

# Research and Implementation of dynamic DNS-based Enterprise Instant Messaging System

Nan Zhao

Department of Information Engineering  
Beijing University of Posts and Telecommunications  
Beijing, P.R.China  
sunflower47@gmail.com

LuMing Chen

Department of Information Engineering  
Beijing University of Posts and Telecommunications  
Beijing, P.R.China  
lmchen@bupt.edu.cn

Gang He

Department of Information Engineering  
Beijing University of Posts and Telecommunications  
Beijing, P.R.China  
brainhe@bupt.edu.cn

ZhiMin Yang

Department of Information Engineering  
Beijing University of Posts and Telecommunications  
Beijing, P.R.China  
010-62283123

ZhenMing Lei

Department of Information Engineering  
Beijing University of Posts and Telecommunications  
Beijing, P.R.China  
lzm@kuanguang.com.cn

**Abstract**—Instant messaging (IM) has infiltrated the enterprise in a big way. It is a fast and inexpensive way to communicate with people who are located elsewhere in the enterprise. There are two dominant standards (SIP/SIMPLE and Jabber/XMPP) and various proprietary IM solutions (MSN, Yahoo, AOL, etc). In this paper the existing standards and peripheral protocols are introduced and studied, and then special considerations on deploying IM services in enterprises are discussed. Moreover, taken into account of the dynamic nature of central Web server's IP address, we add the dynamic DNS functionality into our IM system. Finally, our design and implementation of an enterprise instant messaging system, which is based on the former study and analysis of current IM protocols, is introduced.

**Keywords**—Enterprise Instant Messaging (EIM); SIP/SIPLE; Jabber/XMPP; MSN; group; dynamic DNS

## I. INTRODUCTION

The computer industry has never seen a phenomenon like instant messaging. Instant Messaging (IM) has grown dramatically in popularity among both consumer and corporate users. IM has the potential to enhance productivity and cut costs by accelerating workflow, improving employee performance, reducing telecommunication expenses, and bridging communication gaps between departments and sites. IM makes the real-time enterprise a reality. Industry luminaries see IM technology as the foundation for a new generation of collaborative applications. Moreover, EIM

makes the instant messaging solution as a part of company identity.

In addition, EIM is able to provide converged services integrating video conferencing, text messaging, whiteboard, and VoIP altogether.

However, business-class, enterprise-grade IM is not the same as the popular consumer application—a true enterprise instant messaging (EIM) solution will maintain or even increase the benefits received from consumer IM, while mitigating or eliminating the risks. [8] In addition to core IM and text conferencing capabilities, enterprises also need the ability to control the operation and content of their IM and extend presence and messaging into other applications. [8]

Furthermore, the world of the Internet has become a much more dynamic place. Most large corporations use DHCP to assign IP addresses dynamically. Nearly all ISPs assign addresses to dialup and cable modem customers using DHCP. So enterprise instant messaging system have to adapt to this environment and offers dynamic DNS functionality.

In this paper, we describe first our study and analysis of existing IM protocols, then the special considerations required for deploying an EIM, and finally the design and implementation of our EIM.

## II. IM PROTOCOLS

The leading protocols for enterprise instant messaging (EIM) are the Extensible Messaging and Presence Protocol (XMPP) and Session Initiation Protocol (SIP) for Instant Messaging and Presence Leveraging Extensions (SIMPLE).

### *SIP/SIMPLE*

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls. [1] A session stands for the exchange of data among SIP participants. Data transmission is obtained through TCP or UDP, depending the size of data.

Proxy server is a fundamental element in SIP infrastructure since it is responsible for receiving requests, routing requests to the user's current location, authenticating, implementing policies, and providing features to users. As for users, they are able to upload their current locations to proxy servers through the registration function of SIP. Moreover, proxy servers can help discover the current host(s) at which the destination user is reachable.

SIP can provide user location, availability, capabilities, session setup and management for establishing and terminating multimedia communications:

By using the MESSAGE method, an extension to the SIP is able to facilitate the transfer of Instant Messages whose content is in the form of MIME body parts. However, this IM session is quite different from the session model of SIP. The difference lies in the fact that for one thing, IM session is a media session initiated with an INVITE transaction and terminated with a BYE transaction whereas the session of SIP is an explicit conversation with a clear beginning and end; for another, no dialogs are established in IM sessions while SIP's session model include dialogs. An IM request usually contains following header fields: MESSAGE, Via, Max-Forwards, From, To, Call-ID, CSeq, Content-Type, and Content-Length. The IM mode of SIMPLE is peer-to-peer.

