The ITB Journal

Volume 4 | Issue 1 Article 2

2003

Unified Internet Messaging

Paul Healy

Declan Barber

Unified Internet Messaging

Paul Healy and Declan Barber,

Institute of Technology, Blanchardstown

Paul.Healy@itb.ie

Declan.Barber@itb.ie

Abstract

As telephony services, mobile services and internet services continue to converge, the prospect of providing Unified Messaging and even Unified Communications becomes increasingly achievable. This paper discusses the growing importance of IP-based networks to Unified Messaging developments and examines some of the key services and protocols that are likely to make Unified Messaging more widely available. In this initial paper, we limit ourselves initially to the unification of text-based messaging using SMS and Email. The approach we make is based on the existing Internet Email framework but will take cognisance of the need to add voice and other non-text based messaging and communications at a later stage. Ultimately, the research project aims to build a working prototype of a generic messaging model that works for both non real-time and real-time communications. This model will provide a framework into which existing and future messaging protocols can be plugged. This paper originated from an applied research project which examined the integration possibilities for various messaging and communications protocols

1. Introduction

As traditional telephony services, mobile services and Internet services continue to converge, the prospect of providing true Unified Messaging (UM) and even Unified Communications (UC) becomes increasingly achievable. For the purposes of this paper we will define Messaging as any non-real time or near-real time message transfer service (e.g. email, paging, sms , fax, voicemail) and Communications as any real time, near-real time or non-real time messaging service (e.g. email, sms, fax, voicemail, telephony, computer telephony, video-conferencing). We define the minimum set of features in a truly Unified System (US) to include:

- a unified inbox for each user/profile
- a unified (logical) message store
- a unified directory service
- unified management of either centralised or distributed component architecture
- conversion between different media
- universal access from a broad range of user devices.

The ability to deliver a broad range of services over the ubiquitous Internet Protocol (IP) is driving the convergence of services traditionally provided over voice, data and Internet technologies. The UM of the future will need to be able to support real-time and non real-time messaging and communications based on a variety of media (text, voice, video and indeed multimedia). Traditionally, IP networks have not supported mixed traffic, but protocols are now emerging that can address the class of service issues for different types of traffic. There is widespread support in IP-based networks for email protocols and related extensions for text, voice, graphic and video attachments. Voice has not yet been implemented to the same extent using open standards over IP, but that is rapidly changing. Before we move on to examining specific Internet protocols, it is worth examining the benefits of an IP network.

2. Networks Converge towards IP

A review of the more significant differences between traditional network types is informative in understanding the benefits of using IP-based communications. A brief overview of the relative merits of traditional voice, data and IP networks is shown in Table 1. From the table, it is clear that there are numerous advantages to using IP networks, not least of which is the ability to control bandwidth. As IP protocols continually evolve to support features such as Quality of Service and Traffic Engineering, IP is reaching a stage where it may begin to offer the combination of the quality and reliability of the traditional circuit-switched network with the scalability and flexibility of packet-switched networks.

Arising from these benefits, the telecommunications world is making a shift away from traditional circuit-switched technologies towards a more open packet-switched infrastructure based on routed IP networks. This does not mean that the Internet will replace the traditional telephone and data networks in the foreseeable future but that the convergence of these technologies will lead to increased interoperability and consequently enhanced services. Consequently, our work is firmly focused on IP as the core set of protocols around which we will build our Unified Messaging prototype.

The earliest approaches to unified messaging used proprietary clients which integrated messages in a single application from a number of different message servers, each of which used its own message store and directory. This approach has been superseded by the dominance of clients based on proprietary email systems, such as Microsoft Exchange, with powerful Application Programming Interfaces (APIs). If we consider internet-based unified

messaging, there is the option to use a web-browser based client or widely available open standard email client that conforms to standard email protocols identified in section 5 below. Even the main proprietary email clients now support these standard protocols. Browser based access is server-centric with the web server acting as a proxy client to the end user who views the result in a HTML page. While this may initially provide a degraded email functionality, it does provide the most open, universal access.

