立了专门的工作小组，研究和开发与IM相关的协议。

目前IM有四种协议：即时信息和空间协议(IMPP)、空间和即时信息协议(PRIM)、针对即时通讯和空间平衡扩充的进程开始协议SIP(SIMPLE)以及XMPP。PRIM与XMPP、SIMPLE类似，但已经不再使用了。

## 2.1 MQTT([消息队列](https://cloud.tencent.com/product/cmq?from=10680)遥测传输)

严格来说, MQTT是使用与物联网领域的消息传输协议，但有一些即时通讯系统也使用这个协议进行拓展开发。

MQTT主要有三个特点:

1. 基于发布订阅模式;

2. 为网络不可靠的环境下提供一套消息重传协议.

3. 基于TCP/IP, 消息载体轻, 耗电量小.

优势: 已实现消息丢失重传功能, 消息载体轻, 耗电量小.

缺点: 不是专用于聊天的协议, 登录, 单聊, 群聊, 加好友等功能都需要自己实现. 开发成本高，如要支持多个平台, 每个客户端都需要定制，IM方面的开源社区不活跃，技术文档少。

跨平台: 差, 每个客户端都需要实现MQTT的聊天协议。

## 2.2 XMPP（可扩展消息与存在协议）

是一个成熟开源的即时聊天协议, 基于XMPP的开源技术有openfire(后台端), spark(客户端)

优势: 比较成熟的聊天协议,已在XMPP协议内支持单聊, 群聊, 加好友等功能.满足基本的需要.

缺点: 基于xml 实现的通讯协议, 消息载体比较重, 增加网络流量, 定制困难，需要了解openfire原理, 改造有一定的难度。

跨平台: 一般, 仅基于java平台, 如要实现web 等需要自己使用socket进行xml封装.

## 2.3 ****自定义协议****

可以基于WebSocket， socket.io， 甚至常用的消息队列: RabbitMQ， RocketMQ 等长连接框架上加入聊天的业务，比如登录，单聊，群聊，加好友等功能。

优势: 基于自己熟悉的技术栈，易上手。

缺点: 单聊，群聊，加好友等功能都需要自己实现。

跨平台: 一般，若选用广泛使用于客户端的WebSocket，socket.io，能很好的实现跨平台性。

# 3 WebIM实现技术

## 3.1基于插件的技术

如ActiveX，插件相对稳定，但插件需要用户自己允许并下载安装，而大多数用户担心安装了黑客软件或插件对计算机系统不好而不愿意安装，

并且上网助手等软件也拦截插件，导致很多用户无法使用。另外，ActiveX受平台限制，只能在IE下使用。

## 3.2基于Flash的技术

典型的如 Yahoo web messenger，结合 Flash 和 Ajax，

## 3.3纯粹的基于HTTP的技术

前端使用 Ajax 的 Web IM : meebo, ebuddy, ILoveIm, MSN Web Messenger, KoolIM 等。

后台使用 comet 的 Web IM : meebo, gtalk 等。

只有 ebuddy 支持 wap 。

# 4服务器端实现方式

优点

循环的面向连接的方式 调试、编程、维护、修改容易

1. 服务器一次只能处理一个客户机的请求 ，可能造成客户

端请求的拒绝或响应变慢；

2．客户机无法知道服务器是否收到了其它客户机发给它的信息； 3.服务器对客户机的处理是按照连接请求的先后顺序进行，与一个客户机的通讯故障可能对后续客户造成影响。

缺点

多进程并发方式并行化、简单有序、互不干扰、事务化

1.创建进程需要一定的时间开销；

2.考虑如何实现多个进程之间的互不干扰；

3.占用较多的系统资源

# 5.即时通讯应用技术栈