The exchange of presence information is accomplished through the use of SUBSCRIBE and NOTIFY methods. And a new logical entity is introduced in SIMPLE—presence agent (PA), which is responsible for accepting subscriptions, storing subscription state, and generating notifications when there are changes in presence. [4] Meanwhile, a Presence User Agent is capable of manipulating presence information for present entity. Presence information is pushed into the presence system by PUAs. But PUAs do not receive SUBSCRIBE messages nor do they send NOTIFY messages.

### *Jabber/XMPP*

Jabber/XMPP is an open protocol based on Extensible Markup Language [XML]. Usually it is implemented as client-server architecture and the exchange of messages and presence information is through TCP connections. Security is enhanced by use of TLS and SASL.

A Jabber/XMPP IM system consists of three parts: Jabber servers, Jabber clients and gateways. Gateways are used to

translate XMPP into other protocols such as SIMPLE, MSN, AOL, etc, as well as to translate back.

XML streams and XML stanzas are used in order to exchange structured information rapidly. XML streams starts with an opening XML <stream> tag and ends by a closing XML </stream> tag. In between, there are presence information, messages as well as other useful information which are all tagged with different XML stanzas such as <presence>, <message>, <body>, <iq>, <show>, etc.

Compared with SIP/SIMPLE, Jabber/XMPP is a mature IM technology which offers type identification for message and presence status. As for the message types, they are categorized into five groups: single messages, chat messages, chat room messages, headlines and error messages. In addition, the child elements of message stanza includes <subject/>, <body/>, and <thread/> which is used for tracking an IM session. As for the presence types, they are divided into seven types: unavailable, subscribe, subscribed, unsubscribe, unsubscribed, probe (generated by a server) and error. Moreover, presence stanza has three child elements--<show/>, <status/> and <priority/> which offer a detailed description about the presence state of certain user. To illustrate, four types of tags can be inserted between the show tags to indicate the particular availability status of an entity. These four tags are "away", "chat", "dnd (Do Not Disturb)" and "xa (extended away)". <status/> is usually used with <show/> together to offer a verbal description of the availability state. As for the state of presence subscription, there are four primary states: none, to, from, both which are the attributes of the <item/> element. Thus with the help of these tags demonstrating the specific state of presence information, Jabber is able to offer a well-established presence-aware system. However, the overload of detailed presence information may become a threat to the network traffic.

### *MSN*

In an IM system taking advantages of MSN protocol, there are various distributed notification servers (NS) and switchboard servers (SB) which are two primary elements of the MSN-based IM system. NS is responsible for providing a presence service (handling presence information of all the participants involved in an IM session) while SB is capable of offering an instant messaging service (handling IM sessions between principals). In other words, MSN chat is accomplished via client-server architecture and each person corresponds to a TCP connection in a shared switchboard session.

MSN protocol is characterized by using distinctive commands to establish, handle and terminate IM sessions. To demonstrate, **VER** command notifies the server which protocol versions the client supports; **CVR** command is used for sending client's version information and operating system information to the server; after negotiating with MSN protocol, the server will send a **USR** command and then the client will send a final **USR**; **MSG** command is used for sending messages; **CAL** and **JOI** commands are used for inviting a principal to a session and joining the session; **ACK** and **NAK** commands are sent back from the switchboard to tell the client

if the messages sent was successfully received by its recipients or not.

### III. CONSIDERATIONS ON DEPLOYING EIM

Which protocol to use? The SIP protocol can manage voice, video, application sharing, and more; Jabber is a more mature and open technology; MSN is probably the most popular public IM protocol. However, neither of them can fully meet the requirements for an enterprise IM solution.

Connectivity to public IM networks. The opportunity to talk to friends and families via IM might represent a threat to productivity,"

The management of extended presence. For one thing, Sound balance should be kept between considerable traffic volumes (bandwidth consumption) and effectiveness of presence information. For another, enterprise may require a permissions model for extended presence. Not everyone is authorized to know the presence status of anyone else.

Security issues. Risks such as stealth of information, port sniffing, worms and viruses exit.

