An Intelligent Monitoring and Analysis System for Instant Messaging

Dong Haichao

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2006

Abstract

Instant Messaging (IM) provides a free, real-time and private communication service for remote parties. Hence, its use has gained popularity among the modern generation. However, IM service has inadvertently evolved into a medium exploited for illegitimate information exchanging and criminal activities, such as online sex solicitation, information stealing/leaking and even terrorism network communications. The misuse of IM poses unwarranted threats to unsuspecting users, especially children.

To protect certain groups of users, the IM information exchanging should be closely monitored. Therefore, demand for IM monitoring systems has been increasing dramatically in recent years. However, off-the-shelf IM monitoring systems generally does not have comprehensive message recording capability and lack intelligent chat analysis functions. This research aims at developing an integrated chat message monitoring system which supports real-time chat message recording and provides intelligent chat message analysis services. As a result, this research has achieved the following objectives:

- A client-side personalized IM monitoring approach is proposed. The proposed approach is easily adaptable to IM evolution. In addition, it also incorporates a system hiding mechanism for self-protection from security-related applications.
- An adaptive message transmission control and recovery mechanism is proposed for real-time message transmission. This mechanism tackles the UDP transmission problems of packet loss, out of sequence arrival and duplicated data, while maintaining real-time transmission between the client and server.
- A chat message topic categorization approach is proposed for discovering topical information of chat messages. This approach incorporates the special characteristics of chat messages for topic detection.
- An IM analysis system is proposed. This system provides an interface for supporting both online chat message monitoring and offline chat message analysis including chat message retrieval, social network analysis and topic analysis.

Acknowledgement

I would like to express my sincere gratitude to the following people.

Dr. Hui Siu Cheung, my supervisor, who has been offering his kind guidance and endless patience throughout the project. Without his constant enlightenment, inspiration and motivation, this thesis would not have been possible to be completed. He has extended his help to me, not only as a superior, but also as a friend, which is beyond the call of duty. I am truly grateful for such kindness.

Dr. Chang Kuiyu, my co-supervisor, for the knowledge and advice gained through discussion.

Mr. Teo Choo Eng and Ms. Eng Hui Fang, the laboratory technicians of Database Technology Laboratory, for their support and kindness.

I would like to thank Nanyang Technological University for the sponsorship of this project.

And last but not least, to my wife and my parents, grandparents for their utmost love, support, and encouragement throughout the project development.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the advancement in Internet technology, numerous Internet-based communication services have emerged. Instant Messaging (IM) [1-3] is one of the most rapidly growing services. Millions or even billions of messages are exchanged daily through IM [4-6]. IM has become an electronic heaven for people to discuss virtually unlimited topics for both personal and business purposes. However, IM technology is a double-edged sword and has been misused for illegitimate information exchanging or committing crimes for its anonymity and completely uncontrolled chatting environment. Sexual solicitation [7], online bullying [8], and sensitive or confidential information stealing or leaking have been a great threat to the daily life of people [7], especially for children and teenagers. In addition, IM can also be used by terrorists for making contacts, which pose a great danger for the safety of a society. Therefore, some kinds of control measures such as IM monitoring are highly desirable in order to fight against the misuse of IM.

1.1 Instant Messaging Systems

Instant Messaging is a peer-to-peer service for remote users to communicate with each other, which typically comprises many client-based chat programs and a centralized server. The client programs allow IM users to communicate with direct connections while the server broadcasts the availability of users. There are many IM systems available in the market. Some of the most popular ones are Microsoft's MSN Messenger [1], Yahoo Messenger [2], and America Online's ICQ [3]. The popularity of these IM systems is greatly attributed to its anonymity, privacy and convenience.

- *Anonymity*. IM users are identified by their respective IM accounts, which can be registered for free. There is no direct relation between an IM account and the real identity of the user. It shelters IM users from being discovered.
- *Privacy*. IM users create their own social networks. Individual social networks are confidentially maintained by the IM system. Moreover, participation in an IM

conversation is based on invitation. The conversational contents are only available to those participants who have joined the conversation.

- *Convenience*. IM provides spontaneous communications for users who are connected to the same network. However, they also allow "away" or "offline" messages, which are basically e-mail like communication with information stored for unavailable users to read at a later time.

Although IM is an effective means for communications, it has also created a number of problems. As IM is based on direct communication among users, there is no central authority to monitor chat contents. As a result, the chat environment is completely uncontrolled. The private nature and anonymity of IM make it even harder to maintain a safe environment for communications. Moreover, the virtually unlimited ongoing topics and the great amount of chat messages make it difficult for human monitoring.

1.2 Instant Messaging Monitoring Systems

To help enforce legitimate contents to be communicated in a chat environment, a number of off-the-shelf IM monitoring systems have been developed. These systems can be broadly divided into two categories: network-based and client-based systems. For example, Stellar Internet IM [9] provides server-based monitoring for most popular IM systems over a corporate network, whereas Spector Pro [10] and Chat Watch [11] are client-based systems which allow individual users to monitor chat messages on their own computers. These systems record chat messages and provide facilities to help monitoring authorities to analyze chat messages. In addition to these commercial systems, Chattrack [12] is a prototype system developed for monitoring chat rooms. It supports keyword-based query retrieval and intelligent chat content topic detection. However, these monitoring systems have the following limitations:

- The chat message recording function has many drawbacks. These include the inability to capture encrypted chat messages and the need to update its source code in order to adapt to new versions of IM systems. Moreover, most monitoring systems can also be detected easily and terminated by users.
- Chat message analysis is mainly left as a manual task for the monitoring authorities, which can be very tedious if not impossible. Only very simple analysis functions such as keyword-based search and filtering, and simple statistics on messages are provided.

- Online, real-time chat message monitoring is lacking. Recording is performed in an online manner, while analysis is performed in an offline mode. The advantage of online monitoring is that it can deliver tracking information to users instantly.

1.3 Objectives

To tackle the problems encountered in most IM monitoring systems, the primary objective of this research is to develop an intelligent IM monitoring and analysis system which can support online real-time monitoring and offline intelligent chat message analysis. More specifically, this research will investigate the following issues:

- Investigate a robust IM monitoring technique for real-time message recording and system hiding.
- Investigate a real-time transmission mechanism for transmitting chat messages from monitoring clients to the server for logging and online monitoring.
- Investigate techniques for analyzing the recorded chat messages. In particular, we focus on topic detection for discovering topical information of chat messages.
- Design and implement an IM analysis system that provides an interface for supporting statistical analysis of chat messages, analysis of social relations among IM users and topic detection.

1.4 Proposed Architecture

To fulfill the objectives of this research, we propose a client-server based IM monitoring and analysis system, which is known as IMMonitor. Figure 1-1 shows the client-server system architecture of IMMonitor. The system comprises three major components: IMRecorder, IMServer and IMAnalysis.

- The *IMRecorder* is located at each target client which is a target PC from which IM chat messages are monitored and recorded. The recorded messages are then transmitted to *IMServer* in real-time.
- The *IMServer* then stores the chat message data into the Chat Log Database.
- The *IMAnalysis* is installed at each monitoring client, which is a PC from which the monitoring authorities can perform online monitoring and offline analysis of the recorded chat message data.

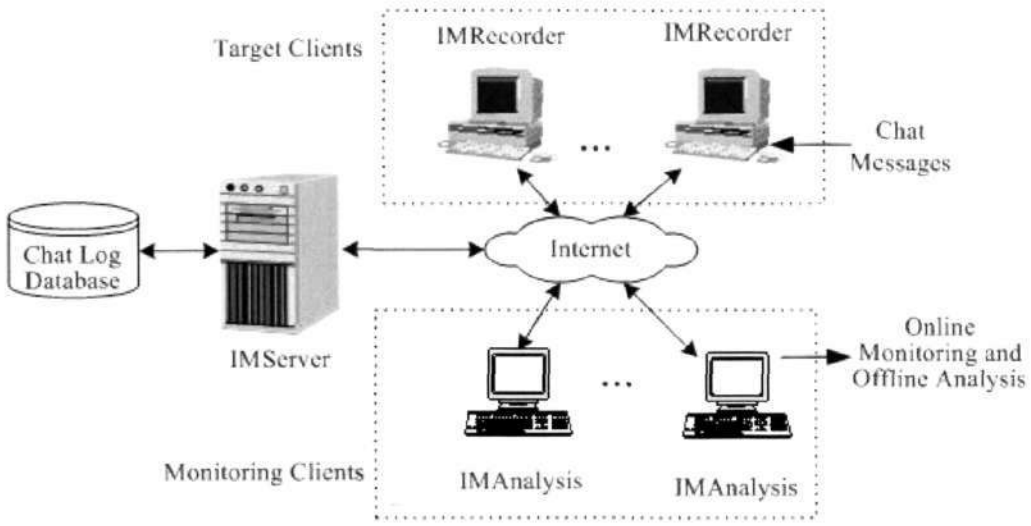


Figure 1-1: System architecture of IMMonitor.

1.5 Contributions

As a result of this research, we have achieved the following objectives:

- A robust IM monitoring technique is proposed for chat message recording and system hiding.
- An adaptive message transmission mechanism is proposed for real-time transmission of chat messages from target clients to the server.
- A classification-based approach is proposed for chat topic detection.
- A client-server based IM monitoring system is developed for supporting real-time, online IM monitoring and offline analysis of chat messages for message statistics, social network analysis and topic detection.

1.6 Organization of the Thesis

This chapter gives the background information and the motivation for our research. The objectives of this research are also discussed. This is followed by the overall architecture of our proposed IM monitoring and analysis system. The rest of the thesis is organized as follows:

- Chapter 2 reviews some of the most popular IM systems. IM monitoring systems will also be discussed based on its recording and analysis features.

- Chapter 3 studies the characteristics of chat messages. It investigates the conversational format of IM systems, message characteristics and supplementary functions of IM systems.
- Chapter 4 proposes an IM monitoring approach. It supports real-time chat message recording and system hiding in order to overcome the drawbacks of current IM monitoring approaches.
- Chapter 5 presents a server-based adaptive redundancy transmission mechanism for chat message transmission from target clients to the server.
- Chapter 6 discusses the chat message topic detection approach. In particular, a classification-based technique is proposed. Moreover, the performance of the proposed topic detection approach is also evaluated in this chapter.
- Chapter 7 describes the design and implementation of the message analysis system. The system provides both offline and online services on automatic chat message analysis for system users.
- Finally, Chapter 8 concludes this research and gives future directions for further research.

Chapter 2

Instant Messaging and Monitoring Systems

In this chapter, we review the features of IM systems based on the four most popular IM systems including MSN Messenger [1], Yahoo Messenger [2], ICQ [3] and QQ [13] (a popular IM system used in China). Then, we discuss some of the existing IM monitoring systems which are available commercially. The different IM systems and IM monitoring systems are also compared and discussed.

2.1 Instant Messaging Systems

This section reviews the features of IM systems in terms of network architecture, operational features, client interface and network protocol.

2.1.1 Network Architecture

Figure 2-1 shows the client-server network architecture of the ICQ system, which is typically employed by most other IM systems such as MSN Messenger, Yahoo Messenger and QQ.

The ICQ system consists of mainly three components, namely ICQ server(s), ICQ clients and the underlying ICQ network. An ICQ client is a program installed at a user PC to enable users to connect to the ICQ server and make contact with other users over the ICQ network. Two users can be connected when both users are valid users and logged onto the same ICQ service network. Once contacted and connected, the communication between the client users is established as direct connections, i.e., the ICQ server is not involved in the processing of chat messages.

In this network architecture, the ICQ server supports the exchange of control information and provides contact information (such as account names, nicknames and status of each contact) for ICQ users for making contacts. The underlying ICQ network is typically formed based on the Internet [14, 15].

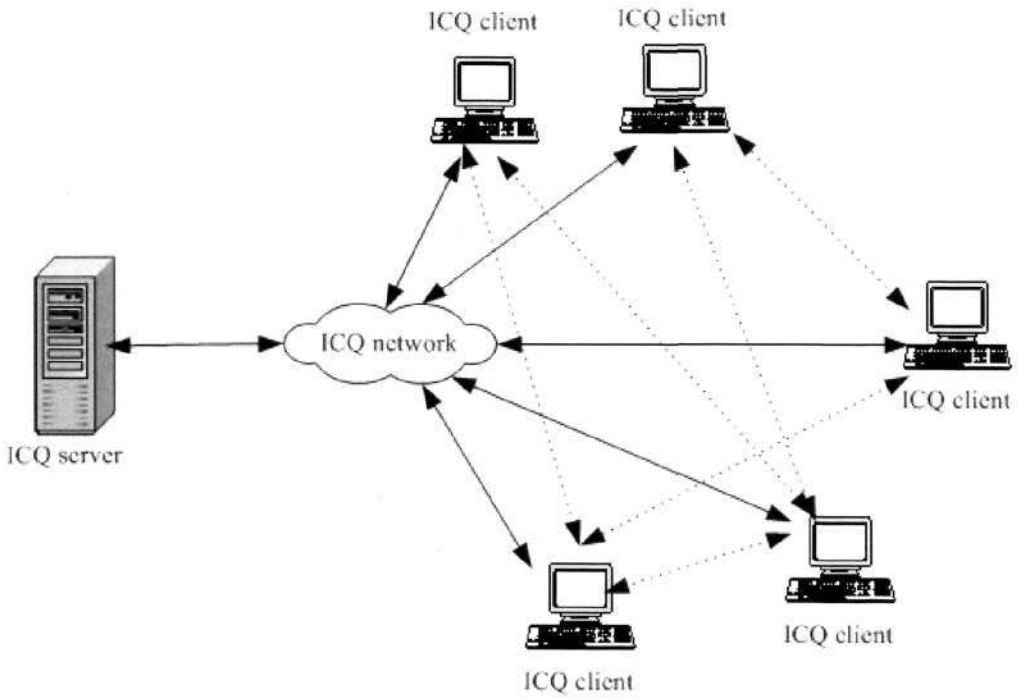


Figure 2-1: The network architecture of ICQ.

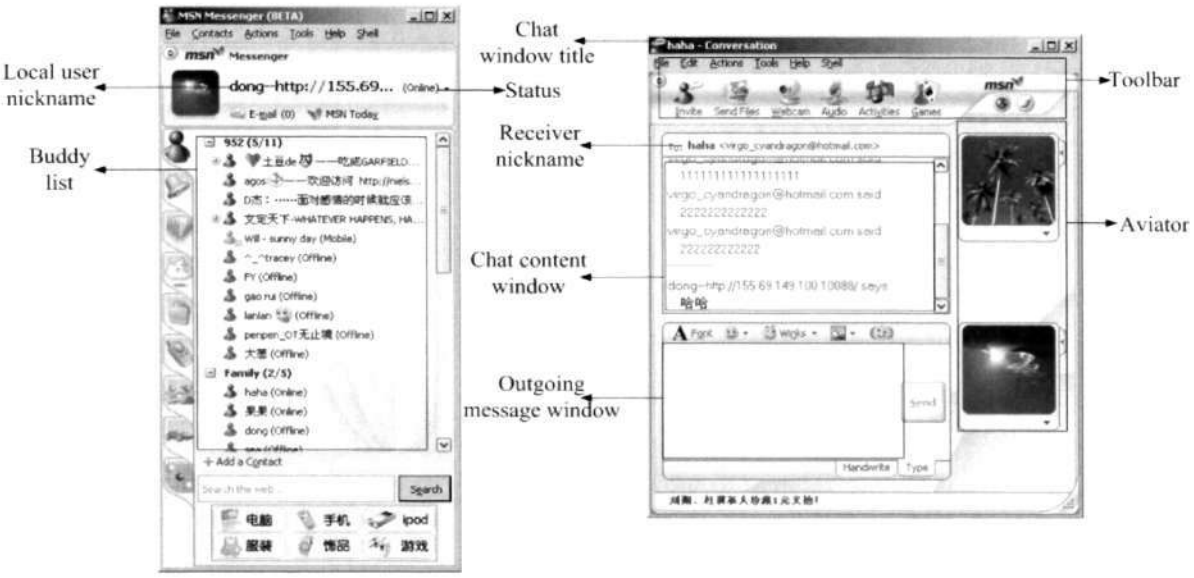
2.1.2 Operational Features

To start using an IM system and making contact with other users, a user must first register to the system with an account, nickname and buddy list.

- *User account.* An IM user account is an unique identity for each authorized IM system user. The registration of IM account is free for most IM systems. As such, a user could register with many different accounts for an IM system.
- *Nickname*. This is an identity of an IM user on the IM network. It shows the availability of the user to other contacts through the associated status (e.g., offline, online, away, etc). A nickname is initially supplied by the user and can be altered later at any time.
- *Buddy list*. This contains a list of contacts that an IM user could interact with. A buddy user can be added to the list only if the IM user knows about the buddy user's account. MSN Messenger and Yahoo Messenger enforce this by requiring authorization from the buddy user, while ICQ and QQ loosen this constraint by allowing messages from unauthorized users. In addition, ICQ and QQ also allow unknown strangers to search for IM users based on nicknames or other personal

particulars in order to make new friends. All systems allow blocking of messages from unwanted users.

2.1.3 Client Interface



(a) Main window. (b) Chat window.

Figure 2-2: MSN Messenger interface.

A client interface provides the features for IM users using the IM system. Figure 2-2 shows an example interface of the MSN Messenger, which is captured using the author's MSN Messenger account. The main window (see Figure 2-2(a)) displays the local user nickname with the associated status and the buddy list that comprises the contacts' nicknames and their associated statuses. The chat window (see Figure 2-2(b)) is created when the local user initiates a conversation with any contacts or accepts a conversation from any contacts. This newly initiated conversation is identified with a title displayed at the upper left-hand corner of the chat window. Each chat window comprises four main components: toolbar, chat content window, outgoing message window and aviator.

- The *Toolbar* area allows users to select different IM functions such as video conferencing, file transfer, etc.
- The *Chat content window* displays the ongoing chat contents. The receiver (remote) contact's nickname and account are displayed above the chat content window.

- The *Outgoing message window* enables the local user to type and send out chat messages.
- The *Aviator* area displays images of participants in a chat session.

2.1.4 Protocol

The four IM systems under review are free IM systems. Although they use the public Internet network infrastructure for communications, each of the systems is built up based on their own proprietary IM network with their own protocol. The protocol of each IM system has evolved rapidly in order to support more IM features and prevent the network from being used by third-party systems. ICQ has the latest version of protocol OSCAR V9 [16] based on Transmission Control Protocol (TCP) [17]. The protocol of MSN Messenger is called MSNP [18], which stands for Mobile Status Notification Protocol. The MSNP versions currently in use include versions 11, 12 and 13 designed for MSN Messenger version 7.0 and later. The protocol can sit on top of both TCP for normal cases and HTTP when dealing with proxies. The protocol of Yahoo Messenger has also been evolved with several versions based on TCP for the past few years. The latest version in use is YMSG12 [19]. QQ developed its protocol based on User Datagram Protocol (UDP) [20]. However, the actual protocol used by QQ is still not known due to its proprietary nature. In short, the IM protocols used by all IM systems are proprietary without official documentation available.

2.1.5 Discussion

Table 2-1: Comparison of four IM systems.

	ICQ	MSN Messenger	Yahoo Messenger	QQ
Network Architecture	Client-server direct IP	Client-server direct IP	Client-server direct IP	Client-server direct IP
User search	Random, filtered	By user account	By user account	Random, filtered
Add contact	By invitation, authorization supported	By invitation, authorization Required	By invitation, authorization required	By invitation, authorization supported
Chat mode	By invitation, stranger's messages allowed	By invitation	By invitation, stranger's messages allowed	By invitation, stranger's messages allowed
Protocol	ICQ V9 TCP based Proprietary	MSNP 11,12,13 TCP or HTTP based Proprietary	YMSG 12 TCP based Proprietary	Unknown UDP based Proprietary

Table 2-1 summarizes and compares the features of the four popular IM systems. We have the following observations:

- *Privacy*. IM conversations are originally designed to be between two parties and later evolved for supporting communications among multiple participants. However, IM conversation is still different from chat-room style of discussion. In IM, chat contents are only available to the invited participants. Moreover, the IM buddy list of each user is confidentially stored in the IM server. Therefore, the IM network contains many private and closed social circles with the exchange of private messages.
- *Anonymity*. The free of charge and free registration nature of IM services separates a user's real identity from his account. The ever-changing nicknames further disguise the user's real identity. Therefore, an IM user is basically anonymous. User identification to ensure good individual behaviour could be quite challenging.
- *Evolution of protocols*. IM protocols evolve very fast in order to support new features and keep the IM service proprietary. Official protocol information is generally not publicly available.
- *Chat content window*. All IM systems have provided a chat window for displaying the ongoing chat conversation.
- *Network architecture*. Most IM systems utilize direct communications for message exchange between the participated users without message processing by IM server. Thus, an IM server may not have the knowledge about a client's chat messages.

2.2 IM Monitoring Systems

IM monitoring systems can be divided into two categories, namely network-based and client-based IM monitoring systems. The former mainly focuses on monitoring IM activities from a centralized server within a corporate network. In this case, multiple IM systems from many client PCs are monitored at the same time. The latter is installed at a single PC and monitor IM activities directly from the PC. In this section, we review some of the most popular commercial IM monitoring systems including Stellar Internet IM [9], Spector Pro [10], Chat Watch [11], Spybuddy [21], ChatBlocker [22] and Invisible Keylogger [23]. The Stellar Internet IM is a network-based monitoring system, while the others are client-based

monitoring systems. The discussion of the systems will focus on the issues of chat message recording, system protection and chat message analysis.

2.2.1 Message Recording

The network-based IM monitoring systems, like Stellar Internet IM, can be used to monitor most popular IM systems such as MSN Messenger, ICQ and Yahoo Messenger. It logs chat records into a database due to the potentially vast amount of chat messages captured from the network. Figure 2-3 shows a sample data log stored in Stellar Internet IM. Each chat message uttered is stored as a record in the database. Each record contains information on message timestamp (the system time when the message was sent out from its originated machine), participants' nicknames, the originated IM system and message contents. The chat messages from different IM users are stored together in temporal order.

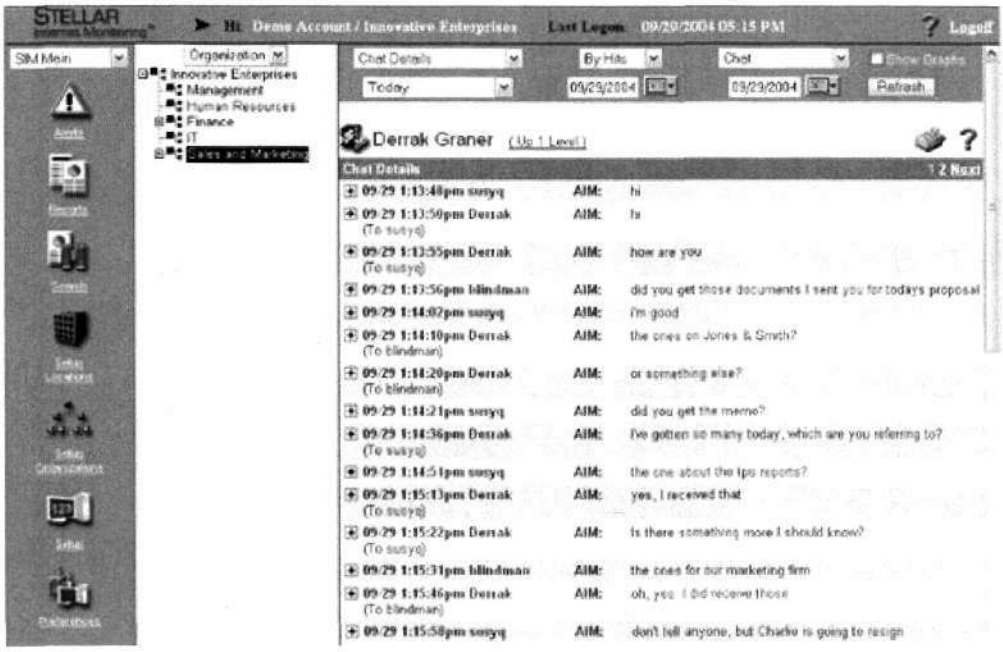


Figure 2-3: Data log of Stellar Internet IM.

The client-based IM monitoring systems, such as Chat Watch, Spector Pro, Spybuddy, ChatBlocker and Invisible Keylogger, store chat messages as text files in the local machine for the relatively small amount of data expected. Figure 2-4 shows a sample data log of Chat Watch. Chat Watch is able to record chat messages from MSN Messenger, ICQ and Yahoo Messenger. In chat log files, the participants' nicknames, timestamp, user account and messages exchanged in each chat session are recorded. Spector Pro, Spybuddy, ChatBlocker

and Invisible Keylogger have similar recording capability as Chat Watch. Chat messages from most popular IM systems such as MSN Messenger, Yahoo Messenger and ICQ can be monitored. For Invisible Keylogger, it can only record outgoing messages of a local user.

2.2.2 System Protection

System protection of IM monitoring systems refers to its hiding ability and security measures that prevent the system from detection, termination and deletion. This will depend on the underlying operating system. As this research focuses on Windows-based systems, we discuss system protection from the viewpoint of Windows operating system. As network-based IM monitoring systems are usually installed at a dedicated machine, it is authorized and physically isolated from all monitored users. To a certain extent, it is perfectly hidden and protected. Therefore, it does not require any further protection mechanisms.

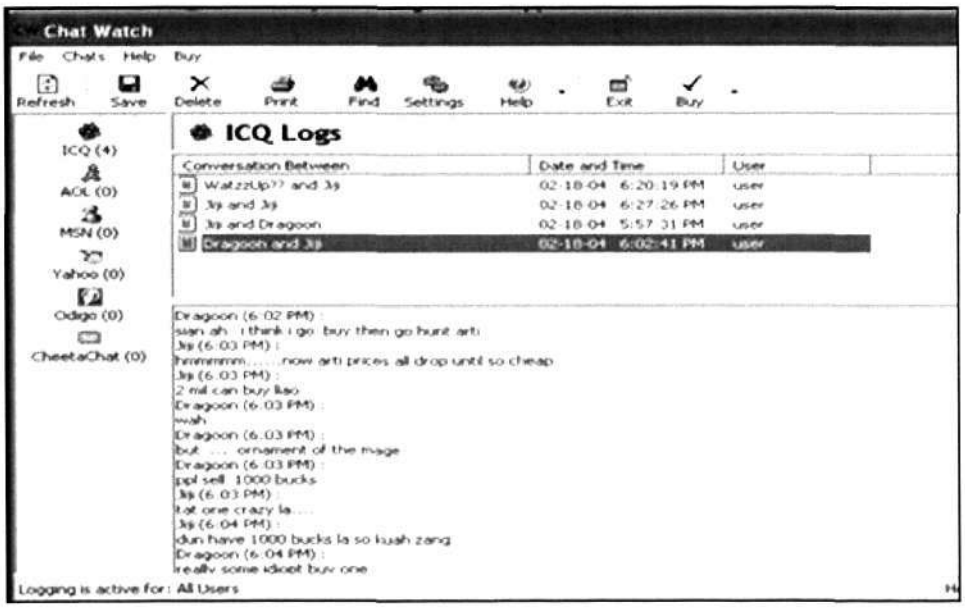


Figure 2-4: Data log of Chat Watch.

In client-based IM monitoring systems, the most common way of hiding is to hide the monitoring process away from the Task Manager. All IM monitoring systems under review have implemented two operational modes: the normal mode and stealth mode. Under the stealth mode, the monitoring systems claimed to be hidden from the Task Manager. Apart from Spector Pro, none of the other systems could hide itself from the Task Manager's process monitoring function.

To prevent from being terminated and deleted by unauthorized users, most IM monitoring systems have incorporated password protection as the primary security measure. Chat Watch even employs password protection on accessing chat logs. Some other systems such as ChatBlocker implement a function to disable user access to the Task Manager. However, this may not be desirable, as the user will be cautious when the Task Manager continues not to function properly, despite its effectiveness in protecting the monitoring systems.

2.2.3 Chat Log Analysis

Chat log analysis aims to browse through the vast amount of messages and filter out information that is of particular interest to the monitoring authorities. Most IM monitoring systems provide the following chat log analysis functions: chat log browsing, chat message retrieval, chat content search and simple statistical reports.

- *Chat log browsing*. Most IM monitoring systems support the viewing of the contents of chat log files.
- *Chat message retrieval*. Most IM monitoring systems only support constrained retrieval of chat messages. Stellar Internet IM performs chat message retrieval based on time or predefined network groups. Similarly, most client-based systems such as SpyBuddy support retrieval of chat logs based on time. The constrained chat message retrieval capability helps users to focus on only a subset of chat logs.
- *Chat content searching/filtering*. Most IM monitoring systems support the keyword-based chat content searching function. Stellar Internet IM performs online chat content filtering based on a user predefined keyword list. Similarly, Chat Watch highlights sensitive keywords in chat logs.
- *Statistical report*. Most IM monitoring systems, especially client-based systems, provide a very simple statistical report on IM monitoring including statistical information on the recorded chat messages. Network-based systems such as Stellar Internet IM are able to produce role-based reports based on predefined network user groups within organizations.

2.2.4 Discussion

Table 2-2: Comparing monitoring capabilities of IM monitoring systems.

	Stellar Internet IM	Spector Pro	Chat Watch	Spybuddy	ChatBlocker	Invisible Keylogger
Supported IM Systems	ICQ, MSN, Yahoo, AOL, Bloomberg	AIM, MSN, ICQ, Yahoo, Trillian and IRC	AIM, MSN, ICQ, Yahoo, and IRC	AIM, MSN, ICQ, Yahoo, Trillian	AIM, ICQ, MSN, Yahoo, IRC, Trillian	Any IM systems
Chat Log Recording	Message-based	Session-based	Session – based	Session -based	Session - based	Message-based
Hiding	NA	Yes, not detectable	Yes, detectable	Yes, detectable	Yes, detectable	Yes, detectable
System Protection	NA	On access password	On access password	On access password	On access password	On access password
Adapting to New Versions	Yes, update required	Yes, update required	Yes, update required	Yes, update required	Yes, update required	Yes, update required

Table 2-2 gives a comparison between different IM monitoring systems in terms of IM monitoring capabilities. We have the following observations:

- *Supported IM systems*. All systems support ICQ, MSN Messenger and Yahoo Messenger. In particular, Invisible Keylogger does not limit itself to any particular IM systems. QQ monitoring is not supported by any of the systems as it is not a popular IM system to non-Chinese users.
- *Information recording*. Network-based systems record chat messages from different users according to its message arrival order. Most client-based systems store chat messages as session-based logs. All IM monitoring systems record similar types of information such as the IM system name, timestamp of the chat session and chat content. Additional information such as the names of participants and title of conversation are also recorded in some systems.
- *Hiding capability*. All client-based monitoring systems except Spector Pro are detectable by the Task Manager.
- *System protection*. All client-based monitoring systems provide password protection measure for protecting the system from being terminated and deleted.

- *Adapting to new versions.* Most monitoring systems are able to monitor new versions of IM systems provided that new updates can be done in the source code to adapt to the new versions. Some systems such as SpyBuddy, ChatBlocker and Invisible Keylogger are unable to deal with multilingual text messages and the chat message recording capability is also not robust.

Table 2-3 gives a comparison of existing IM monitoring systems based on message analysis capabilities. We have the following observations:

- *Chat log analysis.* Very limited message analysis and visualization functions are provided. Only simple functions are available for displaying chat messages recorded in chat logs. However, such functions are insufficient for the processing of a large quantity of chat messages. Human inspection of chat contents could be very tedious.
- *Personal information.* User identification is not well supported in most IM monitoring systems. A few systems make use of nicknames for user tracking. However, as mentioned before, nickname is not a useful means for user identification.

Table 2-3: Comparing message analysis capabilities of IM monitoring systems.

	Stellar Internet IM	Spector Pro	Chat Watch	Spybuddy	ChatBlocker	Invisible Keylogger
Log Browsing	Yes	Yes	Yes	Yes	Yes	Yes
Message Retrieval	IP, Time, Network Account	N.A.	N.A.	Time	N.A.	N.A.
User Tracking	Nickname, Client IP and Network account	Windows account	Nickname	Windows account	Nickname	N.A.
Content Searching	N.A.	Keyword based	Keyword based	N.A.	N.A.	N.A.
Content Filtering	Keyword based (online)	N.A.	N.A.	N.A	N.A.	N.A.
Statistical Report	By IP and network account	N.A.	N.A.	N.A	N.A.	N.A.

2.3 Summary

In this chapter, we have reviewed the features of some of the existing popular IM systems. IM systems support privacy and anonymity. In addition, we have also reviewed some typical IM monitoring systems. These monitoring systems record chat messages and most of them

log messages as sessions based on conversations. And they support the monitoring of most popular IM systems. However, the IM monitoring systems generally do not possess a good hiding ability. Moreover, most systems require updating in its coding in order to adapt to new versions of IM systems. The chat analysis function is not effective in analyzing a large volume of recorded chat messages. The lack of providing personal information for tracking and intelligent message analysis is the biggest challenge faced by IM monitoring systems.

Chapter 3

Chat Message Characterization

In this chapter, we study the characteristics of chat messages from analyzing a collection of 33,121 sample messages gathered from 1700 sessions of conversations of 72 pairs of MSN Messenger users over a 4-month duration from June to September of 2005. These users are haphazard selected from graduate/undergraduate students of Nanyang Technological University and National University of Singapore, Singapore.

The primary objective of chat message characterization is to understand the properties of chat messages for effective message analysis including statistical analysis, social network analysis and message topic detection. The study is carried out based on message contents and is discussed in terms of the following characteristics: chat language, icons, hyperlinks, message length and dynamic content. Before discussing the message characteristics, we review the general conversational format for most popular IM messaging systems including MSN Messenger, QQ, ICQ and Yahoo Messenger. This chapter also summarizes the message characteristics that are important for effective message analysis.