	Voice Networks	**Data Networks**	**IP-Based Networks**
Availability	Very High	Congestion possible	Very High
Reliability	End-to-End	Only within the provider network	End-to-End
Scalability	Moderate	Moderate	Very High
Adaptability /Integration	Low	Low	High
Flexibility	Low	Variable	High
Ability to Control Bandwidth	Low	Low	High
Switching	Circuit-Switched	Packet-Switched	Packet-Switched
Standards	Standards based Equipment and software may be proprietary	Open Standards Based. Interoperable.	Open and Highly interoperable with increasing availability of APIs
Features /Services	Well-developed feature set	Variety of services	Supports multiple services and rich features
Connection	Dedicated and guaranteed Constant Bit Rate	Various connection types possible	Can support both Connection-oriented and Connectionless with varying classes of service.
Relative Ability to withstand Errors	High	Low	Variable
Delay	Low Deterministic	Variable	Traditionally variable. Emerging Protocols (MPLS) make it more deterministic

Table 1: An overview of the relative merits of traditional networks

3. Internet Email - Paradigms

There are a number of different approaches to building a distributed email infrastructure.

- Shared file-system strategies

- Proprietary LAN-based protocols
- X.400 P7 protocol
- Internet message access protocols

The only relevant approach for this project is Internet message access protocols: POP (Post Office Protocol), DMSP (Distributed Mail System Protocol), and IMAP (Internet Message Access Protocol). Of the three, POP is the oldest and consequently the best known. DMSP is largely limited to a single application, PCMAIL is known primarily for its excellent support of "disconnected" operation. IMAP offers a superset of POP and DMSP capabilities, and provides good support for all three modes of remote mailbox access: offline, online, and disconnected.

In the **Offline Paradigm,** mail is delivered to a mail server, and a personal computer user periodically invokes a mail "client" program (e.g., Outlook Express) that connects to the server and downloads all of the pending mail to the user's own machine. Thereafter, all mail processing is local to the client machine. The offline access mode is a kind of store-and-forward service, intended to move mail on demand from the mail server to a single destination workstation. Once delivered to the workstation, the messages are then deleted from the mail server and the user can continue to view and manipulate the messages even while offline.

In the **Online Paradigm** mail is again delivered to a shared server, but the mail client does not copy it all at once and then delete it from the server. This is more of an interactive client-server model, where the client can ask the server for headers, or the bodies of specified messages, or to search for messages meeting certain criteria. Messages in the mail repository can be marked with various status flags (e.g. "deleted" or "answered") and they stay in the repository until explicitly removed by the user, which may not be until a later session.

In the **Disconnected Paradigm**, the disconnected access mode is a hybrid of the offline and online models, and is used by protocols such as PCMAIL. In this model, a client user downloads some set of messages from the server, manipulates them offline, then at some later time uploads the changes. The server remains the authoritative repository of the messages. The problems of synchronisation arising from this mode (particularly when multiple clients are involved) are handled through the means of unique identifiers for each message.

4. Internet Email – Architecture

In Internet based email, the protocols used are highly dependent on the underlying Internet architecture. There is widespread support in IP-based networks for the Simple Mail Transfer

Protocol (SMTP) and the Multi-Purpose Mail Extension (MIME) for text, voice, graphic and video support. These protocols are designed to run over a client/server architecture, where there is an email server component with which the client programs communicate via direct connections such as sockets or remote procedures calls. Internet Email operates using a store and forward process. Both the client and the server must have an appropriate transport layer and a network API in place e.g. TCP/IP, which can be used by a developer to implement the protocols. The message store (or database) needs of an email system are generally simpler than those provided by a relational database and is only accessible via the server program. The store should be transparent to other programs. The email client is typically single-threaded and is only run when needed. It only needs to maintain a single connection to an email server. The server, in contrast, must be running at all times and starts automatically when the server is booted up. It must be capable of supporting multiple connections simultaneously from other clients and servers. The design should be both multitasking and even multithreaded. In such a client/server architecture, the intensive processing, such as for a search, is conducted on the server and only the result is sent back to the client. This minimises the amount of network I/O needed. Client/server are distributed architectures and are generally scalable to millions of users.

