A Performance Study of VoIP Applications: MSN vs. Skype

Wen-Hui Chiang, Wei-Cheng Xiao, and Cheng-Fu Chou
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
Email: {whchiang, garry, ccf}@cmlab.csie.ntu.edu.tw

***Abstract*—Due to the growing demand for VoIP (Voice over Internet Protocol) services, researches on VoIP design have attained more and more attention. Compared with a traditional voice service - PSTN (Public Switched Telephony Network), VoIP is able to provide lower cost and more flexibility. However, there are still many challenging issues to guarantee a consistent and good quality of voice connection over the best-effort Internet. In this work, we use a measurement-based approach to do quantitative evaluation of two most popular VoIP applications, i.e., MSN and Skype. In general, Skype performs better than MSN --- it shows that an up to 47% improvement in the overall throughput and an up to 50% improvement in the MOS. Such performance improvement for Skype is due to its (a) rate control mechanism, (b) voice codec, (c) error-resilience mechanism, and (d) better relaying mechanism to go through NAT servers or firewalls. We believe that this study can be of great use in designing a better voice service in current or next-generation heterogeneous networks.**

Keywords-VoIP; Codec; Skype; MSN; P2P

I. INTRODUCTION

VoIP service is a rapidly emerging technology for voice communication. Different from traditional PSTN, it has several advantages including cost saving, flexibility, and better voice quality. These properties have led to growing demand for development of better VoIP services. Furthermore, recent research works [4][5][6][7] on VoIP design have attained more and more attention.

We note that voice quality is affected by not only bandwidth but also other potential pitfalls. The poor voice quality in PSTN might result from the poor connections or old cables. On the other hand, the voice quality in VoIP is mostly dominated by the characteristics of packet networks such as delay, jitter, and packet loss. Therefore, it is important for us to take characteristics of IP network into consideration when we design a VoIP application.

Skype is able to provide good voice quality under unstable or resource-constrained networks and often outperforms other VoIP applications, e.g., MSN. Now, an interesting problem arises: which mechanism makes Skype superior? Both MSN and Skype provide a variety of functions such as voice calls, instant messaging, audio conferencing, and buddy list. However, their underlying techniques and protocols could be quite different. For example, in [1], they point out that the Skype mechanism, which is used to pass through the NAT, can easily adapt to port constraints on firewalls. Moreover, the wideband codec of Skype results in substantial improvement for the voice quality.

In this work, we use a measurement-based approach to evaluate the performance of MSN and Skype. We first create a voice connection of MSN or Skype between two hosts. In the meanwhile, ethereal is used to collect the online traffic for analysis. To simulate real world traffic, dummynet is used to generate the required bandwidth, add the artificial propagation delay, and provide the specific packet loss rate. Through the experiments, we would like to observe how MSN and Skype react to different network conditions, such as varying bandwidth, different packet loss rates, etc. Moreover, by carefully analyzing the collected data, we can explain which underlying technique, the path selection, or the codec mechanism dominates the performance issues. Hence, we are able to figure out which technique has great impact on the voice quality.

The contributions of this work are as follows. We present a comprehensive study which compares the performance and adaptation characteristics of MSN and Skype. In general, the Skype outperforms MSN. Under public networks, the throughput improvement is at least 13% and the MOS improvement can be up to 50%. In addition, when both hosts are behind NAT servers, the throughput difference is up to 47% or more. Such performance improvement for the Skype application is due to its (a) rate control mechanism, (b) voice codec, (c) error-resilience mechanism, and (d) better relay mechanism for traffic passing NAT servers or firewalls. We believe that such evaluation is important and can be of great help in designing a better voice service across Internet or next-generation heterogeneous networks.

The organization of the paper is as follows. Brief reviews of Skype as well as MSN are given in Section 2. Section 3 describes the details of our measurement-based approach and performance evaluation and comparison. We give some related works in Section 4, and then conclude this paper in Section 5.