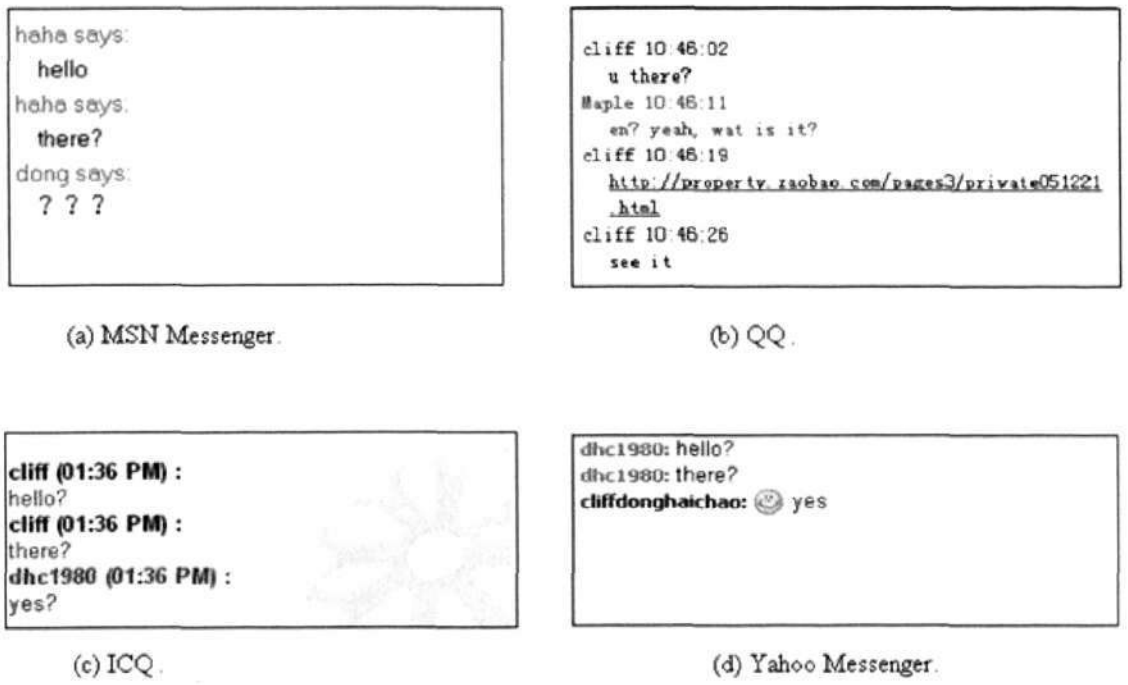
3.1 Conversational Format

Figure 3-1 shows the conversational formats of some of the most popular IM systems such as MSN Messenger, QQ, ICQ and Yahoo Messenger. In general, the conversational format consists of the following three components:

- *Chat participants*. They are IM users participating in the current session of conversation identified by their respective nicknames. For example, "cliff" and "Maple" in Figure 3-1(b) are the two participants of the current session of conversation in QQ. Similarly, "haha" and "dong", "cliff" and "dhc1980", "dhc1980" and "cliffdonghaichao" are the participants in the chat sessions of MSN Messenger, ICQ and Yahoo Messenger respectively.
- *Optional information*. Optional information can be attached at the end of the nicknames of the chat participants. For example, in QQ and ICQ, a timestamp is attached at the end of the participant's nickname. QQ has the timestamp in the format

of *hour:minute:second* (see Figure 3-1(b)), while ICQ displays the timestamp in a different format (see Figure 3-1(c)). Another example of optional information is shown in Figure 3-1(a) in which MSN Messenger attaches the word "says" after the participant's nickname, while Yahoo Messenger does not attach any optional information following the participant's nickname.

- *Chat messages*. The conversational contents of chat messages are displayed or typed after the participant's nickname and the associated optional information. And in most IM systems except Yahoo Messenger, each chat message starts from a new line. Yahoo Messenger displays chat messages immediately after the participant's nickname.



(a) MSN Messenger. (b) QQ.

(c) ICQ. (d) Yahoo Messenger.

Figure 3-1: IM conversational formats.

Figure 3-2 summarizes the general conversational format used by most IM systems. The message contents are displayed according to temporal order. As such, each chat session of conversations preserves the correspondence between chat messages and participant's nicknames, and the temporal sequence of chat messages.

Participant's nickname (optional information)
Message 1
Participant's nickname (optional information)
Message 2
• • •

Figure 3-2: General conversational format.

3.2 Message Characteristics

IM systems are originally designed only for text-based communications. However, in addition to textual contents, contemporary IM systems also support the insertion of icons and hyperlinks into message contents. Moreover, the chat language used by most chat users is also quite different from conventional written English. In this section, we discuss the characteristics of chat messages based on the collected set of 33,121 chat messages.

3.2.1 Chat Language

Chat language is basically written English. However, due to the real-time and informal conversational environment of IM systems, chat messages are written in a very different way from conventional English. Some of the common usage features in chat language include *acronyms*, *short forms*, *polysemes*, *synonyms* and *mis-spelling of terms* [24, 25].

- *Acronyms* are formed by extracting the first letters of a sequence of words. For example, ASAP is an acronym for "As Soon As Possible". Through the inspection of the 33,121 chat messages, we have identified a set of acronyms which is listed in Appendix A. Table 3-1 lists some of the popular acronyms used by the participants.

Table 3-1: Examples of popular acronyms.

Acronym	Equivalent Meaning	Acronym	Equivalent Meaning
ASAP	As Soon As Possible	OTP	On The Phone
ASL	Age Sex Location	POS	Parent Over Shoulder
BRB	Be Right Back	TTYL	Talk To You Later
BF	Boy Friend	U2	You too
GF	Girl Friend	WTH	What The Heck
CU	See You	YW	You are Welcome

- *Short forms* refer to the case in which a lengthy word is replaced with a shorter alternative expression. Table 3-2 shows some example short forms extracted from the 33,122 chat messages.

Table 3-2: Examples of short forms.

Short Form	Equivalent Meaning	Short Form	Equivalent Meaning
L8R	Later	Tom	Tomorrow
Nvr	Never	Btw	Between
Tat	That	Pic	Picture
Nvm	Never-mind	Wlm	Welcome
Frenz	Friends	Congrats	Congratulations
Sth	Something	Eg	Example

- *Polysemes* refer to terms that have more than one interpretation. In chat environment, a term can be either a word or short form. For example, "comp" can refer to "company" or "computer" depending on the context.
- *Synonyms* refer to the case in which terms with similar or same meaning are used interchangeably. For example, "network adaptor", "network interface card", and "NIC" can be used interchangeably during conversation on computer hardware and networking related topics.
- *Mis-spelling of terms* occurs at a higher rate in chat conversations than in traditional published text documents. Due to the real-time and informal nature of chat conversation, mis-spelled words in chat messages often occur. However, there are also cases in which a chat participant purposely mis-spells the words to emphasize the meaning. A common case for mis-spelling is the use of duplicated vowels, such as "sooooo", "noooo" and "thee" instead of "so", "no" and "the" respectively. The number of duplications is not fixed. Another case is the substitution of similar pronounced letters. For example, "today" becomes "todae" and "ok" becomes "okie".

The heavy usage of the above language features in online chat space has resulted in a very large vocabulary that contains many slang words other than the conventional English. The number of occurrences of such words in the sample chat message collection was subsequently counted. There were a total of 14,000 words or 6,000 distinct words after the process of stemming and conventional stop-word removal.

The set of slang words extracted from the sample chat messages is found to be an aggregation of subsets of slang words that can be subjective to specific virtual communities

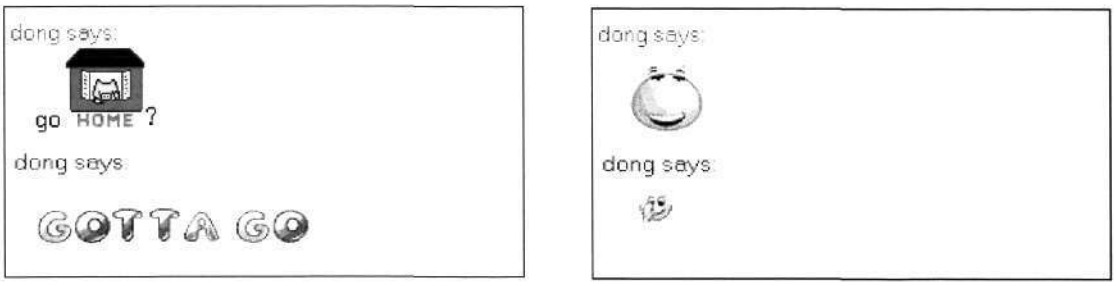
mediated by IM, whereas virtual communities are formed by users who share common interests and their interactions with one another are based on common shared vocabulary [26]. As an example, Chinese students share the onomatopoeic "haha" or "hehe" to represent smiling/laughing, while Japanese use "joujou" instead.

However, the slang words subjective to various communities are likely to either become accepted [27] or get overridden [28] in more general virtual communities such as nation-wide groups. Therefore, slang words originated from different virtual communities tend to become more standardized within a more general community whose members share a common interest as reflected by these standardized slang words.

Nonetheless, a core set of standardized terms is used in chat messages to present a topic. This is similar to traditional written text except that the core set contains many contemporary standardized slang words other than the conventional English. The core vocabulary set is usually sufficient for topic analysis. In addition, many linguistic efforts have been dedicated to capture the slang words used in different topics by different communities [29, 30], which helps to minimize the effect of non-standardized slang words in the analysis of chat messages.

3.2.2 Icons

Icons are images inserted along with the text content. According to the graphical contents, icons can be divided into two groups: text and non-text. *Text icons* are images carrying graphical text. *Non-text icons* contain little textual content. For example, Figure 3-3(a) contains two chat messages composed with text icons. In the first message, the word "HOME" is part of an icon image. In the second message, the icon "GOTTA GO" is formed from images of characters 'G', 'O', 'T', and 'A'. In another example, Figure 3-3(b) shows two smiley icons. The first icon mimics a silly laugh, whereas the second is a farewell icon. Both icons are just images without any textual information associated with it.



(a) Example text icons. (b) Example non-text icons.

Figure 3-3: Icons used in chat messages.

In fact, all IM systems under study have implemented shortcut text for icons. Each icon is associated with a unique sequence of text as shortcut, which will be interpreted and replaced with the corresponding icon when typed. And the text icons are usually associated with the text when they are displayed, i.e., graphical text is usually the same as the shortcut. For example, the first icon in Figure 3-3(a) is associated with the word "home". When the word "home" is typed, it is automatically replaced by the icon without any user actions. Non-text icons, on the other hand, are usually associated with annotations or semantically meaningless shortcuts specified by users. For example, the icons in Figure 3-3(b) are associated with the word "heihei" and the shortcut sequence ":8" respectively.

3.2.3 Hyperlinks

In chat conversations, hyperlinks or URLs (Uniform Resource Locators) can be given to refer other chat participants to web resources such as web pages and files for information sharing. URL must be specified in an absolute form beginning with a scheme name followed by a network name which points to a host server.

Table 3-3: Statistics on URL links.

URL Scheme	# of occurrences	% out of total occurrences	# of sessions	% out of total sessions
http	219	87.25%	162	9.53%
ftp	27	10.76%	14	0.82%
mms	4	1.59%	2	0.12%
rtsp	1	0.40%	1	0.06%
Total	251	100.00%	179	10.53%

Table 3-3 gives the statistics on URL links from the collection of sample chat messages. Some popular URL scheme names used in the collection include *http* (HyperText Transfer

Protocol), *ftp* (File Transfer Protocol), *mms* (Multimedia Message Service) and *rtsp* (Real Time Streaming Protocol). The "# of occurrences" refers to the number of chat messages containing a URL. The percentage of occurrences is given in "% out of total occurrences" from the total of 251 URL occurrences. On the other hand, the "# of sessions" shows the number of chat sessions containing one or more hyperlinks. The percentage of occurrences is given in "% out of total sessions" from the total of 1700 sessions.

According to our study, there are 219 http links, 27 ftp links, 4 mms links and 1 rtsp link. Amongst the 219 http links, 40 of them point to downloadable files such as *mp3*, while the rest are all web pages. Http links are the most popular form of hyperlinks (87.25% out of all URLs) used in chat conversations and occur in 9.53% of the total 1700 sessions. On the other hand, *ftp* links are used for file sharing and occurred in 0.82% of total sessions in the data collection. Streaming media links such as *mms* and *rtsp* rarely occur in chat sessions.

3.2.4 Message Length

Table 3-4 shows the statistics on chat message length from the collection of chat messages. The message length is quite short with about 91.5% of chat messages less than 50 bytes. Table 3-5 shows the statistics on the length of chat sessions of conversations. Most chat sessions have length greater than 50 bytes but less than 500 bytes. However, there are still 34.50% of chat sessions with less than 50 bytes in length. On the other hand, there are also 14.10% chat sessions containing more than 500 bytes but less than 5000 bytes of data.

Table 3-4: Chat message length.

Message length (bytes)	% out of total messages
0- 10	34.40%
11-20	22.40%
21-50	34.70%
51-100	6.90%
101-500	1.40%
501 and above	0.20%

Table 3-5: Chat session length.

Chat session length (bytes)	% out of total sessions
0- 20	14.00%
21-50	20.50%
51-100	21.10%
101-150	12.70%
151-500	17.60%
501-5000	14.10%

Moreover, we have also found that the average time gap between two adjacent chat messages is around 20 seconds and a typical chat session has a duration of 4-20 minutes for conversations greater than 50 bytes but less than 500 bytes.

3.2.5 Chat Topics

Table 3-6: Chat session topics.

# of topics	# of sessions	% out of total sessions
1	1143	67.24%
2	231	13.59%
3	59	3.47%
4	38	2.24%
5	17	1.00%
0	212	12.46%

Due to the interactive and dynamic nature of the IM environment, chat topics (such as Games, Sports, Pornography, etc.) can be changed quite rapidly within a chat session. It is possible that each chat session of conversation contains multiple topics. And a topic may also spread over several sessions of conversations. Table 3-6 shows the statistics on the number of topics discussed in the collected set of chat sessions. 67.24% of all sessions are observed to focus on a single topic. On the other hand, 13.59% of sessions contain two topics in a discussion and a total of about 20.3% of chat sessions contain two or more topics. Moreover, 12.46% of chat sessions are not dedicated to any meaningful topics. These chat sessions are mostly short sessions, which contain messages mainly on greetings.

3.3 Supplementary Functions of IM Systems

Apart from instant messaging, contemporary IM systems typically integrate other supplementary functions to facilitate the communications between chat participants. Audio/video conferencing and file sharing are two most common functions provided by most IM systems.

3.3.1 File Sharing

It is quite common that chat participants sometimes need to share files. The file sharing function is provided almost in all IM systems. Table 3-7 shows the statistics on file sharing

from the collection of chat messages. A total of 195 file sharing has occurred in 122 out of 1700 (or 7.18%) chat sessions of conversations. Among these chat sessions, images (76 occurrences or 38.97% out of all file occurrences) such as *jpg, gif, bmp* files and documents (52 occurrences or 26.67% out of total file sharing) such as PDF, Word and Excel files are most commonly shared. However, images occur only in 1.06% of total chat sessions, while documents occur in 1.94% of total number of sessions. Zip archives (i.e. compressed files) in the format of *zip, rar, gz* have 18.46% occurrences and in about 1.88% of total chat sessions. Multimedia files such as *mp3* and video files have 10.77% of total occurrences and also in 1.29% of the total chat sessions. Finally, program files such as *exe* have 1.54% of total occurrences. All other types of files occupy 3.59% of total occurrences.

Table 3-7: Statistics on file sharing.

Files	# of occurrences	% out of total occurrences	# of sessions	% out of total sessions
Images	76	38.97%	18	1.06%
Documents	52	26.67%	33	1.94%
Zip	36	18.46%	32	1.88%
Multimedia	21	10.77%	22	1.29%
Program files	3	1.54%	10	0.59%
Others	7	3.59%	7	0.41%
Total	195	100.00%	122	7.18%

3.3.2 Audio/Video Conferencing

Audio/video conferencing supports real-time communications of audio and video data, so that participants can speak to and view other chat participants. Due to the requirements of large memory space for storing audio/video conferencing data, we do not collect the actual audio/video data in this study.

In fact, the use of audio/video conferencing is highly restricted by both the environment where the conversation is conducted and the corresponding networking infrastructure. In working places such as office, audio/video conferencing is unlikely to occur. Moreover, the proxy-based LAN connection or network security administrator may also prohibit the direct IP connection for such applications. Besides, the network congestion problem will affect the quality of the transmitted audio/video data. Thus, audio/video conferencing is not considered as a common medium for IM communications. However, it is sometimes used by chat participants as a supplementary tool when there is a need for audio and visual communication during a chat session.

3.4 Summary

In this chapter, we have studied the conversational format, message characteristics and the supplementary functions of IM systems from the collection of 33,121 sample chat messages. We summarize the findings in relation to data analysis as follows:

- *Conversational format.* The conversational format preserves the correspondence between chat messages and participants. As such, statistical analysis and social network analysis based on the recorded chat messages are possible.
- *Chat language*. The chat language used for IM conversations contains acronym, short form, polyseme and mis-spelled words, which make data analysis difficult.
- *Hyperlinks* and *icons*. Hyperlinks and icons contain useful information for instant messaging. Both types of data are important for data analysis.
- *Message length*. Chat messages are short. Each chat session may have one or multiple chat messages. Instead of using each chat message as a unit for data analysis, chat messages can be organized as sessions for analysis.
- *Chat topics*. Each chat session may contain one or more topics. As such, topic detection should be able to identify one or more than one topic.
- *Audio/video conferencing* and *file sharing*. Most IM systems also provide audio/video conferencing and file sharing functions. However, we found that these functions are mainly used as supplementary functions for IM communications. Furthermore, the recording of such media data requires huge memory space and the difficulty in analyzing the data may not justify the need for monitoring such media data.

Chapter 4

Instant Messaging Monitoring

As discussed in Chapter 2, there are two categories of Instant Messaging (IM) monitoring systems: network-based and client-based. The network-based monitoring systems focus on monitoring IM activities from a centralized server within a local network, whereas client-based monitoring systems monitor IM activities directly from a PC. There are two major issues for any monitoring systems, namely message recording and system protection. Message recording is responsible for capturing chat messages from IM systems, while system protection aims to prevent the monitoring system from being detected and terminated. In this chapter, we first discuss the two categories of monitoring approaches based on message recording and system protection capabilities. We then propose a client-based monitoring approach for monitoring IM activities. The performance of the proposed approach is also analyzed.

4.1 Network-based Monitoring

In the network-based approach, the monitoring can be carried out at different locations of a local network (e.g., a gateway server) where IM data traffic can be captured. It then filters the data packets of the entire network and extracts IM data packets based on the format of the corresponding IM protocol.

4.1.1 Message Recording

Packet sniffing is a popular technique for network-based monitoring. In [31], Meehan *et al.* discussed an approach for chat room monitoring using packet sniffing, which is shown in Figure 4-1. In the chat room network, all *Chat Clients* communicate with each other through the *Chat Room Server*. *Packet Sniffer* is tapped at a port of the network where all traffic to/from the *Chat Room Server* can be captured. It then inspects regularly all data packets passing through it, and copies the packets into a separate *Monitoring Server*, where packets will be filtered and depacketized according to the format of the chat room protocol. Then, the chat messages will then be extracted and stored into a database.

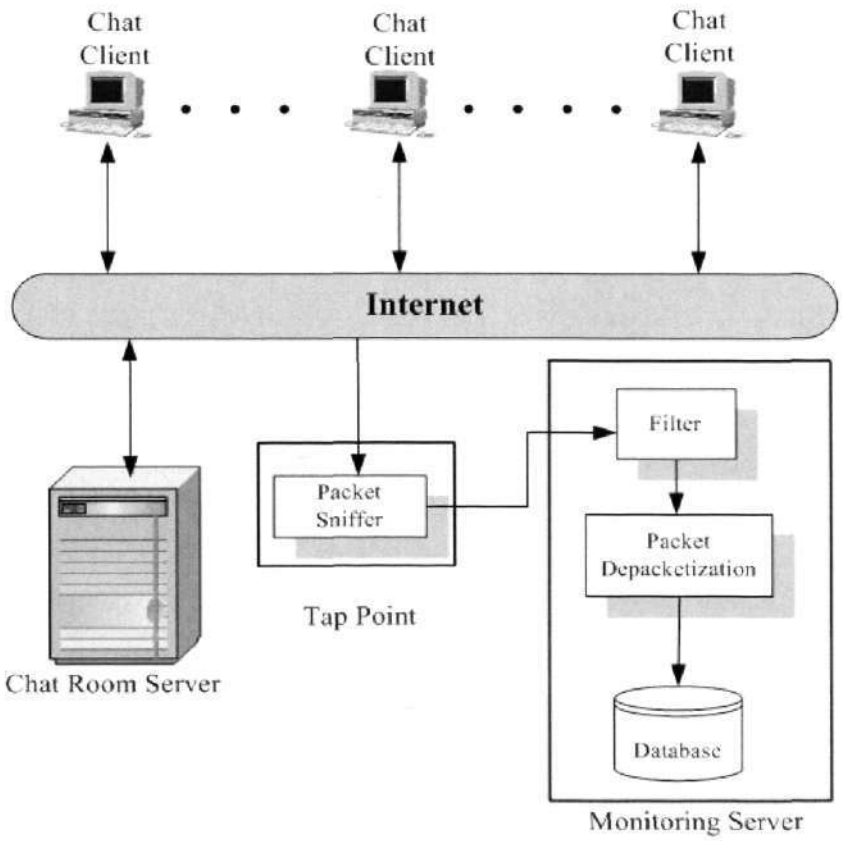


Figure 4-1: Chat room monitoring using packet sniffing.

Many commercial IM monitoring systems such as Stellar Internet IM and Akonix L7 Enterprise [32] employ a configurable gateway between the corporate network and the Internet for IM monitoring. The gateway could be a proxy server, a DNS or an HTTP tunnel. As the gateway server can intercept all inbound or outbound data packets of the corporate network, IM data packets can be captured and filtered. Chat messages are then extracted based on the format of the corresponding IM protocol.

4.1.2 System Protection

As IM monitoring systems based on network-based monitoring approach are physically separated from the monitored targets (i.e., PCs). Therefore, system protection is not really necessary for such systems.

4.1.3 Discussion

In general, network-based monitoring relies mainly on the knowledge on IM protocol for extracting chat messages. These approaches are suitable for monitoring IM messages in a network of client machines as they do not require any pre-installation or reconfiguration of the monitoring systems on the client machines. Moreover, they are quite robust and accurate if the protocol formats are known. However, such approaches still encounter the following problems:

- *Protocol Evolution*. As reviewed in Chapter 2, an IM system has a unique protocol that evolves rapidly over time. However, not much official documentation on IM protocols is publicly available. The available ones are those documented by third parties, which may be either unreliable or not up-to-date. It is a tedious task to investigate the evolved protocol used by an IM system and to re-develop the depacketization process accordingly. It gets even more difficult when different IM systems are considered together for monitoring.
- *Encryption*. An IM system running in a private encrypted network tunnel such as Virtual Private Network (VPN) [33] can possibly escape from network-based monitoring, as the VPN channel can be encrypted to protect the data flow. Apart from network level encryption, there are also third-party security applications available to encrypt data packets at client machines. Therefore, it is not easy to extract the encrypted IM messages for network-based monitoring.
- *Chat Session Reconstruction*. In network-based monitoring, chat messages from different client IPs and chat sessions (possibly from different IM systems) are received and collected together. Therefore, it is necessary to develop a mechanism to reconstruct the chat sessions for different clients. However, the reconstruction task is difficult due to the limited information available in the protocol.

4.2 Client-based Monitoring

Client-based monitoring aims to monitor IM activities from IM systems running on individual client PCs.

4.2.1 Message Recording

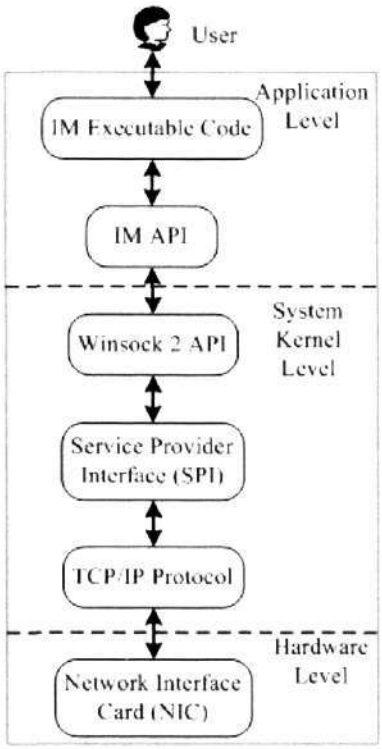


Figure 4-2: Data flow of sending/receiving of chat messages on a PC.

Figure 4-2 shows the data flow of sending/receiving of IM messages on a PC. The message data basically flows through 3 levels of the system architecture, namely Application Level, System Kernel Level and Hardware Level. Each level contains one or more than one layer.

- *Application Level.* It comprises 2 layers, namely IM Executable Code and IM API. IM Executable Code provides an interface that allows an IM user to send/receive messages to/from a contact. The underlying IM API provides functions for supporting backbone services such as packetization and depacketization of chat messages into a particular IM protocol format.

- *System Kernel Level.* It has 3 layers, namely Winsock 2 API, Service Provider Interface (SPI) and TCP/IP Protocol. Winsock 2 API provides socket-based networking infrastructure to the Windows system. It converts packetized/depacketized buffered data into/from network data packets and translates network function calls into/from the corresponding function calls in the TCP/IP Protocol. The TCP/IP Protocol layer handles network packets and provides various transmission control facilities. Service Provider Interface (SPI) is a layered protocol

that provides additional services to the TCP/IP Protocol. If SPI is implemented, data packets will pass through the SPI layer before reaching the TCP/IP Protocol layer.

- *Hardware Level*. It refers to the Network Interface Card (NIC), which does the actual work of sending/receiving packets to/from the network.

Application Level-based Recording

In client-based monitoring, there are three IM message recording techniques implemented at Application Level: Modified COM (Component Object Model) [34], Programmable APIs and Keystroke Logging.

Component Object Model is the architecture in Windows for defining interfaces and interactions among objects implemented by applications. It uses a set of pointer structures storing the addresses of related method implementations that provide services to the objects. The set of pointers are called COM interfaces. Figure 4-3 shows the monitoring of the MSN Messenger using the modified COM interface. In MSN Messenger, its chat window relies on the COM interface *ITextService* of the *RichEdit20.dll* library module, which provides functional support for rich edit text control, to support interactive conversation. In the modified COM approach, the interface pointers such as *TxSendMessage Ptr* of *ITextService* are modified to point to a dummy (recording) function *MySendMessage* that is implemented by the monitoring system. When the chat window requests the *TxSendMessage* service through the COM interface *ITextService* during a chat conversation, the control is then passed to *TxSendMessage Ptr* which in turns passes the control to the recording function *MySendMessage*. The recording function *MySendMessage* is then able to record the message data passed through it. After recording the message data, the recording function *MySendMessage* then redirects the requested *TxSendMessage* service call to *RichEdit20.dll* which then provides the original *TxSendMessage* service to the chat window as per normal.

The modified COM approach captures IM data from the application interface. There is no concern on encryption as data is already in plain text format for display. However, the modified COM approach has to alter the interface pointers to load the recording functions before any interface is initialized. And it requires great programming efforts in order to ensure that the recording function will not trigger a deadlock or race condition in the system.

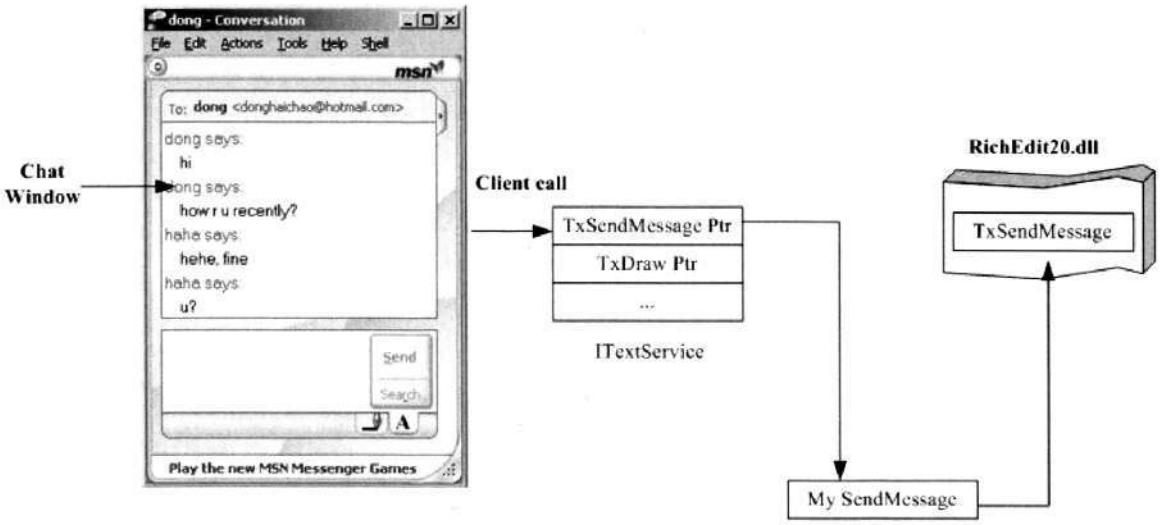


Figure 4-3: MSN Messenger monitoring using COM interface.

Programmable APIs are either provided by the original IM publisher or third-party developers, which can be used to record chat messages for a particular IM system. For example, Microsoft Visual Basic 6 contains the MSN Messenger API library [35], which enables user developed programs to interact with the MSN Messenger. However, the library only contains a limited number of functions that can be used to interact with the latest version of the MSN Messenger. Besides, to use the library interface in programming languages such as Visual C++ instead of Visual Basic 6, only very simple functions are enabled. Thus, the use of MSN API library for implementing recording functions for MSN Messenger is quite limited. There are other third-party APIs for MSN Messenger and Yahoo Messenger. However, they may not be up-to-date and have certain restrictions on the functions that they provide.

Keystroke Logging captures user input keystrokes to a particular IM system to record chat messages. The implementation of keystroke logging usually relies on a Dynamic Link Library (DLL) [36] to capture system events associated with each key pressing event. However, it is obvious that keystroke logging can only record outbound messages, while inbound messages will be completely lost. Hence, the keystroke logging approach is only a partial solution to IM recording, which is unable to reconstruct the whole conversation. As most employers or parents concern more about the information that their employees or children are giving out rather than receiving, the keystroke logging approach is still quite popular and is supported in many monitoring systems.

System Kernel Level-based Recording

In client-based monitoring, some recording techniques capture IM network data packets from different layers of the System Kernel Level on a local machine. These techniques rely on protocol specification information for extracting IM messages. For example, IBM Lotus Workplace [37] implements the IM monitoring service as SPI. Many systems implement a packet sniffer and tap it to the local listening port to log IM messages on the PC with the help of Winsock 2 API. However, such approach encounters the same problems on protocol evolution, data encryption and chat session reconstruction as the network-based monitoring.

4.2.2 System Protection

Client-based monitoring requires system protection techniques to prevent it from being detected and terminated. There are mainly two forms of system protection techniques, namely hiding and prevention techniques. Both techniques can be adopted together for system protection.

- *Hiding Technique*. It aims to create a false impression of non-existence of the monitoring system in the Windows system. The typical hiding techniques hide the monitoring process from desktop by registering the monitoring system as a system service. However, this is only useful for hiding it from novice computer users, as the monitoring process can be detected easily by the Task Manager. Another way to hide IM monitoring systems from the Task Manager is to de-register the IM monitoring system from Windows memory space so that the process information is not reflected by the Windows system. Further, the monitoring system may be disguised as the Windows system process by adopting a process name similar to Windows system services.
- *Prevention Technique*. It aims to prevent users from terminating the monitoring process. A common technique is to provide password protection on access rights for the monitoring system. Only authorized users are allowed to access or terminate the IM monitoring system. Some systems go even further to disable user access to the Task Manager so that IM monitoring process cannot be terminated.

4.2.3 Discussion

In System Kernel Level-based Recording, it still encounters the problems faced by network-based monitoring on encryption and protocol evolution. In Application Level-based Recording, the modified COM interface technique captures displayed messages. Therefore, it does not need to concern about the encryption problem. Instead, it requires very good programming techniques to maintain program stability in order not to affect the system's normal operation. The keystroke logging technique captures user keyboard events. Thus, it is independent of IM systems. It does not face the encryption problem. The IM API libraries only provide a limited number of functions. In terms of system protection, the existing techniques have mainly focused on process hiding and prevention. However, the system protection techniques in most systems, especially hiding, are quite poor in performance.

4.3 Proposed Monitoring Approach

In this research, we propose an IM monitoring approach which aims to tackle the major problems discussed in earlier sections. In particular, we design our approach with the following considerations in mind:

- *Adaptability*. As IM systems and protocols evolve rapidly with added new features and functions, the IM monitoring approach should be easily adapt to new versions and releases.
- *Data Encryption*. As some IM systems may incorporate encryption on data, the IM monitoring approach should also be able to capture such data from those systems.
- *Hiding Ability*. It is important that the IM monitoring approach should have a good hiding ability. So that it will not be easily detected and terminated by users.
- *Personal Information*. It is also very important for the IM monitoring approach to be able to extract user-related information that can then be used for the purpose of user identification during message analysis.
- *Online Monitoring*. IM chat messages should be recorded in real-time for supporting online IM monitoring and analysis.

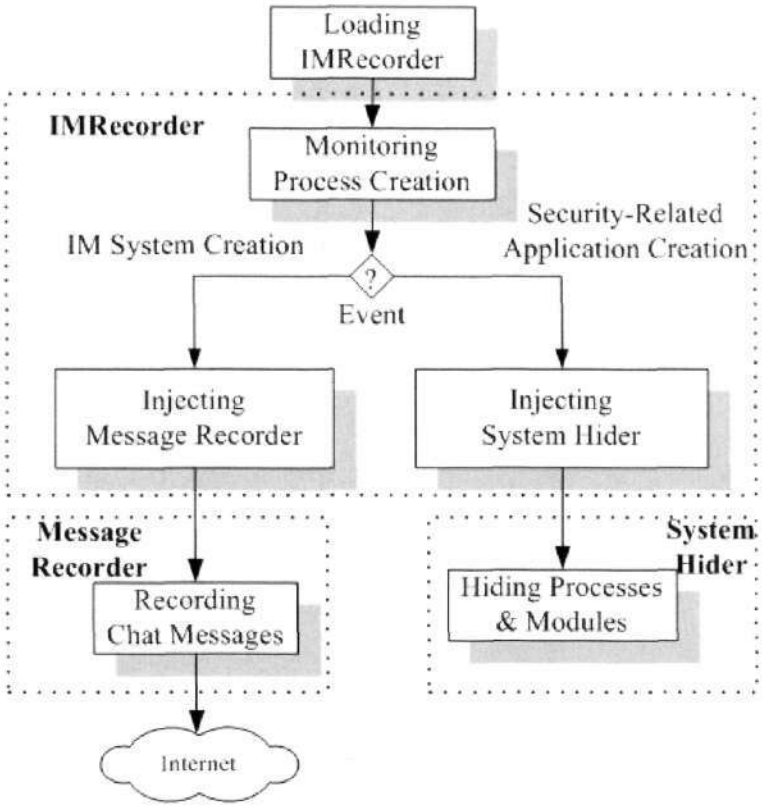


Figure 4-4: Proposed IM monitoring approach.