5. Internet Email – Protocols

To send mail, the email client (or user agent) establishes a network connection to a server and once connected, uses the Simple Mail Transfer Protocol (SMTP) to handle outgoing messages. To retrieve email, a client uses either the Post Office Protocol Version 3 (POP3) or more efficiently, the Internet Message Access Protocol, Version 4 (IMAP4). It is not necessary for the sender and receiver clients to be using the same retrieval protocol. The internal component in the server that collects and distributes the mail to different servers is called the Message Transfer Agent (MTA). It usually runs two main processes: an SMTP process to receive emails from a client and to forward emails to a different email server, and a POP3 or IMAP4 process to support the retrieval of mail. These processes may be run on the same or on different servers. The message store on the server is a specialised free-form textual database. This may be implemented using real database managers or use some internal database functionality. Apart from storing the messages, the MTA may record other useful data, such as if the message has been read or not (although strictly speaking, this is part of the clients responsibility). The message store may be implemented on a separate machine from the MTA or even across a number of different machines. It is utterly transparent form

the client. Message queues, such as an outgoing message queue or failed delivery queue are used to manage messaging. The Multipurpose Internet Mail Extension (MIME) to SMTP allows various kinds of files such a document, image or audio files, to be attached to text mail. It achieves this using binary-to-text encoding. As well as encoding, MIME adds data to the message header, so it is not a process that can be executed in advance of client activation. Rather, to send a MIME attachment, MIME must be integrated into the client. MIME facilitates the insertion of intelligence into the attachment recovery process by identifying the type of attachment to be specified (e.g. audio or graphic). MIME also supports the organisation of messages in various ways (e.g. in parallel or in multiple parts). Although not obliged to, email servers usually process at least some of the MIME header information to optimise performance. MIME is quite complex and as a result, most clients implement only a reasonably small part of the standard.

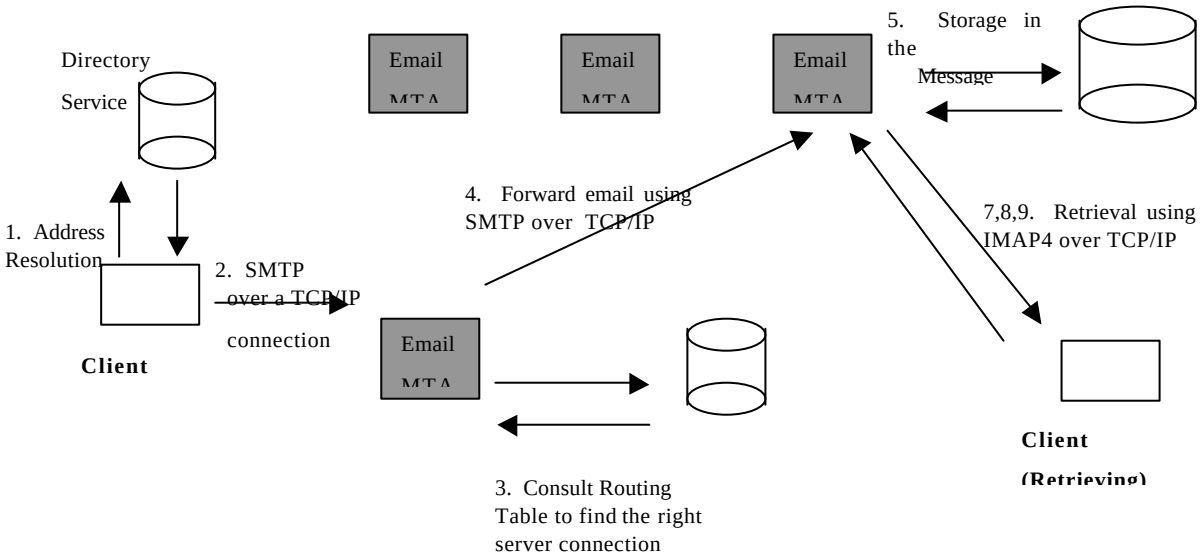

Figure 1: Internet Email Architecture and Operation

6. Comparison of POP3 and IMAP4

POP is the most commonly used Internet mail protocol. This protocol is the easiest to implement and use. Until a few years ago mail servers and client readers only supported POP. POP is still the default protocol for most mail reader clients and is supported by products on PCs Macs and Unix systems. POP is based entirely on the offline paradigm and therefore suffers limitations. It is only suitable for use with one client machine per account because messages are deleted from the server once they are downloaded to a machine. Once a connection is established with POP, all messages and their entire messages bodies, including