II. MEASUREMENT-BASED APPOACHES

In this study, we aim to observe how the performance of VoIP applications is affected by different network conditions. To achieve this goal, we design measurement-based experimental approaches. We use the ethereal to collect voice packets from real VoIP voice connections. Besides, the dummynet is used to simulate the real network conditions such as the bandwidth constraint and events of packet loss. So, we can analyze the collected data and understand the important factors that greatly influence the VoIP design.

A. Experimental Environment

In the following experiments, we use Skype of version 1.3 and MSN of version 7.0. Both Skype and MSN run on Microsoft Windows XP and machine with Intel Pentium 4 1.8 GHz CPU and 256 MB DDR DRAM. In addition, Ethereal of version 0.10.12 is used for packet dumping, and FreeBSD 5.4 is used for dummynet settings.

The experiment environment is shown in Fig. 1. We perform the experiments under two different scenarios. In the first scenario, both clients are assigned public IP addresses. In the second scenario, one is assigned a private IP address while the other is assigned a public IP address. In both scenarios, we use dummynet to simulate required network conditions, such as bottleneck capacity, network delay and events of packet loss. With above mechanisms, we can simulate the real network environment and observe how Skype and MSN react to different network conditions. Ethereal is installed as well to monitor the traffic for analyzing. We also record the conversation in each experiment and the recorded data will be graded based on the perception of voice quality. Note that all experiments were performed from August 15th to August 31st, 2005.

B. Measured Experiments

To observe the effect caused by a single factor in the VoIP service, i.e., to isolate the effect from other factors, we design 3 different experiments. In the first experiment, public IP addresses are assigned to both clients. The second experiment is similar to the first one while we add the background traffic into the network to compete with the voice connection. Different from the first two experiments, in the final one, a public IP and a private IP are respectively assigned to these two clients. We do analyses and compare the performance of Skype and MSN in all these experiments.

1) Direct Connection – Both Clients with Public IP:

This is a simple case in the Internet. Clients with public IP can connect with each other directly. In this case, we focus on the effect of codec and basic design issues such as the rate control and the error recovery mechanisms.

This experiment is performed in LAN. In the beginning, we setup a bottleneck link to observe how Skype and MSN react to the bandwidth-constrained network. In other words, the goal is to examine the rate control mechanism of Skype and MSN. Next, packet loss events are manually generated to test the resilience mechanism; that is, to see whether these applications can maintain the acceptable voice quality with the

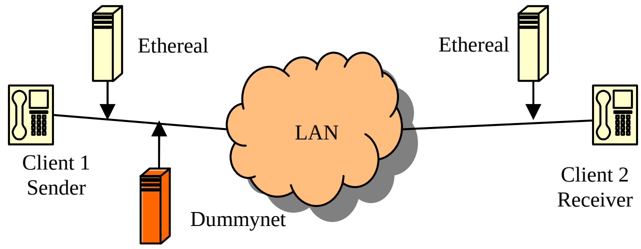


Figure 1. Experimental environment

codec recovery and retransmission mechanisms. To avoid the transient behavior of the environment, we wait for at least 3 minutes between two conjunctive experiments.

2) Direct Connection with Background Traffic – TCP-Friendly or not:

In this experiment, we observe that the voice connections compete resources of bandwidth with FTP sessions. The motivation is to examine whether the Skype or MSN voice connection is TCP-friendly or not. In the beginning, we start a FTP session, which occupies most of the link resources. After a while, a Skype or MSN voice connection is then set up to see how it reacts to the FTP session.

3) Connection through NAT – One with Public IP and the other with Private IP:

Owing to the shortage of IP addresses, the number of private network is increasing. Thus, the problem - how to go through NAT - becomes an important issue now. This testing case is similar to the "Direct Connection" experiment while the difference is that one of the clients uses a private IP. Through this experiment, we would like to investigate the impact caused by NAT on the performance of Skype and MSN.