To achieve the stated purposes, we propose a client-side IM monitoring approach. The proposed approach consists of the following three program modules: *IMRecorder*, *Message Recorder* and *System Hider*, which are shown in Figure 4-4.

IMRecorder is an executable code (with file extension *.exe*) which is loaded during Windows' initialization stage (i.e., *Loading IMRecorder* in Figure 4-4). The *IMRecorder* module carries out the following tasks:

- *IMRecorder* monitors process creation of any IM systems and injects the *Message Recorder* module into the newly created IM system memory space to obtain the access right to the IM system's memory.
- At the same time, *IMRecorder* monitors process creation of any security-related applications (such as Task Manager) and loads the *System Hider* module into their memory space for monitoring system protection.
- In addition, *IMRecorder* also retrieves user's personal information which includes client PC's IP address and user's Windows account.

To inject *Message Recorder* and *System Hider* into the appropriate processes, *IMRecorder* sets up a system-wide hook to monitor process creation events. A hook [38] is a

trap in the system message-handling mechanism, which employs a hooking function to act on events. When an event creation of the IM system or security-related application occurs, *Message Recorder* or *System Hider* module is then mapped into the respective memory space by the hook. Thus, the modules become part of the processes and obtain access right to alter the corresponding processes' memory content.

The *Message Recorder* and *System Hider* will be discussed in the following two sections.

4.4 Message Recorder

The *Message Recorder* module, which is a DLL (Dynamic Link Library) module, is responsible for the detection, extraction and transmission of messages from IM systems. In this module, both the System Kernel Level-based Recording (or protocol-based) and Application Level-based Recording (or chat window-based) methods are proposed for message extraction. The two methods can be used to support monitoring for different IM systems. Figure 4-5 shows the *Message Recorder* module which performs the following three steps: Packet Detection, Message Extraction and Transmission:

- *Packet Detection*. This step detects the existence of IM data packets in real-time in order to support online monitoring.
- *Message Extraction*. This step extracts newly available message data. There are two methods for message extraction: protocol-based and chat window-based. In Protocol-based Message Extraction, IM chat messages are extracted based on IM protocol format. In addition, session detection is also incorporated to stamp each message with a session identifier. A session contains all chat messages from a chat window creation to its closure. In Chat Window-based Message Extraction, chat messages are extracted based on the hierarchical data structure of chat window interface. It also supports session detection.
- *Message Transmission*. This step packetizes the extracted messages and session identifier of each client machine for transmission to the centralized server for storage. The message transmission mechanism will be discussed in Chapter 4.

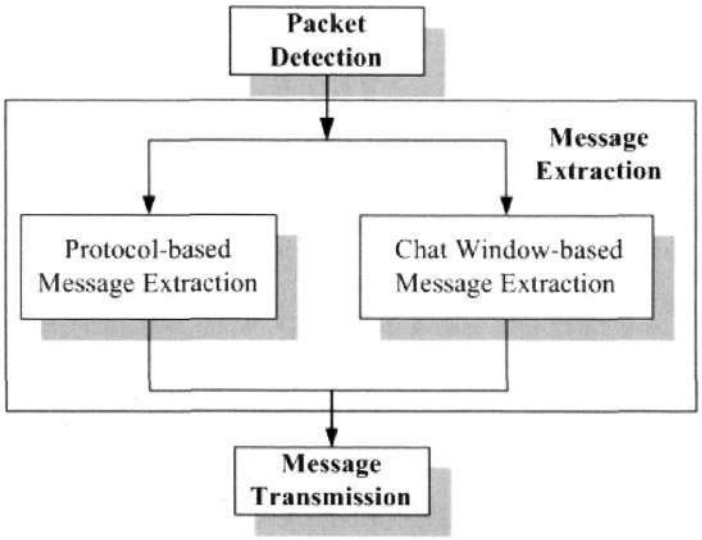


Figure 4-5: Message Recorder.

4.4.1 Packet Detection

In this step, *Message Recorder* employs the Winsock 2 API interception technique to detect the availability of IM data packets in real-time. This technique alters IM process memory contents and obtains the control of IM systems' function call to Winsock 2 API for Packet Detection. In particular, the network data packet sending and receiving functions provided by Winsock 2 API are to be intercepted.

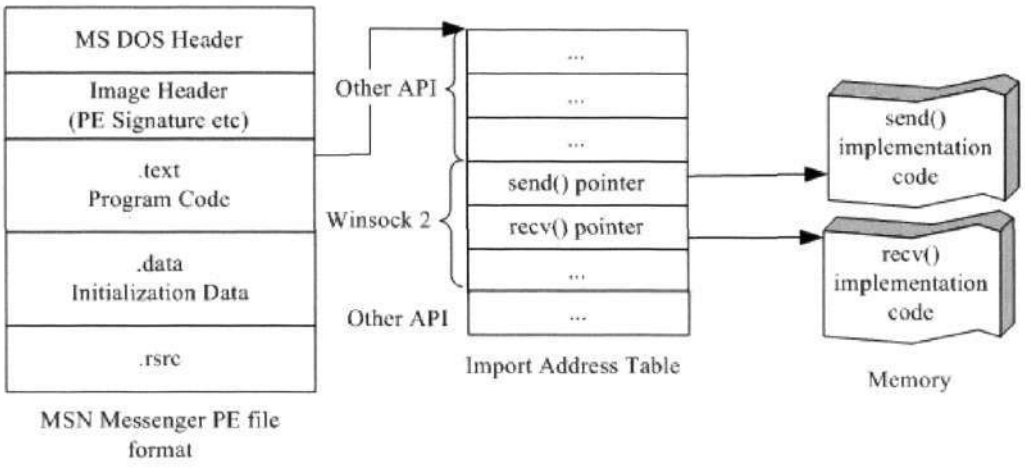


Figure 4-6: Structures for Portable Executable and Import Address Table.

The Winsock 2 API interception technique relies on understanding Portable Executable (PE) [39, 40] file format, which is commonly used for Windows applications on all supported platforms. A PE file is a concatenation of data stored in sections. Figure 4-6 shows the MSN Messenger's PE file format which contains 3 sections, namely *.text*, *.data* and *.rsrc*. The *.text*

section contains an Import Address Table (IAT) to store pointers to all imported API function implementations in the memory as shown in Figure 4-6. The addresses in these entries are used to locate the actual implementation of an imported function and perform the corresponding action during runtime.

To perform the interception, the IAT table entries corresponding to the IM network sending and receiving function pointers are replaced with addresses pointing to dummy (recording) functions implemented in the *Message Recorder*. By this way, the *Message Recorder* can obtain the control of chat data when the IM system performs the sending or receiving function call to Winsock 2.

Figure 4-7 compares the IM data flow of MSN Messenger before and after Winsock 2 API interception. The Winsock 2 API interception inserts the *Message Recorder* between the MSN Messenger's executable code and Winsock 2 module. After thorough inspection, we found that the Winsock 2 functions send() and recv() are used in MSN Messenger for network packet sending and receiving respectively. Therefore, when MSN Messenger makes a function call to send() or recv(), the chat messages are directed to the *Message Recorder*'s dummy functions *mysend()* and *myrecv()* for message recording. After that, the function calls will be redirected back to the original Winsock 2 function implementation in memory to perform the actual networking function. Table 4-1 shows the network sending and receiving functions to be intercepted for different IM systems including MSN Messenger, Yahoo Messenger, ICQ and QQ.

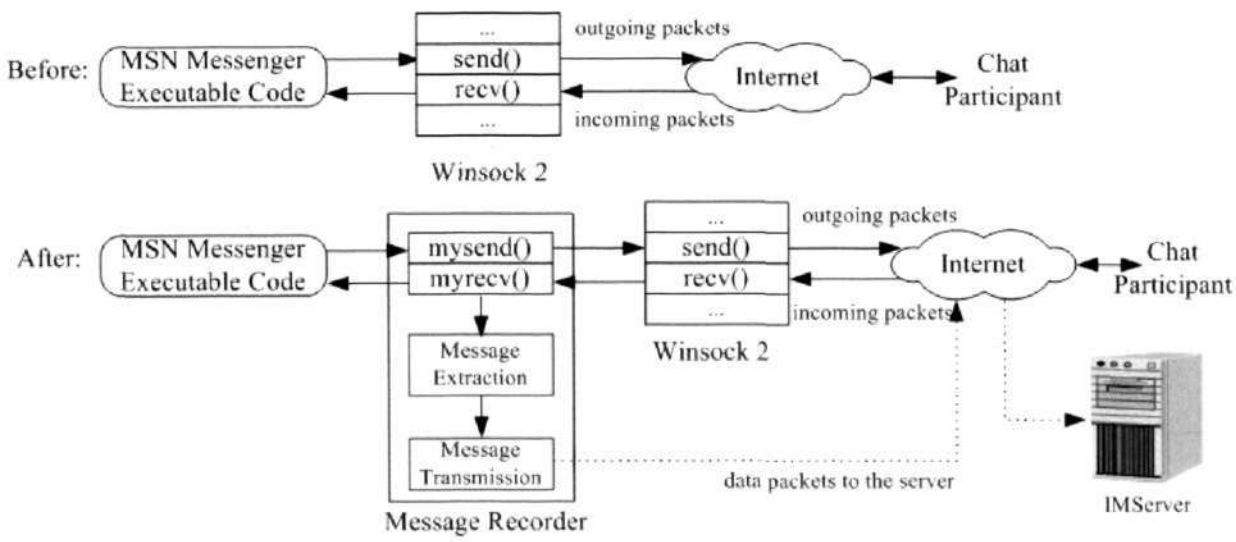


Figure 4-7: Winsock 2 interception for MSN Messenger.

Table 4-1: IM Winsock 2 functions.

	MSN Messenger	Yahoo Messenger	ICQ	QQ
Sending	send()	send()	send()	sendto()
Receiving	recv()	WSARecv()	WSARecv()	recv()

4.4.2 Message Extraction

Message Extraction receives IM data packets detected from Packet Detection in real-time. Two methods, namely protocol-based and chat window-based, are proposed for extracting newly available chat messages. As the names imply, the former extracts chat messages directly from the data packets based on the IM protocol format, while the latter extracts chat messages from chat windows.

Protocol-based Message Extraction

The Winsock 2 API interception for detecting chat messages has given the *Message Recorder* the direct access to the IM data packets. Therefore, Message Extraction can inspect the data packets to extract chat messages. Apart from chat messages, session information will also be identified for the purpose of reconstructing chat sessions at the server.

The protocol-based message extraction and session detection method act on three types of events, which occur during the lifespan of a chat session. The three types of events are Session Initiation, Message Exchanging and Session Termination.

- *Session Initiation*. This occurs when a chat session is newly initiated.
- *Message Exchanging*. This occurs when chat messages are exchanged between participants after a chat session has already established.
- *Session Termination*. This occurs when a chat session ends.

Session detection is performed with the help of a Session Mapping Table in the memory. Each entry in the table contains 5 fields storing information that uniquely identifies a chat session:

- *Local Account*. It refers to the monitored IM user account in the target client PC.
- *Remote Account*. It is the participant account with whom the Local Account has a chat session.
- *Remote Nickname*. It is the nickname of the Remote Account and can be optionally stored.
- *Channel*. It is a unique channel through which the two parties chat with each other.

- *Session Identifier*. It combines a timestamp with an integer counter to assign each chat session with a unique identifier.

Figure 4-8 gives the protocol-based message extraction method which comprises 4 steps, namely Packet Inspection, Session Initialization, Message Exchanging and Session Termination. The following discussion will use MSN Messenger as an example to illustrate the various steps of the protocol-based message extraction method.

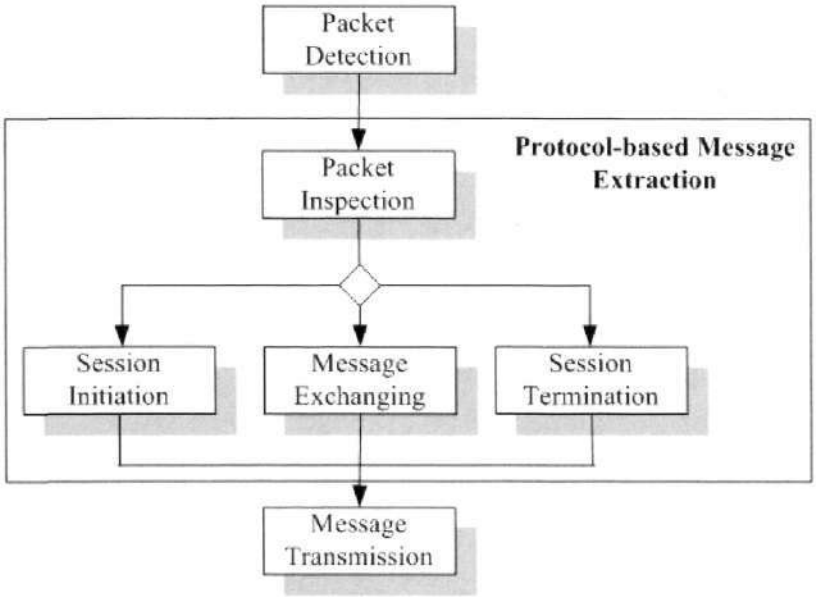


Figure 4-8: Protocol-based message extraction.

Packet Inspection. This examines input IM data packet sequence from Packet Detection to identify the current event type based on an IM packet identifier and filter out other data packets. An IM packet identifier is a fixed data field located at the beginning of a data packet (excluding the TCP/IP header). It allows IM systems or servers to determine the packet type and depacketize the packet. For example, Figure 4-9 shows the unique MSN Messenger packet sequence exchanged for an active session connection initiated by the monitored user. The italicized characters are packet identifiers. In MSN Messenger chat session initiation, the first two *XFR* packets (packets 1 and 2) set up a Switchboard (SB) session from the server, and the following two *USR* packets (packets 3 and 4) establish a TCP connection channel for the conversation. Finally, the *CAL* packets (packets 5 and 6) are used to invite "remote@hotmail.com" to join the chat session connection and the *JOI* packet (packet 7) is a confirmation of this request. Similarly, Figure 4-10 shows the unique packet sequence of a passive session connection initiated by a contact received by the monitored user using MSN Messenger. The identifiers are italicized as well. Figure 4-11 and Figure 4-12 give two

example MSN message data packets for the receiving and sending functions respectively. Both have *MSG* as the packet identifiers.

The inspection of packet identifier does not rely on the IM protocol format. Since packet identifier is a fixed data field, it has a fixed set of possible values to provide information for data depacketization by IM systems or servers. Therefore, the set of packet identifier values can always be studied in a heuristic manner for determining the event types.

```
1: XFR 15 SB
2: XFR 15 SB 207.46.108.39:1863 CKI 165140.1115642417.12194
3: USR 2 local@hotmail.com 165140.1115642417.12194
4: USR 2 OK local@hotmail.com local_name
5: CAL 4 remote@hotmail.com
6: CAL 4 RINGING 165140
7: JOI remote@hotmail.com remote_name
```

Figure 4-9: Data packet sequence for MSN Messenger active chat session.

```
1: RNG 110391 207.46.108.135:1863 CKI 1115633441.4717
   remote@hotmail.com remote_name
2: ANS 1 local@hotmail.com 1115633441.4717 110391
3: IRO 1 1 1 remote@hotmail.com remote_name
4: ANS 1 OK
```

Figure 4-10: Data packet sequence for MSN Messenger passive chat session.

```
MSG remote@hotmail.com remote_name 135
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
X-MMS-IM-Format: FN=MS%20Shell%20Dlg; EF=; CO=0; CS=86; PF=0
11111
```

Figure 4-11: An example MSN Messenger message receiving packet.

```
MSG 4 N 133
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
X-MMS-IM-Format: FN=MS%20Shell%20Dlg; EF=; CO=0; CS=86; PF=0
11111
```

Figure 4-12: An example MSN Messenger message sending packet.

Session Initiation. When session initiation occurs, this step creates a new session entry into the Session Mapping Table. The creation of a new entry is based on the IM protocol format. For example, when the active session initiation packet sequence (as shown in Figure 4-9) occurs in MSN Messenger, the Local Account (local@hotmail.com) and the Remote Account (remote@hotmail.com) as highlighted in packets 4 and 7 can be extracted based on the IM protocol format. The Channel field contains the handle of the current chat window, which uniquely indicates the existence of the chat session. The Session Identifier for the newly created session is formed by combining the system time of initiating the session

together with a session counter. As a result, a new entry with all the necessary information is created in the Session Mapping Table. When a passive session shown in Figure 4-10 is received, another entry creation will be performed in a similar manner.

Message Exchanging. This step is invoked when message exchange occurs. It extracts newly available chat message content from the chat message data packets, after which it attaches a session identifier retrieved from the Session Mapping Table to that particular chat message. The IM protocol format plays a key role in this step. For a received chat message data packet shown in Figure 4-11, the chat message content "11111" will be extracted and the Remote Account (remote@hotmail.com) will be used to look up the mapping table for Session Identifier. On the other hand, the sent data packet in Figure 4-12 does not contain explicit information for session identification. Thus, the active chat window's handle value at the time of message sending will be used to search the Channel field of the Session Mapping Table and retrieve the corresponding Session Identifier.

Session Termination. This step deletes a session entry from the Session Mapping Table when a session is terminated. In this case, a chat window closing event is captured by the system-wide hook. The closing window handle can be obtained by searching the Channel field of the Session Mapping Table and the corresponding session entry is then deleted.

Chat Window-based Message Extraction

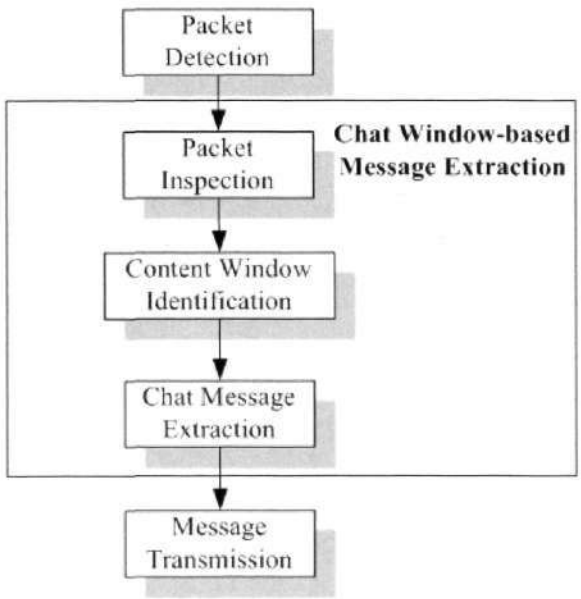


Figure 4-13: Chat window-based message extraction.

This method extracts chat messages from the application level, in particular, the chat content window of chat dialog window. It does not rely on the IM protocol format. Therefore, it does not face the data encryption problem encountered by the protocol-based message extraction method. Figure 4-13 shows the chat window-based message extraction method, which consists of the following steps: Packet Inspection, Content Window Identification and Chat Message Extraction.

Packet Inspection. This step is similar to its counterpart in the protocol-based message extraction method except that it only examines the existence of data packets containing chat messages in order to trigger Content Window Identification and Chat Message Extraction in real-time. The data packets from Packet Detection are then discarded.

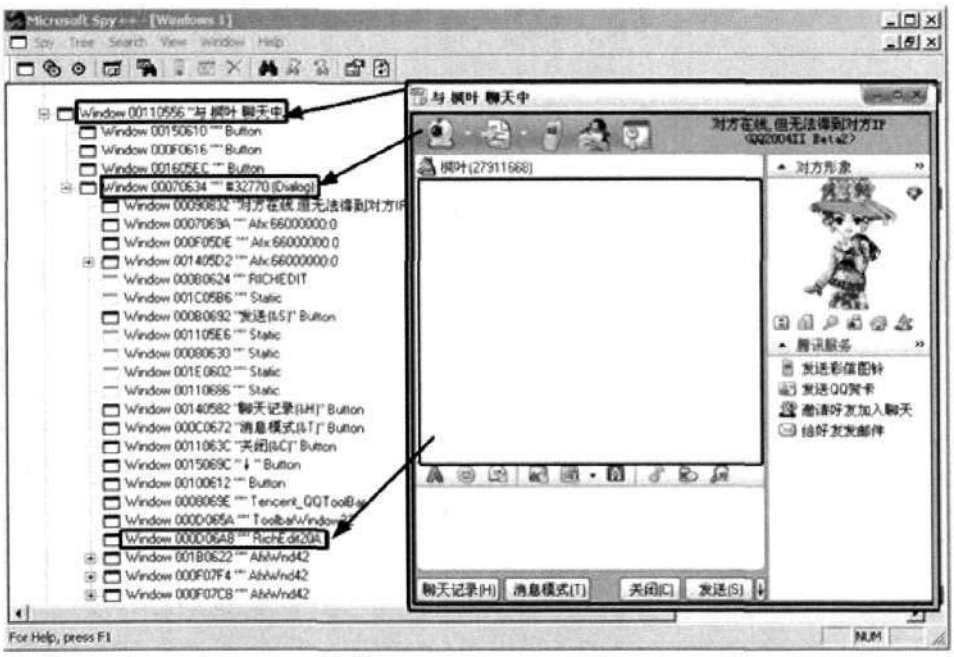


Figure 4-14: QQ chat interface and window class inspected by Spy++.

Content Window Identification. This step extracts the chat content window component from IM chat window. It relies on the knowledge of the hierarchical IM chat window structure, which can be easily identified. For example, a QQ chat window structure examined by Microsoft Spy++ [41] is shown as a tree in Figure 4-14. Each child window object in QQ chat window is displayed as a tree node labelled with its corresponding class and the window handle value. The root node highlighted at the top corresponds to the QQ chat window. It has four first level child nodes including three "*Buttons*" and a *"#32770"* (or dialog). This dialog object in turn has multiple child windows as leaf nodes, one of which is an instance of the

class *RichEdit20A* as highlighted in the Figure 4-14. This parent-child chat window hierarchical structure provides a solution for automatic chat content window identification.

Chat Message Extraction. This step aims to extract newly available chat messages contained in the identified chat content window. The chat window-based message extraction method is IM system dependant, as the chat content windows of different IM systems used for message display are different. Table 4-2 gives the class names of chat content windows used by different IM systems. Nevertheless, the messages contained in the chat content window can be obtained with Windows tools such as API library and the messaging mechanism. In addition, each chat message will be stamped with a unique chat window handle value as session identifier for the purpose of session reconstruction at the server end.

Table 4-2: IM chat content windows.

	Yahoo Messenger	ICQ	QQ	MSN Messenger
Class name	Internet Explorer_server	RichEdit	RichEdit20A	DirectUI

Discussion

The protocol-based message extraction method has the advantages in simple chat message extraction (directly from data packets) and preservation of chat sessions with some additional efforts in developing the session reconstruction mechanism. However, it still suffers from the problems on data encryption and protocol evolution. On the contrary, the chat window-based message extraction method is independent of protocol formats. Therefore, it does not need to handle data encryption. It also does not require much effort in terms of session reconstruction. But instead, it is necessary to develop programs for automatic content window identification and chat message extraction from the interface level even though the interface architecture can be easily examined.

Table 4-3 gives the applicability of both message extraction methods for different IM systems. As shown, the protocol-based method can be applied to extract ICQ, Yahoo Messenger and MSN Messenger. However, it does not apply to QQ because QQ protocol is not publicly available and QQ data packets are encrypted. On the other hand, the chat window-based method can be applied to extract chat messages from any IM systems except MSN Messenger. As shown in Table 4-2, MSN Messenger is examined to use an instance of the class *DirectUI* for displaying chat content. However, the wrapper class *DirectUI* is not documented by Microsoft. And there is no information on how to access the wrapper content.

Nonetheless, the combination of two chat message methods can be applied to monitor personalized chat messages with the preservation of natural session coherency in a real-time manner and easy adaptation to evolutions of IM systems.

Table 4-3: Applicability of the two message extraction methods for IM systems.

	Protocol-based	Chat Window-based
ICQ	Yes	Yes
Yahoo Messenger	Yes	Yes
MSN Messenger	Yes	No
QQ	No	Yes

4.5 System Hider

The System Hider is a DLL module, which is loaded into security-related applications' memory space. It aims to protect the client-side IM monitoring system from detection and termination. It adopts Windows Native API and GDI (Graphics Device Interface) interception to hide the active process *IMRecorder* and DLL modules (the *Message Recorder* and *System Hider* itself) from security-related applications. Windows Native APIs contain functions for internal system use, whereas GDI provides the graphical and textual display infrastructure of Windows.

As many existing IM monitoring systems are unable to hide themselves well from the Task Manager, we examine the process monitoring service of Task Manager which is shown in Figure 4-15. The process monitoring service comprises two major steps, namely *Process Information Retrieval* and *Display*. In *Process Information Retrieval*, it obtains a *process list* by capturing a system-wide snapshot. The process list consists of all processes running in the system's current memory space. Then, it traverses the process list to retrieve specific information of each process for *Display*. The Task Manager periodically performs the two steps to dynamically monitor system processes until the Task Manager itself is terminated.

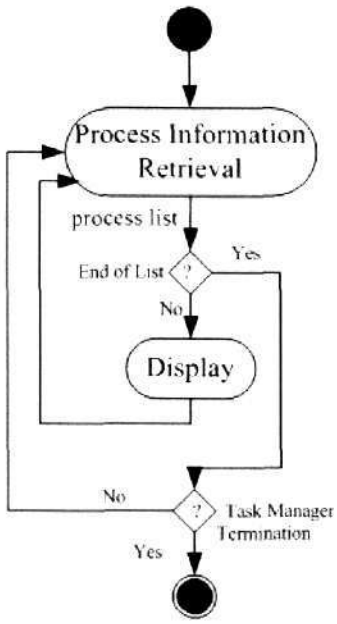
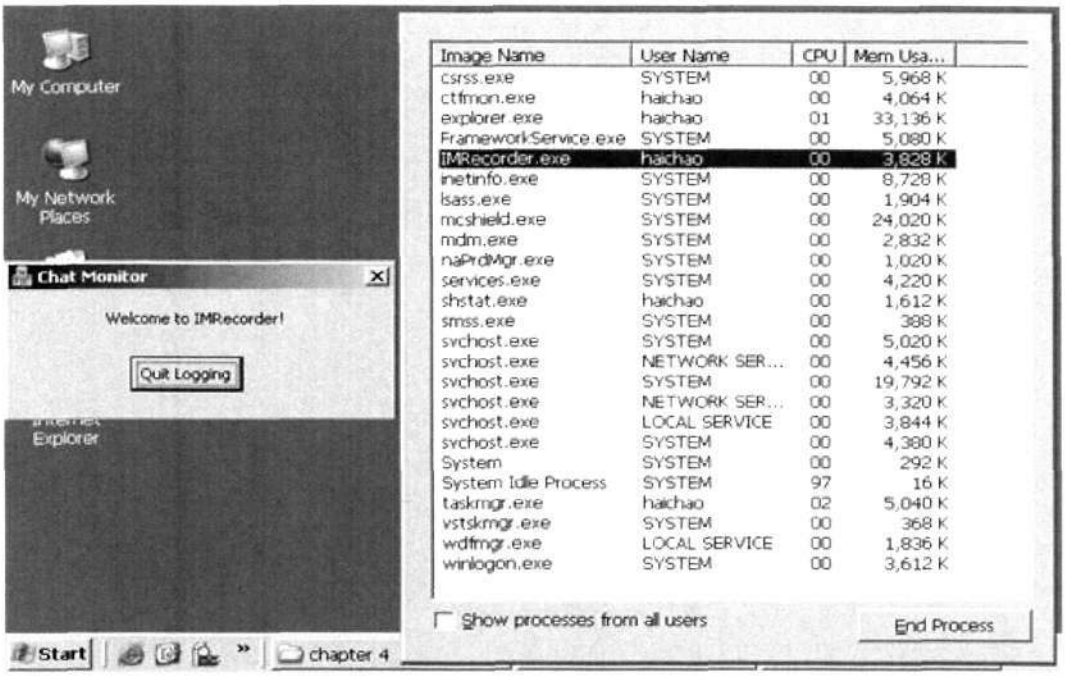


Figure 4-15: Process display in Task Manager.

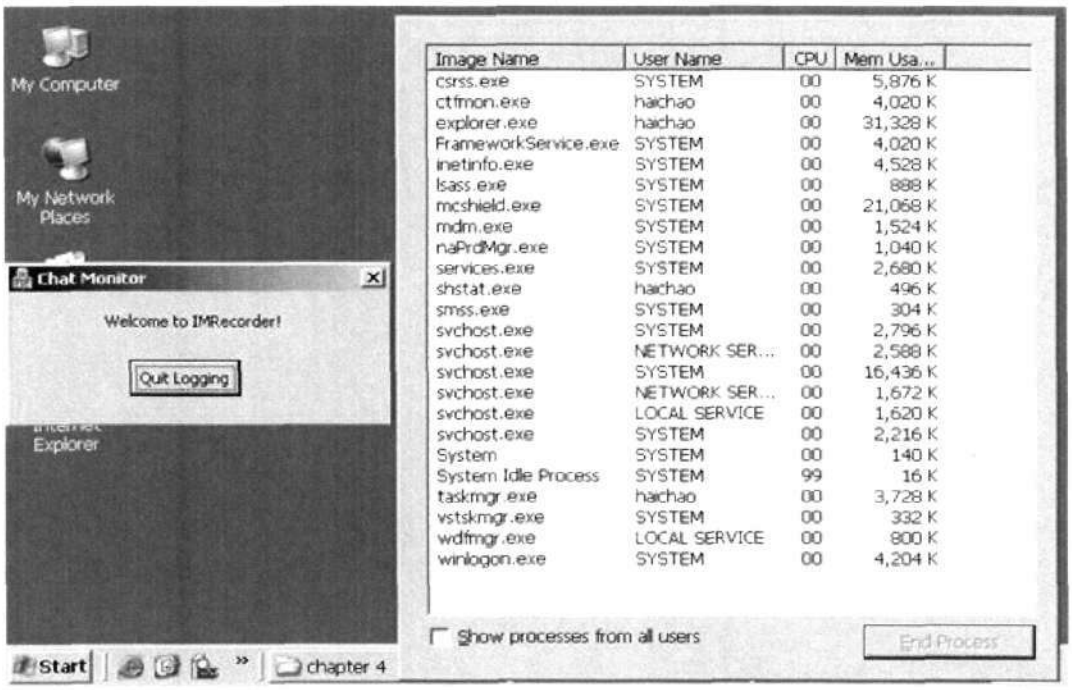
In the System Hider, both Native API and GDI interceptions are performed in order to manipulate the information flow of the two major steps of process display of the Task Manager:

- *Windows Native API Interception*. Through heuristic inspection, we found that the Task Manager employs Windows Native API function *NtQuerySystemInformation()* to obtain the process list. Therefore, Windows Native API Interception can intercept *NtQuerySystemInformation()* with a dummy function that filters out the *IMRecorder* process from the process list.
- *GDI Interception*. Again, through heuristic inspection, we found that Task Manager uses the GDI function *ExtTextOutW()* for process information display. Therefore, GDI Interception can intercept *ExtTextOutW()* so that the *IMRecorder* information is omitted for display.

Most security-related applications have similar monitoring mechanisms to the Task Manager of the Windows system for the detection of processes or library modules. Therefore, both system hiding methods can be adapted to hide the *IMRecorder* process and DLL modules from security-related applications. Moreover, the System Hider can also be used to hide program modules from Windows Explorer and Registry Editor. In such cases, it will be more difficult for users to detect the presence of the IM message monitoring components in the system.



(a) Screen shot before enabling hiding.



(b) Screen shot after enabling hiding.

Figure 4-16: Screen shot for IMRecorder hiding from Task Manager.

Figure 4-16 shows an example on illustrating the system hiding capability for *IMRecorder*. In Figure 4-16(a), the left window with the title "Chat Monitor" displays a running instance of the *IMRecorder* process which is developed for illustration purpose. The

Task Manager displays the information of all processes running on the system including the *IMRecorder* process (highlighted as "IMRecorder.exe") when hiding is not enabled. In Figure 4-16(b), the Task Manager displays the process list that does not have the corresponding "IMRecorder.exe" process after hiding is enabled.

4.6 Performance Analysis

In this section, we will discuss the performance of the proposed chat monitoring client comprising *IMRecorder*, *Message Recorder* and *System Hider*. As discussed in Section 4.4, the *Message Recorder* DLL intercepts network API calls of IM systems. This interception is carried out by the system service *IMRecorder*. Effectively, a low-resource gateway is installed for each IM system and the chat monitoring client is able to capture all chat messages. In terms of CPU utilization, the event-driven client-side components only require the minimum of system resources and do not incur any obvious CPU overhead. As a result, the evaluation in this section will only focus on memory utilization of the individual client-side monitoring components.

To measure memory utilization, the IM monitoring system is firstly started to monitor MSN Messenger's chat messages in a Pentium 4 3.0 Gigahertz PC with 1 Gigabyte RAM running under the Windows XP system. During the monitoring, the MSN Messenger continuously sends out chat messages with maximal allowable length (400 bytes) at a rate of 1 message per second. This is much faster compared with the message rate of 20 seconds (i.e. 0.05 message per second) between messages discussed in Chapter 3. As the *Message Recorder* and *System Hider* are library modules injected respectively into the memory space of the MSN Messenger and Task Manager, the modules' memory utilization is measured by comparing the memory utilization of MSN Messenger and Task Manager before and after the respective modules get injected. The results for the individual monitoring system components are obtained based on an average value of 20 measurements.

Table 4-4: Statistics on memory usage for individual client-side monitoring components.

	Memory Usage (KB)	Percentage (from 256MB RAM)
IMRecorder	804	0.31%
Message Recorder	404	0.15%
System Hider	72	0.03%
Total	1280	0.49%

Table 4-4 shows the statistics on memory utilization for the individual client-side monitoring components. As shown in the table, *IMRecorder* has an average memory usage of 804 KB. The *Message Recorder* typically requires less than 404 KB memory, which is far greater than the *Message Recorder* file size stored on hard drive (100 KB). This is probably due to the memory reservation for dynamic data members holding the maximal allowable chat messages in memory. The *System Hider*, on the other hand, requires only 72 KB memory.

