A Trace-Based Feasibility Study of Infrastructure-Less VoIP

KUN-CHAN LAM AND TSUNG-HSUNG WU
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, 804 Taiwan

Voice over IP (VoIP) is one of the most popular Internet applications. Many VoIP protocols (e.g., SIP) are based on a central-server scheme in which a caller needs to query the central server first for the callee’s information (e.g., IP address) before it can connect to them. However, such a centralized architecture has some drawbacks, such as an unbalanced load and a single point of the failure. As a result, some researchers have proposed the use of peer-to-peer (P2P) techniques for VoIP communication. However, compared to a centralized approach, setting up a VoIP connection over multiple hops could potentially take a longer time and might discourage users from utilizing this service. In this paper, we set out to study the performance of existing P2P protocols for VoIP applications in a realistic setting based on traces collected from the Skype network. The Skype traces are used as an input to model the VoIP network topology. We evaluated the performance of three types of DHT protocols, namely, flat DHT, hierarchical DHT, and proximity DHT, in terms of their lookup latency for the connection setup phase of VoIP applications. In addition, to understand the feasibility of using infrastructure-less VoIP in the real world, we adopted pair comparison techniques to evaluate the quality of the user experience. Finally, we proposed a new hybrid protocol for the infrastructure-less VoIP communication. We concluded that current P2P protocols for VoIP are not satisfactory as compared to the traditional centralized approach and there is still a lot of room for improvement.

Keywords: Skype, peer-to-peer, paired comparison, VoIP, measurement

1. INTRODUCTION

Voice of IP (VoIP) is one of the most popular Internet applications, with many people making VoIP calls for business or personal purposes on a daily basis. VoIP involves sending voice transmissions as data packets using the Internet Protocol. The user’s voice is first converted into a digital signal, compressed and split into a series of packets. The packets are then transported over IP networks andreassembled and decoded at the receiving side.

VoIP communication typically consists of two phases. The first is the signaling phase and the second is the communication phase. The signaling phase is generally for call setup, such as obtaining the callee’s IP address. After a call is set up, the caller then starts communicating with the callee over the IP network.

Traditional VoIP protocols, such as Session Initial Protocol (SIP) [1] and H.323 (ITU recommendation), work in a centralized manner. For example, SIP employs a central server for storing the caller’s and callee’s information. Such a centralized approach is generally considered to have several drawbacks, such as load imbalance, lack of scalability, and the potential to develop a single point of failure.

Received October 3, 2010; revised December 4, 2010; accepted January 30, 2011.
Communicated by Ren-Hung Hwang.
Some decentralized VoIP systems have proposed the use of peer-to-peer techniques for the signaling phase. However, a P2P approach could potentially increase the call setup time, an important factor in people’s decisions whether or not to use a VoIP service, since the query information might have to traverse multiple hops. A long lookup latency could potentially discourage users from using a decentralized VoIP system. In this work we set out to study this problem by surveying existing P2P approaches and compare their performance against the centralized approach using real world traces. We collected Skype traces and used them to simulate a real-world VoIP network topology. We then compared the performance of three existing P2P techniques, namely, flat DHT, hierarchical DHT, and proximity DHT. Furthermore, based on our simulation results, we proposed a new hybrid protocol for infrastructure-less VoIP communication. Finally, we utilized the paired comparison technique [2] to understand the user-perceived quality of the experience with these techniques, as compared to the centralized approach.

The rest of this paper is organized as follows. In section 2, we provide an overview of the related works. We describe the collection and analysis of Skype data in section 3. In section 4, we compare the lookup latency of different P2P techniques based on the collected Skype topology. In section 5, we use the paired comparison technique to evaluate the feasibility of using existing P2P approaches to replace the centralized approach. We offer the conclusions and discuss directions for future work in section 6.

2. RELATED WORK

2.1 Skype

Skype is currently the most popular VoIP software in the world. Skype allows two users to establish audio/video streams with each other and offers a variety of communications services, such as placing a call to another Skype user, voice conference calls, placing a call to traditional telephone lines (SkypeOut), receiving calls from traditional telephone lines (SkypeIn), providing instant messaging for groups of up to 48 participants, cross-platform file transfer, and presence management. Some of Skype’s services are similar to those offered with MSN and Yahoo messenger, but their underlying techniques and protocols are different.