III. EVALUATIONS

A. Experimental Settings

The experiment environment is illustrated in Fig. 1. We do the experiments under various environment settings with path capacity from 32 Kbit/s to unlimited and with path loss rate ranging from 0% to 40%.

B. Performance Metrics

The performance metrics used in our experiments includes 1) throughput, 2) mean, variance and distribution of packet inter-arrival time, and 3) MOS (Mean Opinion Score) [3].

To some extent, throughput can reflect the bandwidth requirement and the achieved voice quality of a VoIP application. For an application, higher throughput often results in better quality. From the distribution of packet inter-arrival time, we can see if the traffic generated by Skype or MSN is bursty or stable. In addition, we invited 20 students to grade the received voice quality by MOS, in which the score is ranged from 1 (unacceptable) to 5 (excellent). The MOS results directly reflect voice quality, which almost determines user's will to use the application.

C. Direct Connection

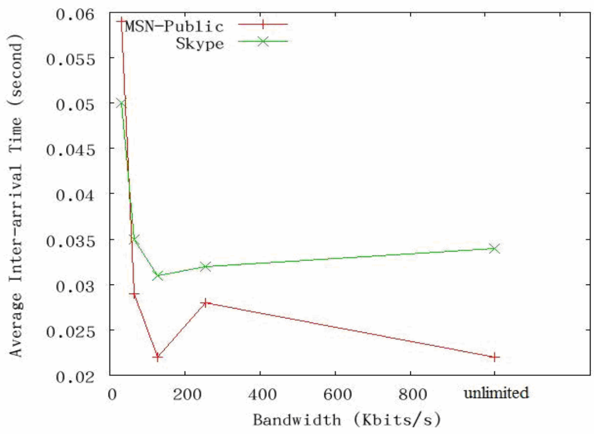
In this experiment, both clients use a public IP to set up a direct voice connection. Specifically, we would like to investigate and compare the performance of (1) the rate control mechanism and (2) the error resilience mechanism in both Skype and MSN applications under different network conditions.

1) Rate Control Scheme – The influence of different bandwidth: The motivation of this experiment is to measure the performance of Skype and MSN under bandwidth-constrained networks. As shown in Fig. 2(a), when the available bandwidth is higher than the requirement of voice data (e.g., 64 Kbits/s or higher), Skype has larger average packet inter-arrival time than MSN does. This is also illustrated in the Fig. 3. For example, as the bottleneck bandwidth is higher than 64Kbit/s, almost 100% of MSN packets have their inter-arrival time less than 50ms while only around 90% of Skype packets do. Of course, as the bottleneck bandwidth is less than the requirement, e.g., 32 Kbit/s, average packet inter-arrival time of both Skype and MSN increases. However, the increasing amount of the inter-arrival time of Skype is smaller than that of MSN. In addition, an interesting and important observation is that no matter what the bottleneck bandwidth is, the variance of packet inter-arrival time in Skype is much smaller than that of MSN as shown in Fig. 2(b) and Fig. 3.

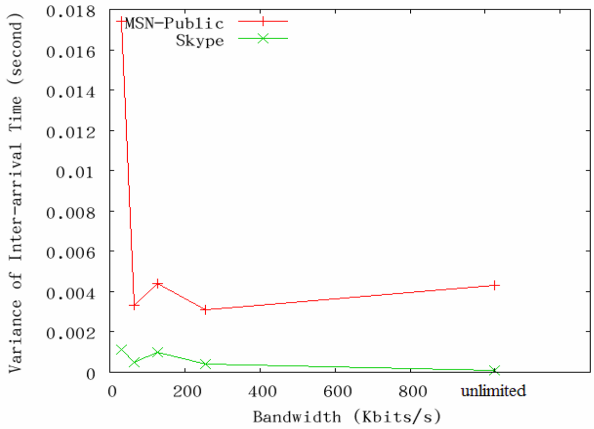
The above observations directly explain why the MOS of Skype is higher than that of the MSN - Skype keeps smooth transmission, which results in the smaller variance of the packet inter-arrival time. Especially, when the available bandwidth is as low as 32 Kbit/s, to maintain a smooth transmission, Skype not only reduces its sending rate but shrinks its packet size from 150 Bytes to 87 Bytes. On the other hand, MSN still keeps its sending rate and packet size. This also explains why the average and variance of packet inter-arrival time in MSN increase as the path capacity drops to 32 Kbit/s. Therefore, the performance improvement of the Skype in Fig. 2(c) is mostly contributed from the better rate control mechanism and voice codec. On the other hand, higher variance in MSN may result from the silence suppression mechanism, which we will investigate later.