The last column of Table 4-4 calculates the percentage of each component's memory usage based on RAM size of 256 MB, which is the minimal configuration for modern computers. The three client monitoring components are shown to consume 0.31%, 0.15% and 0.03% of the total memory respectively, which sum up to 0.49% in total. Therefore, the memory usage of the client side IM monitoring system is insignificant and negligible.

4.7 Summary

In this chapter, we have reviewed the existing IM system monitoring techniques. However, the existing monitoring techniques face problems on data encryption, protocol evolution and session reconstruction. Moreover, they also lack of effective hiding techniques to hide the system from detection and termination. To overcome these problems, we have proposed an IM monitoring approach consisting of three program modules, namely *IMRecorder*, *Message Recorder* and *System Hider*, to support real-time message recording and hiding. The *Message Recorder* employs both protocol-based and chat-window based methods for the extraction of chat messages. The proposed chat-window based method is able to overcome both the data encryption and adaptability problems. Session detection is also incorporated in order for session reconstruction. In addition, personal information such as client IP, user Windows account and monitored IM user account are also recorded for user identification. The *System Hider* incorporates both Windows native API and GDI interception techniques to provide hiding capability from the Windows system and security-related applications. Finally, the memory utilization of the client-side IM monitoring system has been analyzed, which has shown that the proposed client-side IM monitoring approach is resource efficient.

Chapter 5

Adaptive Message Transmission

The popularity of Instant Messaging (IM) systems is greatly attributed to its real-time communication capability provided for users. For client-server based IM monitoring systems, it is necessary to transmit messages to the server for storage and at the same time, monitor chat conversations online. As such, an efficient message transmission mechanism is required. In this chapter, we propose a server-based adaptive message transmission mechanism for transmitting monitored chat messages from each target client to a server.

5.1 Network Protocols

Network protocols establish the communication channel between client and server. It facilitates the exchange of data between them in a format which is understandable by both parties. In this section, we review several basic network protocols.

5.1.1 Transmission Control Protocol (TCP)

TCP [17] is a connection-oriented network transport protocol providing reliable transmission of packets between two end-points on the network. It is responsible for assembling data passed from higher layer applications into TCP packets and ensuring that the data is transmitted and received correctly and in its entirety. The delivery of packets is guaranteed to be in sequence. If a packet is corrupted or lost, TCP will retransmit the packet to ensure in-sequence arrival and error free. This is achieved by using acknowledgement and packet sequence numbers. TCP usually runs on top of IP (Internet Protocol) for transmitting data packets on the Internet.

5.1.2 User Datagram Protocol (UDP)

UDP [20] is a connectionless network transport protocol for the delivery of data packets. It runs on top of the IP protocol. Unlike TCP/IP, UDP/IP provides very few error recovery services. As such, packet loss may arise, and it is not as reliable as TCP. On the other hand, UDP does not dedicate a network path to send and receive datagrams over an IP network.

5.1.3 Real-time Transport Protocol (RTP)

RTP [42] is a protocol designed for the delivery of data for real-time services, which runs on top of UDP to make use of its multiplexing and checksum services. In [43], RTP for real-time text transmission is specified. It assumes that each text data packet is small in size, typically with a few characters. The data must be encoded using the UTF-8 standard. However, RTP does not provide any mechanisms to ensure quality of services (QoS). In addition, RTP does not guarantee packet delivery in sequence.

5.1.4 Discussion

TCP/IP dedicates a network channel for reliable communication. When errors occur, subsequent transmission of data packets will be delayed until the error is recovered. In addition, the communication channel established does not guarantee the shortest network path. Therefore, TCP/IP does not provide a fast transmission channel although it is reliable. This is especially undesirable for online, real-time transmission. Besides, when delay occurs, subsequent data packets are queued up waiting for transmission, resulting in higher memory usage at the client side.

UDP/IP, on the other hand, lacks of error recovery and in-sequence delivery guarantee, UDP data packets are transmitted immediately without being queued up for retransmission of previous packets when errors occur. Therefore, UDP guarantees the fastest communication on the current network situation. However, packet loss will be inevitable in UDP transmission. Further, packets may arrive out of sequence and duplicated data packets may also occur.

RTP is originally designed for real-time audio and video communications. It sits on top of UDP without guaranteeing QoS and in sequence arrival.

5.2 Considerations for Message Transmission

For client-server based IM monitoring systems that require to transmit chat message data from the target client to the server for logging and online monitoring, it is necessary to design a message transmission mechanism with the following considerations:.

- *Online delivery*. The recorded messages should be delivered on time to the server to cater for the need of online chat monitoring and analysis. In addition, it is necessary

to hide the monitoring component from detection, the monitoring component should minimize the use of resources at the target clients. As such, the monitoring component should transmit chat messages out to the server as soon as possible.

- *Reliability*. As network traffic varies from time to time, packet loss may occur at times and important information may be lost. Thus, a reliable transmission mechanism is necessary in order to minimize packet loss.

5.3 Adaptive Transmission Control Mechanism

Research on adaptive redundancy transmission control and recovery mechanism is traditionally carried out mainly on audio, video and fax data in order to support real-time transmission requirement in many Internet multimedia applications [44-46]. Our client-server based IM monitoring system requires a similar real-time communications over the Internet for the transmission of chat messages from a monitored target client to the server. At the same time, the packet loss should be minimized. However, there are differences between the client-server based IM monitoring system and the existing multimedia communication applications.

In the existing multimedia applications, either a point-to-point communication or a central server catering for receiving clients that are independent is used. The clients in these applications are dedicated to transmission of data packets. Thus, the adaptive transmission control mechanisms are focused on using client-side adaptive control mechanisms, where the workload of adaptive transmission control and recovery is mainly performed at the client-side machine. Moreover, the packet loss recovery mechanism is based on a point-to-point approach.

In contrast, in our client-server based IM monitoring system, the monitoring components resides at target clients, which are not dedicated for monitoring purposes. And it is crucial for the monitoring component to minimize the utilization of resources such as memory and CPU at the target client. Moreover, the IM monitoring system is an integrated data recording application that consists of a single receiving end (i.e., server) with multiple data sources (i.e., target client set) sharing a common network. Also the communication data is very sparse and light. As such, a mechanism that recovers packet loss for all clients aggregately is more appropriate.

Due to the different requirements of chat message communication from audio and video communication, we propose a new server-based adaptive redundancy transmission control and recovery mechanism for chat message transmission.

5.4 Data Packet Format

To cater for real-time data packet delivery, we select UDP as the base protocol, since UDP is appropriate for on-time delivery that does not wait for acknowledgement. To facilitate the transmission between the client and server, the message packet format is based on UDP packet format which is given in Figure 5-1. The message data packet is started with a normal UDP header followed by the Redundancy Transmission Header and Message Data Field. Redundancy Transmission Header stores information about the data messages assembled in the Message Data Field.

5.4.1 Redundancy Transmission Header

The Redundancy Transmission Header consists of 4 fields, namely, Sequence No, Redundancy Flag, Compression Flag, and Segment No.

- *Sequence No*. The sequence number indicates the data packet's sequence from a particular client (i.e., sender).
- *Redundancy Flag*. The byte-sized redundancy flag reflects whether redundancy transmission is incorporated and the number of redundancy data that should be assembled in the packet.
- *Compression Flag*. This field indicates the data compression method used to compress the redundancy data if multiple compression methods are available.
- *Segment No*. This segment number is used when a long message is divided into a series of smaller packets for transmission. The segmentation number starts from 1, a 0 in the field indicates non-segmented data.

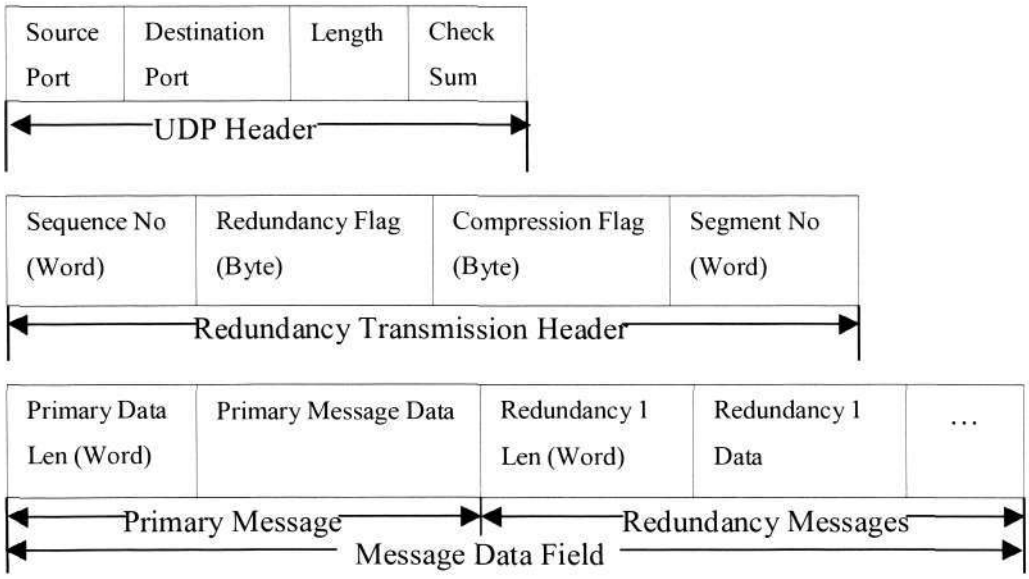


Figure 5-1: Message packet format.

5.4.2 Message Data Field

Message Data Field is assembled with primary data and redundancy data, if any. It typically contains a few fields, namely, Primary Data Len, Primary Message Data, and redundancy data including Redundancy Sequence No, Redundancy Len and Redundancy Data (repeat for n redundancy data).

- *Primary Data Len*. Primary data length indicates the length of primary data to be transmitted. The primary data refers to new chat message to be transmitted, but it may not be the entire chat session due to the need for segmentation of long messages.
- *Primary Message Data*. This field contains the primary message.
- *Redundancy Message Len*. This field contains the length of redundancy message.
- *Redundancy Message Data*. This field contains the redundancy data content.

The number of redundancy messages should comply with the redundancy flag. Although the number of redundancy message is not limited, the total buffer storage of redundancy data will be maintained such that it is much smaller than the primary data.

5.5 Proposed Message Transmission Mechanism

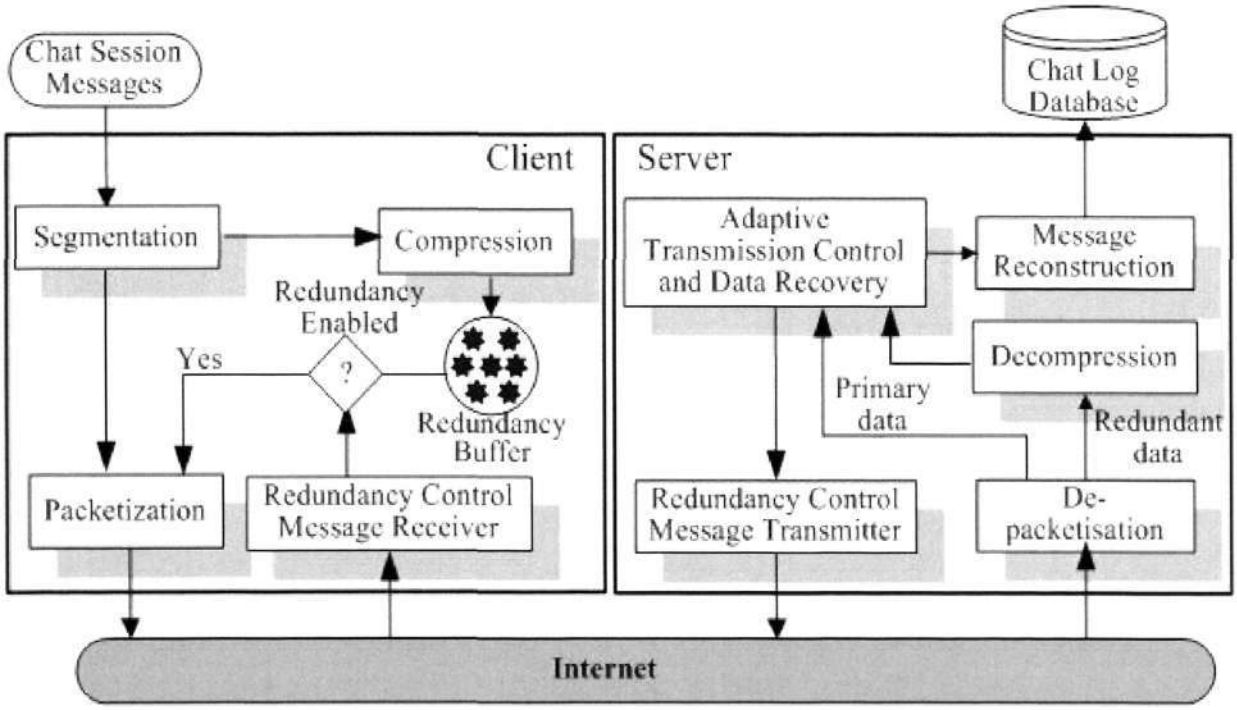


Figure 5-2: Adaptive Redundancy Transmission Control and Recovery Mechanism.

To tackle the packet loss situation during UDP transmission, a Server-based Adaptive Redundancy Transmission Control and Recovery mechanism is proposed for real-time chat message transmission between a server and multiple clients. Figure 5-2 shows the architecture for the proposed server-based adaptive redundancy transmission control and recovery mechanism. The server is a receiving end and a client is a data source. With the proposed mechanism, packet loss is minimized by controlling the redundancy transmission dynamically from the server based on the evaluation of the network congestion level.

A client comprises four processes, namely Segmentation, Compression, Redundancy Control Message Receiver and Packetization. In addition, a Redundancy Ring Buffer is also reserved in each target client.

- *Segmentation*. This process breaks down long message data extracted by the monitoring component into a series of consecutive segmented data for transmission.
- *Compression*. This process compresses primary messages using text compression algorithms to save storage space and bandwidth in redundancy transmission.

- *Redundancy Ring Buffer*. A ring buffer is used to store compressed chat messages in memory space for redundancy transmission support.
- *Redundancy Control Message Receiver*. Each client keeps a redundancy control flag which adjusts the number of redundancy data to be assembled in a packet. This receiver receives a control packet from the server and set the redundancy control flag accordingly.
- *Packetization*. The packetization process encodes the primary message together with header information into a message packet. If redundancy transmission is required, redundancy messages will be retrieved from the redundancy buffer and assembled into the packet according to the redundancy control flag.

The server comprises five processes, namely De-packetization, Decompression, Adaptive Transmission Control and Data Recovery, Redundancy Control Message Transmitter and Message Reconstruction. They are briefly discussed as follows:

- *De-packetization*. This process depacketizes a received message packet according to the header information.
- *Decompression*. If any redundancy data is present in the decoded message packet, the redundancy data will be decompressed according to the compression algorithm indicated by the compression flag.
- *Adaptive Transmission Control and Data Recovery*. This process performs data recovery if necessary. At the same time, this process evaluates the packet loss information and determines the overall network congestion level, based on which the number of redundancy transmission data will be determined.
- *Redundancy Control Message Transmitter*. The transmitter packetizes the output obtained from the adaptive transmission control process into a packet and broadcasts the packet to all clients.
- *Message Reconstruction*. As the name implies, this process updates Chat Log database with correctly received data packets.

5.6 Client Processes

In this section, we discuss the client processes including Segmentation, Compression, Redundancy Ring Buffer, Redundancy Control Message Receiver and Packetization.

5.6.1 Segmentation

A long data packet of tens of kilobytes in size is more prone to errors during transmission. Moreover, transmission of such long packet is undesirable over a 56K modem line. Therefore, the segmentation mechanism is provided to break down long messages into smaller message segments. From observations, we found that the maximal chat message length of ICQ and QQ are roughly 7000 bytes, whereas maximal MSN Messenger and Yahoo Messenger chat message length is 400 bytes (200 characters in UNICODE). As such, the segmentation will be applied to ICQ and QQ to break chat messages into segments with maximal 200-byte lengths.

5.6.2 Compression

Compression aims to save memory space and bandwidth when storing and transmitting redundancy data respectively. Two compression algorithms are selected as candidates: the LZW [47] and SCSU (a standard compression scheme for UNICODE) [48]. LZW is a classical lossless compression algorithm and SCSU is suitable for short UNICODE text message compression. Compression will be applied to redundancy data, while the primary data is not compressed.

5.6.3 Redundancy Ring Buffer

The Redundancy Ring Buffer stores the compressed redundant data for redundancy transmission. To store redundant data into the ring buffer, the buffer slot to be replaced by the new redundant data is calculated using the following equation:

$$Slot\_no = Sequence\_no \ \ mod \ Buffer\_size \qquad (5.1)$$

where *Sequence_no* is the current packet's sequence number, and *Buffer_size* is the size for the redundancy buffer.

5.6.4 Redundancy Control Message Receiver

Each client maintains a *redundancy control flag* which can be used to adjust the number of redundant data to be assembled into a packet. The Redundancy Control Message Receiver will receive a control packet from the server and set the value of the redundancy control flag accordingly.

5.6.5 Packetization

This process forms a UDP data packet for transmission by assembling chat messages and the header information. The sequence number is increased by 1 each time the packetization process is invoked. A compression flag is set to indicate the compression algorithm used for compressing redundancy data. A redundancy flag is set to indicate the number of redundancy messages. For segmented data packets, a segmentation number is also used in addition to the sequence number.

Figure 5-3 shows the packetization process. For illustration purpose, a maximum of two redundant messages are assembled into the packet. The current primary data has a sequence number *N*. To packetize a message packet, the *redundancy control flag* value will be examined. If the control flag is 0, redundancy transmission is disabled. The packetization process will simply assemble the primary data into a message packet. If the control flag is 1, one redundant message is required. The redundant data (N-1) will be retrieved from the redundancy buffer. If the control flag is 2, the redundant data (N-1) and (N-2) will be retrieved from the redundancy buffer and assembled into the message packet for transmission. A redundancy flag greater than 2 can be realized in a similar manner.

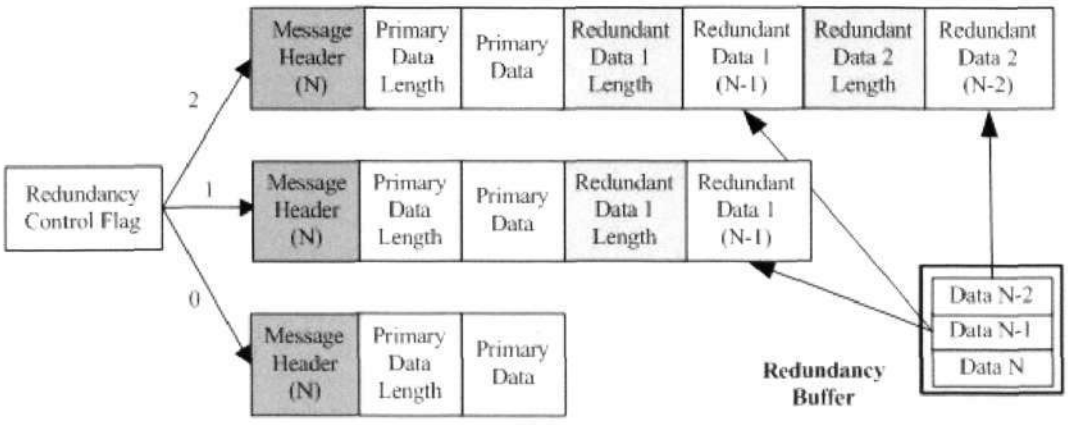


Figure 5-3: Redundant message packetization.

5.7 Server Processes

This section discusses the server processes including De-packetization and Decompression, Adaptive Transmission Control and Data Recovery, and Redundancy Control Message Transmitter.

5.7.1 De-packetization and Decompression

The De-packetization and Decompression processes are just the reverse processes of packetization and compression processes in the client. The de-packetization process reads the message header and extracts the primary message and redundant messages (if any). The extracted messages are stored into different data buffers according to target clients (based on IP addresses). Message packet data are inserted into the corresponding data buffer in the order according to its sequence number. The redundant data will be decompressed according to the compression method indicated by the compression flag before being inserted into the data buffer.

5.7.2 Adaptive Transmission Control and Data Recovery

Figure 5-4 shows the Adaptive Transmission Control and Data Recovery process, which takes depacketized messages from the De-packetization and Decompression processes as inputs. The Adaptive Transmission Control and Data Recovery process comprises 4 steps: IP Dispatcher, Packet Information Counting, Packet Loss Analysis, and Control Message Generation.

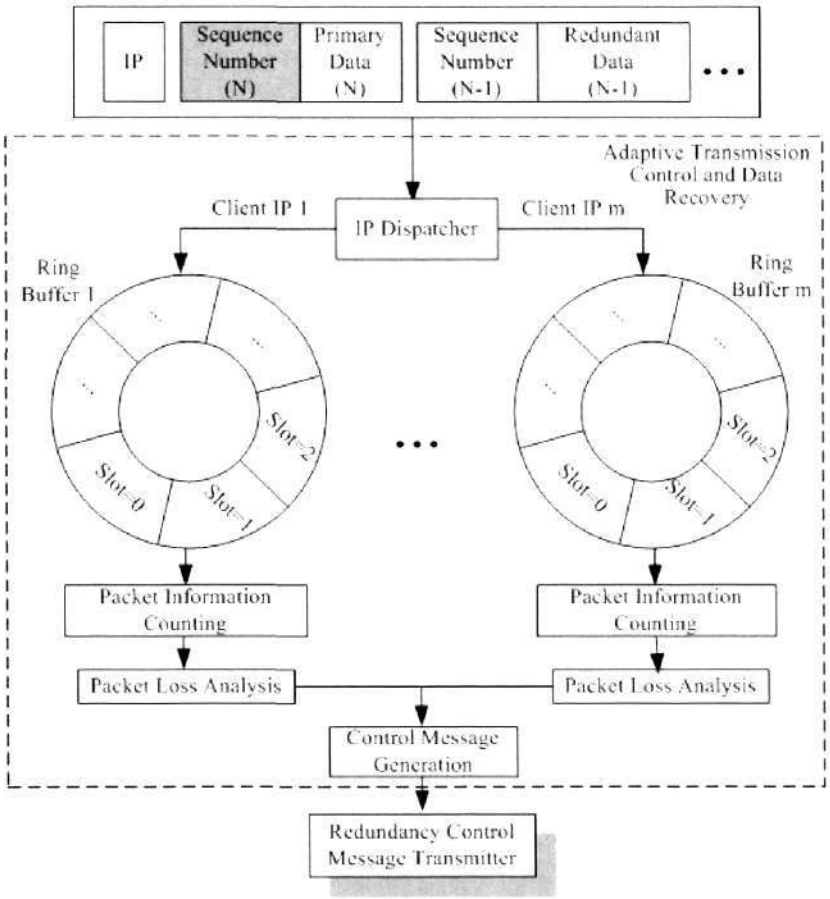


Figure 5-4: Adaptive transmission control and data recovery.

IP Dispatcher

The depacketized messages will be dispatched by the IP Dispatcher according to its originating IP address. A ring buffer is used for storing the data for each client IP in the server. The buffer slot that is used to store the data in the corresponding ring buffer is calculated as follows:

$$Slot\_no = Sequence\_no \ \ mod \ Buffer\_size \qquad (5.2)$$

where *Sequence_no* is the packet's sequence number and *Buffer_size* is the size for the ring buffer. Packet loss can be recovered using redundant data. A redundant data will not overwrite an existing data in the same slot if they have the same sequence number. As such, the primary data will not be overwritten by redundant data in order to avoid information loss due to compression.

Packet Information Counting

This step collects 2 types of data for use in Packet Loss Analysis. Firstly, it keeps track of the total number of non-duplicate packets received from each target client in a periodic manner. Secondly, the maximal sequence numbers received from each target client during each time period are also recorded.

Packet Loss Analysis

The server calculates the packet loss rate for each client periodically using the data collected in the Packet Information Counting step. We first calculate the packet loss $PL_i(n)$ from a particular client *i* during time period *n* using the following equations:

$$PN_i(n) = MAXSN_i(n) - MAXSN_i(n-1) \qquad (5.3)$$

$$PL_i(n) = PN_i(n) - APN_i(n) \qquad (5.4)$$

where $PN_i(n)$ is the total number of packets sent, which is computed by taking the difference between the maximal packet sequence numbers received in time *n* and *n-1* respectively (i.e. $MAXSN_i(n)$ and $MAXSN_i(n-1)$). Equation (5.4) obtains the number of packets lost by subtracting the total number of non-duplicate data packets received ($APN_i(n)$) from the number of packets sent ($PN_i(n)$).

Based on the packet lost information, the *current packet loss rate* ($P_i(n)$) can be calculated using Equation (5.5).

$$P_i(n) = PL_i(n) / PN_i(n) \qquad (5.5)$$

$$\lambda_i(n) = (1-\alpha) \times \lambda_i(n-1) + \alpha \times P_i(n) \qquad (5.6)$$

The current packet loss rate will then be smoothed with a low pass filter given in Equation (5.6), where α is a constant between 0 and 1 to indicate the influence of the current packet loss rate on the smoothed loss rate. $\lambda_i(n-1)$ is the smoothed loss rate for the previous time period. A moderate value of 0.3 is suitable for α as discussed in Busse *et al.* [49].

Control Message Generation

Control Message Generation determines the appropriate number of redundancy transmission using an estimation of the overall *network congestion level* (*CL*) for the near future, i.e., for the time period *n+1*. The estimation of overall network congestion level is based on the current loss rate $P_i(n)$ and smoothed loss rate $\lambda_i(n)$ for each client *i* during time *n* obtained from Packet Loss Analysis.

Firstly, we define the *current network congestion level* (*CCL*) by taking the largest value of the current packet loss rates for all clients. This is given in Equation (5.7). The *CCL* is calculated for the current period *n*.

$$CCL(n) = \max_i \{P_i(n)\} \tag{5.7}$$

Equation (5.8) gives the *CCL estimation* for period *n+1*, represented by $\overline{CCL(n+1)}$. It adds the difference between the average values of current packet loss rates during time *n* and *n-1* (i.e., $\underset{i}{avg}\{P_i(n)\}$ and $\underset{i}{avg}\{P_i(n-1)$ respectively) to $CCL(n)$.

$$\overline{CCL(n+1)} = CCL(n) + [\underset{i}{avg}\{P_i(n)\} - \underset{i}{avg}\{P_i(n-1)\}] \tag{5.8}$$

Similarly, the *smoothed network congestion level* (*SCL*) is calculated by taking the maximal of all smoothed packet loss rates for all clients during time *n*. An estimation on $SCL(n+1)$, represented as $\overline{SCL(n+1)}$, can be obtained similarly to $\overline{CCL(n+1)}$ except that the current loss rates are replaced with the smoothed loss rates. Equation (5.9) and Equation (5.10) give the calculation.

$$SCL(n) = \max_i \{\lambda_i(n)\} \tag{5.9}$$

$$\overline{SCL(n+1)} = SCL(n) + [\underset{i}{avg}\{\lambda_i(n)\} - \underset{i}{avg}\{\lambda_i(n-1)\}] \tag{5.10}$$

Finally, Equation (5.11) gives the estimated overall network congestion level for time *n+1* represented by $\overline{CL(n+1)}$. It smoothes the estimation based on both past estimated value and current estimation (the larger value between $\overline{CCL(n+1)}$ and $\overline{SCL(n+1)}$) into consideration. The β value reflects how much effect the current estimation will have on the

estimated overall network condition for the next period. A moderate value of 0.5 is suitable for β.

$$\overline{CL(n+1)} = \beta \times \max\left\{\overline{CCL(n+1)}, \overline{SCL(n+1)}\right\} + (1-\beta) \times \overline{CL(n)} \qquad (5.11)$$

The reason why the maximal value is used to calculate the *CCL* and *SCL* is that all target clients can be considered as sharing the same network. As data transmission is based on the UDP protocol, the message packets from each target client are transmitted across the entire Internet and pass through every route to reach the server in order to guarantee the shortest delivery time. Thus, different clients can be considered as residing at the same network in broad sense. The maximal packet loss rate then reflects the overall network condition shared by all clients.

Both the current and smooth estimations are considered for calculating the number of redundant data to be used, since the estimations reflect different network conditions. The current network congestion level reflects a short-term reception condition by using the current packet loss values. An increase in the current loss rate would indicate the start of congestion state. On the other hand, a sharp decrease in the current loss rate would reflect a recovery from congestion temporarily. The smoothed counterpart reflects long-term reception condition by using accumulated past loss rates, i.e., the smoothed packet loss rate.

With the estimated network congestion level, the Control Message Generation generates a control packet to control redundancy data transmission from target clients, which aims to reduce the packet loss rate from all clients to be within λ_u. λ_u is a packet loss threshold. Acceptable chat session messages can be reconstructed if the packet loss rate is below this value. The value of λ_u is determined experimentally.

Figure 5-5 shows how to determine the number of redundancy data to be transmitted from the client side based on the estimated overall network congestion level for time $n+1$, i.e., $\overline{CL(n+1)}$. The value of $\overline{CL(n+1)}$ will be measured against two boundary loss values, namely Upper Loss Limit (U) and Lower Loss Limit (L), to determine the number of redundancy data to be transmitted. The relation between U, L and λ_u is shown in Equations (5.12) and (5.13).

$$L = \lambda_u \qquad (5.12)$$

$$(1-U) - U(1-U) = 1 - \lambda_u \Rightarrow U = \sqrt{\lambda_u} \qquad (5.13)$$

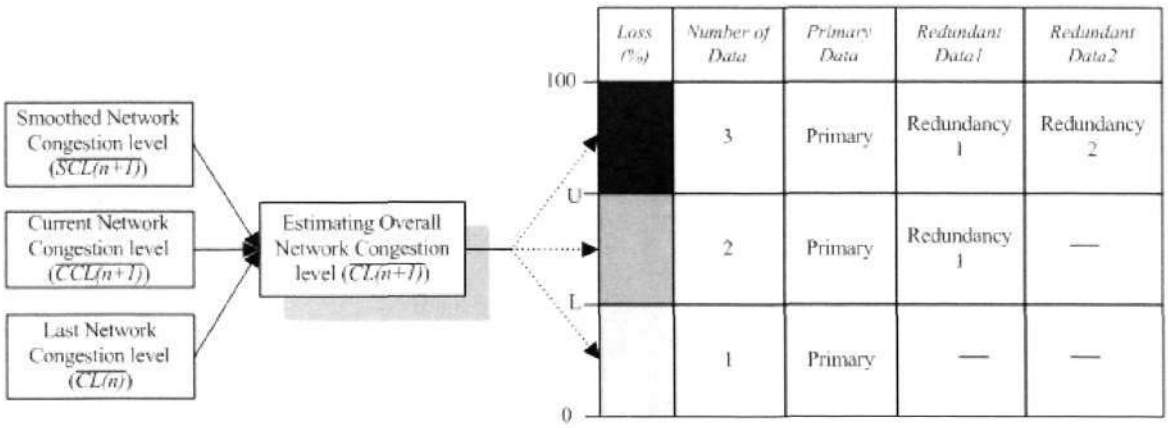


Figure 5-5: Determination of number of redundancy data.

5.8 Performance Analysis

This section discusses the performance of the Server-based Adaptive Redundancy Transmission Control and Recovery mechanism. The simulated experiments focus on evaluating the effectiveness of the proposed mechanism for packet loss recovery under two different network traffic conditions: low and heavy network load conditions. In heavy network load condition, the overall network packet loss rate is simulated with an average of 20%, whereas in the low network load condition, it simulates an average packet loss rate of less than 5%.

In this experiment, the equipment used includes 10 sets of NEC Powermate workstations with 1 Gigabytes of RAM running Windows XP. One of the workstations is dedicated as the server while the others are set up as clients. Each client has 1000 packets with random length queued up for transmission, which is to be sent out at 600 ms interval. The redundancy level is initially set to 2 and feedback interval is set to 30 seconds.

Table 5-1 records the number of packets (*# of Packets*) and the corresponding average packet loss rate (*Average Loss Rate*) for all 9 clients before and after enabling the adaptive redundancy recovery mechanism during heavy network load condition. By comparing packet loss rates before and after recovery, it is observed that the recovery mechanism is able to reduce the packet loss rate to less than 1% for most clients (except Clients 4 and 7). Nevertheless, all clients have average packet loss rates of less than 2%. Figure 5-6 gives the packet loss rates before and after enabling the recovery mechanism for the 9 clients during heavy network load condition. As shown in the figure, the packet loss rate for each client has substantially been reduced.

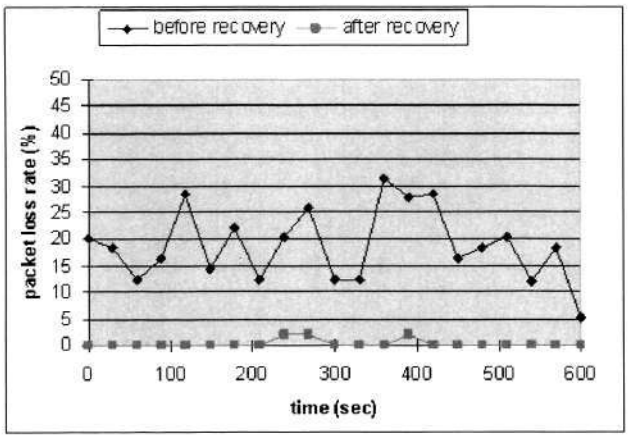
Table 5-2 shows the packet loss statistics before and after enabling the adaptive redundancy recovery mechanism during low network load condition for all 9 clients. By comparing packet loss rates before and after recovery, it is shown that the majority of clients (Clients 2, 3, 5, 6 and 8) have 0% average packet loss rate. The rest have average packet loss rates reduced close to 0%. Figure 5-7 gives the packet loss rates for each client before and after enabling the recovery mechanism during low network load condition. With the help of redundancy packets during transmission, most clients have packet loss rates around 0% after using the recovery mechanism.

5.9 Summary

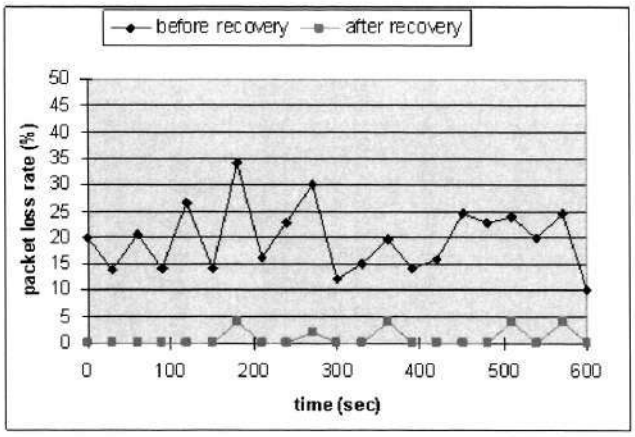
In this chapter, a server-based Adaptive Redundancy Transmission Control and Recovery mechanism has been proposed to provide real-time transmission of chat messages from target clients to the server. UDP is used as the base protocol. With the proposed mechanism, the problems on packet loss, out of sequence and duplication are minimized. The performance of the proposed mechanism is evaluated based on different network packet loss situations. The experimental results have shown that the mechanism is very effective for the recovery of UDP packet loss during real-time transmission.

