

# Understanding Instant Messaging Traffic Characteristics

Zhen Xiao<sup>1</sup>, Lei Guo<sup>2</sup>, and John Tracey<sup>1</sup>

<sup>1</sup>IBM T. J. Watson Research Center  
19 Skyline Drive  
Hawthorne, NY 10532, USA  
{xiaozen, traceyj}@us.ibm.com

<sup>2</sup>Department of Computer Science  
The Ohio State University  
Columbus, Ohio 43210, USA  
lguo@cse.ohio-state.edu

## Abstract

*Instant messaging (IM) has become increasingly popular due to its quick response time, its ease of use, and possibility of multitasking. It is estimated that there are several millions of instant messaging users who use IM for various purposes: simple requests and responses, scheduling face to face meetings, or just to check the availability of colleagues and friends.*

*Despite its popularity and user base, little has been done to characterize IM traffic. One reason might be its relatively small traffic volume, although this is changing as more users start using video or voice chats and file attachments. Moreover, all major instant messaging systems route text messages through central servers. While this facilitates firewall traversal and gives instant messaging companies more control, it creates a potential bottleneck at the instant messaging servers. This is especially so for large instant messaging operators with tens of millions of users and during flash crowd events. Another reason for the lack of previous studies is the difficulty in getting access to instant messaging traces due to privacy concerns.*

*In this paper, we analyze the traffic of two popular instant messaging systems, AOL Instant Messenger (AIM) and MSN/Windows Live Messenger, from thousands of employees in a large enterprise. We found that most instant messaging traffic is due to presence, hints, or other extraneous traffic. Chat messages constitute only a small percentage of the total IM traffic. This means, during overload, IM servers can protect the instantaneous nature of the communication by dropping extraneous traffic. We also found that the social network of IM users does not follow a power law distribution. Our analysis sheds light on instant messaging system design and optimization and provides a scientific basis for instant messaging workload generation.*

## 1 Introduction

Instant messaging (IM) has become increasingly popular. It is estimated that there are several millions of instant messaging users all over the world. Teenagers use instant messaging to keep in touch with their friends and families, while corporate users exchange IMs to discuss work. Compared to other methods of communication, instant messaging offers several advantages: its almost synchronous nature makes it ideal for simple requests and responses. It provides presence and event notification which make it easy to keep track of the availability of colleagues and friends (“buddies” in IM terminology). In addition, most IM systems today incorporate support for voice or video chats as well as file transfer, making it an integrated environment for a wide variety of communication needs. The popularity of IM is expected to continue increasing in the foreseeable future.

Despite its huge popularity and user base, little work has been done to understand the workload generated by this important application. The limited existing work in this area focuses mostly on understanding the social behavior of instant messaging users. For example, the authors of [10] interviewed twenty IM users and found that IMs are often used to negotiate the availability of co-workers who may then switch to a different media (e.g., phone) for complex discussions. Work in [2] studies IM usage from sixteen teenagers and found different social behaviors between high school and college students. Since the bulk data in those studies came from surveys and interviews, their scopes are inevitably limited to small user samples (often fewer than fifty) and to the subjective descriptions of those users. The only work we are aware of that studies IM usage in a relatively large scale is [5] which collected IM logs from 437 users. However, they developed a specialized instant messenger, called “Hubbub”, to facilitate trace collection and studied the social behavior of that particular user community. While providing similar functionality, their instant messenger is incompatible with popular instant messaging applications, such as AOL Instant Messenger. As a result, it

is not clear to what degree their results can be extended to the general IM population. Like previous work, their focus was to understand the social behavior, not the traffic patterns, of IM users.

Other previous work includes [4] which provides an overview of the architecture and protocols of three popular instant messengers (from AOL, MSN, and Yahoo) and compares their features. [8], [9] and [12] investigate the security vulnerabilities of IM networks in the face of viruses and worms. To the best of our knowledge, none of the existing work focuses on the characteristics of instant messaging traffic.

One reason for the lack of study might be the relatively small traffic volume of IM applications, since most text messages are short. In fact, some systems (e.g., AOL) put a limit on both the size and the rate of instant messages their users can send and use “rate throttling” to restrain uncooperative users. However, with more and more use of video and voice chats and file attachments, traffic volume is likely to increase substantially. Another reason for the lack of previous studies is the difficulty in getting access to instant messaging traces due to privacy concerns.