2) Error Resilience Mechanism – The Influence of Packet Lost: In this experiment, we want to explore the resilient capability of the VoIP software under a loss-prone network. We use dummynet to introduce packet loss events in the voice connection with loss rate ranging from 0% to 40%. As shown in Fig. 4, no matter what the packet loss rate is, the average packet inter-arrival time of Skype is larger than that of MSN while the variance of the packet inter-arrival time of Skype is much smaller.

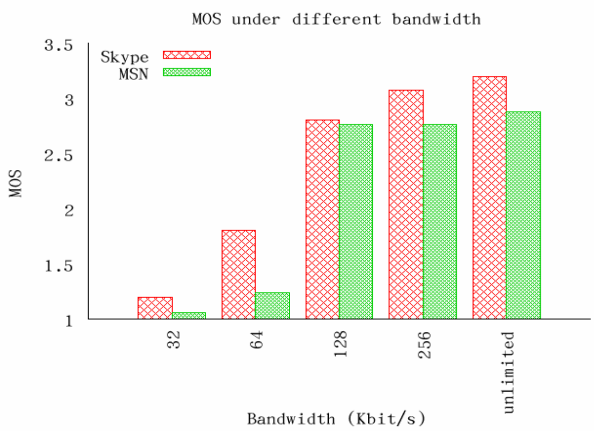
Moreover, as the packet loss rate increases, Fig. 5(a) shows that the throughput of the MSN voice connection decreases as we expect. On the contrary, the throughput of the Skype connection increases a lot as packet loss rate is higher than 10%. This is because Skype will send more packets and increase the packet size from one hundred bytes to two or three hundred bytes when loss events occur. Also, this can illustrate how the error-recovery mechanism works in the



(a) Inter-arrival time



(b) Variance of inter-arrival time



(C) MOS

Figure 2. (a) The average Inter-arrival time (b) The variance of inter-arrival time (c) The MOS information

presence of loss events. On the other hand, based on our collected data, we cannot find any error resilience scheme in the MSN application. This explains why MSN's throughput drops as the packet loss rate increases.

Next, we would like to investigate how good the error-recovery mechanism of Skype is. As shown in the Fig. 5(a)