Table 5-1: Packet loss statistics for the 9 clients during heavy network load condition.

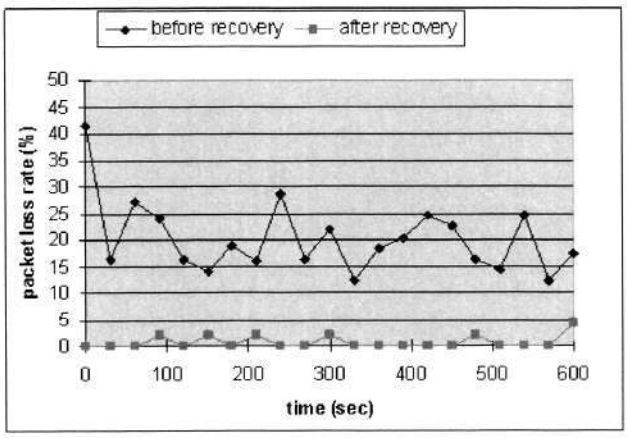
Clients	Before Recovery		After Recovery	
	# of Packets	Average Loss Rate	# of Packets	Average Loss Rate
1	809	19.1%	998	0.2%
2	802	19.8%	992	0.8%
3	800	20.0%	994	0.6%
4	751	24.9%	987	1.3%
5	784	21.6%	991	0.9%
6	791	20.9%	993	0.7%
7	791	20.9%	981	1.9%
8	803	19.7%	991	0.9%
9	799	20.1%	991	0.9%



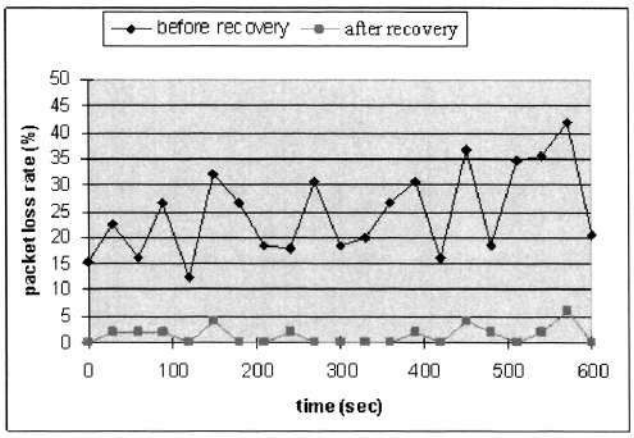
(1) Client 1.



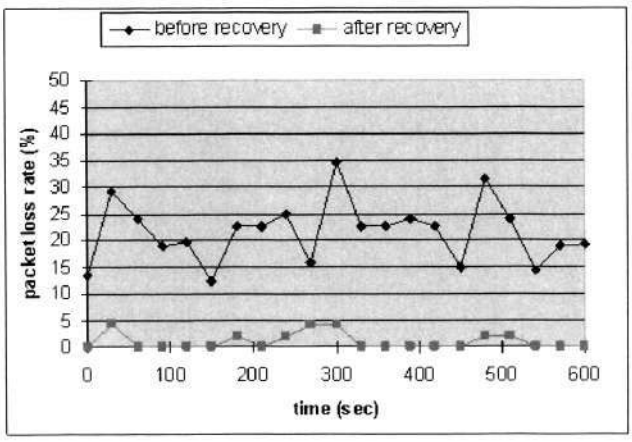
(2) Client 2.



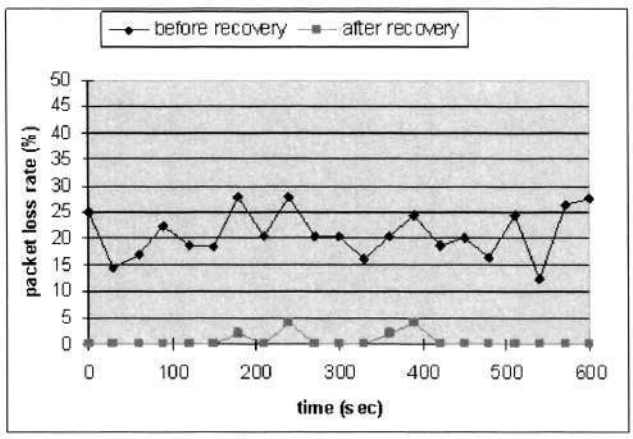
(3) Client 3.



(4) Client 4.

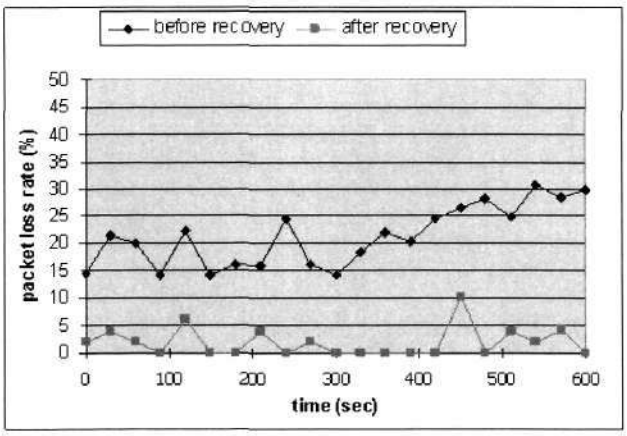


(5) Client 5.

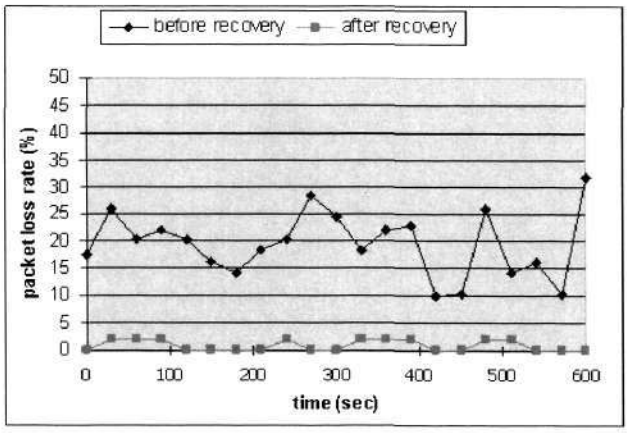


(6) Client 6.

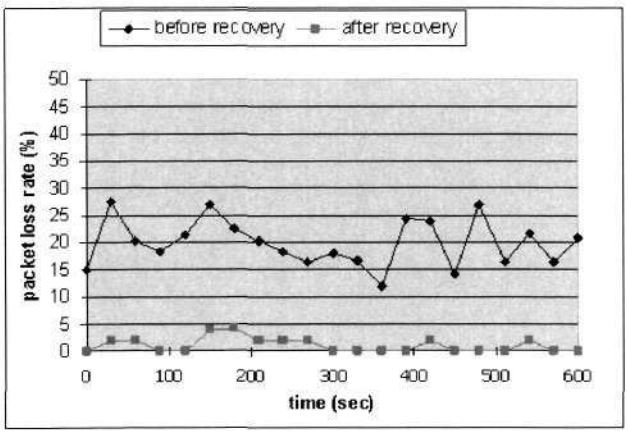
Figure 5-6: Packet loss rates during heavy network load condition.



(7) Client 7.



(8) Client 8.

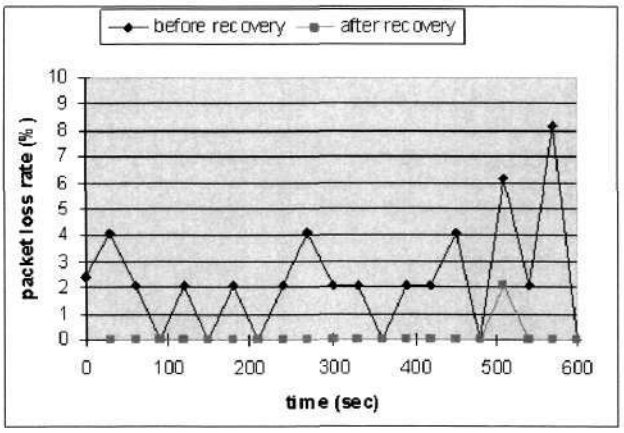


(9) Client 9.

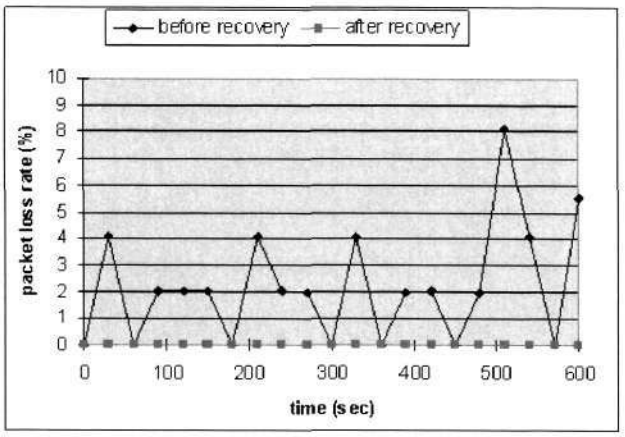
Figure 5-6: Packet loss rates during heavy network load condition (Cont.).

Table 5-2: Statistics of packet loss recovery of 9 clients in low network load condition.

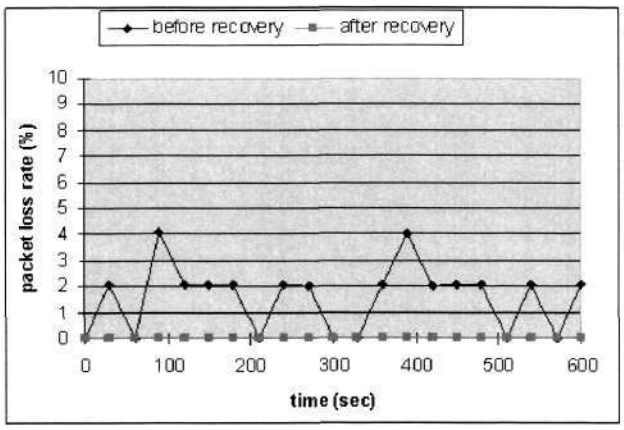
Clients	Before Recovery		After Recovery	
	# of Packets	Average Loss Rate	# of Packets	Average Loss Rate
1	977	2.3%	999	0.1%
2	978	2.2%	1000	0.0%
3	984	1.6%	1000	0.0%
4	977	2.3%	999	0.1%
5	985	1.5%	1000	0.0%
6	986	1.4%	1000	0.0%
7	972	2.8%	996	0.4%
8	981	1.9%	1000	0.0%
9	983	1.7%	999	0.1%



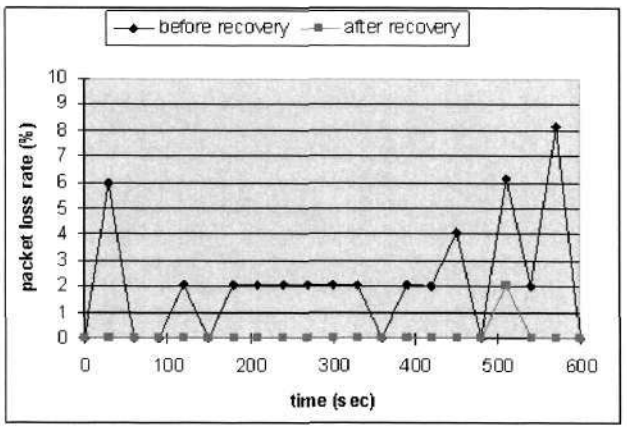
(1) Client 1.



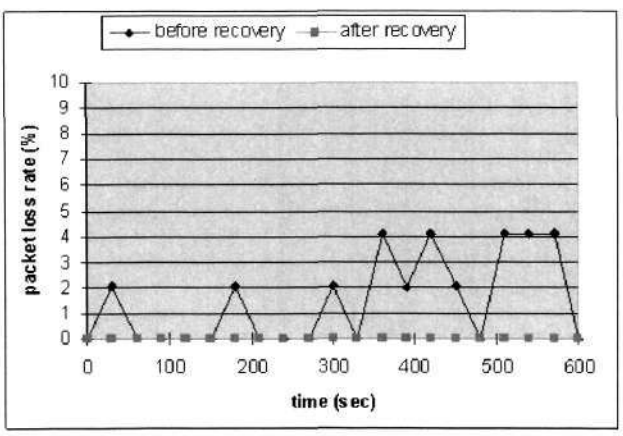
(2) Client 2.



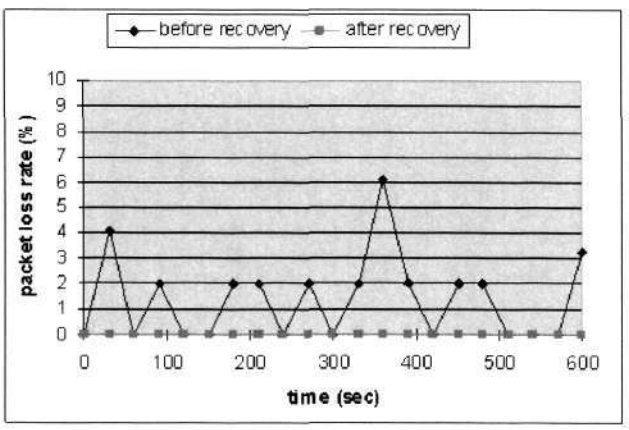
(3) Client 3.



(4) Client 4.

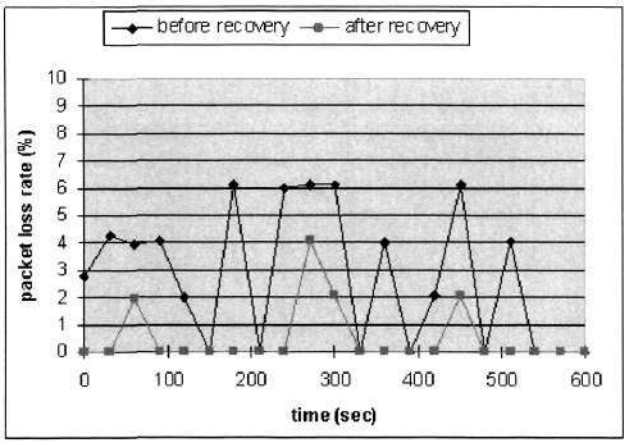


(5) Client 5.

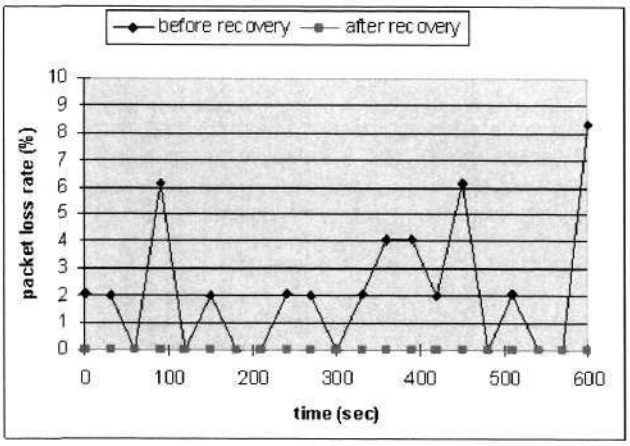


(6) Client 6.

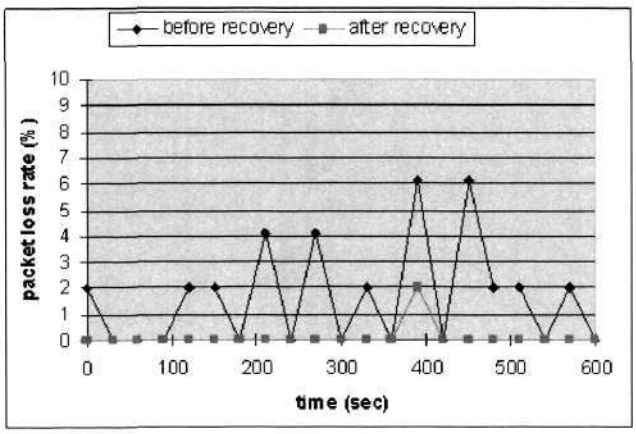
Figure 5-7: Packet loss rate during low network load condition.



(7) Client 7.



(8) Client 8.



(9) Client 9.

Figure 5-7: Packet loss rate during low network load condition (Cont.).

Chapter 6

Chat Topic Detection

The monitoring of Instant Messaging (IM) systems and chat rooms has recently been investigated for preventing the potential misuse of such systems. To support IM monitoring, it is necessary to have effective techniques for analysing the recorded chat messages. Topic detection [50-52] is one of the important research areas on chat message analysis, which aims to analyze the chat content for identifying the topics that are under discussion among users.

In this chapter, we propose an efficient and effective approach for chat topic detection based on classification techniques. The proposed technique aims to support chat topic detection for both online topic monitoring and offline topic analysis. The performance of the proposed approach is also evaluated.

6.1 Topic Detection

Topic detection, which identifies conceptual topics discussed among text documents, is a challenging problem in text mining [59, 60, 70, 71, 72]. For the past few years, many research works [54-60] on topic detection have been conducted mainly based on conventional text documents such as news articles (e.g., Reuters-21578 news articles [61]), web documents (e.g., web pages from Open Directory Project [62]) and newsgroups (e.g., the 20 newsgroups [63]). These conventional text documents are written in standard language (such as English) and usually have self-contained and coherent content. Recently, with the proliferation of Instant Messaging systems and the need for monitoring the contents of such systems, a few research works [50-52] have been started on analysing conversational messages such as chat messages, which are also considered as written speech transcript, for topic detection. The language structure of chat messages is quite different from conventional text documents in terms of chat language usage, incompleteness, shortness, interwoven topics and multimedia context.

Topic detection approaches can basically be classified into supervised and unsupervised. Supervised approaches require domain experts for training text documents on pre-defined conceptual topics, and prediction on topic labels can then be made on unknown data objects.

Unsupervised approaches, on the other hand, clusters text documents into different groups according to the similarity of its contents without involving domain experts for the purpose of retrieving text documents of the same or similar topics. In this section, we review the related works on topic detection on both conventional text documents and chat messages.

6.1.1 Supervised Approaches

Supervised approaches are mainly based on classification techniques for topic detection. Different classification techniques including regression models [53, 54], nearest neighbours classification [55, 56], Bayesian probabilistic approaches [57, 58], decision trees [64, 65], inductive rule learning [67-69], neural networks [66, 70] and Support Vector Machines (SVM) [71] have been investigated for topic detection.

Conventional Text Documents

Yang and Liu [72] conducted a comparative study on topic detection approaches using different classification algorithms based on the Reuters-21578 data set. In the study, vectors are generated from Reuters news articles using TFIDF (Term Frequency Inverse Document Frequency) [73]. Single words of news articles are selected as features based on the χ^2 statistic or *Information Gain* (IG) [74] criterion. The performance of the classification algorithms, including linear SVM, *K* Nearest Neighbours (k-NN) [55] and Naïve Bayes [57], is evaluated based on each topic category using recall, precision and F-measure. The global performance across topic categories is evaluated using micro-average and macro-average of the respective measures. In this study, SVM is evaluated as the best classifier, followed by k-NN and Naïve Bayes.

Dumais *et al.* [75] also presented a similar comparative study on the Reuters-21578 data set. In this work, Reuters articles are formed as binary weighted feature vectors composing single word features selected with the *Mutual Information* (MI) [74] criterion. Classification techniques including Decision Tree, Naïve Bayes and linear SVM are used for topic detection. The performance is evaluated based on each single topic category using the recall and precision measures. The global performance is evaluated using micro-average of the recall and precision measures. From the experimental results, SVM again outperforms others for the Reuters data set despite of the difference in data preparation and representation from that of Yang and Liu [72].

Antonie and Zaiane [76] used association rules for topic classification, i.e., associative classification, based on the Reuters-21578 data set. In this research, the documents are pre-processed with stop-word removal and pruned based on *TFIDF* to remove least representative words for topics. The performance is evaluated using the precision and recall measures. The micro-average and macro-average of the recall and precision measures are also calculated across the top ten Reuters categories for the evaluation of global performance. Antonie and Zaiane claimed that associative classification outperforms most of the conventional methods except SVM.

In addition, some other research works [77] adopted ontology [78] as domain knowledge for automatic topic identification on web documents by mapping document keywords into the corresponding ontology concepts. However, document classification based on simple keyword mapping does not produce impressive results.

Chat Messages

Elnhrawy [50] presented an offline topic categorization approach for analysing chat conversations related to criminal activities. In this approach, conversation logs are used for analysis and categorization. The logs are pre-processed with stop-word removal and converted into term frequency weighted vectors. Classification techniques including k-NN, Naïve Bayes and linear SVM are used for topic classification. The performance of the different classification techniques is evaluated based on web chat logs using a measure on "average accuracy". From the performance results, the Naïve Bayes classifier has performed "surprisingly well" and has "significantly" outperformed k-NN and SVM on categorising chat messages. However, the evaluation is based only on a small data set and the single measure on "average accuracy", which is not clearly defined and could be biased.

Bengel *et al.* [12] also adopted a categorization approach for analysing chat messages from Internet Relay Chat (IRC) [79]. In this work, the archived chat messages are filtered based on time, chat room channel or chat message authors. The resultant collections of chat messages are grouped as "sessions" for processing and categorization. Each of these "sessions" is pre-processed with stop-word removal and stemming, and then represented using *TFIDF* weight scheme for classification. Instead of using chat messages, the classifier is trained based on web pages obtained from ODP (Open Directory Project) for pre-defined topics. However, the performance of the categorization approach is not given in the paper.

6.1.2 Unsupervised Approaches

Unsupervised approaches on conventional text documents are mainly based on clustering techniques. Clustering techniques such as k-Means [80], Agglomerative Hierarchical Clustering (AHC) [80] and Expectation Maximization (EM) [81] have been applied for topic detection. On the other hand, unsupervised approaches for chat message topic detection are mostly based on signal processing techniques [51, 52]. These techniques treat text documents or chat sessions as a mixture of sources, i.e., topics. The textual terms are observations of signal sources. The objective is to separate the topic sources based on the observations of textual terms.

Conventional Text Documents

Young and Sycara [60] compared the performance of clustering-based topic detection approaches based on 20 newsgroups and the TDT pilot corpus [82]. In this work, text documents are pre-processed with stop-word removal and least frequent-word removal based on term frequency to obtain the feature words, which are then used to form the feature vectors using a variant of the TFIDF weighting scheme. Clustering algorithms including AHC, k-Means, EM, and the combined approaches such as AHC with K-means, and AHC with EM are employed for topic detection. The performance is evaluated based on each document cluster using the precision, recall, F-measure, miss and false alarm [73] measures. The cross-clustering evaluation is based on the micro-average of the respective measures. Based on the experimental results, the hierarchical clustering algorithm (AHC) is found to produce stable results but is very time consuming. On the contrary, the performance of iterative clustering algorithms such as k-Means and EM depends on the initial cluster centres. However, the performance of clustering-based topic detection approaches based on the text data sets does not yield promising results.

Yang *et al.* [83] proposed another clustering-based approach for news event detection based on the TDT corpus. In this work, each TDT article is represented as a TFIDF weight vector with k top-ranking keywords, where k is empirically chosen. To discover topics, the approach employed an agglomerative clustering algorithm and a single pass clustering algorithm developed with considerations on temporal event distributions such as temporal proximity of news articles, time gap between news article bursts and also vocabulary shift in terms of frequency distribution. The performance is evaluated using the recall, precision,

miss and false alarm measures. The micro-average and macro-average of the F-measure are also presented. From the performance results, topic detection using clustering algorithms can be highly effective when the problems are well-defined, and the content and temporal information are properly used. In addition, the single pass clustering algorithm is also used for online topic detection. However, online topic clustering does not yield impressive performance results.

In another work, Chung and Mcleod [84] proposed an incremental hierarchical clustering technique for online event detection of CNN [85] news data. In this approach, each incoming news article is pre-processed with stop-word removal and stemming, and represented as a TFIDF feature vector. The vector is then assigned to the nearest cluster within the defined neighbourhood range or else it forms a singleton cluster. The performance of the proposed approach is compared with k-Means based on the average of precision and recall. From the performance results, the approach is able to produce high-quality clusters. However, the performance is not evaluated based on any benchmarking data sets.

Clifton and Cooley [86] adopted the Natural Language Processing (NLP) technique for topic detection based on the TDT2 [87] news articles. In this work, each news article is processed with the NLP technique to identify a set of named entities. The named entities are grouped as frequent itemsets or clustered by the hypergraph clustering algorithm [88]. As a result, each group of named entities represents a particular topic. From the experimental results, the approach has a low false alarm score but a high miss score, which is probably due to the restriction on assigning each document to only one topic group.

Chat Messages

Kolenda *et al.* [51] applied Independent Component Analysis (ICA) for chat room topic detection. In this approach, the chat messages are first pre-processed with stop-word removal, and then partitioned into "sessions" by overlapping fix-sized windows. Latent Semantic Indexing (LSI) is then applied to the "sessions" for dimension reduction before ICA is carried out for topic detection. This approach is said to be able to detect four highly relevant dynamic topics. However, as the performance is not given in the paper, it is not very clear whether the proposed approach is effective. Further, human interpretation is required in order to label the topics with automatic generation of indicative terms.

Bingham *et al.* [52] proposed a similar chat room topic detection approach to that of Kolenda *et al.* except that Complexity Pursuit is used instead of ICA. The Complexity

Pursuit algorithm separates interesting components from a time series of chat message data and identifies the hidden topics. It is claimed to have the best performance compared with ICA-based approaches. However, the approach has encountered the same problem as that of ICA in terms of topic interpretation. Besides, the performance evaluation is conducted on the standard 20 newsgroups instead of chat messages based on a single measure on "total error", which could be biased.

6.1.3 Discussion

The supervised approaches suffer from the major drawback that great effort is required for training classifiers for detecting topics from text documents. However, once the classifier has been trained, topic detection will be efficient and effective. On the contrary, the unsupervised approaches detect all possible topical groups from text documents. However, unsupervised approaches simply group text documents discussing similar topics and present the overall context structure. Human effort is required for labelling the topics for interpretation. Besides, the effectiveness of unsupervised approaches, especially online topic detection, on text documents is generally not satisfactory.

In chat topic detection, we aim to identify chat sessions discussing only a limited number of important topics with high accuracy. In addition, topic detection will also be incorporated into online monitoring, in which online topic detection is needed. Therefore, a supervised topic detection approach will be more suitable than an unsupervised approach for meeting our objective. As such, in this research we investigate a supervised approach for topic detection.

According to the review on topic classification approaches, SVM, Naïve Bayes and associative classification are some of the most commonly used classification techniques which can achieve good performance for conventional text documents. In this research, we will adopt the three classification algorithms for our proposed chat message topic detection approach and evaluate the performance of the three techniques using recall, precision, F-measure, accuracy and macro-average based on two data sets of web documents and chat messages.

6.2 Proposed Topic Detection Approach

In this section, we propose an approach for chat topic detection based on classification techniques. The proposed approach, which uses topic indicative terms, supports multi-label categorization. The multi-label categorization enables chat sessions to be classified with multiple class labels. Topic indicative terms are predefined for each topic and used in the classification process. As each chat session will be classified by each topic classifier, topic indicative terms can be used to limit the number of key terms to be used for classification purposes, thereby improving the efficiency of the categorization process. This is particularly important for online topic detection. In addition, as the indicative terms are identified by an experimental study on sample training data, the use of indicative terms should also improve the performance of the categorization process. In this approach, we have used the Naïve Bayes (NB), Support Vector Machine (SVM) and Associative Classification (AC) for the classification engine.

Table 6-1: A survey on teenager's most popular chatted topics by PEW.

Topic	Total	Topic	Total
Boyfriend/girlfriend	38%	Sports	21%
Other friends	36%	Secret things	20%
Life in general	35%	Current events	19%
Someone to date	35%	School / grades	18%
Music	30%	Jobs	16%
Sex	27%	Deep feelings	15%
Gossip	26%	College	12%
The future	25%	Fashion	12%
Next weekend	25%	Video games/computers	11%
Last weekend	23%	Parents	9%
Movies/TV shows	21%	Celebrities	7%

Before discussing the proposed approach, we first need to determine the topics that should be detected in chat conversations. In [89], the most popular topics chatted amongst most teenagers is identified, which is shown in Table 6-1. However, some of the popular topics are either very general (e.g. Gossip, Life in general, etc.) or not well-defined (e.g. Next weekend, The future, etc.). In this research, we focus on five useful topics for investigation. The topics include Sports, Games (i.e., Video games/computers), Entertainment, Travel (i.e., Someone to date and weekend), and Pornography (i.e., Sex and Secret things). The selection of these five topics is based on our major interest for child monitoring. The first four topics are common topics amongst teenagers while the last one is an objectionable topic.

Nevertheless, other topic categories of chat messages can also be defined for other monitoring requirements.

Figure 6-1 shows the proposed classification-based approach for chat topic detection. In the proposed approach, each chat session can be categorized into more than one topic. The proposed approach comprises the following major components: Sessionalization, Feature Extraction, Feature Selection and Topic Categorization.

- *Sessionalization*. It groups a collection of related chat messages into sessions for processing and categorization.
- *Feature Extraction*. It extracts features such as textual contents, icon text and URL contents from chat sessions.
- *Feature Selection*. It selects chat features for categorization based on indicative terms stored in the Indicative Term Dictionaries.
- *Topic Categorization*. It categorizes chat sessions into one or more topics using a set of topic classifiers. The topic classifiers are built based on classification techniques including Naïve Bayes, Support Vector Machine and Associative Classification.

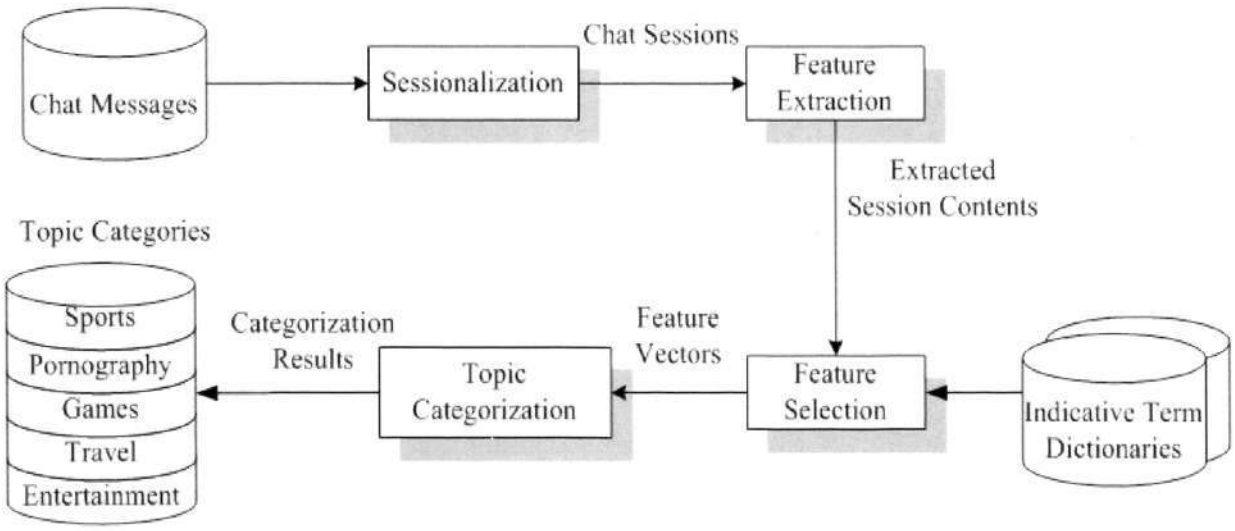


Figure 6-1: The proposed classification-based approach for chat topic detection.

6.2.1 Sessionalization

Due to the nature of chat messages that is short and concise, a single chat message is typically less than 10 words. This poses a big challenge for topic detection. To tackle this problem, a collection of chat messages are gathered as the basic processing and categorization unit or *session*. A session is defined as a sequence of chat messages exchanged

within the lifespan of a chat dialog window. The assumption is that the dialog window will be closed when the conversation ends naturally. However, there are exceptions:

- A user may close a dialog window after a few messages are exchanged. This may happen when the user sees the boss coming towards him at that time.
- A user may also leave a chat window running for a long time without closing it even though all conversations have ended. This happens quite commonly in places where PCs are not shut down.

The above two situations cause problems to our initial definition on sessions. To resolve this, we refine the session definition with the following two heuristics:

- Merge sessions of the same participants with temporal proximity for their potentially coherent content. In this research, we set the temporal proximity boundary to be 10 minutes empirically. In other words, two chat sessions occurred between the same participants with an interval less than 10 minutes will be merged.
- Split messages of a chat session that have a long time gap between them into two or more sessions. As discussed in Chapter 3, typical chat sessions last between 4 to 20 minutes. For example, if two chat messages have a time gap of more than 40 minutes, we will consider the gap as long and the chat messages will be divided into two sessions.

6.2.2 Feature Extraction

Apart from textual contents, a chat message may also contain icons and even URLs. In Feature Extraction, it extracts both icon text and web page contents of the corresponding URLs given in chat sessions. For web page contents, information displayed in the viewable body text and several other locations such as the title of the web page, and the meta data of "description" and "keywords" are extracted. The textual contents, the icon text and the web page contents are combined together to form the chat session content. Figure 6-2 shows an example for the Feature Extraction process.

6.2.3 Feature Selection

In Feature Selection, we have used indicative terms stored in the Indicative Terms Dictionaries for selecting appropriate features for classification purposes. This approach is based on our observation that chat conversations on a particular subject/discussion (called

topic) usually contain a set of words, known as *indicative terms* (or *topic keywords*) that characterize that particular topic. This set of indicative terms is considered to be highly representative for all conversations on the same topic. Therefore, indicative terms can be treated as a unique collection of features characterizing the chat contents belonging to that topic. Indicative terms are not limited to terms with single words, it could also refer to terms with phrases. With indicative terms predefined as features for selection, it can also help solve the dimensionality problems during the classification process.