The lack of systematic study of instant messaging applications impede the design, implementation, and performance optimizations of such systems. For example, almost all major instant messaging systems route text messages through central servers. While this facilitates firewall traversal and gives instant messaging companies more control, it creates a potential bottleneck. This is especially so for large instant messaging operators with tens of millions of users and during flash crowd events. In addition, it impedes development of realistic workload generators for IM traffic and makes standardization efforts across different IM implementations difficult.

We are fortunate to get access to instant messaging traffic from thousands of employees in a large enterprise. Compared to previous studies, our workload not only covers a much larger user base but also has much higher traffic volume. In this paper, we focus on the traffic of two popular instant messaging systems: AOL Instant Messenger (AIM) and Microsoft Messenger (MSN). Details of our finding will be presented in later sections, but here are some highlights:

1. Most instant messaging traffic is due to presence, hints, or other *extraneous* traffic. Chat messages constitute only a small percentage of the total instant messaging traffic. During overload, therefore, IM servers can protect the instantaneous nature of the communication by dropping extraneous traffic.
2. Unlike previous studies, we found that the social network of IM users does not follow a power law distribution. It can be characterized by a Weibull distribution instead.

Our analysis sheds light on instant messaging system design and optimization and provides a scientific basis for instant messaging workload generation.

The rest of the paper is organized as follows. Section 2 introduces some basic background of IM systems and protocols. Section 3 describes our workload collection methodology and gives an overview of our collected workload. Section 4 analyzes the online behaviors of AIM and MSN Messenger users. Section 5 studies the IM traffic in details from the message level, the conversation level, and the user level. Section 6 summarizes other related work. Section 7 concludes.

## 2 Background on Instant Message protocols

In this section, we provide the basic background on IM protocols, focusing on two popular IM systems, AIM and MSN.

Each IM user has a unique *screen name* in the system and a list of *buddies*. When a user is online, the IM system notifies the presence of her buddies to her as well as notifying her presence to her buddies. Most instant messaging systems are client-server based and route chat messages through central servers. Chat messages are typically small, and the IM system may limit the size of each message. Besides the chat messages, the IM client software also generates certain *hint* messages when a user is typing or editing a message. This feedback can be useful, for example, to prompt her buddy to wait for her response. IM users can also select a favorite picture as their *icons* which will be displayed when chatting with their buddies. In both AIM and MSN Messenger, a user can send messages to a buddy only when the buddy is online. In contrast, some IM systems such as QQ support offline messaging where messages sent to offline users are buffered at the servers and delivered when those users come online.

Besides text-based messaging, most IM systems today support voice/video chat and file transfers in an integrated environment. While IM servers are involved in the initial establishment of a voice/video call or a file transfer, subsequent communications are often conducted between the two end users directly in a peer-to-peer fashion. However, when the caller and callee are both shielded by a firewall or NAT router, their communication must be relayed by IM servers. The focus of this study is the communication between IM clients and servers. We do not analyze peer to peer IM traffic.

### 2.1 Server Architecture

Both AIM and MSN use an asymmetric architecture. In AIM, an authentication server processes user logins and

redirects users to messaging servers, called BOS (Basic Oscar Services) servers. A user disconnects from the authentication server as soon as it connects to a BOS server. It remains connected to the BOS server until it logs off or it is redirected to another BOS server due to, for example, server load redistribution. A user sends messages to and receives messages from her buddy via her BOS server. For group chats, the initiator creates a room in a chat room server, and invites other users to join. As an integrated messaging environment, there are also some other servers such as email servers and buddy icon servers in the AIM system.

With the MSN messaging system, a user contacts a dispatch server (DP server) first, which redirects the user to a notification server (NS server). The NS server is responsible for notifying the user of various events, such as buddy presence, instant message calls, and email arrivals. Similar to a user’s connection to her BOS server in AIM, a user’s connection to her NS server is persistent throughout her entire online time. When one user wants to chat with another, it sends a request to the NS server, which then redirects the user to a switchboard server (SB server). Correspondingly, the callee of an IM chat receives a notification from her NS server, and then is redirected to the same SB server. All conversations between the two users are then relayed by the SB server. The advantage of this architecture is that an instant message is relayed by a single server (i.e., the SB server). In contrast, an instant message in AIM is relayed by two servers: the BOS server of the caller and that of the callee. Another difference is: an AIM user sends and receives all chat messages over a single connection to her BOS server, while a MSN user communicates with different buddies over separate connections to SB servers. When the two MSN users become silent for a long time without sending any message, they are automatically logged off from the SB server, but not from their NS servers. Group chats are conducted in the same way as two-user chats in MSN.