### IV. THE DESIGN AND IMPLEMENTATION OF OUR EIM SYSTEM

#### *OUR DESIGN GOALS:*

The functionalities offered in our EIM system are as follows: the creation, deletion and web-based management of instant communication groups as well as temporary groups; single point-to-point chatting; multicasting for sending announcements to persons in a certain work group; single point to multi point chatting (chatting in a meeting mode); peer-to-peer file transferring; peer-to-peer audio streaming; support for dynamic domain name.

#### *GENERAL DESIGN AND IMPLEMENTATION CONSIDERATIONS*

As far as the special features and requirements of enterprise instant messaging are concerned, we introduce a concept of *group* which is independent of the concept of department in an enterprise. There are two types of groups: instant communication groups and temporary groups. The former one is set up by the administrator of the EIM system according to the work division and different sections of the company, while the latter one can be set up by any person using this system. Therefore, people are encouraged to chat with people who are from different department and usually have distinctive background and expertise. To illustrate, people in a company are not restricted to communicate only to the people in their department, but can talk to anyone that have business connection with them or anyone who share the same interest with. For instance, a technical staff in the tech-section of the company can communicate with a sales man in the sales department through our EIM solution.

Moreover, in order to make the best of JAVA technology and the convenience of web applications, we implement the user interface of this EIM as an applet which is downloaded to the browser as a plug-in. The applet is digitally signed in order

to accomplish the file transfer and audio functionalities. The `Top_applet` demonstrates all the presence information about the colleagues, instant communication groups, temporary groups as well as the public notices and announcements. When double clicking one person or group whose presence information shows that it is available right now, a pop-up frame will show, which serves as the container for any further instant messaging.

As for the file transfer and audio streaming, they are both implemented via peer-to-peer mode in order to alleviate the burden of central server and take the advantage of P2P technology.

Furthermore, this system is implemented via client-server architecture, and the central server is responsible for exchanging presence information and instant messages.

As for the security issues, we use triple-DES to encrypt and decrypt the primitives sent in the network.

#### *Protocol design and Implementation*

The design of this EIM system is based on the former study of current IM standards and proprietary protocols. Also certain adaptation and modification is made in order to meet the need of an enterprise IM. Since this EIM system is special for its *group* concept, the protocol design is centered on *group*, providing presence and instant messaging for different types of groups.

SIP/SIMPLE, based on peer-to-peer architecture and UDP, is able to provide converged services and connect seamlessly to the VoIP telephony world; Jabber, based on client-server architecture and TCP, uses XML-streaming technology for data exchange. Each of them has its own features and advantages. As for an enterprise instant messaging system, its local network can ensure most of the traffic and since relatively higher control over the instant messages as well as the presence information is required, so we use client-server architecture and UDP for our EIM.

Moreover, in order to make the best of the VoIP feature provided by SIP, we use jain-sip, an open source java implementation of RFC 3261 for peer-to-peer audio. Also, we learn integration details from sip-communicator, an application that allows audio/video conversations over the Internet through the SIP protocol.

UDP is used for the exchange of primitives between the clients and servers. Primitives are strings which contain the protocol identifier, command, useful information for identifying sessions and the messages being sent. The format of the primitive is similar to that of Jabber and MSN, but without the XML stanza. A typical primitive is like:

```
eim command length callerid receiverid groupid port  
[messages]
```

But the elements of primitives may vary according to different commands and situations. Frequently-used commands include: login, logout, group, tempgroup, host, thost(temphost), pp\_connect\_caller, pp\_connect\_receiver, pp\_content, pp\_disconnect, pp\_file, pm\_content (multicast message content), mm\_connect, mm\_connect\_caller,

mm\_connect\_receiver, mm\_content, mm\_disconnect, user\_status, user\_add, user\_delete, group\_add, group\_delete, group\_status, tempgroup\_create, tempgroup\_add, tempgroup\_delete, tempgroup\_status, keepalive.

## V. DYNAMIC DNS AND GNUMDIP

Bind 9 supports the dynamic update facility described in RFC 2136. This permits authorized updaters to add and delete resource records from a zone for which a name server is authoritative. Updaters can add or delete individual resource records, delete Rrsets (a set of resource records with the same domain name, class, and type), or even delete all records associated with a given domain name. Dynamic updating is achieved through the command facility 'nsupdate' included in the Bind distribution. When a name server processes a dynamic update, it is changing a zone and must increment that zone's serial number to signal the change to the zone's slaves. In order to enhance the security of dynamic DNS, TSIG-Signed updates and digitally-signed updates are introduced. Thus the zone administrators can choose either of them to ensure secure updates.