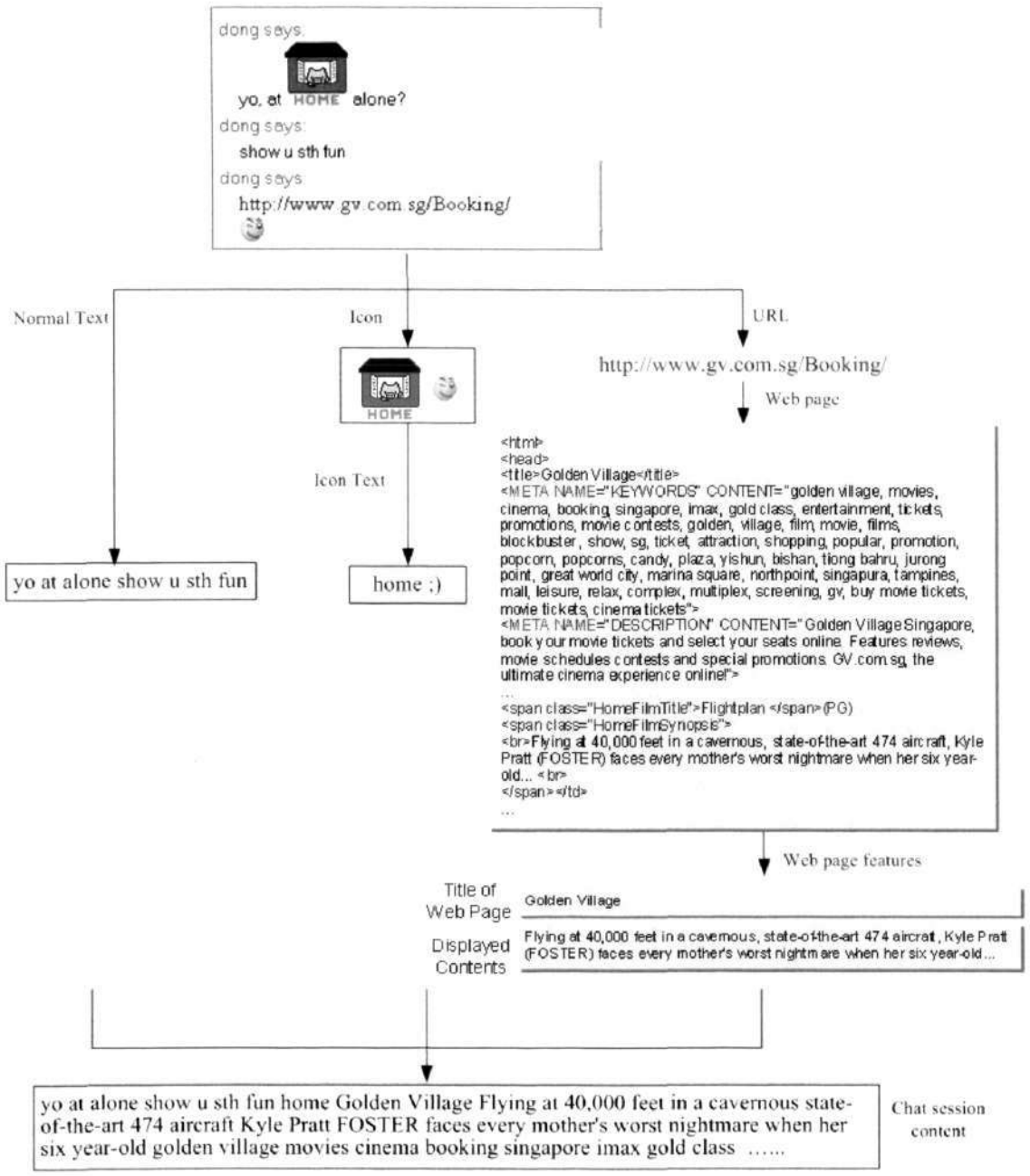


Figure 6-2: Feature extraction.

Figure 6-3 shows the Feature Selection process which is divided into two steps: Tokenization and Indicative Terms Identification. Tokenization simply breaks the chat session content into a list of single words or tokens while preserving the relative ordering of the tokens. Each term is also converted into lower cases for processing. Indicative Terms Identification then selects the set of terms from the tokens for each topic category based on the Indicative Terms Dictionaries, which stores a predefined set of indicative terms for each topic category. After Indicative Terms Identification, a set of indicative terms occurred in the chat session will be incorporated into a feature vector which will be used as the input to the different topic classifiers. The weightage of each occurred indicative term will be assigned based on a binary value, i.e., if an indicative term is present, the corresponding weight is assigned to "1" in the feature vector. Otherwise, "0" will be assigned.

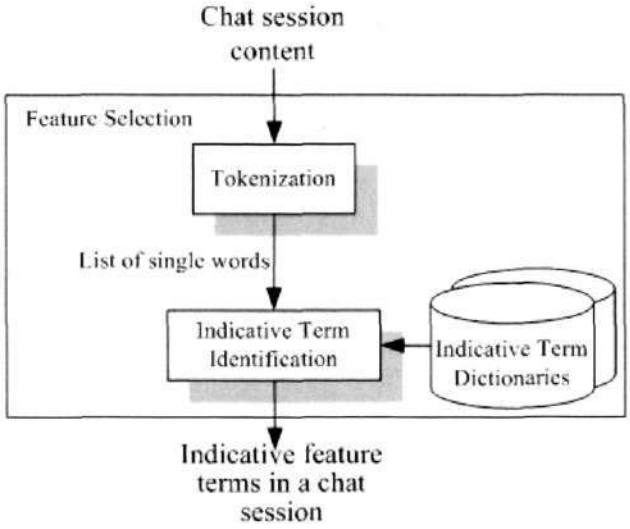


Figure 6-3: Feature Selection.

As only the indicative terms will be used for the categorization process, it is of the utmost importance to choose the most representative set of indicative terms for each topic category. The list should not be too long or short. If the list is too long, it will introduce noise (irrelevant words) and overheads. However, if the list is too short, it will cause performance degradation. To compile a dictionary that covers extensively most of the keywords and key-phrases for each topic, we collected and analyzed the statistical data regarding the usage of indicative terms that can be commonly found for each of the five topic categories.

Table 6-2 gives some example indicative terms for the Games category stored in the Indicative Term Dictionaries. As shown in the figure, each row represents a unique indicative term indexed by the first column. Different entries in the same row represent all the possible

variations of the unique term. Appendix B gives the full list of the indicative term dictionary for the Games category. The identification of indicative terms has taken into the consideration of the characteristics of chat message language such as short-forms, acronyms and polysemic words as discussed in Chapter 3. During Indicative Term Identification, any matches of the indicative terms in the same row will contribute to one occurrence of that unique feature represented by the row. The acronyms and the corresponding full names such as "cs" and "counter strike" are considered. Similarly, short-forms such as "graphics card" and "gfx card" are also considered. Polysemic words such as "computer game" and "pc game" are treated as the same feature term. The morphing of terms is also considered such as "game" and "games", and "gamer" and "gamers".

Table 6-2: A sample indicative term dictionary for "Games".

Index	Term 1	Term 2	Term 3	Term 4	...
1	cs	counter strike	counterstrike		
2	game	games	gaming	gamer	gamers
3	graphics card	graphics cards	gfx card	gfx cards	
4	multiplay	multiplayer	multiplayers	multi-play	multi-player
5	pc game	computer game	video game	pc games	computer games
	...				

6.2.4 Topic Categorization

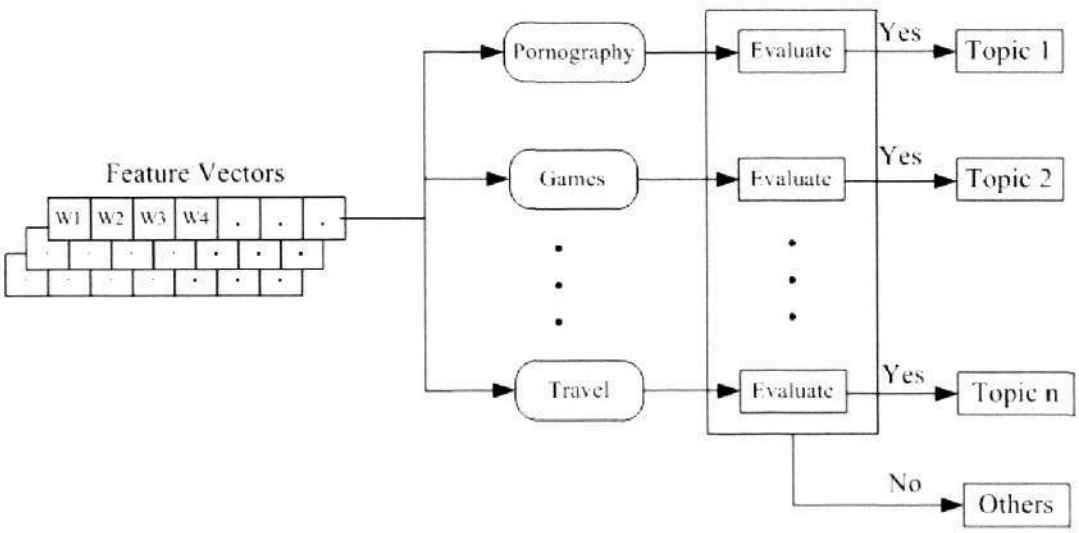


Figure 6-4: The topic categorization process.

In Topic Categorization, we identify chat sessions with one or possibly multiple topics. To do this, we construct chat topic classifiers for the detection of each particular topic. In the topic

categorization process which is given in Figure 6-4, it classifies the feature vector of each chat session using the set of topic classifiers. For each topic classifier, it will determine whether the chat session contains contents that belong to that topic based on the feature vector of the chat session. As such, each chat session can be classified into one or more topics. If the feature vector is not classified into one of the topics, then we will treat the chat session contains contents that belong to other topic categories. One of the major advantages of this categorization process is that new topic classifier can be easily incorporated when the classification of new topics is needed.

6.3 Performance Evaluation

This section evaluates the performance of the proposed chat topic detection approach. The performance is measured based on the following three classification techniques, Naïve Bayes (NB), Associative Classification (AC) and Support Vector Machine (SVM), for topic categorization. All experiments are conducted on an Intel Pentium 4 3.0 Gigahertz machine with 1 Gigabytes of memory running Microsoft Windows XP operating system.

6.3.1 Experiments and Data Sets

The experiments are conducted using two sets of data. One is based on web page documents and the other is based on chat messages. Web page documents and chat messages represent two different types of text documents. Web pages are static documents that have relatively long and coherent contents using the common English language, while chat messages are shorter and much more dynamic for its conversational nature. Therefore, apart from evaluating the proposed approach based on chat messages, we will also evaluate the performance of the proposed approach for web page documents for comparison purposes.

Table 6-3: The statistics for the first data set of web page documents.

Category	No. of web pages	Size (MB)
Sports	779	27.3
Pornography	782	16.4
Games	787	17.8
Travel	800	27.8
Entertainment	777	47.3
Others	3228	61.4

The first data set contains web pages mainly downloaded from the ODP (Open Directory Project) directory [61]. We collect five subsets of data samples from ODP with each corresponding to each of the following five topic categories: Sports, Pornography, Games, Travel and Entertainment. However, as ODP does not contain any sample web pages for the topic Pornography, web pages from Pornography are collected manually from the web. Web pages under the Movies and TV directories of ODP are collected for the topic Entertainment. Moreover, web pages from other topics such as Education, Employment, Shopping and Health are also downloaded from ODP to form another subset of data samples called "Others", which will be used for training purposes. Table 6-3 shows the statistics on the data set of web pages. Each subset of topic category contains about 800 web pages.

The second data set on chat messages is downloaded from some web chat sites including *UGroups.com* [90], *jolt.co.uk* [91] and *AdultfriendFinder* [92]. Similar to the data set on web pages, we also collect five subsets of chat messages, with each corresponding to one of the five topic categories under evaluation. For each subset of the data set, the structural information of web chats, such as reply, quote and author, is removed to retain only the chat contents. In addition, the "Others" subset is also collected for training purposes. Each subset of the data set also has about 800 sessions for each category. Table 6-4 shows the statistics on the data set of chat messages.

Table 6-4: The statistics on the data set of chat messages.

Category	No. of chat sessions	Size (MB)
Sports	792	1.1
Pornography	807	2.5
Games	789	1.8
Travel	784	3.2
Entertainment	753	2.5
Others	1118	5.6

The classifiers are required to undergo a training process before they can be used for topic detection. Therefore, two training data sets based on web pages and chat messages are formed by taking approximately 70% of documents for each topical category of the collected data sets. The remaining 30% of documents from each topical category of each data set will then be used as the testing data sets for performance evaluation of topic categorization.

6.3.2 Evaluation Measures

In the proposed topic categorization approach, each binary classifier will determine whether an incoming document or chat session belongs to a particular topic category C_i. Training samples belonging to a particular topic category are positive samples, and the rest are negative samples. Table 6-5 shows the confusion matrix. The four values *a, b, c* and *d* are to be counted for a topic category C_i, where

a = the number of correctly predicted positive samples;

b = the number of incorrectly predicted positive samples;

c = the number of incorrectly predicted negative samples; and

d = the number of correctly predicted negative samples.

Table 6-5: The confusion matrix.

	Positive Samples	Negative Samples
Positive Prediction	a	b
Negative Prediction	c	d

To evaluate the performance for each topic category C_i, we use the measures *precision (p)*, *recall (r)*, *F-measure* (F_1) and *accuracy* [93] defined as follows:

Precision = the proportion of correctly predicted positive samples in all positive samples;

$$= \frac{a}{a+b}.$$

Recall = the proportion of correctly predicted samples in all positive predicated samples;

$$= \frac{a}{a+c}.$$

F-measure $= \frac{2rp}{r+p}$, which measures the balance between precision and recall.

Accuracy = the proportion of total correct prediction in all samples;

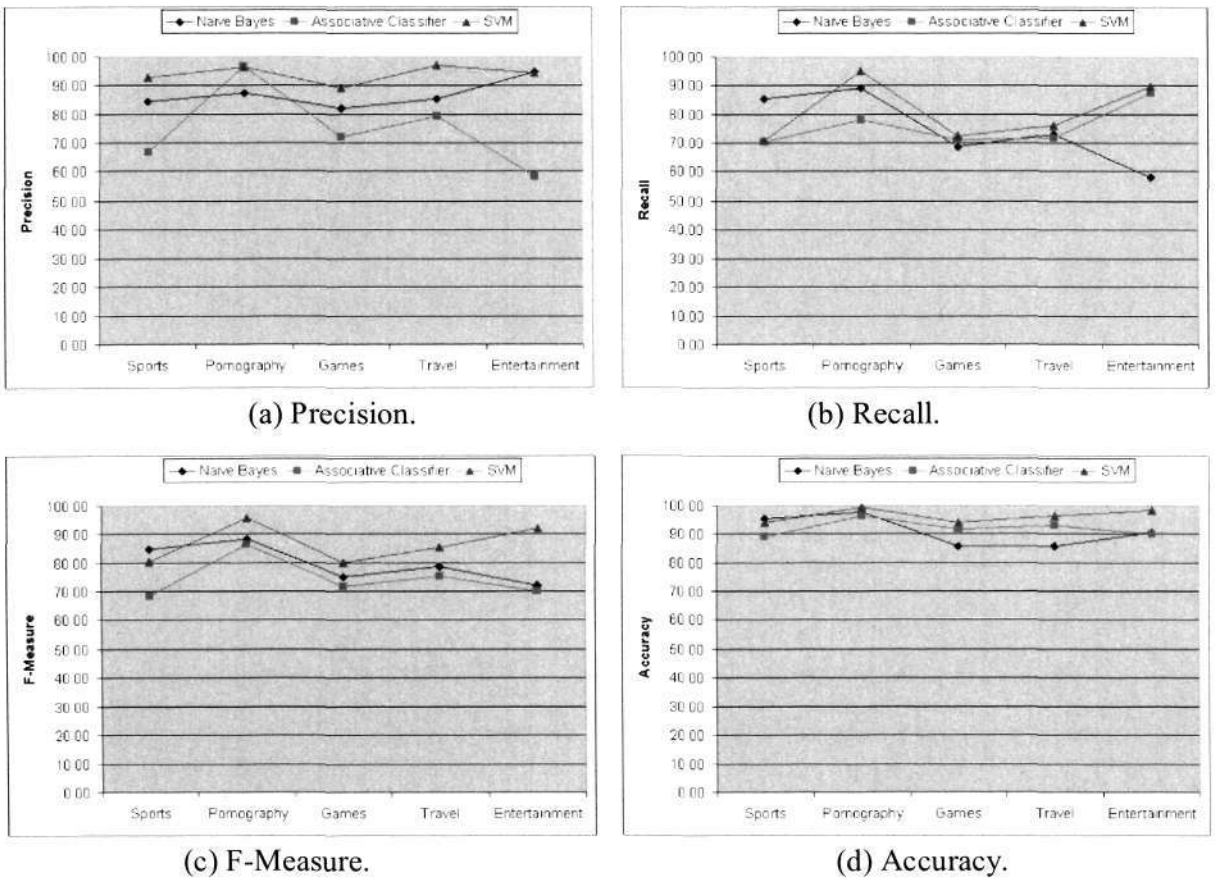
$$= \frac{a+d}{a+b+c+d}.$$

In addition, we also use *macro-average* [93] to calculate the average value of all topic categories under evaluation for each measure in order to evaluate the performance across all topic categories.

6.3.3 Training Performance

In this section, we measure the training performance of the proposed categorization approach using the three classifiers, Naïve Bayes (NB), Associative Classification (AC) and Support Vector Machine (SVM), based on the two data sets of web page documents and chat messages.

Web Page Documents



(a) Precision.

(b) Recall.

(c) F-Measure.

(d) Accuracy.

Figure 6-5: Training performance results based on the training data set of web page documents.

Figure 6-5 presents the training performance results of NB, AC and SVM based on precision, recall, F-measure and accuracy using the training data set of web page documents. As shown in the figure, SVM performs the best and outperforms the other two classifiers on all measures across most categories. It has also achieved very high precision and accuracy for all categories. NB performs better than AC in precision and F-measure. Among all topic categories, the classification of Pornography documents has obtained the best performance

for all three classifiers, while the classification of Games documents has obtained relatively poorer performance for all three classifiers. This is mainly attributed to the use of a better indicative term dictionary for Pornography than Games, as more related terms can be defined in the Pornography topic.

Table 6-6 shows the global training performance results based on the macro-average of performance measures across all topic categories. From the table, SVM has achieved the best training performance for all measures with especially high precision (93.85%) and accuracy (96.46%). It is followed by NB with good performance in precision and F-measure. All classifiers have achieved good performance on accuracy. SVM and NB have also achieved good precision. However, the recall scores are relatively lower with 80.91%, 75.82% and 74.88% for SVM, AC and NB respectively.

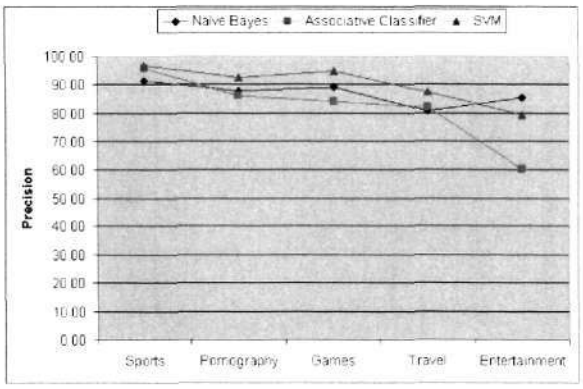
Table 6-6: Training performance results based on macro-average of performance measures.

	Precision (%)	Recall (%)	F-Measure (%)	Accuracy (%)
NB	86.82	74.88	79.75	91.05
AC	74.79	75.82	74.49	91.98
SVM	93.85	80.91	86.66	96.46

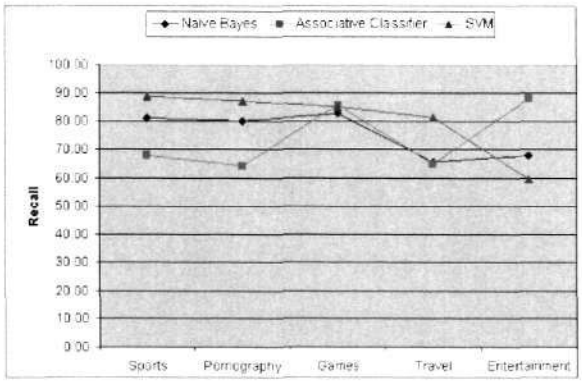
Chat Messages

Figure 6-6 shows the training performance results of NB, AC and SVM on each category of training data set of chat messages. SVM outperforms the other two classifiers in precision, F-measure and accuracy. It has achieved very good performance in precision and accuracy across all five topic categories. However, all classifiers have performed relatively poor in recall. Among all topic categories, the classification of the Sports category has achieved the best performance, while the classification performance of the Entertainment and Travel categories has generally obtained poorer performance. As the Entertainment topic covers a broad range of sources from TV/movies to sports, games, and even travel, it is difficult to compile a small set of indicative terms that is capable of covering all sources. Therefore, it is difficult to achieve good classification performance for the Entertainment category.

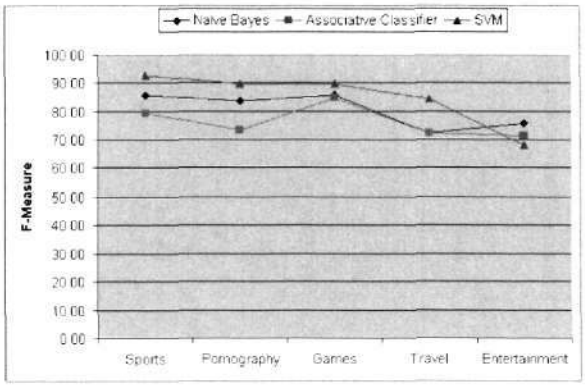
Table 6-7 shows the global training performance results based on the macro-average of the performance measures. From the table, SVM has achieved the best performance in all measures. The high precision (90.02%) and accuracy (95.33%) across all five topic categories are especially impressive. SVM and NB have obtained relatively lower scores in recall (80.52% and 75.52% for SVM and NB respectively) compared with other measures.



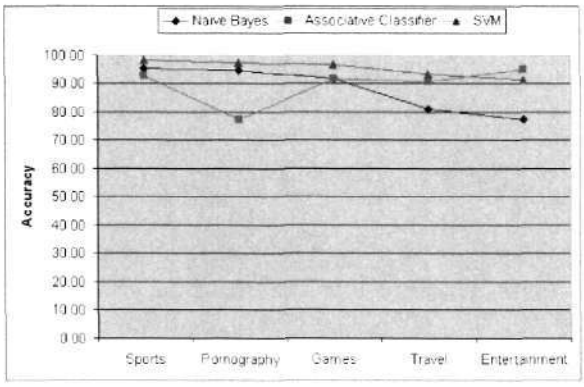
(a) Precision.



(b) Recall.



(c) F-Measure.



(d) Accuracy.

Figure 6-6: Training performance results based on the training data set of chat messages.

Table 6-7: Training performance results based on macro-average of performance measures.

	Precision (%)	Recall (%)	F-Measure (%)	Accuracy (%)
NB	86.81	75.52	80.70	88.10
AC	81.51	74.19	76.33	89.44
SVM	90.02	80.52	84.88	95.33

6.3.4 Categorization Performance

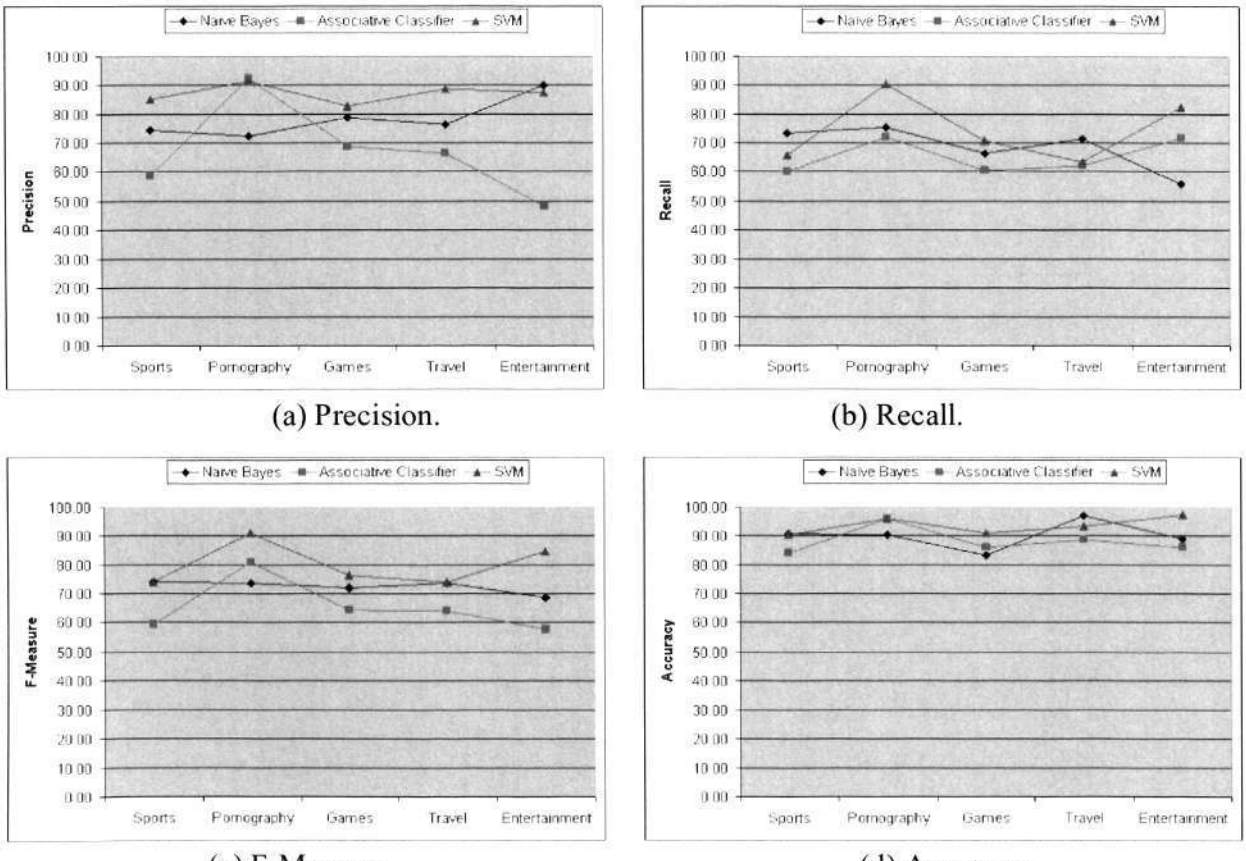
In this section, we measure the performance of the proposed topic categorization approach based on the three classifiers, Naïve Bayes (NB), Support Vector Machine (SVM) and Associative Classification (AC) using two sets of testing data on web pages and chat messages.

Web Page Documents

Figure 6-7 shows the topic categorization performance results of the three classifiers on each category of the testing data set of web page documents. Similar to the training performance,

SVM outperforms the other two classifiers in almost all categories for precision, recall, F-measure and accuracy. The accuracy of SVM is above 90% across all five topic categories. In general, AC has the poorest performance among the three classifiers in all performance measures. Among the five categories, SVM and AC have the best performance on the Pornography topic category. NB has the best performance on the Travel category in F-measure and accuracy. Similar to the training performance, the classification of the Games documents generally has achieved the poorest performance in all three classifiers.

Table 6-8 presents the global categorization performance based on the macro-average of the performance measures across the five topic categories. From the table, SVM has obtained the best overall performance in all measures. The precision (87.02%) and accuracy (93.67%) are high. The categorization performance of SVM is comparable to its training performance. NB has achieved better performance than AC. All three classifiers have achieved good accuracy. However, they have obtained relatively low scores in recall.



(a) Precision.

(b) Recall.

(c) F-Measure.

(d) Accuracy.

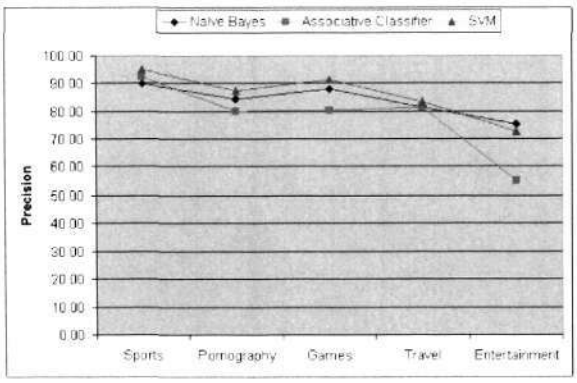
Figure 6-7: Categorization performance results based on the testing data set of web page documents.

Table 6-8: Categorization performance results based on macro-average of performance measures.

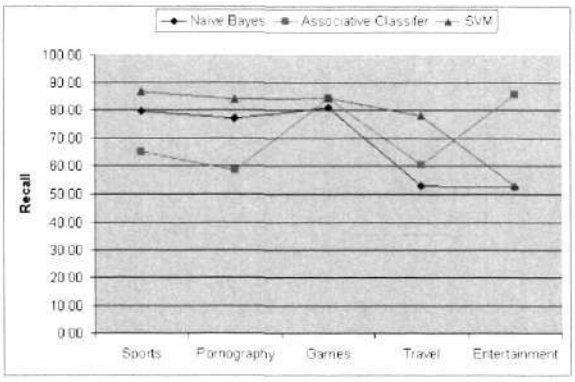
	Precision (%)	Recall (%)	F-Measure (%)	Accuracy (%)
NB	78.47	68.51	72.56	90.12
AC	66.90	65.32	65.34	88.09
SVM	87.02	74.52	80.00	93.67

Chat Messages

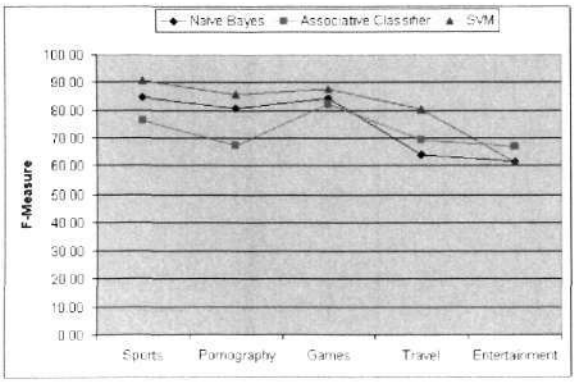
Figure 6-8 shows the topic categorization performance results of NB, AC and SVM on each category of the testing data set of chat messages. SVM outperforms the other two classifiers in precision, F-measure and accuracy. Among the five categories, all three classifiers, SVM, NB and AC, have the best performance on the Sports category. And the classifiers have obtained poorest performance on the Entertainment category. In general, the categorization results correspond to the results of the training performance.



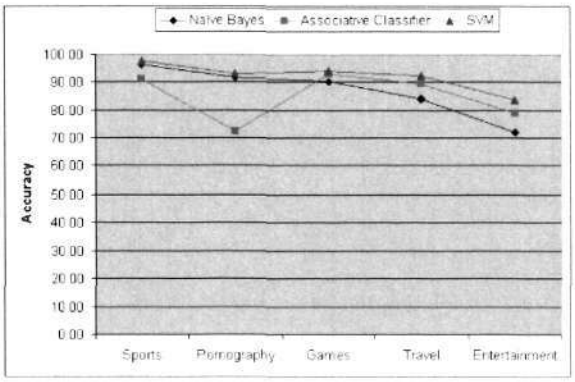
(a) Precision.



(b) Recall.



(c) F-Measure.



(d) Accuracy.

Figure 6-8: Categorization performance results based on the testing data set of chat messages.

Table 6-9 presents the global performance of the three classifiers based on the macro-average of all measures across the five categories. From the table, SVM has obtained the best performance in almost all measures except recall. The high precision (87.25%) and accuracy (92.14%) across the five topic categories are especially impressive, which are comparable to its training performance. NB has achieved better performance than AC. However, all classifiers have obtained low scores in recall with 77.16%, 70.80% and 68.65% for SVM, AC and NB respectively.

Table 6-9: Categorization performance results based on macro-average of performance measures.

	Precision (%)	Recall (%)	F-Measure (%)	Accuracy (%)
NB	84.10	68.65	75.14	86.91
AC	77.83	70.80	72.53	85.01
SVM	87.25	77.16	81.19	92.14

Discussion

From the performance evaluation, SVM generally outperforms the other two classifiers across all categories on both data sets. It is followed by NB, and then AC. And the performance on recall for both data sets is generally lower than that of precision. In general, the construction of indicative terms for topic categories has great effect on the performance of the proposed approach. For example, the indicative terms for the Pornography category are generally easier to identify and more accurate than that of the Entertainment category. As a result, the performance based on the Pornography category is generally better than that of the Entertainment category.