In the Skype overlay network, there are two kinds of nodes, ordinary and super nodes. An ordinary node is a normal Skype client that can make voice calls and send text messages. When an ordinary node has a public IP address and is equipped with rich computing resources, such as powerful CPU, large memory and sufficient network bandwidth, it could be randomly selected to become a super node. A super node will share some of its resources to help maintain the Skype overlay network. Each ordinary node has to connect to at least one super node. Therefore, each node maintains a table of some Skype super nodes called a “host cache”. The host cache file lists a set of IP addresses and port numbers of super nodes. In its newer versions, Skype has encrypted the IP addresses in the host cache.

2.2 Peer-to-Peer Network

Peer-to-peer (P2P) is a popular network technique, and numerous P2P protocols have
been proposed in the literature and implemented in practice [3]. P2P overlay networks are distributed systems without centralized control. In a P2P network, peers form self-organizing networks that are overlaid on top of the IP networks. P2P offers numerous applications, such as robust wide-area routing architecture, efficient search of data items, selection of nearby peers, redundant storage, and network fault tolerance. P2P overlay systems are not like traditional client-server systems. P2P peers have symmetry in roles where a client may also be a server. A peer permits access to its resources by other systems and supports resource sharing, which requires fault-tolerance, self-organization, and massive scalability properties. An important issue in a P2P system is to assign and locate objects among nodes, and this is typically achieved by a P2P lookup service. In the current P2P research, there are two classes of P2P overlay network: structured and unstructured. In a structured P2P network, the topology is tightly controlled and the contents are placed at some specific nodes following certain rules to make subsequent queries more efficient. A structured P2P system typically uses a Distributed Hash Table (DHT). There are numerous existing DHT works, such as Chord [4], Pastry [5], CAN [6], and Tapestry [7]. An unstructured P2P system, on the other hand, is composed of nodes that join the network with some loose rules without any prior knowledge of the topology. Nodes use a flooding mechanism to send queries across the overlay with a limited scope.

Table 1. Comparison of unstructured and structured P2P networks.

<table>
<thead>
<tr>
<th>Finding specific object</th>
<th>Unstructured (e.g. Gnutella)</th>
<th>Structured (e.g. Chord)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flooding (O(n))</td>
<td>Unreliable</td>
<td>Reliable</td>
</tr>
<tr>
<td>Hash function (O(\log n))</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1 shows a comparison between unstructured and structured network approaches. The unstructured approach cannot find a specific object efficiently due to the use of flooding and its performance is \(O(n)\), where \(n\) is the number of nodes in the overlay system. In addition, in an unstructured network, there is no guarantee that a specific object can eventually be found, because there is no rule for placing/locating an object. On the other hand, in a structured network, it is guaranteed that a specific object can be found in a bounded time \(O(\log n)\). Therefore, we conclude that a structured P2P system might be more suitable for VoIP.

2.3 Infrastructure-Less VoIP

Some previous studies have suggested running a VoIP application on top of a P2P network. Kundan Singh [8] proposed a pure P2P architecture for the SIP-based IP telephony systems. SIP is a signaling protocol, and generally used to control multimedia com-
munication sessions like VoIP. Their P2P-SIP architecture supports basic user registration and call setup, as well as advanced services such as offline message delivery, voice/video mails and multi-party conferencing. SOSIMPLE [9] is a fully decentralized, P2P, standards-based approach to communications. By building on top of the existing SIP/SIMPLE infrastructure for VoIP and IM, SOSIMPLE avoids the traditional centralized architectures, eliminates dependency on constant Internet connectivity, and supports ad hoc groups. SOSIMPLE is implemented as a DHT overlay based on Chord using SIP messages. In addition, it replicates location information for the purpose of reliability. The DHT is used only for lookups, and actual communication is passed directly between clients. However, the feasibility of the above approaches when used in a real world scenario where there are a large number of nodes is not yet clear.

2.4 Quality of Experience

For evaluating the perceived quality of experience, Mean Opinion Score (MOS) is the most widely used technique for multimedia, such as audio, video, and voice telephony. MOS has a numerical indication of the quality of received media. The reason why MOS scoring has been widely adopted is because it is simple and intuitive, but it has many problems [10]. In this work, we thus adopt the pair comparison technique [10] to evaluate the user-perceived quality of VoIP for different P2P architectures. The paired comparison method can generate a ratio-scale score and the execution of such a method is easy for participants. The only thing that the participants need to do is to compare two options and decide which one has the better quality. Unlike MOS, the participants will not have difficulty in deciding which rating they should give when using the paired comparison methodology. After finishing the comparison of all pairs, the Bradley-Terry-Luce (BTL) model is then used to generate ratio-scale scores based on the results of pair comparison, as shown in the Eq. (1). Here the $T_i$ stands for the option $i$. The $\pi(T_i)$ is the estimated score of $T_i$ and the $P_{ij}$ is the probability of choosing $T_i$ over $T_j$. The advantages of the paired comparison methodology over the traditional MOS ratings are (1) the rating procedure is simpler thus less burden is on experiment participants; (2) it derives ratio-scale scores; and (3) it enables systematic verification of participants’ inputs.