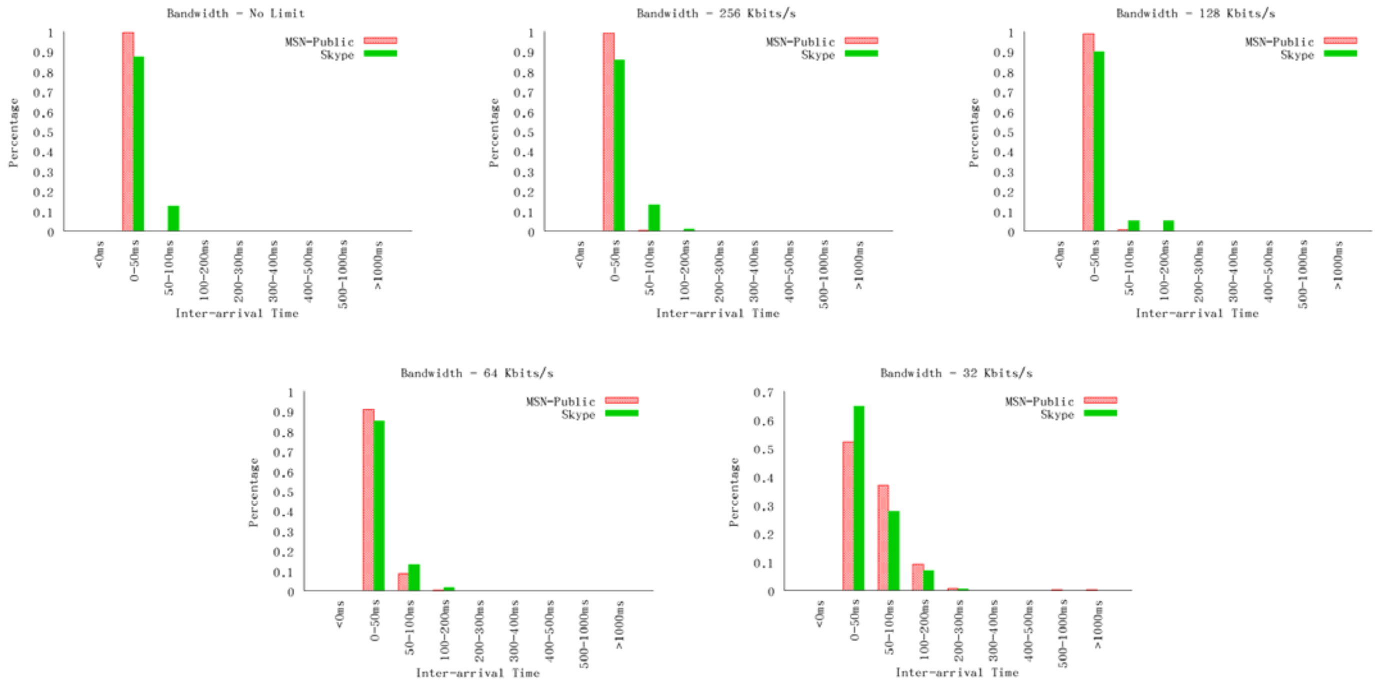


Figure 3. Relation between the inter-arrival time and bottleneck bandwidth

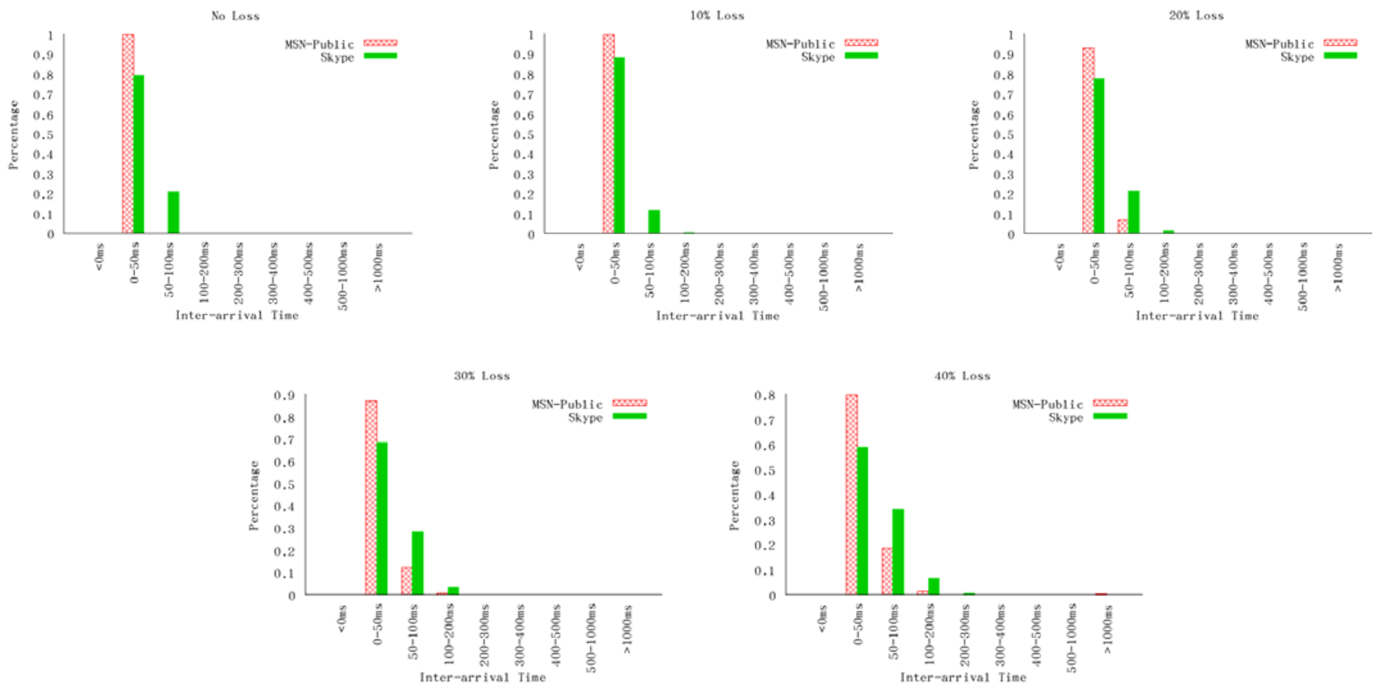
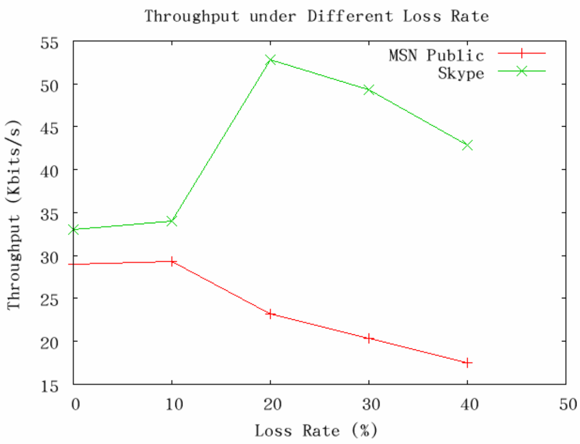


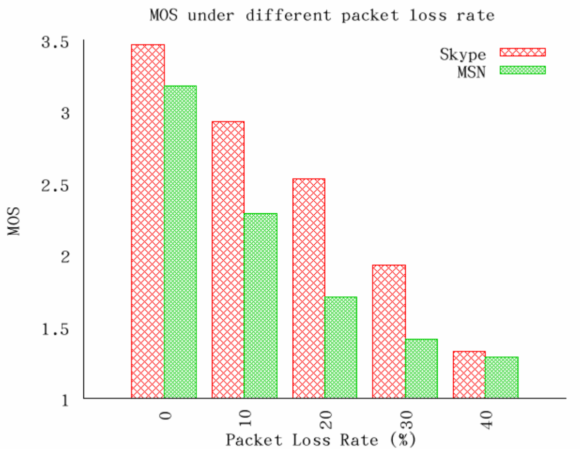
Figure 4. Relation between the inter-arrival time and loss rate

and 5(b), when the packet loss rate is less than 10%, the throughput of Skype increases slightly and the MOS still keeps higher than 3. When the packet loss rate becomes larger than 10%, the throughput of Skype increases a lot while the MOS still decreases. In other words, as the packet loss rate is larger than 10%, the effect of the error-resilience scheme of Skype is not significant and this error-resilience scheme consumes a lot of network resources as well. Therefore, the error-resilience mechanism of Skype works well as the packet loss rate is under 10%. On the other hand, as the packet loss rate is larger than 10%, this error-recovery mechanism might have to be re-designed.

3) Silence Supression: From the collected data, we can find that the silence suppression scheme is supported in the MSN voice service but not in Skype. However, sometimes this scheme might have negative effects on the voice quality.