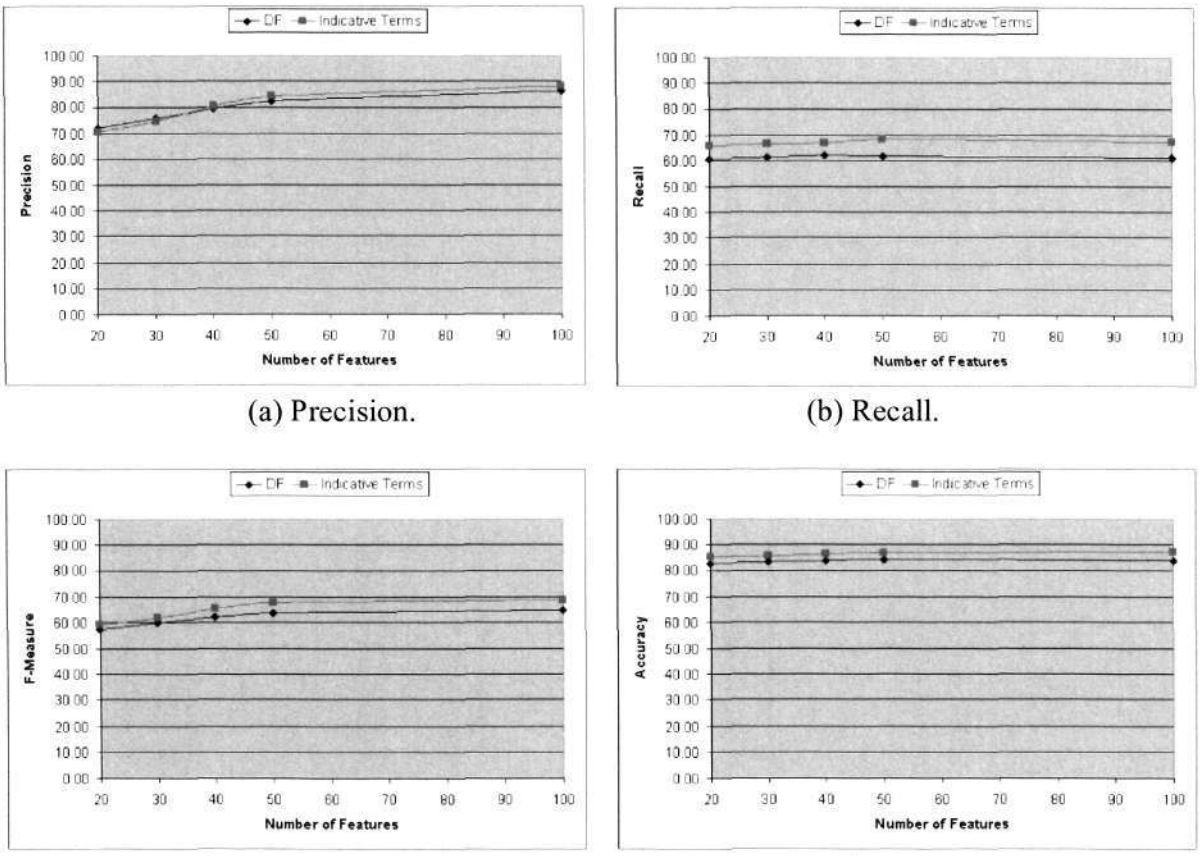
The overall performance of AC is not as promising as that given in [76] using news articles in their evaluation. This is probably due to the fact that we have adopted indicative terms for forming a very small feature set, which do not contribute to the selection of high quality rules for classification. Nevertheless, the high precision and accuracy achieved by the classifiers of the proposed categorization approach across different categories from the two sets of data are promising for topic detection.

6.3.5 Comparison with Document Frequency Based Approach

In this section, we compare the performance of using *document frequency* (DF) thresholding feature selection [94] approach with our proposed indicative terms based approach for

categorization. Document frequency is one of the best feature selection approaches for classification [95].

In this experiment, both feature selection approaches are evaluated based on the same specified numbers of different feature sets (i.e., 20, 30, 40, 50 and 100). For the DF-based approach, the features are selected according to the specified number of most representative words, whereas the specified number of most representative indicative terms from the ranked list of indicative terms for each topic is selected as features for the indicative terms based approach. Naïve Bayes is used as the classifier in this experiment. The evaluation is based on the macro-average of precision, recall, F-measure and accuracy for all categories on the chat message data set.



(a) Precision. (b) Recall.

(c) F-Measure. (d) Accuracy.

Figure 6-9: Performance results of indicative terms based and DF-based approaches for chat topic categorization with Naïve Bayes classifier.

Figure 6-9 shows the topic categorization performance results of Naïve Bayes using different number of feature sets for DF-based and indicative terms based feature selection approaches. As shown from the figure, the indicative terms based approach has achieved

better performance than the DF-based approach in almost all measures except precision (when a small number of features is used). An important advantage of the indicative terms based feature selection approach is its relative high and stable performance with the limited number of features. This results in high computational efficiency while maintaining a satisfactory classification performance, which is especially important for the purpose of online monitoring.

6.4 Summary

In this chapter, we have proposed an indicative terms based categorization approach for chat topic detection. The proposed approach is suitable for both online and offline topic detection. In the proposed approach, we have incorporated different techniques such as sessionalization of chat messages and the extraction of features from icon text and URLs for pre-processing. And indicative terms are used as features for the classification of topic categories. Different classification techniques such as Naïve Bayes, Associative Classification and Support Vector Machine are employed as classifiers for categorizing topics from chat sessions. The performance of the proposed approach is evaluated based on precision, recall, F-measure and accuracy from the data sets of web page documents and chat messages. From the experimental results, the proposed approach has achieved good performance on precision and accuracy. Among all classifiers under evaluation, SVM has achieved the best performance from both sets of test data.

Chapter 7

Instant Message Analysis System

This chapter discusses the implementation of the intelligent chat message analysis system which is known as IMAnalysis. The IMAnalysis system supports both online and offline chat message analysis. For offline chat message analysis, three major functions, namely chat message retrieval, social network analysis and topic analysis are provided. For online chat message analysis, only the online message display and topic analysis are provided. In this chapter, we first present the overall system architecture. This is followed by the system implementation for both offline and online message analysis. In addition, an application scenario based on parental control over child online safety is also given.

7.1 System Architecture

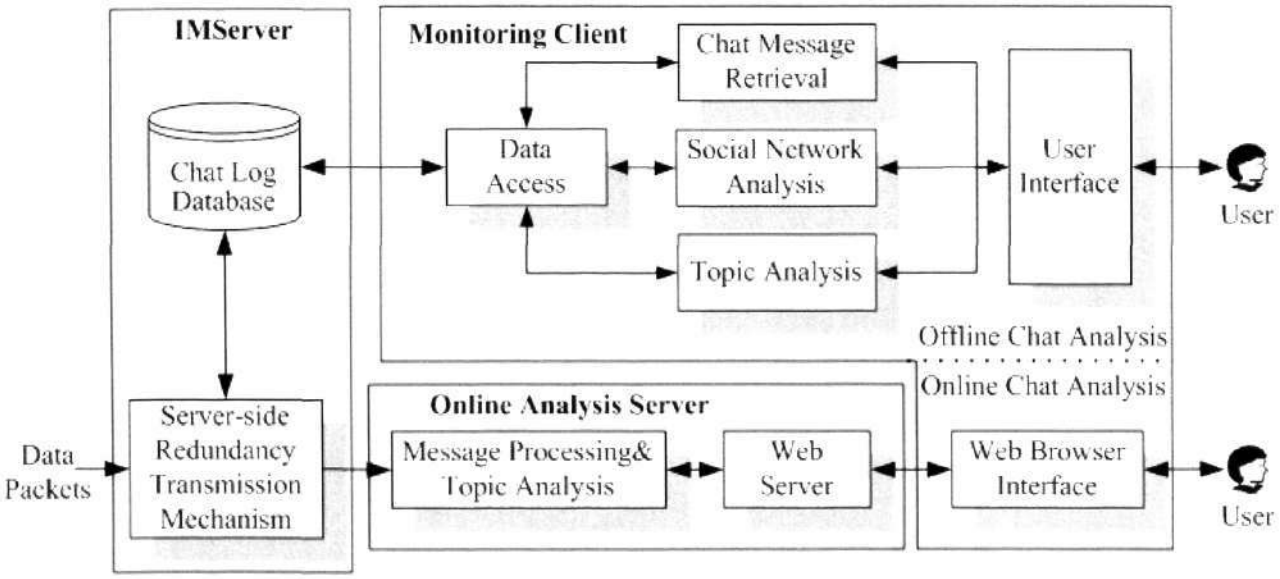


Figure 7-1 shows the system architecture of IMAnalysis which supports both Offline and Online Chat Analysis. During monitoring, chat data packets are transmitted to the IMServer through the Server-side Redundancy Transmission Mechanism which stores the received chat data into Chat Log Database. For Offline Chat Analysis, chat data from Chat Log Database are accessed for chat analysis. For Online Chat Analysis, the Server-side Redundancy

Transmission Mechanism forwards the received chat data to Online Analysis Server for online monitoring.

Offline Chat Analysis supports the following three major functions:

- *Chat Message Retrieval*. This allows users to browse and retrieve chat sessions. It also displays statistical data on chat activities of the monitored IM users.
- *Social Network Analysis*. This extracts sender-receiver pairs of chat messages and constructs the social interaction network of the monitored IM users.
- *Topic Analysis*. This incorporates topic detection for analysing session topics of the monitored IM users.

Online Chat Analysis allows users to view the latest session data online in real-time. Users can also browse information of the monitored session data and the corresponding identified topics through the Web Browser Interface which is provided through a web server located in the Online Analysis Server.

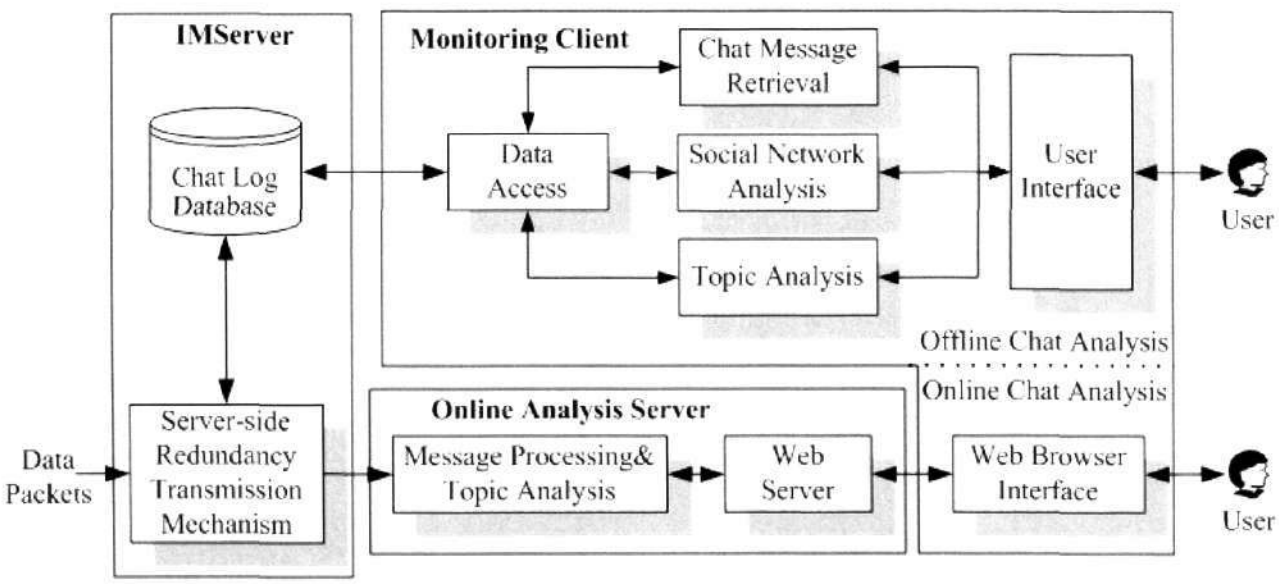


Figure 7-1: System architecture of IMAnalysis.

7.2 Offline Chat Analysis

Offline Chat Analysis provides three major chat analysis functions: Chat Message Retrieval, Social Network Analysis and Topic Analysis.

7.2.1 Chat Message Retrieval

Chat Message Retrieval supports the browsing and retrieval of chat session data archived in the Chat Log Database. Chat Message Retrieval provides three main functions, namely Statistics Generation, Chat Message Browsing and Chat Message Searching.

Statistics Generation

Figure 7-2 shows the interface for Statistics Generation. Users can first specify the target monitored users and the duration on which the chat session data will be analysed. Next, the users press the tab on Statistics. The statistics data will then be displayed according to the chat sessions extracted from the specified time duration and the way it will be counted (i.e., daily, weekly or monthly). For example, as shown in Figure 7-2, the user has selected a target user 155.69.144.204a@hotmail.com, where a user is represented by both the IP address (155.69.144.204) and IM account (a@hotmail.com). The time duration and the way it will be counted are also specified as shown in Figure 7-2. Then, the average usage statistics, number of messages, and average message length (in terms of number of words) are generated from the specified chat sessions and displayed to the users.

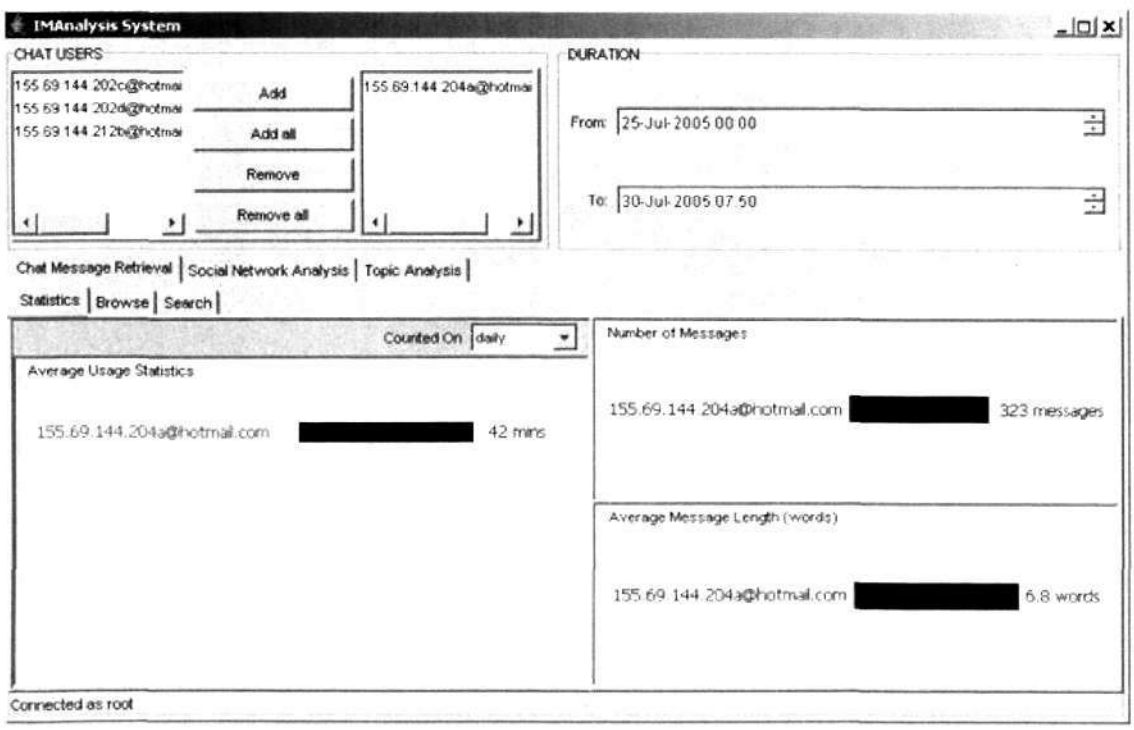


Figure 7-2: Statistics Generation.

Chat Message Browsing

This allows users to browse through the retrieved chat session data. Figure 7-3 shows the interface for Chat Message Browsing. After the user has specified the criteria on the target users and time duration, the chat sessions that satisfied the criteria are extracted from Chat Log Database and displayed on the screen. The user can then select a session and browse through the messages of the session. In addition, the user can also specify the keywords to be highlighted during browsing.

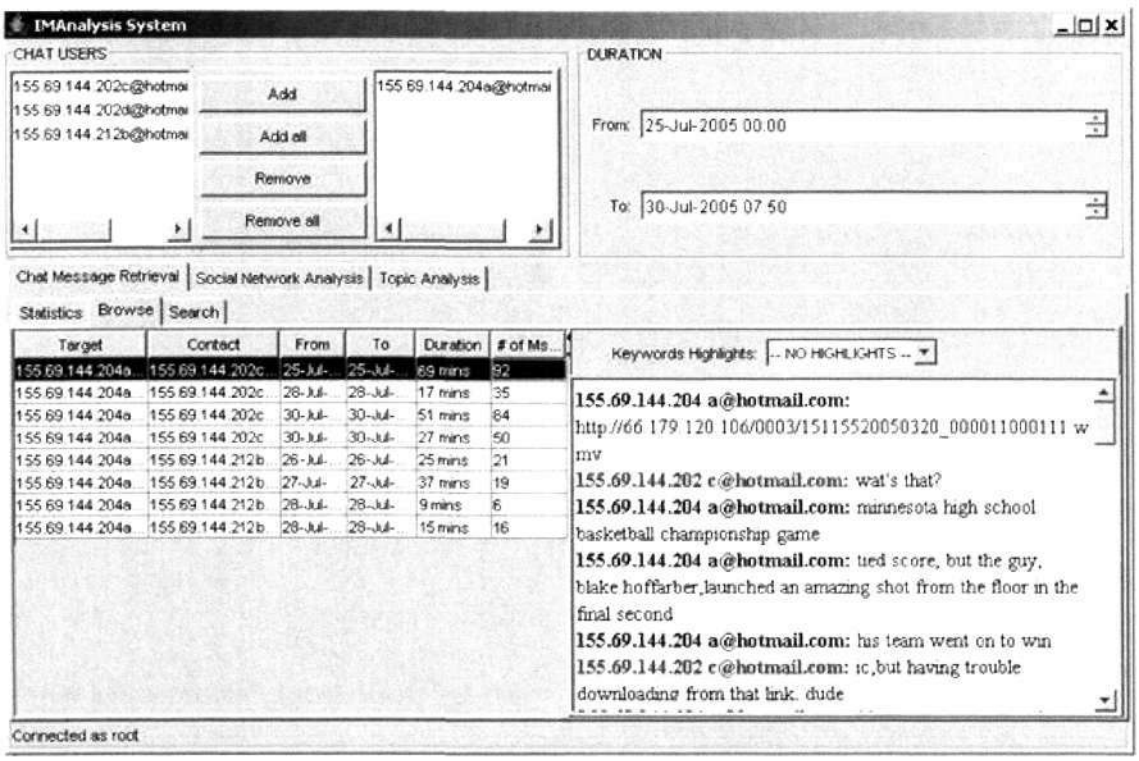


Figure 7-3: Chat Message Browsing.

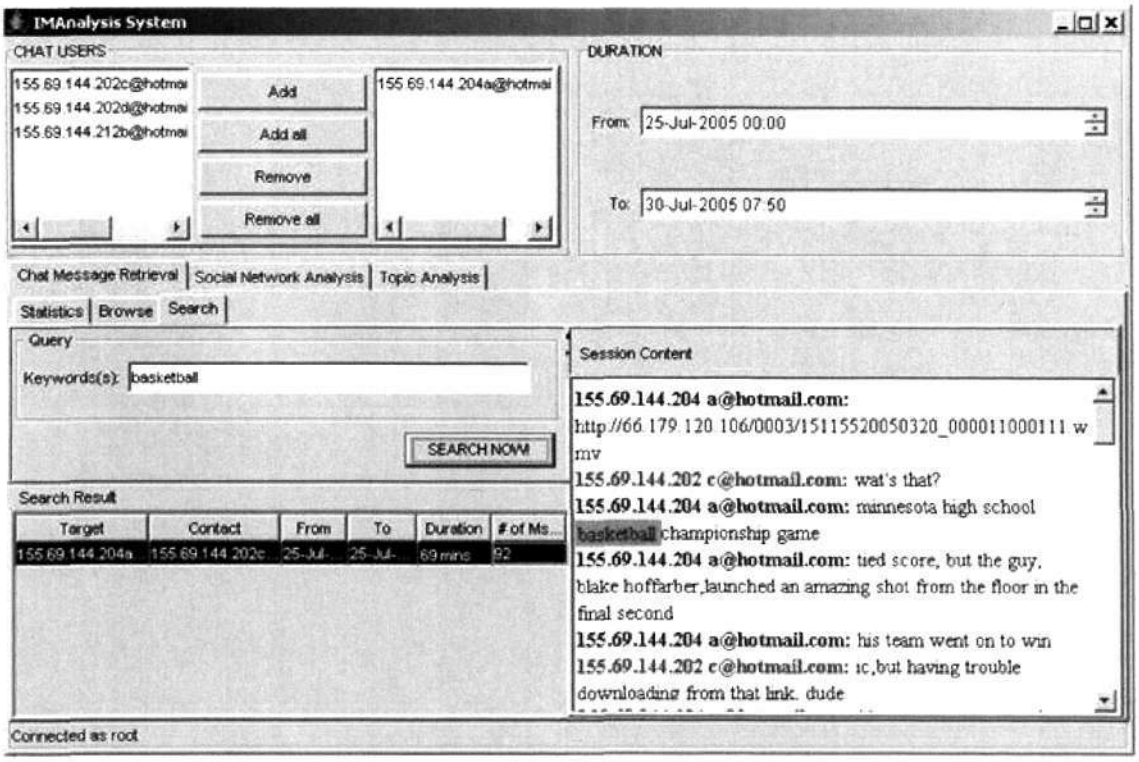


Figure 7-4: Chat Message Searching.

Chat Message Searching

This allows users to search for chat sessions according to their interest. This is done in a similar manner as Chat Message Browsing. Figure 7-4 shows the interface for Chat Message Retrieval. The user first specifies the criteria for the target users, time duration and the search keywords. Chat sessions that satisfy the specified criteria are then retrieved from Chat Log Database and displayed on the screen. A simple Boolean search is currently used for message searching. For example, as shown in Figure 7-4, after specifying the target monitored users and time duration, a search based on the keyword "basketball" is performed and the search results are then displayed on the screen.

7.2.2 Social Network Analysis

Social Network Analysis [96-99] gives the social interactions between the target monitored users and their contacts from the buddy lists. It extracts sender-receiver information of chat messages according to the specified criteria on target users and time duration. The social interactions between the sender-receiver pair are described in terms of quantity and direction.

If there are many chat messages exchanged between the sender and receiver, it implies a close relationship or strong tie between the sender and receiver. The inbound or outbound message direction indicates the receiving or sending of messages of the target user. Moreover, the target users may also interact with contacts from the buddy list that may or may not be under monitoring.

Figure 7-5 shows the interface of Social Network Analysis. After the user has selected the target monitored user—155.69.144.204a@hotmail.com, a star-like social network of the target user is displayed. As shown in the figure, the target user has exchanged chat messages with two contacts. Chat users (both the target user and his contacts) are represented as nodes and the sender-receiver relationships (i.e., messages are exchanged between them) are represented as links. We use the thickness of links to indicate the amount of messages exchanged between the target user and the contact.

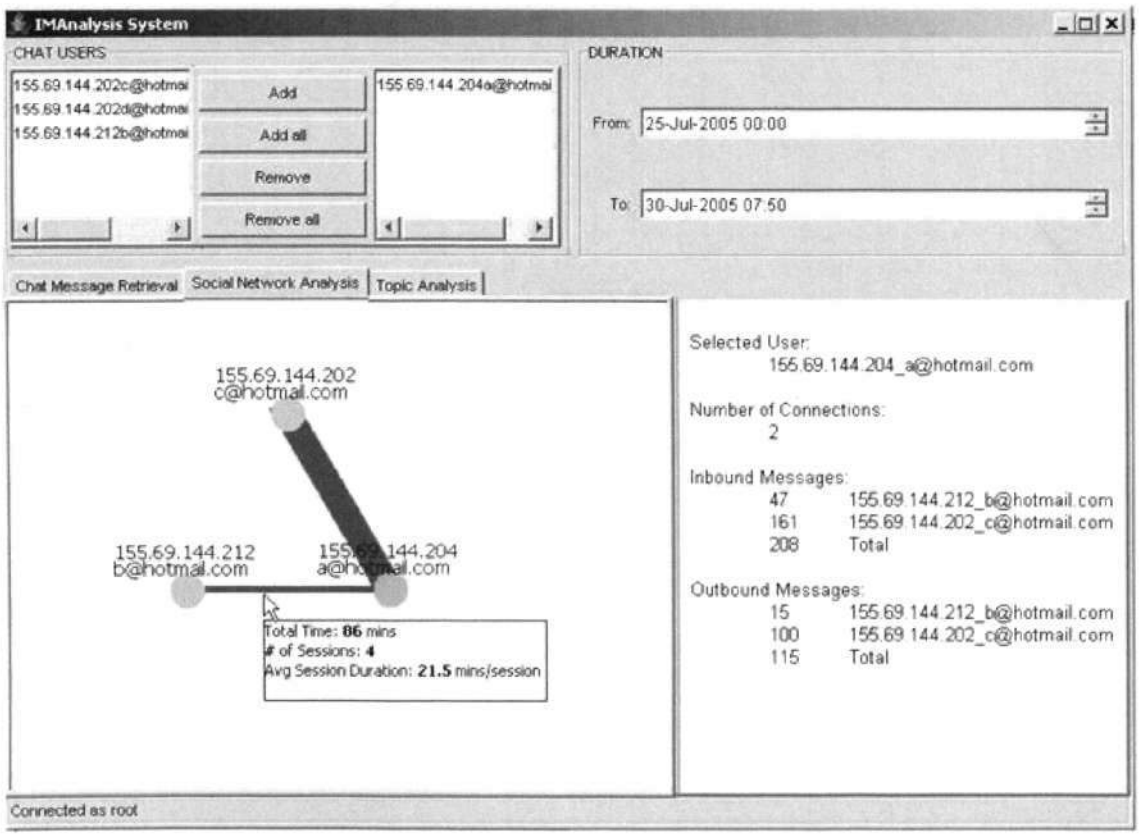


Figure 7-5: Social Network Analysis.

When the system user wants to know the statistical information of a social relationship between the target user and a contact, he can just either click on a link or a node. When a link is selected, the information for a social relationship between the target user and the contact is

displayed. This information includes the total amount of time spent, the total number of sessions occurred and the average time spent per session. When a node is selected, the information on the target user is then displayed. This information includes the number of connections, the number of inbound messages, and the number of outbound messages of the target user.

7.2.3 Topic Analysis

Topic Analysis displays the topics that are detected from the chat sessions of the target monitored users. It adopts the topic detection approach proposed in Chapter 6 for detecting topics for chat sessions based on the five categories of Pornography, Sports, Games, Travel and Entertainment. If a chat session does not belong to one of the five predefined categories, it is labelled as "Others". Figure 7-6 shows the interface for Topic Analysis. After the user has specified the target monitored users and time duration, the identified session data and its detected topics are then displayed on the screen. The user can select to view the details of a chat session. In addition, the displayed information can also be sorted according to different attributes. This will enable users to sort the chat sessions according to topics.

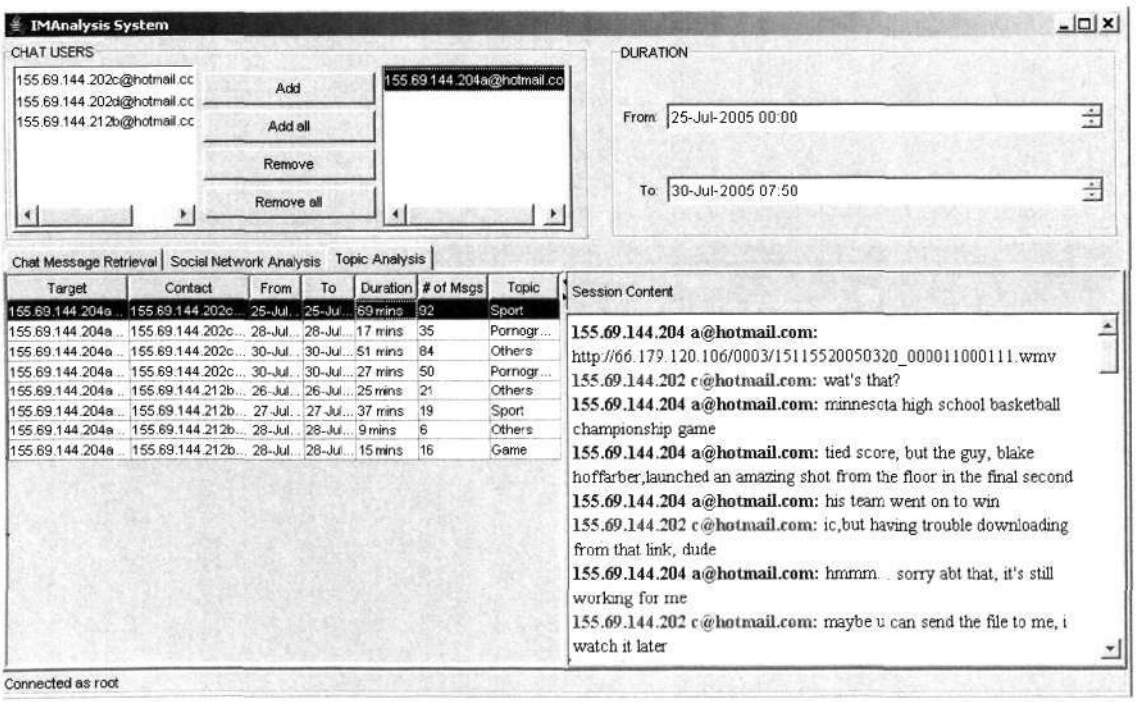


Figure 7-6: Offline topic distribution display.

7.3 Online Chat Analysis

Online Chat Analysis allows users to monitor target IM users online at anytime and anywhere through the Web Browser Interface. The message processing and topic analysis is done in the Online Analysis Server. Figure 7-7 gives the Chat Monitoring Main Page which displays all the target monitored users with current chat messages. The chat messages are transmitted to the Online Analysis Server from the IMServer for topic analysis and display. The user can also view all the monitored sessions by selecting the target monitored user. This is shown in Figure 7-8. In this web page, monitored information including the start time of current online monitoring, the total number of sessions and the time duration are displayed. In addition, the monitored user's chat sessions are also displayed. The user can also select a session and view the details of the selected session. The chat topics detected from the chat sessions are also displayed.

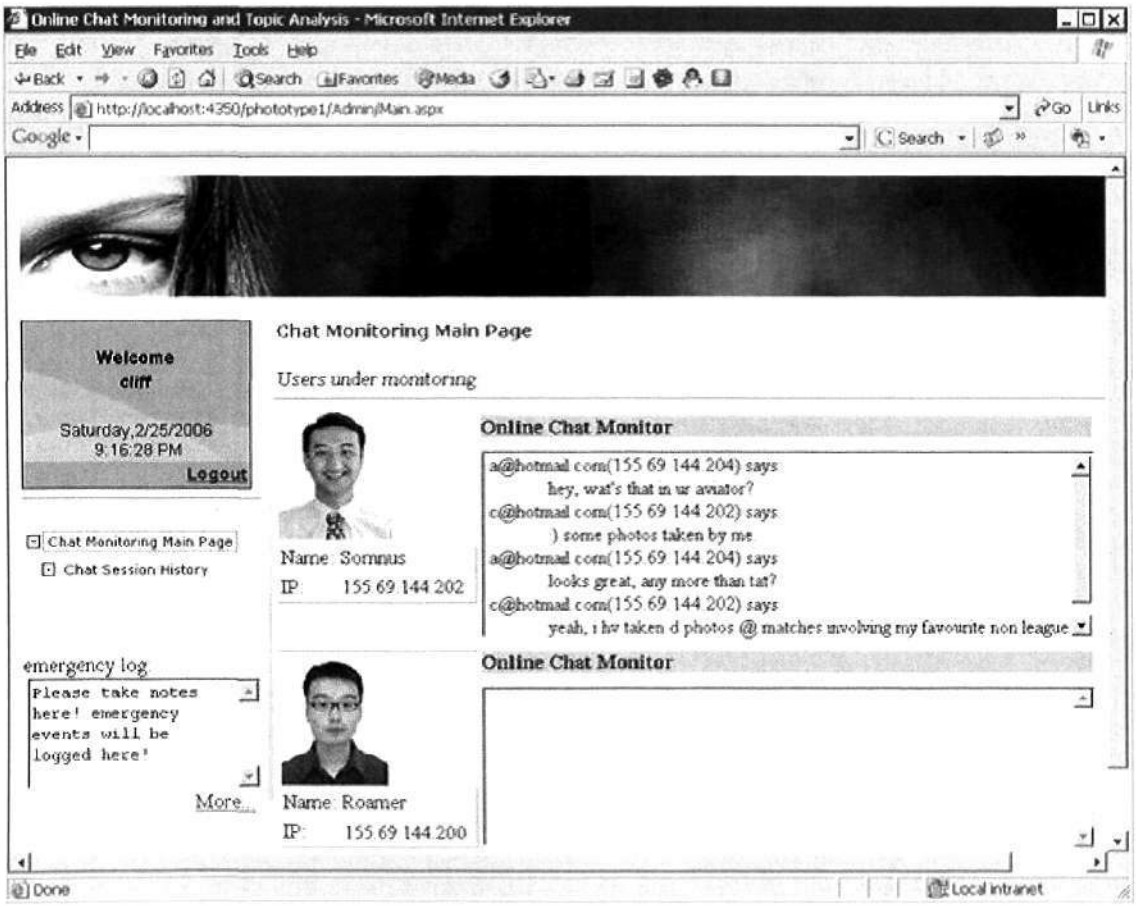


Figure 7-7: Online chat monitoring main page.

7.4 An Application Scenario

As discussed in Chapter 1, instant messaging may be misused and poses a great threat to children using IM. This is especially true for cases on sex solicitation. Here, we give a discussion on some general chatting behaviour exhibited by children who might be exposed to online sex predation [100-103].

- *Unusual chatting behaviour*. Most children that fall victims to online sex predators have shown excessive chat usage [101]. Moreover, they usually chat during the time when their parents are not around or at night.
- *Exposing to inappropriate information especially sex explicit contents*. Pornography is often used in online sex victimization. During a grooming process of sex predation [103], nudity and erotic contents are also used to expose and desensitize the target child. Moreover, discussion on private body parts is also common in online sex solicitation chat sessions.
- *Chatting with strangers*. Sex-offenders are initially strangers to children, and they continuously agree and claim they share the same interest with the children for trust-building or grooming. In other words, sex predators are usually stranger contacts. They progressively exchange chat messages with the target child in great amount of messages.
- *Withdrawn social relationships*. Children being groomed by online sex offenders tend to be socially isolated from peers and even family members [100, 101]. They gradually decrease the amount of messages communicated with their contacts.
- *Leaking personal information*. Children may leak out important personal information such as name, address and telephone. In addition, sex offenders may also give out their telephone number for their target contacts [100, 101].
- *Exchanging images*. In online sex-offence, children are often exposed to images with explicit sex contents.