attachments, are downloaded to the client application. This creates a time and bandwidth overhead for unwanted messages. However POP is very easy to implement from both the client and server ends and is widely supported. POP is most suitable for home users that use just one machine for mail delivery.

IMAP is based on the online and disconnected paradigms where all the mailbox and messages are maintained on the server. It may also be set up for offline use if desired. IMAP allows the client to access and manipulate remote messages as if they were stored locally. The client issues commands to download them or delete them, access and set message state information, but the server always maintains the information. IMAP allows for faster service because the client reads all of the message headers without having to download all of the actual messages. For example, if a mailbox has ten messages in it and two of those messages has 200kb attachments then obviously it would be preferable to view the headers for these messages first and then decide if it is worth spending extra time and bandwidth downloading the files.

IMAP vs. POP:

Advantages of POP:

- Simple protocol to implement.
- More supporting client software available.

Advantages of IMAP:

- Can manipulate persistent message status flags.
- Stores messages as well as fetch them.
- Support for multiple mailbox management.
- Support for concurrent updates and access to shared mailboxes.
- Suitable for accessing non-email data; e.g., NetNews, documents.
- Also suitable for offline paradigm, for minimum connection time and disk use.
- Online performance optimisation, especially over low-speed links.

POP and IMAP can basically be compared based on the suitability of the supporting message access paradigms. So while offline and online mailers both allow access to new incoming messages on the mail server from a variety of different client platforms, the similarities stop there. The two paradigms reflect different requirements and styles of use and they do not mix very well. Offline works best for people who use a single client machine all the time. It is not well-suited for the goals of accessing an inbox of recent messages or saved-message folders from different machines at different times. This is because the use of offline ("download and delete") mail access from different computers at different times, tends to scatter the mail

across the different computers, unless they are all linked to a common network file system (in which case the access mode is really more online than offline.) On the other hand, the chief virtue of offline access is that it minimises the use of server resources and connect time when used via dialup.

IMAP (online) is the preferred access method for web-browsers because it only downloads the headers, whereas POP (offline) will store the entire messages in the browser cache. This can lead to out-dated files being kept on the local machine even after the user has deleted them from their main local message store.

7. Internet Directory Services

The ability to map a list of people/names to a relevant address is a critical part of the Internet structure. This need is analogous to the need for a telephone directory for PSTN numbers. In relation to email, we can consider this address mapping at locations: in the client, in the enterprise and globally. Proprietary email systems have developed directory services using a centralised approach to support all but the latter extremely well. The Internet has traditionally been weak in this area. A strong standard, X.500, was developed by the ISO to keep track of email addresses for X.400 mail but it requires the use of an ISO network stack and is therefore not appropriate for the Internet. A standard called the Lightweight Directory Access Protocol (LDAP), which is based on part of the X.500 standard but which runs on a TCP/IP stack, was developed for use on the Internet to manage internet-based email addresses. Most email clients, both proprietary and Internet–based, now support LDAP and a global network of LDAP servers and databases can be developed to facilitate finding someone's email address.

8. Short Message Service (SMS)

Developed in 1991, SMS (more commonly known as Text Messaging) is a globally accepted wireless service that enables the transmission of alphanumeric messages between mobile subscribers and external systems like email, paging, and voice mail systems. With SMS, an active mobile handset can receive or submit a short message at any time, even if a voice or data call is in progress. SMS also guarantees delivery of the short message by the network. Temporary failures due to unavailable receiving stations are identified, and the short message is stored until the destination device becomes available.