(a) Throughput under different loss rate



(b) MOS

Figure 5. Experiment results under different packet loss rate: (a) throughput (b) MOS

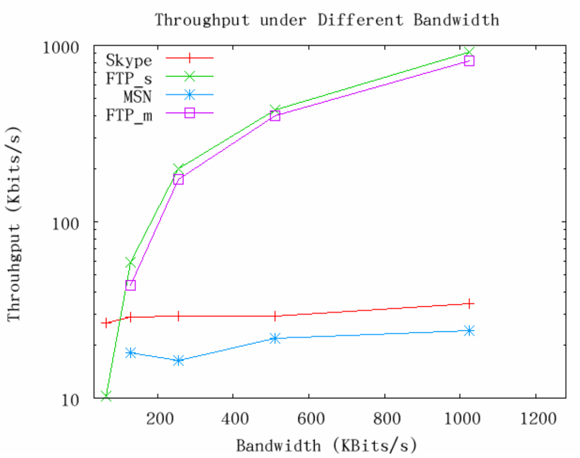


Figure 6. The competition between FTP and voice connection
FTP_s is the FTP traffic which competes with Skype.
FTP_m is the FTP traffic which competes with MSN.

For instance, Skype delivers packets regularly even if the user does not speak. This also explains why the Skype voice service has the low variance of the inter-arrival time of packets.

D. Direct Connection with Background Traffic: TCP-Friendly or not?

This experiment is to examine whether the two VoIP applications have any congestion control mechanism, i.e., we are interested in investigating TCP-friendliness of Skype and MSN. As shown in Fig. 6, as bottleneck bandwidth is larger than 128Kbits/s, the throughput of Skype is around 30Kbit/s and that of MSN is around 20Kbit/s. In this case, we note that the available bandwidth can both meet the demands of FTP and VoIP software. When we limit the bandwidth to 64Kbit/s, the MSN voice connection consumes almost all the bandwidth, which results that we cannot start another FTP connection. Similarly, the Skype connection uses most of the link resources and leaves only 1Kbit/s for the FTP. From the above observations, we can see that both Skype and MSN do not have implemented the congestion control mechanism, i.e., they are not TCP-friendly. The reason why the Skype leave 1Kbit/s for the FTP could be the smooth packet delivery of the Skype connection. Thus, the FTP packets are not dropped at all. On the contrary, the bursty traffic in the MSN connection does not give any chance to the FTP session to transmit a packet. This is why we cannot start another FTP connection simultaneously with the MSN voice connection and do not depict the point for the MSN connection when the bandwidth is 64Kbits/s.

E. Connection Through NAT: The Relay Node Mechanism

According to the research work in [1], we know that the Skype can go through port-restricted NAT directly with the help of relay nodes to start connections. When Skype goes across a UDP-restricted NAT, one of the super nodes will help relay voice packets via a TCP session. In MSN, when the client behind a firewall wants to set up a voice connection, the connection can be indirect, i.e., the voice packets will be relayed through a certain node in the US even if both clients are located in Taiwan. Therefore, the voice quality of the conversation will drop down due to the long route through the US. Table I includes the throughput ratio of Skype to MSN under different network conditions. In every experiment, the throughput of MSN is normalized to 1.0. And then we use this normalization to compute the throughput ratio of Skype to MSN under the same condition. For example, when the voice packets are transmitted through NAT and the link loss rate is 0%, throughput ratio of Skype to MSN is 1.47:1. However, if the voice connection is direct, this ratio is only 1.13:1. From the results in Table I, we can see that Skype always has higher throughput than MSN, and that when the voice packets go through NAT, MSN uses some relay node such that the achieved throughput is even lower than that of direct connection, which can lead to worse voice quality.

TABLE I. Throughput Ratio of Skype to MSN

Type of Connection	Loss Rate (%)				
	0	10	20	30	40
Direct	1.13	1.13	2.25	2.43	2.51
Through NAT	1.47	1.97	3.08	3.67	4.11

To sum up, under a NAT or a firewall network environment, the voice quality of the MSN connection becomes worse since the voice packets are re-routed through a remote host. On the other hand, the Skype connection can set up its voice packets without the help of super node even under the NAT or the port-restricted firewall network environment. Finally, we note that such relay node mechanism, i.e., going through the NAT or the firewall, has a significant impact on the performance of the VoIP applications.