$$P_{ij} = \frac{\pi(T_i)}{\pi(T_i) + \pi(T_j)}$$

$T_i$: $i$th stimuli
$\pi(T_i)$: the estimated score of $T_i$
$P_{ij}$: probability of selecting $T_i$ over $T_j$

3. COLLECTION AND ANALYSIS OF SKYPE TRACES

In order to understand the performance of different P2P protocols in a more realistic setting, we collected traces from the Skype network and used the collected data to model the topology of a VoIP network. In this section, we first describe how we collected Skype data. We then provide an analysis of Skype topology and user session time. The results of
our work might be useful to people who want to implement a realistic topology generator for their VoIP simulations.

3.1 Data Collection

We collected traces from the Skype network because it is currently the most popular VoIP software and has the largest user population in the world. As mentioned previously, the Skype overlay network has two types of nodes, ordinary and super. All the ordinary nodes need to connect to a super node before they initiate/receive a call. Therefore, every ordinary node has a list of IP addresses of super nodes. Since a Skype super node is also a normal Skype user, by collecting the lists from ordinary nodes, we could obtain a list of active users on the Skype network.

3.1.1 Details of the experiment

In this section, we describe the process of collecting the Skype user topology. When a Skype node is active, it maintains a table (called host cache) of other Skype nodes. The host cache consists of IP addresses and port numbers of a set of super nodes, and it refreshes constantly. The host cache is implemented as an XML file named “shared.xml” on the Linux system. Note that we used an older version of the Skype software for our experiment, since the current version of Skype software encrypts the IP information of the super nodes. Because the host cache is only refreshed when one starts the Skype software, we periodically re-started the Skype program to get the list of currently active nodes. We continuously collected the Skype traces for one week and discovered 164,377 super nodes, which is by far the largest Skype traces as far as we know [11]. We then used MaxMind GeoIP [12] to determine the latitudes and longitudes of the collected nodes. However, we found only 12,372 unique (latitude, longitude) locations due to the limitations of GeoIP, in which the IPs belonging to a physical subnet could have the same latitude and longitude information. Most of the Skype nodes are concentrated in American and Europe, which is consistent with the results of a previous study of Skype network distribution [13]. We also observed that the country that has the most super nodes is America (40,169 super nodes) and the city with most super nodes is Taipei (8,206 nodes), as shown in Figs. 1 and 2.

![Fig. 1. Top 10 countries with the most Skype super nodes.](image1)

![Fig. 2. Top 10 cities with the most Skype super nodes.](image2)
3.2 Node Density

The performance of using P2P protocols for a VoIP network could be strongly affected by the underlying overlay topology. In this section, we perform an analysis of the node density distribution and how long a node remains active based on the collected Skype traces. After collecting the Skype super nodes data, we try to analyze the data and model the Skype network topology. Our results could be useful for designing a realistic topology generator for VoIP network simulations. Here we define the node density as the number of nodes in one fixed area. We look at three different sized areas: 10,000 km$^2$, 100 km$^2$ and 1 km$^2$.

We plot the distribution of node density, as shown in Fig. 3. The y-axis is the probability that the node density is larger than $x$ (i.e. $P[X > x]$). The x-axis is the number of nodes in a fixed area. For example, as shown in Fig. 3, when the size of the area is 1 km$^2$, the probability that the node density is larger than 10 is 0.17. When the area size is 10,000 km$^2$, the probability that the node density is larger than 10 is 0.52.