### 3 Trace Collection

#### 3.1 Experimental Setup

We have installed a sniffer machine at the Internet gateway of a large enterprise network with more than four thousand employees. The sniffer machine is equipped with an additional network card connected to the network switch of the gateway, which forwards packets it sends to or receives from the Internet to the sniffer network card by port mirroring. We have reverse engineered two popular instant messenger protocols, AIM<sup>1</sup> and MSN Messenger. While some

<sup>1</sup>In August 2006, AOL released its new instant messenger software, AIM Triton, with new protocols (correspondingly, the traditional AIM software and protocol are called AIM Classic). We have not reverse-engineered this new protocol. However, the traffic of AIM Triton only

**Table 1. Overview of IM traffic**

	AIM	MSN	Yahoo	GTalk/Jabber	IRC
Port number	5190	1863	5050	5222	6667
# of servers	1177	600	140	29	79
# of requests	24460	29888	8743	6009	24072
Failed requests	2.8%	1.0%	0.5%	0.7%	96.5%
Outbound bytes	185M	347M	249M	124M	24M
Inbound bytes	442M	597M	121M	207M	66M
% of IM traffic	26.0%	39.1%	15.3%	15.9%	0.04%

open source implementation of IM clients exist, they typically only implement a particular (often out of date) version of the protocol and only a subset of the functionality. Although an IM client can talk to the server using the protocol it chooses, our IM sniffer in the middle of the network must be able to understand various versions of IM protocols and be able to parse their messages correctly. We developed the sniffer software to collect IM packets online without violating user privacy. Written with C and pcap libraries, the sniffer monitors packets that are passed from the OS kernel, identifies IM packets and hashes privacy sensitive information, such as user screen names and chat messages, then dumps the hashed IM packets in pcap format.

IM traffic is identified by port number as shown in Table 1. However, an IM port may also be used as the port of a TCP client. To reduce the overhead of IM packet parsing, the sniffer filters out all TCP packets with a port number less than 1024 before further processing, since in this case the IM port is not used as a server port (all IM port numbers in our study are greater than 1024).

All dumped packets in pcap format include timestamps on when they are captured. Due to the online packet reconstruction, the dumped packets may be out of order. When a dumped file is closed, post-processing is performed to sort the packets by their timestamps.

#### 3.2 Workload Description

We have collected the AIM and MSN Messenger packets with our IM sniffer software for nearly one month. The workload covers a 26-day period from 2006-10-14 20:34 to 2006-11-06 20:43. Our workload includes 469 AIM users and 408 MSN users, with more than 20,000 user conversations and millions of IM messages. We have also collected the IP and TCP headers of three other popular instant messenger traffic: Yahoo Messenger, GTalk/Jabber, and IRC Chat. To our best knowledge, this is the largest scale instant messaging measurement to date.

Table 1 shows an overview of IM traffic in this workload based on the TCP/IP header analysis. To avoid false positive

accounts for about 10% of all AOL instant messenger traffic.

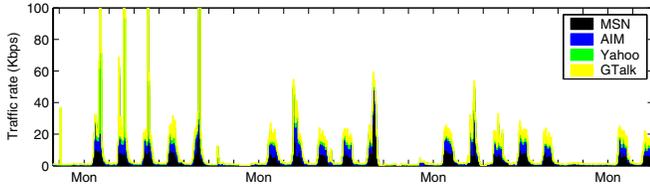


Figure 1. IM traffic rate

identification of IM traffic, servers and clients are identified by IP address and port number in TCP packets with the SYN flags. We can see AIM and MSN account for the the majority of IM traffic. Yahoo Messenger and GTalk/Jabber are next. From this table, we can see the number of IM servers is large. That means, although IM accounts for only a small percentage of Internet traffic, the server capacity requirement of an instant messaging service is large. Indeed, despite having many IM servers, the percentage of failed IM requests are non-trivial for all IM systems. The high percentage of IRC request failure is due to a single unreachable IRC server. We also observe for IM systems such as AIM, MSN, GTalk, and IRC, the inbound traffic (from server to client) is much higher than outbound traffic (from client to server).