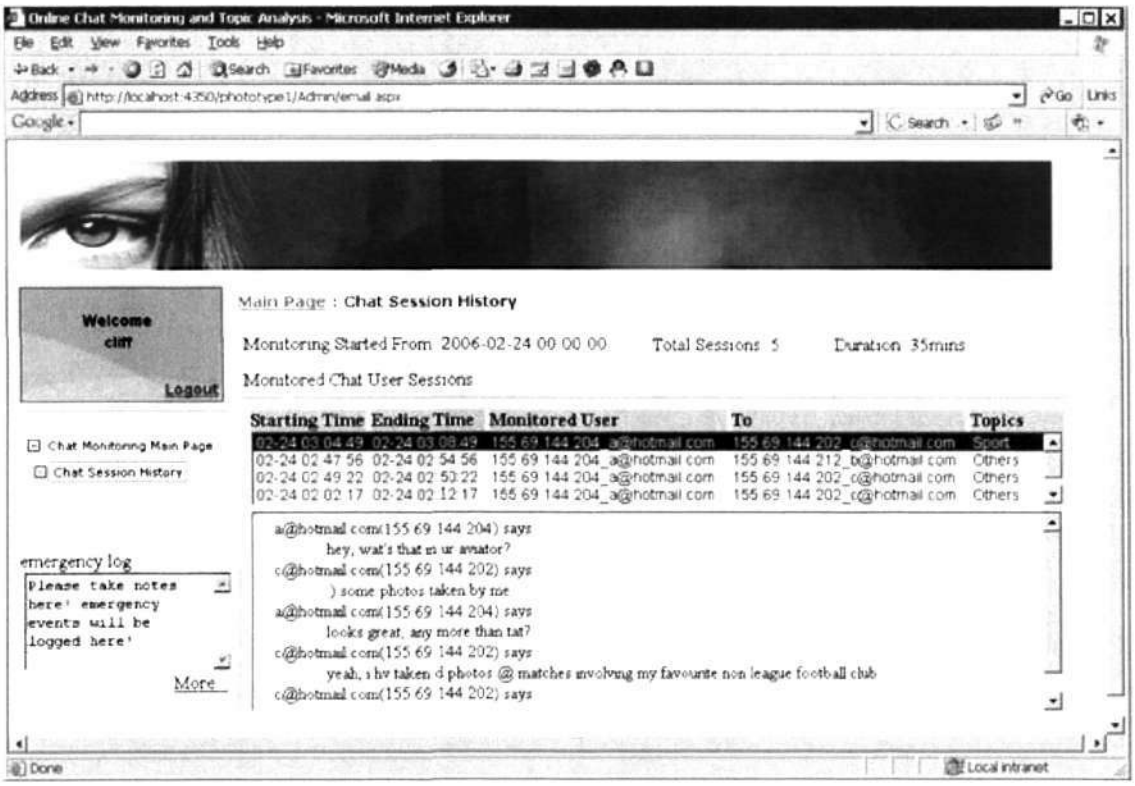


Figure 7-8: Online chat session details.

In order to help protect online safety for chat users, especially children, the IMAnalysis system can be used to monitor the possible signs that might lead a user to become a victim under sex predation. And pre-cautious measures can then be taken to prevent such potential dangers from happening.

- *Chat Message Retrieval*. The message statistics information helps detect unusual chat usage of a user and the time periods the user using IM. In addition, the chat message browsing and searching functions can also help to examine the potential leakage of personal information to strangers.
- *Social Network Analysis*. This helps display the social interactions between a target user and any stranger contacts. For example, if a child is gradually withdrawn from social relationships, his social network will show a few and weak linkages to other contacts except the potential sex predator. In addition, the message direction between the frequently contacted strangers implies whether the child is being exposed to great amount of messages sending to him.
- *Topic Analysis*. This helps search for exposure to inappropriate chat messages. The chat content can be used to determine the topics of chat sessions and thereby deciding

whether the child is being approached by sex predators. Or at least, whether the child is sexually curious and requires further attention from the parents.

For example, let us consider the monitored user 155.69.144.204a@hotmail.com in Figure 7-2, who is in fact a child user.

- Figure 7-2 gives the daily statistics information which has shown that the user has used IM excessively.
- Figure 7-3 shows the message browsing function that has further shown that most IM sessions between this child and 155.69.144.202c@hotmail.com have occurred either during office hours or at evening time when his parents are not around. From the observations based on Chat Message Retrieval, the child user has shown an unusual high IM usage and the parents are advised to perform further analysis on the online behaviour of the child.
- Figure 7-5 shows the social network analysis for the child. It has shown that the child has only interacted with two contacts and exchanged a lot of messages with the contact 155.69.144.202c@hotmail.com. The child is therefore potentially socially isolated and 155.69.144.202c@hotmail.com is a suspect of sex predator even though they exchanged a balanced amount of inbound and outbound messages.
- Figure 7-6 shows the topic analysis results of chat sessions carried out by the child. It has discovered that two chat sessions on Pornography have occurred between the child and the identified contact 155.69.144.202c@hotmail.com.

Therefore, the parent is able to identify and determine the sex predator. Moreover, the sex explicit chat sessions can be saved as evidence for legal action against the suspect.

On the other hand, Online Chat Analysis can also help to view the chat messages of the target monitored user that are exchanged with his contacts online through a web browser. It is a very useful tool to help parents to watch out for inappropriate information exposure of their children especially when they are not at home. The viewing of chat messages online can help identify whether a child is being approached by online sex predators. This can also be used in conjunction with Offline Chat Analysis for detailed message analysis.

7.5 Summary

In this chapter, we have developed an intelligent chat message analysis system, IMAnalysis, which provides the interface to support both offline and online chat message analysis

services. The Offline Chat Analysis comprises three message analysis functions including Chat Message Retrieval, Social Network Analysis and Topic Analysis. In particular, the Chat Message Retrieval generates statistical report on the chat data and allows users to filter chat logs for manual content study. Social Network Analysis discovers the social interactions of a monitored IM user with his contacts. And Topic Analysis detects the topics that the monitored IM users are involved in. The Online Chat Analysis allows users to monitor target IM users' discussion contents in real-time and analyze the chat topics at anytime and anywhere through web browsers. In addition, we have also presented a scenario in using the IMAnalysis system for real-world application. In this application, we have demonstrated how to use the system to facilitate parental control in understanding a child's online discussion and thereby protecting the child from online sex predation.

Chapter 8

Conclusion and Future Work

8.1 Summary and Conclusion

Instant Messaging systems have gained much popularity in recent years as a free, real-time and private communication medium over the Internet between parties in far and remote locations. IM systems also have enhanced features where multimedia content can be exchanged. However, this medium may be exploited as a tool for sex solicitation, information stealing and terrorism network communication. Thus, it has inadvertently evolved into potential menace to unsuspecting individuals, corporate companies and even countries in general.

Such types of illegal information exchange should be monitored in order to protect certain groups of users, such as children who are vulnerable to online sex solicitations. This is where an IM monitoring system will be proved to be useful. However, existing IM monitoring systems generally do not possess a good message recording capability in capturing chat messages. Moreover, there is a lack of intelligent chat analysis functions, which are mainly due to the characteristics of written speech like chat messages that are concise, incomplete, high in vocabulary size, varied language usage, and interwoven with multiple topics. These pose great challenges to the automatic IM chat message analysis task. These problems and challenges, however, have initiated our desire to propose a new chat message monitoring system which is superior in terms of chat recording, system hiding and above all, providing intelligent online monitoring and offline analysis services.

The major contributions of this research work are summarized as follows:

- *A client-side IM monitoring approach is proposed for personalized chat message recording*. The approach adopts both protocol-based and chat window-based methods to record chat messages in real-time. The recording approach has the advantages in overcoming the data encryption and easy adaptation to IM system evolutions. The natural coherency among chat messages in individual chat sessions is preserved by session detection mechanism. Besides, personal information such as client IP, user

Windows account and monitored IM user account are recorded for user identification and tracking purpose. Moreover, the IM monitoring approach has the hiding ability in self-protection from the Windows system and security-related applications.

- *A server-based Adaptive Redundancy Transmission Control and Recovery mechanism is proposed for real-time message transmission from monitored clients to a server for archival* [104]. This mechanism makes use of UDP as the base protocol. It minimizes UDP transmission problems on packet loss, out of sequence and duplication while maintaining high efficiency for real-time communication.
- *A classification-based topic detection approach is proposed for conceptual understanding of chat message topics*. This approach tackles the IM topic detection problem by handling chat message characteristics. In order to overcome the shortness problem, refined sessions of chat messages are used as basic processing unit. The information exchanged as icon text and URLs in addition to text messages is also captured. More importantly, indicative terms are adopted for feature selection, which tackles the high vocabulary size and high noise level problems and, at the same time, provides high computational efficiency for online monitoring. Finally, effective classifiers are employed to identify chat sessions into multiple topics and easy incorporation of new models for detection.
- *An instant message analysis system is developed for intelligent chat message analysis*. This system comprises a set of monitoring applications and tools that support both offline IM monitoring and online analysis of chat messages. The offline analysis integrates Chat Message Retrieval, Social Network Analysis and Topic Analysis in order to provide system users with the understanding on the statistical properties, social relationships and semantic distribution of the monitored users' chat space. The web-based online chat analysis allows system users to monitor the discussion contents and analyze the chat topics at anywhere.

8.2 Ethical Issues

In this research, our main focus is on investigation of techniques for IM monitoring. The use of IM monitoring should improve productivity and the safety of an organization or home. Although the collection of data under an organization's own premises is legalized [105], and when we apply such technology to work places or homes, privacy of individuals should also

be respected and observed. Employees or individuals should thus be informed about the installation of the monitoring software [105-107]. Guidelines on the use of IM systems in organizations should also be made know to their employees. More importantly, the recorded chat data should not be released and used for other illegitimate purposes [105-107]. Hopefully, such new technology can be used to help to make our chat activities safer than before.

8.3 Future Work

In this research, we have developed an IM monitoring system for real-time personal chat message recording and intelligent chat message analysis. This research can be further extended in the following directions: Multilingual Chat Message Analysis, Pattern Analysis, Unsupervised Chat Analysis, Multimedia Content Analysis and Further Topics Detection.

8.3.1 Multilingual Chat Message Analysis

Currently, the IM monitoring systems and techniques are developed targeting mainly at English chat sessions. However, Instant Messaging environment is typically multilingual context especially in Asian countries. Chat messages in Chinese language, as an example, are exchanged extensively by the Chinese-speaking population. Thus, the current IM monitoring systems, including IMAnlaysis, are ineffective in analyzing the semantic information composed in such languages.

Therefore, we can extend IMAnalysis to classify English-Chinese bilingual content to address the multilingual chat message analysis problem. A possible solution to this problem is to perform Pre-processing and Feature Extraction separately for English and Chinese text. English content will be stemmed while Chinese will be segmented. Keyword dictionaries can be compiled containing both English and Chinese indicative keywords for a particular category to identify high quality features for each document, which shall be combined to form a single document vector representation for topic classification. Moreover, the distinct nature of Chinese language shall be considered during the data preparation stage.

8.3.2 Pattern Analysis

Pattern analysis constructs an accurate profile to reveal a user's IM usage. An abrupt change in the user's IM usage would imply the odd behaviour and should be brought to the attention of monitoring authority.

In this research, the topic classification approach has presented the monitored target's interest space in IM discussion. A simple statistical based topic trend detection that can reflect the evolvement of IM user interest can be developed based on the average number of chat sessions dedicated to various chat topics over a period of time. The user behaviour on chat messages can thus be analyzed. More advanced techniques such as association rule mining [108] and sequential pattern mining [109] can be adopted in chat message pattern analysis as well. These advanced data mining techniques are capable of extracting frequent user involvement in various chat topic sequence patterns or behaviour according to the time-of-day or day-of-week criteria.

8.3.3 Unsupervised Analysis

In this research, we have adopted classification algorithms for topic detection purpose. Unsupervised algorithms such as clustering techniques should also be investigated for providing an automatic visualization of chat messages session groups and a way for retrieving chat sessions with semantically similar contents. However, the difficulty in applying clustering techniques is the compromise in dealing with real-time incremental chat message data while maintaining acceptable high accuracy. Besides, the small set of available features in each chat message session and high vocabulary size in chat space will also pose a problem in using clustering techniques. Nonetheless, it is an interesting direction to investigate how the clustering techniques can perform on chat message data.

8.3.4 Multimedia Content Analysis

IM applications have been evolved into a multimedia communication medium, where audio/video conferencing and image/audio/video files are utilized for information exchange. To carry out automatic analysis of such multimedia content, the system will need to have the image and audio/video analysis, and classification functionality. This can be realised using classification models with image feature extraction mechanisms. The classification algorithms can then be trained to classify the images according to the features extracted. A

typical application in online sex solicitation material may be supported by sampling of human skin tone and body shapes on the image, since sexually explicit images usually show images with nude body parts.

8.3.5 Further Topic Detection

The current IMAnalysis system only targets at detecting five categories of chat messages. Nevertheless, the approach used by the system which is based on indicative terms can be extended to include other conceptual domains, such as terrorism and bomb-making for preserving the peace and harmony of the society. Moreover, we can also adopt conceptual hierarchical structure into indicative terms for detailed profiling of monitored user's interests in online discussions.

References

[1] MSN.com, "MSN Messenger version - 7.5", available at http://messenger.msn.com/.

[2] Yahoo.com, "Yahoo! Messenger with voice – chat, call, share photos, and more", available at http://messenger.yahoo.com/.

[3] ICQ.com, "ICQ.com – community, people search and messaging service!" available at http://www.icq.com.

[4] Microsoft PressPass, "100 million customers and counting: MSN messenger extends worldwide lead among Instant Messaging providers", available at http://www.microsoft.com/presspass/press/2003/may03/05-12100millionPR.asp.

[5] B. Woods, "IM more popular than ever at work – study", Nov 15 2001, available at http://boston.internet.com/news/article.php/923611.

[6] ZDNet Research, "MSN Messenger overtakes AOL AIM and ICQ as top instant messenger", available at http://www.itfacts.biz/index.php?id=P1613.

[7] K. Thomas, "Kids need enduring smarts with instant messaging", available at http://www.usatoday.com/tech/news/2001-07-10-instant-messaging-safety.htm.

[8] ParentHood.com, "Cyber Brats: Bullies who taunt their peers with the click of a mouse", available at http://www.parenthood.com/articles.html?article_id=4335.

[9] Stellar Technologies, "Stellar Technologies, Inc. – web, e-mail & IM management Stellar Internet global employee management", available at http://www.stellarim.com/.

[10] Spectorsoft.com, "SpectorSoft.com – Spector Pro for Windows", available at http://www.spectorsoft.com/products/SpectorPro_Windows/index.html.

[11] Zemerick Software Inc., "Zemerick software – home of Chat Watch", available at http://www.zemericks.com/.

[12] J. Bengel, S. Gauch, E. Mittur, and R. Vijayaraghavan, "ChatTrack: chat room topic detection using classification", in *2nd Symposium on Intelligence and Security Informatics*, 2004.

[13] Tencent.com, "Tencent homepage", available at http://www.qq.com.

[14] Controlling the Virtual World, Chat Systems, available at http://www-cs-education.stanford.edu/class/cs201/projects-98-99/controlling-the-virtual-world/technology/realtime.html.

[15] M. Mannan and P.C. van Oorschot, "Secure Public Instant Messaging: A Survey", in proceedings of 2nd Annual Conference on Privacy, Sep 2004.

[16] IServerd project, "OSCAR (ICQ v7/v8/v9) protocol documentation", available at http://iserverd.khstu.ru/oscar/.

[17] J. B. Postel, "Transmission Control Protocol", *RFC793*, Sep 1981.

[18] Wikipedia the free encyclopedia, "Mobile Status Notification Protocol", available at http://en.wikipedia.org/wiki/MSNP.

[19] Wikipedia the free encyclopedia, "YMSG", avaiable at http://en.wikipedia.org/wiki/YMSG.

[20] J. B. Postel, "User Datagram Protocol", *RFC768*, Aug 1980.

[21] Exploreanywhere.com, "SpyBuddy spy software", available at http://www.exploreanywhere.com/sb-intro.php.

[22] Exploreanywhere.com, "Chat Blocker Spy Software", available at http://www.exploreanywhere.com/cb-intro.php.

[23] InvisibleKeylogger.com, "Invisible Keylogger – the perfect stealth keylogger" available at http://www.invisiblekeylogger.com/.

[24] V. Tuulos and H. Tirri, "Combining topic models and social networks for Chat Data mining", in *Proceedings of the IEEE/WIC/ACM Conference on Web Intelligence (WI 2004)*, pp. 206-213.

[25] B. Ripplinger and P. Schimidt, "Automatic Multilingual Indexing and Natural Language Processing", in SIGIR, 2001.

[26] C. E. Porter, "A Typology of Virtual Communities: A Multi-Disciplinary Foundation for Future Research", in *Journal of Computer-Mediated Communication*, vol. 10(1), 2004.

[27] H. Wentworth and F. B. Stuart, *Dictionary of American Slang,* Thomas Y, Crowell Company, New York, 1975.

[28] R. L. Chapman, *New Dictionary of American Slang,* Harper and Row, New York, 1986.

[29] UrbanDictionary, available at www.UrbanDictionary.com.

[30] P. Dickson, *Slang: The Authoritative Topic-by-Topic Dictionary of American Lingoes from All Walks of Life,* Pocket Books, New York, 1998.

[31] A. Meehan, G. Manes, L. Davis, J. Hale, and S. Shenoi, "Packet sniffing for automated chat room monitoring and evidence preservation", in *Proceedings of the Second annual*

IEEE System, Man, and Cybernetics Information Assurance Workshop, West point, New York, Jun 2001.

[32] Akonix Systems Inc., "Aknoix – enterprise Instant Messaging security & management solutions", available at http://www.akonix.com/.

[33] Wikipedia the free encyclopedia, "Virtual private network", available at http://en.wikipedia.org/wiki/VPN.

[34] Z. F. Zhang, "COM interface hooking and its application - part I", available at http://www.codeproject.com/com/cominterfacehookingpart.asp.

[35] Microsoft MSDN, "Windows messenger client reference", available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/WinMessenger/winmessenger/reference/messengeruasdk/cpp_client_entry.asp.

[36] PestPatrol, "About keyloggers", available at http://www.pestpatrol.com/Support/About/About_KeyLoggers.asp.

[37] R. Sahasrabudhe, "Exploring the application development options for IBM Lotus Workplace 2.0", available at http://www-128.ibm.com/developerworks/lotus/library/lwp-api/, 29 Nov 2004.

[38] K. Marsh, "Win32 hooks", Microsoft Developer Network Technology Group, pp. 1-14, Jul 1993 (revised Feb.1994).

[39] M. Pietrek, "An in-depth look into the Win32 Portable Executable file format-part 1", MSDN Magazine, Microsoft, Feb 2002.

[40] M. Pietrek, "An in-depth look into the Win32 Portable Executable file format-part 2", MSDN Magazine, Microsoft, Feb 2002.

[41] Microsoft MSDN, "Homepage: Spy++", available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vcug98/html/_asug_home_page.3a_.spy.2b2b.asp.

[42] A. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications", *RFC 1889*, Jan 1996.

[43] G. Hellstrom, "RTP payload for text conversation", *RFC 2739*, May 2000.

[44] K.V. Chin, S.C. Hui, and S. Foo, "Packet voice recovery techniques for real-time Internet voice communication", in *Proceedings of 4th Asia-Pacific Conference on Comunications/6th Singapore International Conference on Communications Systems (APCC/ICCS'98)*, Singapore, pp. 122-126, 1998.

[45] S. C. Hui, C. K. Yeo, and K. Jin, "RTFaxing: Internet real-time faxing system", *Networking and Information Systems*, vol. 2(3), pp. 323-344, 1999.

[46] S. C. Hui and F. Wang, "Remote video monitoring over the WWW", *Multimedia Tools and Applications*, vol. 21, pp. 173-195, 2003.

[47] C. Y. Cheng, "Introduction on text compression using Lempel, Ziv, Welch (LZW) method", available at http://www.geocities.com/yccheok/lzw/lzw.html.

[48] Unicode, "Unicode Technical Standard #6, A Standard Compression Scheme for Unicode", technical report, available at http://www.unicode.org/unicode/reports/tr6/.

[49] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic Qos control of multimedia applications based on RTP", *Computer Communications*, vol. 19, pp. 49-58, 1996.

[50] E. Elnahrawy, "Log-based chat room monitoring using text categorization: a comparative study", in *Proceedings of the IASTED International Conference on Information and Knowledge Sharing (IKS 2002)*, St. Thomas, US Virgin Islands, 2002.

[51] T. Kolenda, L. K. Hansen, and J. Larsen, "Signal detection using ICA: application to chat room topic spotting", in *Proceedings Of the 3rd International Conference on Independent Component Analysis and Blind Source Separation*, pp. 540-545, 2001.

[52] E. Bingham, A. Kab, and M. Girolami, "Topic identification in dynamical text by complexity pursuit", in *Neural Processing Letters*, vol. 17, pp. 69-83, 2003.

[53] N. Fuhr, S. Hartmann, G. Lustig, M. Schewantner, K. Tzeras and G. Knorz, "AIR/X – a rule-based multistage indexing system for large subject fields", in *Proceedings of RIAO' 91*, pp. 606-623, 1991.

[54] Y. Yang and C. G. Chute, "An example-based mapping method for text categorization and retrieval", *ACM transaction on Information Systems (TOIS)*, vol. 12(3), pp. 252-277, 1994.

[55] B. Masand, G. Linoff, and D. Waltz, "Classifying news stories using memory based reasoning", in *15th Ann International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92)*, pp. 59-64,1992.

[56] Y. Yang, "Expert network: effective and efficient learning from human decisions in text categorization and retrieval", in *17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pp. 13-22, 1994.

[57] K. Tzeras and S. Hartman, "Automatic indexing based on Bayesian inference networks", in *16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, pp. 22-34, 1993.

[58] I. Moulinier, "Is learning bias an issue on the text categorization problem?" technical report, LAFORIA-LIP6, Universite Paris VI, 1997.

[59] Y. Yang, T. Pierce, and J. Carbonell, "A study on retrospective and on-line event detection", in *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 28-36,1998.

[60] W.S. Young and K. Sycara, "Text clustering for topic detection", technical report CMU-RI-TR-04-03, Robotics Institute, Carnegie Mellon University, January, 2004.

[61] Text REtrieval Conference (TREC), "Reuters Corpora @ NIST", available at http://trec.nist.gov/data/reuters/reuters.html.

[62] Open Directory Project, "ODP – Open Directory Project", available at http://dmoz.org/.

[63] 20 Newsgroup, "Home page for 20 Newsgroup data set", available at http://people.csail.mit.edu/jrennie/20Newsgroups/.

[64] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization", in *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, pp. 81-93, 1994.

[65] C. Apte, F. Damerau, and S. Weiss, "Text mining with decision rules and decision trees", in *Proceedings of the Conference on Automated Learning and Discovery, Workshop 6: Learning from Text and the Web*, 1998.

[66] H. T. Ng, W. B. Goh, and K. L. Low, "Feature selection, perception learning, and a usability case study for text categorization", in *20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, pp. 67-73, 1997.

[67] C. Apte, F. Damerau, and S. Weiss, "Towards language independent automated learning of text categorization models", in *Proceedings of the 17th Annual ACM/SIGIR conference*, pp. 23-30, 1994.

[68] W. Cohen and Y. Singer, "Context-sensitive learning methods for text categorization", in *SIGIR'96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 307-315, 1996.

[69] I. Moulinier, G. Raskinis, and J.-G. Ganascia, "Text categorization: a symbolic approach", in *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval*, pp. 87-99, Las Vegas, US, 1996.

[70] E. D. Wiener, J. Pedersen, and A. Weigend, "A neural network approach to topic spotting", in *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, pp. 317-332, Nevada, Las Vegas, 1995.

[71] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", in *European Conference on Machine Learning (ECML)*, pp. 137-142, Berlin, Springer, 1998.

[72] Y. Yang and X. Liu, "A re-examination of text categorization methods", in *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp. 42-49, 1999.

[73] C. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*, the MIT Press, Cambridge, Massachusetts, 1999.

[74] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization", in *the 14th International Conference on Machine Learning*, pp. 412-420, Morgan Kaufmann, 1997.

[75] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization", in *Proceedings of the 1998 ACM CIKM International Conference on Information and Knowledge Management*, pp. 148-155, 1998.

[76] M.-L. Antonie and O. R. Zaiane, "Text Document Categorization by Term Association", in *IEEE International Conference on Data Mining*, pp. 19-26, 2002.

[77] S. Tiun, R. Abdullah, and T. E. Kong, "Automatic Topic Identification Using Ontology Hierarchy", in *Proceeding of the 2nd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2001)*, 2001.

[78] N. Guarino and P. Giaretta, "Ontologies and Knowledge Bases: Towards a Terminological Clarification", in *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, Amsterdam, Holland: IOS Press, 1995.

[79] IRC.org, "IRC.org – home of IRC", available at http://www.irc.org.

[80] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1998.

[81] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society*, vol. 39, pp.1-38, 1977.

[82] Linguistic Data Consortium, "TDT-pilot", available at http://www.ldc.upenn.edu/Projects/TDT-Pilot.

[83] Y. Yang, T. Pierce, and J. Carbonell, "A study on retrospective and on-line event detection", in *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, 1998.

[84] S. Chung and D. McLeod, "Dynamic topic mining from news stream data", in *Proceedings of ODBASE*, 2003.

[85] CNN.com, "CNN.com – breaking news, U.S., world, weather, entertainment & video news", available at http://www.cnn.com.

[86] C. Clifton and R. Cooley, "TopCat: data mining for topic identification in a text corpus", in *Proceedings of the 3rd European Conference of Principles and Practice of Knowledge Discovery in Database*, Prague, Czech Republic, 1999.

[87] NIST - National Institute of Standards and Technology, "1998 topic detection and tracking project (TDT-2)", available at http://www.nist.gov/speech/tests/tdt/tdt98/index.htm.

[88] E. H. Han, G. Karypis, and V. Kumar, "Clustering based on association rule hypergraphs", in *Proceedings of SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery*, ACM, 1997.

[89] R. Lee, "Teenage life online: the rise of the networked generation", Youth.Net Conference, Singapore August, 2003, available at http://www.pewinternet.org/ppt/2003%208.14.03%20--%20Singapore%20Youth.Net%20Conference.ppt.

[90] Ugroups.com, "UGroups: free web access to the Usenet newsgroups and forums", available at http://www.ugroups.com.

[91] Jolt.co.uk online game forum, "jolt.com.uk public forums – powered by vBulletin", available at http://forums.jolt.co.uk.

[92] AdultFriendFinder.com, "Adult friendfinder – the world's largest sex personal sites", available at http://www.adultfriendfinder.com.

[93] C. J. van Rijsbergen, *Information Retrieval*, Butterworths, London, 1979.

[94] C. Apte, F. Damerau, and S. Weiss, "Automated learning of decision rules for text categorization", *ACM Transactions on Information Systems*, vol. 12(3), pp. 233-251, 1994.

[95] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization", in *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pp. 412-420, 1997.

[96] Orgnet.com, "An introduction to Social Network Analysis", available at http://www.orgnet.com/sna.html.

[97] J. Resig, S. Dawara, C. Homan, and A. Teredesai, "Extracting social networks from Instant Messaging populations", in *KDD 04 Link Discovery Workshop (LinkKDD 2004)*, 2004.

[98] P. Mutton, "Inferring and visualizing social networks on Internet Relay Chat", available at http://www.jibble.org/piespy/.

[99] S. A. Çamtepe, M. Goldberg, M. Magdon-Ismail, and M. Krishn, "Detecting conversing groups of chatters: a model, algorithms, and tests", in *IADIS AC 2005*, pp. 89-96, 2005.

[100] Children and Family Canada, "Protecting Your Children from Online Predators" , available at http://www.cfc-efc.ca/docs/mnet/00001239.htm.

[101] U.S. Department of Justice, "A parent's guide to Internet safety", Federal Bureau of Investigation publications, available at http://www.fbi.gov/publications/pguide/pguidee.htm.

[102] M. Sulliven, *Safety Monitor: How to Protect Your Kids Online*, Bonus Books Inc, Aug, 2002.

[103] I. R. Berson, "Grooming cybervictims: the psychosocial effects of online exploitation for youth", *Journal of School Violence*, pp. 1-9, 2003.

[104] H. C. Dong, S. C. Hui, and K. Y. Chang, "An adaptive message transmission mechanism for real-time monitoring over the Internet", in *Proceedings of the IEEE International Conference on Communication and Information*, Beijing China, pp. 802-806, 2005.

[105] G. Dodig-Crnkovic, "Privacy and Protection of Personal Integrity in the Working Place", ZiF-Workshop Privacy and Surveillance, ZiF: Zentrum für interdisziplinäre Forschung, Uni, February, 2006.

[106] N. Flynn, *Instant Messaging Rules: A Business Guide to Managing Policies, Security, and Legal Issues for Safe IM Communication*, AMACOM, 2004.

[107] D. Kawamoto, "Mind those IMs—your Cubicle's walls have eyes", available at http://news.com.com/Mind+those+IMs--your+cubicle's+walls+have+eyes/2100-1014_3-5423220.html, CNet News.com.

[108] Y. Li, P. Ning, X. S. Wang, and S. Jajodia, "Discovering calendar-based temporal association rules", in *Proceedings of the 8th International Symposium on Temporal Representation and Reasoning*, 2001.

[109] B. Y. Zhou, S. C. Hui and A. C. M. Fong, "CS-mine: an efficient WAP-tree mining for web access pattern", in *Proceedings of Sixth Asia Pacific Web Conference*, Hangzhou, China, pp. 523-532, 2004.

Appendix A

Common Acronyms for Instant Messages

Table A-1: Acronyms and equivalent phrases.

Acronym	Equivalent Phrase
A/S/L	Age/Sex/Location
ABT2	About To
AFAIK	As Far As I Know
AFAYC	As Far As You're Concerned
AFINIAFI	A Friend In Need Is A Friend Indeed
ASAP	As Soon As Possible
ASL	Age/Sex/Location
ATST	At The Same Time
AYK	As You Know
B4N	Bye For Now
B4U	Before You
B4YKI	Before You Know It
BB4N	Bye-Bye for Now
BBL	Be Back Later
BBS	Be Back Soon
BF	Boy Friend
BR	Bathroom
BRB	Be Right Back
BTA	But Then Again
BTW	By The Way
BW	Best Wishes
CID	Consider It Done
CMF	Count My Fingers
CSL	Can't Stop Laughing
CU	See You
CUL	See You Later
CUL8R	See You Later
CYL	See You Later
CYT	See You Tomorrow
DF	Dear Friend
DGA	Don't Go Anywhere
DGT	Don't Go There
DH	Dear Hubby or Husband
DIY	Do It Yourself
DLTM	Don't Lie To Me
DMI	Don't Mention It
DQMOT	Don't Quote Me On This
EM?	Excuse Me?
EOD	End Of Day
EOM	End Of Message

Acronym	Equivalent Phrase
F2F	Face-to-Face
FF	Friends Forever
FO	F*** Off
FTTB	For The Time Being
FYI	For Your Information
GA	Go Ahead
GAL	Get A Life
GG	Gotta Go
GL	Good Luck
GF	Girl Friend
GMTA	Great Minds Think Alike
GT	Go To
GTG	Got To Go
GTH	Go To Hell
HAK	Hugs And Kisses
HIH	Hope It Helps
I 1-D-R	I Wonder
IAC	In Any Case
IAE	In Any Event
IC	I see
ICBW	I Could Be Wrong
IDK	I Don't Know
IK	I Know
ILU	I Love You
ILY	I Love You
IM	Instant Messaging
IMO	In My Opinion
IMS	I Am Sorry
IOW	In Other Words
JIC	Just In Case
JK	Just Kidding
JTLYK	Just To Let You Know
KISS	Keep It Simple Stupid
KIT	Keep In Touch
KOTC	Kiss On The Cheek
KOTL	Kiss On The Lips
KPC	Keeping Parents Clueless
KWIM	Know What I Mean
LD	Long Distance
LMK	Let Me Know
LOL	Laughing Out Loud

Acronym	Equivalent Phrase
LTNS	Long Time No See
MYOB	Mind Your Own Business
N/A	Not Applicable
N/T	No Text
N1	Nice One
N2M	Not To Mention
NBD	No Big Deal
NCG	New College Graduate
NM	Never Mind or Nothing Much
NMP	Not My Problem
NOYB	None Of Your Business
NP	No Problem
NRN	No Reply Necessary
NTK	Nice To Know
NTYMI	Now That You Mention It
NUFF	Enough Said
NVM	Never Mind
NW	No Way!
NYCFS	New York City Finger Salute
OIC	Oh, I see
OMG	Oh My Gosh
ONNA	Oh No, Not Again
OOI	Out Of Interest
OOO	Out Of Office
OS	Operating System
OT	OverTime
OTOH	On the Other Hand
OTP	On the Phone
OUSU	Oh, You Shut Up
P&C	Private and Confidential
POS	Parent Over Shoulder
POV	Point of View
QL	Quit Laughing!
R&D	Research & Development
R&R	Rest & Relaxation
RL	Real Life
RLF	Real Life Friend

Acronym	Equivalent Phrase
RN	Right Now!
ROFL (ROTFL)	Rolling on the Floor Laughing
RU	Are You?
RUMORF	Are You Male Or Female
RUOK	Are you Okay?
S2R	Send To Receive
SEP	Somebody Else's Problem
SorG	Straight or Gay?
SWDYT	So What Do You Think?
SYS	See You Soon
TBH	To Be Honest
TM	Trust Me
TTG	Time to Go
TTUL	Talk to You Later
TTYL	Talk To You Later
TY	Thank You
TYVM	Thank You Very Much
U2	You Too?
U-L	You Will
U R	You Are
VC	Venture Capital
WAD	Without A Doubt
WB	Welcome Back or Write Back
WFM	Works For Me
WTF	What The F***
WTG	Way To Go!
WTH	What the Heck
WU?	What is Up
XME	Excuse Me
YDKM	You Don't Know Me
YGBK	You Gotta Be Kiddin'
YKW?	You Know What?
YNK	You Never Know
YW	You are Welcome
2B or not 2B	To Be Or Not To Be

Appendix B

Indicative Term Dictionary for Games Category

Table B-1: Indicative terms for Games.

Index	Indicative Terms	Index	Indicative Terms
1	account | accounts	31	pc game | computer game | video game | pc games | computer games |video games
2	admin	32	play | played | playing | plays
3	battlefield | battle field	33	player | players
4	blood	34	quakenet | quake net | quakenets | quake nets
5	campaign | campaigns | cmpn | cmpgn	35	quest | quests
6	cheat | cheaters | cheating | cheats	41	release | releases
7	clan | clans	42	updates | update
8	command | commands	43	version
9	complete | completed	44	war
10	cs | counter strike | counterstrike	45	world
11	custom	46	xbox
12	demo | demos		
13	directx		
14	download | downloads | downloading | downloaded		
15	file		
16	game | games | gaming | gamer | gamers		
17	goal | goals		
18	graphics card | graphics cards| gfx card | gfx card | graphic card| graphic cards		
19	install | installs | installed | installation		
20	join | joins		
21	kills | killing		
22	latest		
23	level | levels | lvl | lvls		
24	map | maps		
25	mod |mods		
26	move | moves		
27	multiplay | multiplayer | multiplayers | multi-play | multi-player		
28	network | networks		
29	nintendo		
30	patch | patches		