Most corporations use DHCP to allocate IP addresses and the central server of the enterprises tend to be connected to the Internet through ADSL which means that the IP addresses are dynamic. Moreover our enterprise instant messaging system is delivered through Web, therefore, in order to offer a wholesome enterprise solution, we use GnuDip—an open source project, to provide dynamic domain name update functionalities.

GnuDip consists of two parts—the client application and the server group which includes BIND, MYSQL, APACHE, and SENDMAIL (optional). Bind offers the commands 'nsupdate' and 'dnssec-keygen' to accomplish high-security updates; MYSQL is the database used for storing information such as the current domain names and their corresponding IP addresses, as well as the name and key of the server administrator; APACHE provides the cgi-based web interface to ease the management and updating operations of the administrator; SENDMAIL is responsible to use the email address of the administrators to verify their authority, thus making sure that the update queries come from the reliable users.

The administration of GnuDip server can be achieved either with the help of the web tool or through the command 'gdipc.pl' in the client application.

The architecture of our EIM is illustrated in Figure 1.

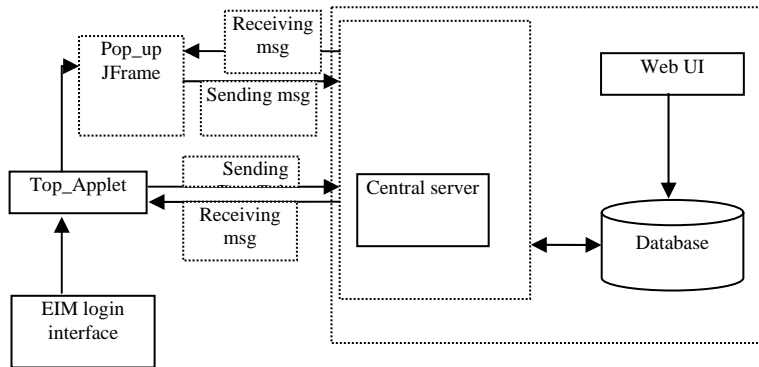


Figure1. The architecture of EIM

## VI. FUTURE WORK AND CONCLUSIONS

Our design and implementation of dynamic DNS-based EIM reflects the convenience and flexibility provided by the IM in enterprise. Meanwhile, our dynamic-DNS based solution can quickly adapt to the contemporary environment of enterprise network. Also we introduce the concept of group used especially for the protocol design as well as the interface design.

In the next step, we plan to solve the interoperability of this system, and research on the pros and cons of integrating compatibility to the public IM services. Meanwhile we intend to research on the management and traffic engineering of EIM.

## REFERENCE

- [1] J. Rosenberg et al., "Session Initiation Protocol," IETF RFC3261, June 2002.
- [2] B. Campbell et al., "Session Initiation Protocol (SIP) Extension for Instant Messaging," IETF RFC 3428, Dec. 2002.
- [3] J. Rosenberg, "A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)," IETF RFC 3857, Aug. 2004.
- [4] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)," IETF RFC 3856, Aug. 2004
- [5] P.Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence," IETF RFC 3921, Oct. 2004.
- [6] P.Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core," IETF RFC 3921, Oct. 2004.
- [7] J. Rosenberg, "An Extensible Markup Language (XML) Based Format for Watcher Information," IETF RFC 3858, Aug. 2004.
- [8] Jabber.Inc, "How to Evaluate an EIM solution," Jabber Whitepaper, 2005
- [9] Gnudip, <http://gnudip2.sourceforge.net/>
- [10] P. Vixie, Editor et al., "Dynamic Updates in the Domain Name System (DNS UPDATE)," IETF RFC2136, April 1997.
- [11] D. Eastlake 3<sup>rd</sup>, "Secure Domain Name System Dynamic Update," IETF RFC2137, April 1997.
- [12] Paul Albitz, Cricket Liu, "DNS and Bind", 4<sup>th</sup> edition, O'Reilly, 2002.12
- [13] sip-communicator, <https://sip-communicator.dev.java.net/instructions.html>
- [14] jain-sip, <https://jain-sip.dev.java.net/>
- [15] Mockapetris, P., and K. Dunlap. "Development of the Domain Name System." *Proc ACM SIGCOMM* (August 1988). Stanford, CA.