Initial applications of SMS focused on eliminating alphanumeric pagers by permitting two-way general-purpose messaging and notification services, primarily for voice mail. As technology and networks evolved, a variety of services were introduced, including interactive banking, information services such as stock quotes, integration with Internet-based applications, and email, fax, and paging integration. In addition, integration with the Internet spurred the development of Web-based messaging and other interactive applications such as instant messaging, gaming, and chatting.

SMS is currently more popular with users than the WAP protocol. It is also a simpler protocol for which to create an application. The following are some reasons why SMS is more suitable than WAP for value-added services:

- Large number of legacy (non-WAP enabled) phones
- WAPs uncertain future
- Lack of widespread WAP content
- SMS is suitable for meeting major market needs
- SMS can be used as a kind of 'push technology', meaning the user does not have to request delivery of information. It can be automatically sent.

9. SMS Connectivity:

This project will submit SMS messages from web-browsers. There are three distinct approaches to this:

- To **send an SMS from an attached GSM modem** over the normal air interface. This has the advantage of being able to send and receive SMS messages directly but is not very scalable and is disregarded for that reason.

- **HTTP Post Method:** to simply post the message data from the browser directly to the project web server. The web server will then forward this data to the core SMSC, where it will be processed and sent on to the appropriate mobile number(s).

- **TCP/IP Connection**: to post the message data to the project web server where a server-side application will establish a TCP/IP connection with the SMSC and transmit the message over IP.

The **HTTP method** of SMS communication allows for one-way transmission of single or multiple SMS messages from a web browser to mobile handsets. This is the limit of

functionality of this method. The browser on the client machine downloads the web page from a local or remote server, fills in the message and recipient details and then submits the message details as an HTTP form. However the form details are not submitted back to the server from where the web-page was downloaded. Instead it posts the message data to a third-party SMS handler, which then forwards the message data on to the SMSC. The SMSC is responsible for propagating the message(s) on to the recipient(s). This is usually done for some nominal fee.

Figure 2 depicts the process described above. The single-way arrows connecting the client machine, third-party server and the SMSC depict the communication limitation to this approach. The only return from a message submission is either confirmation of receipt at the SMSC or one of a host of possible error codes. The third-party server and the SMSC may in some instances belong to the same organisation. This does not affect the message submission process.