Figure 1 shows the hourly IM traffic rate in our collected TCP/IP packet header workload. We can see that in general, instant messaging traffic has a clear diurnal pattern and weekly pattern, which correspond to the user activities in this enterprise network. The overall traffic rate of IM is quite small, about 8.9 Kbps. However, there are a few traffic rate spikes in the workload, which cannot be completely shown in the figure due to the scale of the y-axis. Analyzing the workload in detail, we found these spikes are caused by a small number of TCP connections (about 20–30). In some TCP connections, the traffic rate spike lasts less than one minute, but can be as high as 500 Kbps to 1.5 Mbps, which is likely due to file transfers. In other TCP connections, the traffic rate remains at about 100 Kbps for a long time (more than 20 minutes), which is possibly due to video chat. Because the enterprise network is shielded by a firewall, if the other side of a file transfer or voice/video chat is also shielded by a firewall, the transfer cannot be performed in a peer-to-peer fashion and must be relayed by an IM server. We plotted the hourly traffic rate of AIM, MSN, Yahoo, and GTalk/Jabber separately (figures omitted due to lack of space). We found that the traffic rate of GTalk has a clearer diurnal pattern and weekly pattern without spikes, which is in contrast to that of MSN. This is because the GTalk client software embedded in the GMail Web page supports only text-based chat<sup>2</sup>. The traffic peak time for all

<sup>2</sup>Google has also released the GTalk software which supports video chat and file transfers, but it needs separate installation.

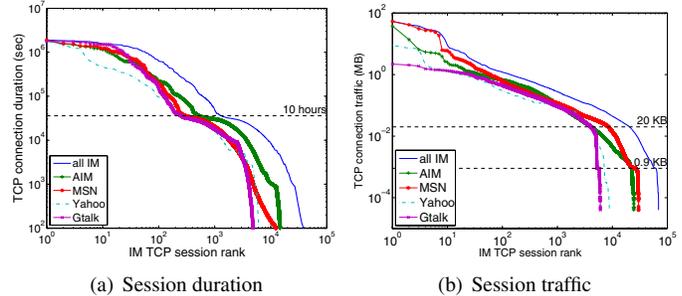


Figure 2. IM TCP session

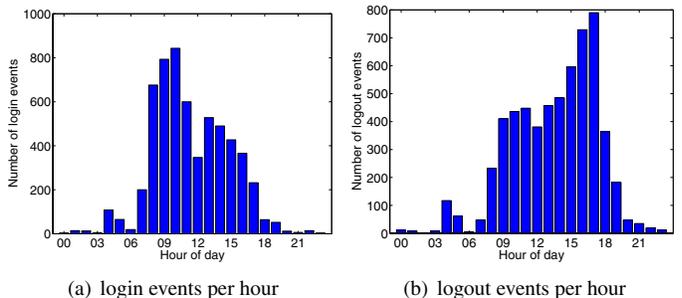


Figure 3. AIM user login/logout time

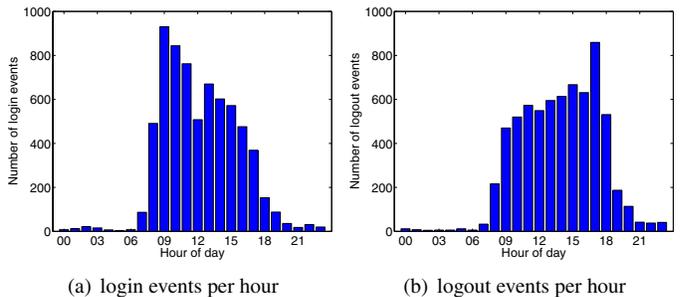


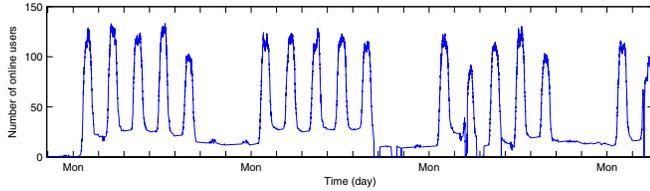
Figure 4. MSN user login/logout time

IM systems is about 2–3 PM.

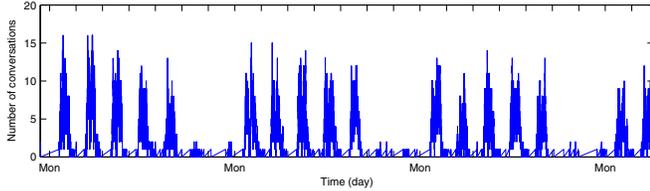
Figure 2 shows the duration and traffic volume of each TCP connection in the IM workload. The distribution of IM TCP duration is heavy-tailed: most TCP durations are short; however, there are some TCP sessions longer than 10 hours, which correspond to IM users who do not turn off the computer and do not sign out for a long time. Similarly, the traffic volume distribution is also heavy-tailed.