IV. RELATED WORK

In recent years, VoIP service becomes more and more popular, and there have been many applications that allow users to send voice calls or instant messages to their friends on the Internet. Some well-known applications include Skype, MSN Messenger, Yahoo Messenger, Google Talk, etc. In early days, MSN and Yahoo Messenger mainly focus on delivering instant messages instead of voice call. People can save their friends' information in the buddy list and send messages to them when they are online. However, like Skype, MSN Messenger and Yahoo Messenger are also supporting voice calls now. Moreover, Skype also provides service of calling ordinary phone numbers around the world, which is named "SkypeOut", and service of accepting call from a real phone number, which is named "SkypeIn."

There have been some researches [1][2] discussing the key components of Skype. As in [1], they include a) super node: there are some nodes called the "super node" that help relay voice packets and request when end hosts are behind restricted networks, b) host cache: host cache is a list of super nodes cached in the Windows Registry. It helps a Skype client build connections with others, c) codec: Skype uses some wideband codec to encode voices, and d) NAT and firewall: when clients reside behind NAT or firewall, Skype will try to go through the firewall with a variation of the STUN and TURN protocols.

In this paper, however, we focus on the performance and quality that Skype can reach, compare it to another popular application – MSN Messenger under various network conditions, and then discuss about the metrics that affect the performance of VoIP applications.

V. CONCLUSION

We compare the most popular VoIP applications, Skype and MSN, to observe how they react to different network conditions. At the beginning, we observe the original behaviors of Skype and MSN in normal network environments. Then, we test the influence of insufficient bandwidth, packet loss events, and the competition with other traffic. We also try to compare the difference between setting the clients behind NAT and exposing them to the Internet.

- In general, we know that MSN sends smaller packets with higher sending rate than Skype. Owing to the effect of silence suppression, MSN has higher variance in packet's inter-arrival time while Skype sends packets in a stable fashion. The packet size of Skype varies with the network condition while that in MSN is almost fixed.
- In Skype, the bandwidth insufficiency has little influence on the inter-arrival time; that is, the variance of inter-arrival time in Skype is very low. Skype also has better performance than MSN in MOS score.
- Skype increases its throughput to recover missed data when suffering from higher loss rate. On the other hand, the throughput of MSN drops when facing the same situation.
- Skype has better mechanisms to go through NAT. This avoids packets traveling long route through relay nodes and decreases the influence of background traffic. MSN does not address this issue much.

From the differences of Skype and MSN, it is obvious that Skype is trying to improve the overall quality via some mechanisms. We believe that these analytical results are helpful if they can be applied to further design in VoIP applications, and then these VoIP applications can have voice connections as stable as PSTN.

REFERENCES

[1] S. A. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol", Columbia University Technical Report CUCS-039-04, Sept. 2004.

[2] D. Bergstrom, "An Analysis of the Skype VoIP application for use in a corporate environment", Oct. 2004.

[3] ITU-T Recommendation P.800, "Methods for Subjective Determination of Transmission Quality", Aug. 1996.

[4] H Zlatokrilov and H Levy, "Packet Dispersion and the Quality of Voice over IP Applications in IP networks", IEEE INFOCOM, 2004

[5] L. Sun and E. Ifeachor, "New models for perceived voice quality prediction and their applications in playout buffer optimization for VoIP networks," in Proc. ICC, June 2004.

[6] R. Cole and J. Rosenbluth, "Voice over IP performance monitoring," Journal on Computer Commun. Review, vol.31, Apr. 2001.

[7] A. Kansal and A. Karandikar, "Adaptive delay estimation for low jitter audio over Internet," IEEE GLOBECOM, vol.4, pp.2591-2595, 2001.