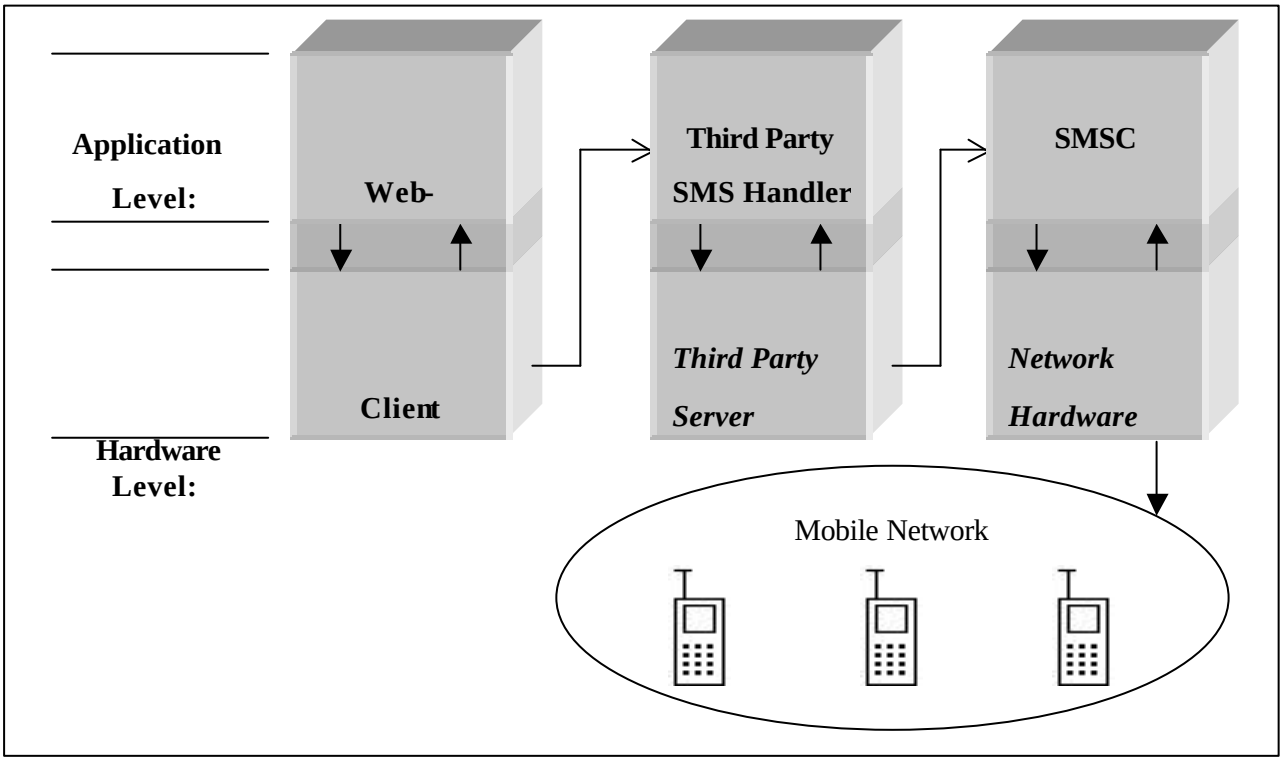

Figure 2: Diagram of HTTP Post Method

TCP/IP Transmit Method:

The TCP/IP method requires a program running on the project server which can access the SMSC directly without using a third party. The web page is downloaded into users browser from the project server, the message details are filled out as before but this time they are

posted back to the project server from which the page was downloaded. A program running on the server will then establish a TCP/IP connection (possibly synchronous depending on requirements) with the SMSC. The message data is streamed across this link to the centre where it is processed and forwarded to the relevant mobile handset(s).

There are a number of advantages to this method. Firstly the third party, which is effectively a "middle man", is removed from the equation so the cost is reduced. There is also no restriction to one-way messaging using this method. When a connection is established between the project server and the SMSC, it is possible to build or edit distribution lists for different user groups. The most useful feature of this type of connection is two-way messaging. It is possible to bind to the SMSC using a range of mobile numbers and then retrieve all waiting messages for the bound set of numbers. So messages can be retrieved by the project server and forwarded to the user in the form of a web page. This web interface may then allow the user to process these messages as if they were emails. They can be stored permanently in a database and retrieved at any time through this interface. If all messages are stored on the project server database then there is no restriction based on SIM card size so any number of messages can be kept.