#### 4 Online activity of IM users

Figure 3 shows the number of login and logout events in different times of the day for AIM users. Figure 4 shows the results for MSN users. In both figures, the peak time of user login is around 9 AM, while the peak time of user logout is

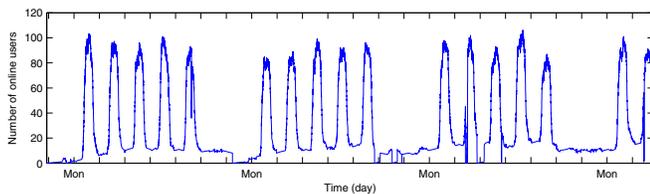


(a) Number of online users

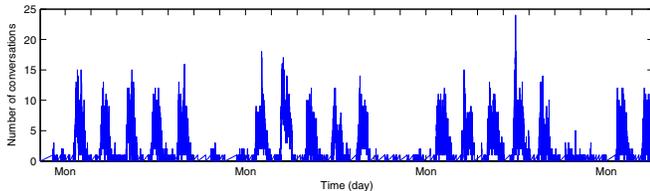


(b) Number of concurrent conversations

**Figure 5. AIM user activity**



(a) Number of online users



(b) Number of concurrent conversations

**Figure 6. MSN user activity**

around 5 PM, reflecting employee work hours.

We have investigated the number of online users and number of user conversations at the same time. The online time is computed based on the duration of BOS server TCP connections in AIM or the duration of NS server TCP connections in MSN. This corresponds to the duration from when a user logs in to when she logs out. In MSN, each conversation is hosted by a SB server in a new TCP connection, and the conversation duration is the duration of this TCP connection. When the user is idle for a long time (about several minutes), the conversation session times out. In AIM, because all chat messages are forwarded by a single BOS connection, we group messages into conversations based on message intervals: if the interval between two messages is greater than a threshold (5 minutes), we assume they are in different conversations. Figure 5(a) and 6(a) show the num-

ber of online users for AIM and MSN, respectively. Figure 5(b) and 6(b) show the number of concurrent conversations for AIM and MSN messenger systems in our network, respectively. The number of online users has clear diurnal and weekly patterns for both AIM and MSN. The peak time of online users is around 2 PM for both AIM and MSN. For MSN, the maximum number of online users is about 90–100, and the average per day is about 66. For AIM, the maximum is about 110–130, and the average per day is about 78. The peak time of IM conversations is also around 2 PM for both AIM and MSN. The number of concurrent conversations is much smaller than that of online users. For MSN, the maximum number of concurrent conversations is about 15, and the average per day is about 5. For AIM, the maximum is about 20–25, and the average per day is about 5. This indicates that most IM users are not active during their online time, and that the most active period of a day is after lunch.

Figure 7 shows the CDF and Weibull fit of online duration for AIM and MSN users. We found that they are very close. This indicates that users of different IM systems have similar online time. Compared with Figure 2(a), we can see there is a clear cut between IM sessions longer than 10 hours and those shorter than 10 hours. To figure out the reason behind this distribution, we plot the login time and logout time for IM sessions with durations between 9–11 hours (figures omitted due to lack of space). The login time is highly concentrated in 8–9 AM, while the logout time is highly concentrated in 6–7 PM. This indicates that the online durations of IM user are due to user behavior rather than system behavior. Figure 7 also shows that the online durations of AIM and MSN users fit Weibull distributions roughly (except for online sessions less than one second, which are likely due to login failure) which has been reported by a P2P study [11].

## 5 Analysis of IM Traffic

### 5.1 Message level analysis

We classify IM messages into five categories:

- Chat messages a user sends or receives.
- Hint messages generated by the IM client software when a user is typing or editing a message.
- Presence messages used to notify the status of buddies in a user’s contact list.
- Icon and binary messages used to upload a user defined picture to the IM server, to download the picture of buddies from the IM server, or to deliver voice/video chat and file transfer packets when the two users cannot communicate directly.