![Fig. 3. Node density.](image)

We then draw log-log plots of the node density distributions and find they can be approximated to straight lines, as shown in Fig. 4, which indicates that the node density follows a power-law distribution (also often referred as heavy-tail distribution). A log-log plot is a common way to find out if the data follows the power-law distribution. To be more specific, a power-law distribution has the mathematical form $s = c t^\alpha$. If we take the log of both sides of this equation $s = c t^\alpha$, then we would obtain $\log s = \alpha \log t + \log c$. When we consider $\log s = y$, $\log t = x$ and $\log c = k$, we could obtain a linear equation $y = \alpha x + k$. In our case, the Eq. (2) can be mapped to the form $s = c t^\alpha$. Taking 1 km$^2$ data for example, its log-log plot can be approximated to the line $y = -0.8036x - 0.0439$ and the correlation coefficient (i.e. $R$) is 0.986. Here $\alpha = -0.8036$ and $\log c = -0.0439$ (and $c = 0.904$).

$$P[X > x] \sim c x^\alpha$$ (2)
The power-law distributions concisely depict a skewed distribution and it has been shown that wide varieties of phenomena in computer network follow a power law [14-16]. A power-law tends to imply that small occurrences are extremely common, whereas large instances are extremely rare. In our case, our data suggests that most of the Skype super nodes only have a few neighbors (i.e. other Skype super nodes) but a very small set of Skype super nodes are connected to many nodes. This insight could be useful for people who want to model a more realistic VoIP topology in their simulations.

3.3 Session Time

We define the session time as the duration that a Skype user remains online. Some prior work used ‘pings’ to check if a Skype user is online [12]. However, a user might have their machine on but already have exited from the Skype program. Therefore, we checked if a Skype user is online by repeatedly creating a socket connection to the collected (IP, port) pair. If the connection is successful, we assume the Skype user is online. We then closed the socket connection immediately. We repeated this process until the connection fails. The session time is calculated as the duration during which the socket connection is successful. Note that it is tricky to decide how frequent to ping the Skype host in order to check if the Skype user is still online. Pining too frequently might waste system resource and potentially result in denial-of-service attacks. On the other hand, checking only sporadically might mistakenly combine several short sessions into a longer session.

As shown in Fig. 5, about 50% of nodes have a session time of less than 16 hours. Fig. 6 zooms in on the part in Fig. 5 from 0 to 24 hours. The probability of nodes having a session time greater than 1 hour is 90%. We also obtained an average session time of 30 hours. In addition, we found that the distribution of session time can be also approximated as having a power-law distribution, as shown in Fig. 7. The correlation coefficient is 0.958. If we try to fit the session time distribution into Eq. (2), then we obtain $\alpha = 0.5814$, and $c = 0.0869 (\log c)$.  

In this section, we demonstrate that the node density and the duration of user sessions can be modeled as power-law distributions, based on our collected traces. This insight might be useful in generating a realistic network topology in VoIP simulations.
4. COMPARING DIFFERENT DHT TECHNOLOGIES FOR VOIP COMMUNICATION

In this section, we compare the performance of using various DHT technologies for the signaling phase of VoIP communication in a trace-based setting. The performance metric we consider in this work is the lookup latency for a caller to locate its callee IP in a DHT overlay network. We wrote a program using our Skype-trace-based topology to simulate the query routes of different DHT methodologies as described below. In a DHT network, the information is evenly distributed among all the overlay nodes. To locate the information, a lookup query might need to traverse multiple nodes in the DHT overlay. Here we assume the lookup latency is mainly contributed by the propagation delay and queuing delay that a packet might incur when it travels from the source (i.e., the node initiating the query) to the destination (i.e., the node in which the queried information is stored).

We model the queuing delay by adopting a similar approach as in a prior work [17]. We estimate the propagation delay based on our collected Skype traces. Using our traces, we can estimate the geographic distance between any two nodes and their corresponding propagation delay. More specifically, we first collected IPs of Skype nodes and then obtained (latitude, longitude) of collected IPs from Maxmind GeoIP database. The geographic distance of any two nodes is calculated based on their (latitude, longitude) information and their propagation delay is calculated by dividing the distance by the speed of light. Note that the way we calculated the propagation delay is just an approximation since it is...
difficult to infer the exact route a packet actually travels between any two arbitrary nodes due to the limitation of our traces. In a DHT-based VoIP network, to locate the IP address of the callee, the caller first initiates a query through the DHT overlay. The query packet might travel several hops to reach the node where the callee’s IP information is stored. For example, for a query route as shown in Fig. 8, the query packet initiated by the caller (i.e., node A) might have to first travel through node B and C before it finally arrives the node where the callee’s information is stored (i.e., node D). In this example, the lookup latency for node A to locate its callee is $P_1 + Q_1 + P_2 + Q_2 + P_3 + Q_3 + P_4$, where $P_1$, $P_2$, $P_3$, and $P_4$ are the propagation delays and $Q_1$, $Q_2$, and $Q_3$ are the queuing delays. Here we assume the destination (i.e., node D) can directly return the callee’s IP information to the source (i.e., node A) [4].