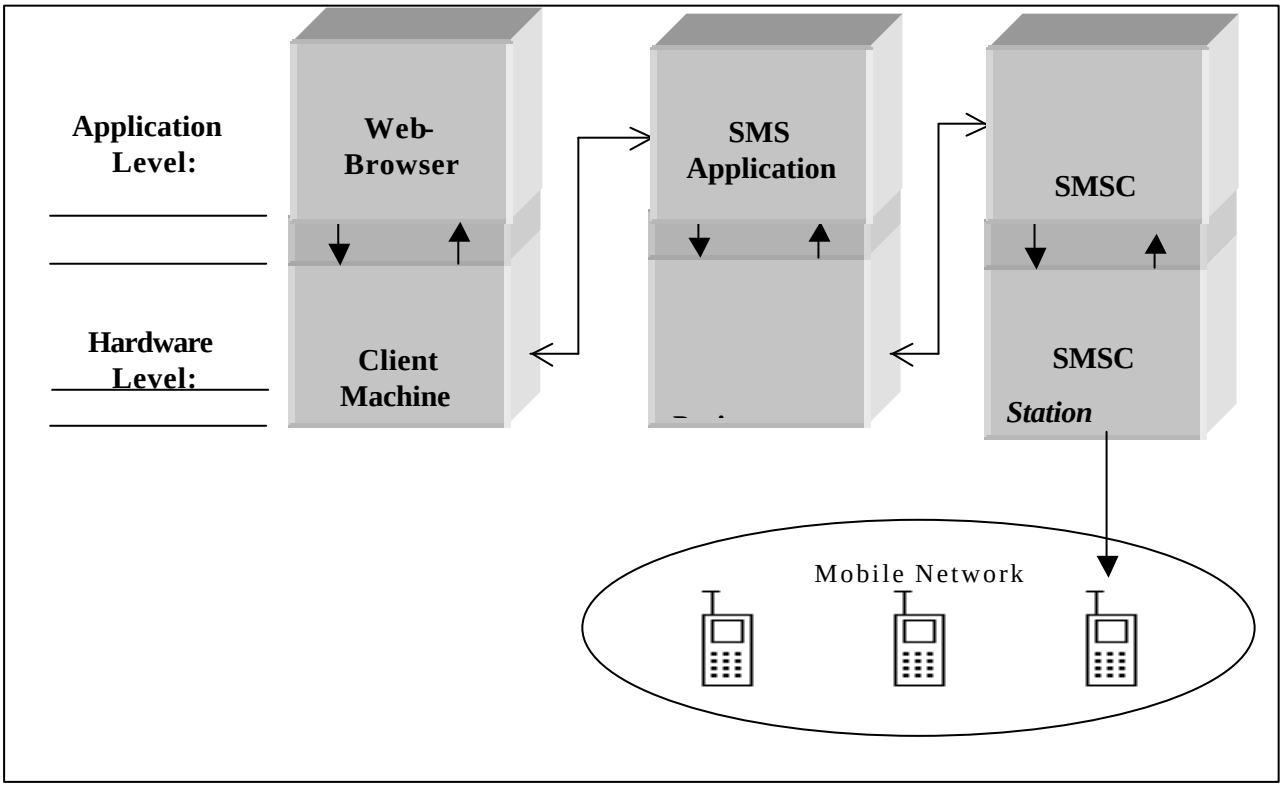

Figure 3: Diagram of the TCP/IP Method

Another advantage of this approach is paging functionality. Short Message Service and Number/Message Paging are based on the same basic technology. SMS is merely a functional superset of paging. Therefore it would be simple to incorporate a paging module to coincide with the SMS module. Figure 3 has bi-directional arrows connecting the client, project server and SMSC together. These show the two-way communication possible over the TCP/IP link as opposed to the one-way "message submission" method possible over HTTP alone.

Advantages of HTTP Method:

- Easy and very fast to implement.
- No responsibility for quality of service.

Advantages of TCP/IP method:

- Two-way messaging.
- Possible to retrieve messages on computer instead of mobile handset.
- Possible to store more messages on database than normally possible with mobile SIM cards.
- Possible to extend use of stored messages, e.g., forward messages on as email without re-writing.
- Extend functionality of SMS server application to also include the paging protocol.
- Two-way messaging via one web interface is the core idea of this project.

10. Preliminary Conclusions

The convergence of telephony, mobile services and the Internet strongly suggests that future UMS design should be based around IP. There is already widespread support for text based messaging in IP networks. The online email paradigm seems well suited not just to email bit to integrated text messaging. Research indicates that SMTP with MIME, IMAP4 and LDAP will be core protocols in our overall design. With well-established IP gateways to the Mobile Networks, integration of SMS with Email will be very achievable. Our design will primarily rely on the use of TCP/IP connections but either the HTTP or air interface connection could be used for reliability. The next stage of this research will examine the integration of voice-based messaging into our design and establish an overall design framework for a system that unifies both real and non real-time communications.

References:

1. Avison, David and Shah, Hanifa (1997) **The Information Systems Development Life Cycle: A First Course in Information Systems**, McGrawHill
2. Ford, Andrew, (2000), **Apache - Pocket Reference**, First Edition, O'Reilly
3. Laurie, Ben and Laurie, Peter, (1997), **Apache – The Definitive Guide**, First Edition, O'Reilly

4. Dubois, Paul, (2000), **MySQL**, New Riders
5. Crispin, M., (1994), **RFC-1733 - Distributed Electronic Mail Models in IMAP4**, Network Working Group
6. http://www.cae.wisc.edu/facstaff/imap/mail.html, Computer Aided Engineering Centre
7. http://www.imap.org/, The IMAP Connection
8. http://www.noctor.com/smsjdk.htm, Noctor Consulting