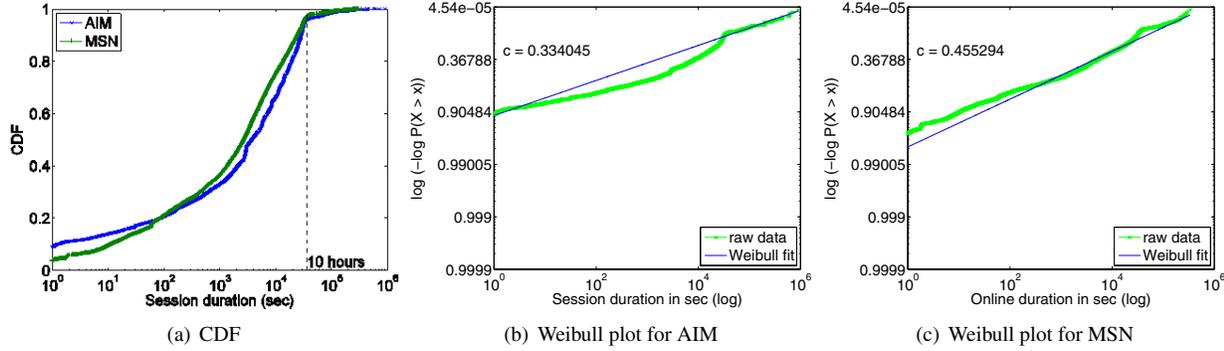


Figure 7. Online duration of IM users

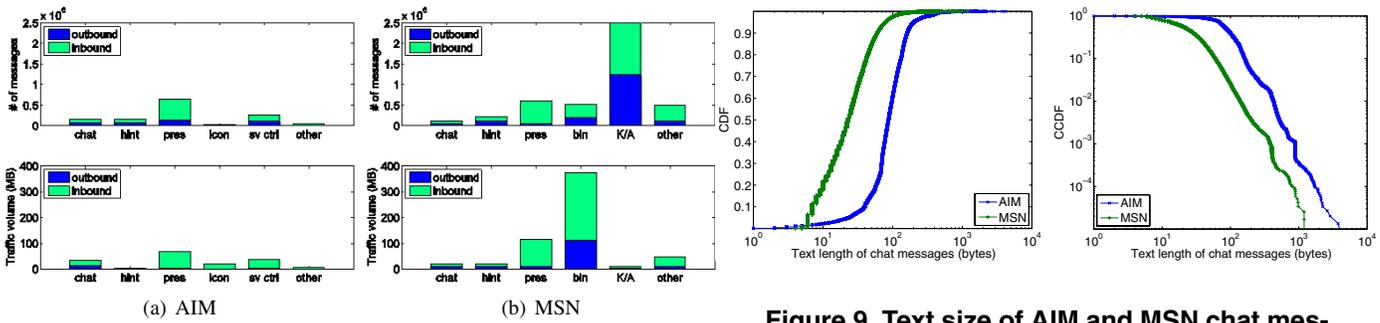


Figure 8. Breakdown of IM message types

Figure 9. Text size of AIM and MSN chat messages (CDF and CCDF)

- Service control messages including those for log in and log out, server redirection, application level keep alive, etc.
- Other management messages

Figure 8 shows the number of messages and traffic volume for AIM and MSN. For both protocols, chat messages account for only a small percentage of IM messages, either by number or by volume. The number of hint messages is greater than the number of chat messages, and the number of presence messages is greater than that of hint messages. For AIM, the volume of hint messages is smaller than that of chat messages probably due to the binary nature of the protocol. In contrast, the protocol for MSN Messenger is text-based, and the message headers account for the majority of a hint message. The volume of binary messages accounts for the majority of MSN Messenger traffic, because it contains file transfer and voice/video chat traffic. However, we cannot distinguish them from the binary messages for user icons because they have the same message type. The number of those messages is not very large, but the packet sizes are much larger than other messages.

We also observe that the chat message size distributions of AIM and MSN Messenger are similar, as shown in Figure 9. Chat messages in AIM tend to be larger because most

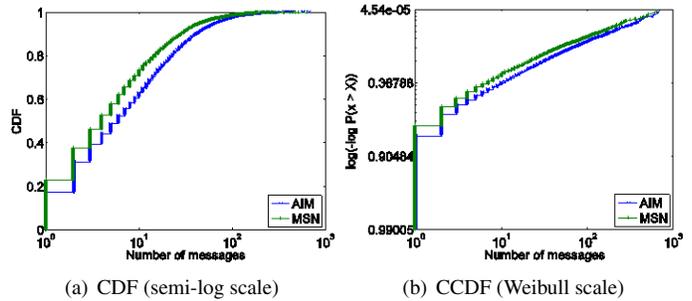


Figure 10. Number of messages in each IM conversation

of them are HTML-based and we did not strip the HTML formatting. In contrast, the format of a chat message (such as font size and font style) in MSN Messenger is encoded in the message header.

## 5.2 Conversation level analysis

Figure 10 shows the number of messages in each IM conversation for AIM and MSN Messenger with different plot scales. We observe that it does not follow a power law dis-

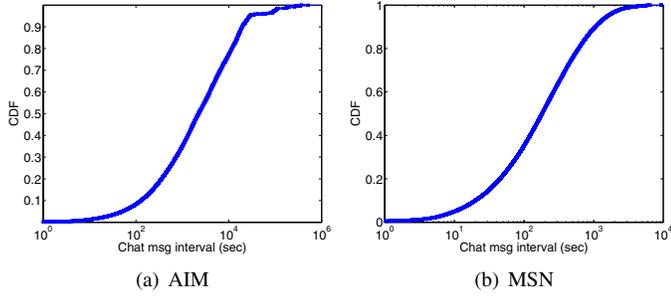


Figure 11. Chat interval distribution

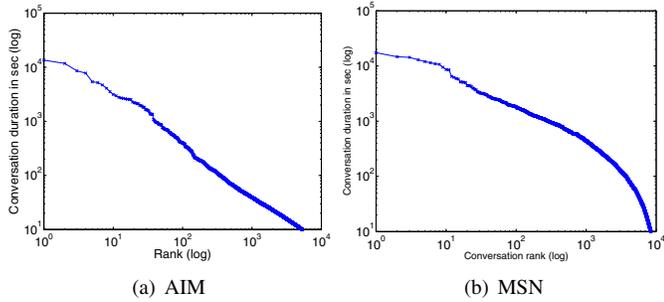


Figure 12. IM conversation duration

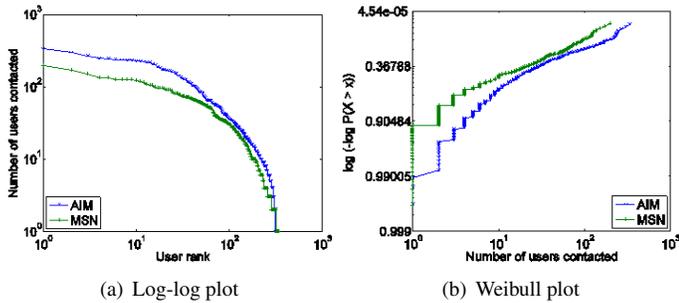


Figure 13. Number of buddies for IM users

tribution. Instead, the CCDF follows a Weibull distribution. Figure 11 shows the CDF distribution of chat message intervals for AIM and MSN.

Figure 12 shows the distribution of IM conversation duration for AIM and MSN users. The conversation duration is the duration from the first IM chat message to the last IM chat message in the conversation (conversations less than 10 seconds are excluded). The distributions of conversation duration for AIM and MSN look differently.

### 5.3 User level analysis

Buddy lists reflect the social network of IM users. Although we cannot rebuild the contact network of IM systems with only a small subset of IM users, the rank distribu-

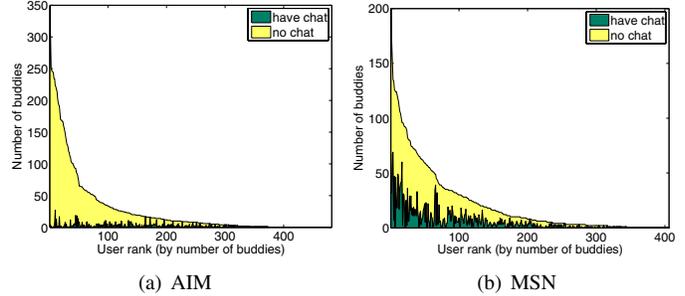


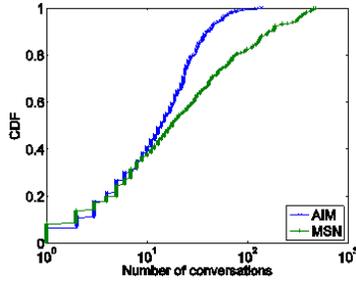
Figure 14. Number of buddies an IM user chats with

tion of the number of buddies for IM users can still give an outline of the IM social network. Figure 13 shows the distribution of the number of buddies for AIM users and MSN users. Surprisingly, it does not follow a power law distribution as reported in [9]. Instead, the number of buddies for MSN users follows a Weibull distribution very well. The distribution for AIM users is a little rough, but deviates from power law model significantly. In general, power law cannot characterize the social network of IM users. Although AIM has an upper bound on the number of buddies for a user, this upper bound (300) is not reached for AIM users in our study.

Since the node degree distribution in an IM network is not power law, the “super nodes” in such a network are not as important as those in peer-to-peer networks. Actually, although worms and viruses in IM networks have been reported for a long time, they have not been a big threat to the Internet, possibly because the propagation of virus on IM networks is slow due to the network topology.

Figure 14 shows the number of buddies a user chats with in our workload. For AIM systems, a user chats with only 1.9 users on average, about 7% buddies in her buddy list. For MSN systems, a user chats with 5.5 users on average, about 25% buddies in her buddy list. In our network, AIM users seem less active than MSN users, possibly due to different user population. In either case, a user only contacts a small percentage of users in her buddy list during the one month workload. Thus, the social network of IM users has a small working set. For example, the 406 MSN users chat with only 2053 out of the 8456 buddies in their buddy lists.

Figure 15 shows the number of conversations for AIM and MSN users. The number of conversations for AIM users is much smaller than that for MSN users: as shown in the CDF figure, 10% AIM users have more than 30 conversations, while 18% MSN users have more than 100 conversations. In both AIM and MSN systems, a small fraction of users contribute most conversations, and consume most



**Figure 15. Number of conversations of IM users**

server resources of the IM system.

## 6 Other related work

In addition to instant messaging systems, work [13] presents a measurement study on the Short Message Service (SMS) in cellular networks. Work [1] presents a measurement study on an Internet chat system, which can be regarded as an early instant messaging system and has many similarities with modern IM systems.

More remotely related are workload characterization studies on peer to peer systems. Gummadi et al. characterized the active session length, download sizes, and evolution of object popularity in KaZaa traffic collected from a university campus [3]. More recently, Klemm et al. proposed a synthetic workload model for the query behaviors in large scale peer to peer systems distributed across three continents [6]. It observed that the distribution of query popularity has a “flattened head” when aggregated over multiple days.

Lindemann et al. proposed a simulation model for disseminating presence information in mobile ad hoc networks [7], which can be employed for workload generation of IM traffic as well.

## 7 Conclusion

Understanding the characterization of instant messaging traffic is essential to its system design and workload generation. This paper analyzes the traffic of the two most popular instant messaging systems from thousands of employees in a large enterprise. In the future, we plan to extend the scope of our study by analyzing traces from other user population.

## Acknowledgments

We are very grateful to Erich Nahum, Charles Wright, William Cornejo, Arup Acharya, and Xiping Wang for

many insightful discussions on this work. We would like to thank the anonymous reviewers for comments on an early draft of the paper.

## References

- [1] C. Dewes, A. Wichmann, and A. Feldmann. An analysis of internet chat systems. In *Proc. of ACM SIGCOMM IMC*, Oct. 2003.
- [2] R. E. Grinter and L. Palen. Instant messaging in teen life. In *Proc. of ACM Computer Supported Cooperative Work (CSCW)*, Nov. 2002.
- [3] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proc. of ACM Symposium on Operating Systems Principles (SOSP)*, Oct. 2003.
- [4] R. B. J. III, E. M. Nahum, D. B. Olshefski, D. Saha, Z.-Y. Shae, and C. Waters. A study of Internet instant messaging and chat protocols. *IEEE Network*, 20(4):16–21, July-August 2006.
- [5] E. Isaacs, A. Walendowski, S. Whittaker, D. J. Schiano, and C. Kamm. The character, functions, and styles of instant messaging in the workplace. In *Proc. of ACM Computer Supported Cooperative Work (CSCW)*, Nov. 2002.
- [6] A. Klemm, C. Lindemann, M. K. Vernon, and O. P. Waldhorst. Characterizing the query behavior in peer-to-peer file sharing system. In *Proc. of ACM SIGCOMM IMC*, Oct. 2004.
- [7] C. Lindemann and O. P. Waldhorst. Effective dissemination of presence information in highly partitioned mobile ad hoc networks. In *Proc. IEEE SECON*, 2006.
- [8] M. Mannan and P. C. van Oorschot. On instant messaging worms, analysis and countermeasures. In *Proc. of ACM WORM*, Nov. 2005.
- [9] C. D. Morse and H. Wang. The structure of an instant messenger network and its vulnerability to malicious codes. In *Proc. of ACM SIGCOMM*, Aug. 2005.
- [10] B. A. Nardi, S. Whittaker, and E. Bradner. Interaction and outercation: Instant messaging in action. In *Proc. of ACM Computer Supported Cooperative Work (CSCW)*, Dec. 2000.
- [11] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In *Proc. of ACM SIGCOMM IMC*, Oct. 2006.
- [12] M. M. Williamson, A. Parry, and A. Byde. Virus throttling for instant messaging. Technical Report HPL-2004-81, HP Laboratories, May 2004.
- [13] P. Zerfos, X. Meng, and S. H. Wong. A study of the short message service of a nationwide cellular network. In *Proc. of ACM SIGCOMM IMC*, Oct. 2006.