Intuitively, the lookup latency of a centralized approach will be shorter than the distributed approaches based on the P2P protocols, given that a query does not need to traverse multiple hops when a centralized approach is used. Therefore, we first compute the lookup latency of the centralized approach as a baseline before we compare different DHT technologies for VoIP communication. We picked a Skype server (195.215.8.145) which is located in Denmark [18] and computed the propagation delay of every node (in our collected traces) to this server. The results are shown, as the line indicated as ‘Centralized’, in Fig. 11. We find that half of the nodes have a lookup latency of less than 30ms. Using this as a baseline, we then compare how far the performance of three different DHT technologies, namely, flat DHT (FDHT), hierarchical DHT (HDHT) and proximity DHT (PDHT), is from the centralized approach’s.

4.1 Flat DHT

We select Chord as an example of flat DHT. Chord uses a finger table to locate the next hop toward the destination. As shown in Fig. 9, there are 10 nodes on the DHT overlay ($N_1$, $N_8$, $N_{14}$, ...). The finger table on $N_8$ contains six entries. The first three entries point to $N_{14}$, because $N_{14}$ is the first that succeeds $N(8 + 2^0)$, $N(8 + 2^1)$ and $N(8 + 2^2)$. Assuming that $N_8$ wants to search key 54, it will find in its finger table that the closest node that precedes 54 is $N_{42}$, and then forward the query to $N_{42}$. $N_{42}$ and $N_{51}$ will repeat the same process by looking up their finger tables to find the next successors, until finally the query reaches $N_{56}$, in which information is stored.

We use the SHA-1 hash function to hash the IP addresses into their corresponding IDs. All the nodes are arranged into a ring ordered by their IDs. We randomly choose a caller and a callee from the collected Skype traces, and use the Chord routing protocol to find the callee’s IP address. We repeat the experiment 10,000 times, and the results are shown, as the line indicated as ‘FDHT’, in Fig. 11. Clearly, the lookup latency using FDHT is significantly worse than that of the centralized approach.
4.2 Hierarchical DHT

In hierarchical DHT [19], nodes are organized into groups, and each group has its own intra-group overlay network. The groups are organized as a top-level overlay network. Fig. 10 illustrates a two-tier hierarchical DHT. To locate the node which is responsible for a key, one first finds the group responsible for the key in the top-level overlay. The selected group then uses its intra-group overlay to find the node which is responsible for the key. In the top-level overlay network, some inter-group routing can be used to locate the responsible group. In the lower-level overlay network, some intra-group routing is issued to find the node that has the key.

4.3 Proximity DHT

We next consider the property of node proximity into the design of HDHT. Specifically, we group nodes which have the same IP prefix (e.g., 140.116.x.x) into the same group. Here we assume that nodes which are in the same subnet might be physically close to each other. As shown, as the line indicated as “PDHT”, in Fig. 11, considering the proximity of nodes does improve the performance of HDHT to some extent. On the other hand, the lookup latencies of three DHT technologies are still significantly longer than that obtained when using the centralized approach.

In PDHT, we find that a significant portion of the lookup latency is contributed by the time when a packet travels across different groups in the top-level overlay network. In
practice, the nodes in the top-level overlay are typically selected from nodes that are more persistent (i.e. have a longer session time), so that the topology will remain more stable and result in better network performance.

4.4 Hybrid Approach

Based on the above observation, we propose a new design by modifying the routing mechanism of the top-level nodes. Specifically, similar to some prior work, such as the OneHop approach [20], each node in the top-level overlay maintains a routing table that contains complete information to reach any other nodes in the top-level overlay. Therefore, each lookup in the top-level overlay takes only one hop. The overhead here is mainly the cost of maintaining a routing table that has the information of every node in the top-level overlay. If the network “churn” is higher, it will consume more bandwidth to keep the routing table up to date. However, in the top-level overlay network, the nodes are selected based on their stability. For nodes in the traces we collected, the probability of having a session time greater than 1 hour was 90% and average session time was 30 hours, which suggests that it is feasible to implement an OneHop-like technique for the top-level overlay. The simulation results are shown in Fig. 11, in which the top-level overlay is implemented using OneHop and the lower-level overlay is implemented with Chord, and we thus call our proposed approach a “Hybrid” one. In our simulation, the number of supernodes in the top-level overlay is 410. As shown in Fig. 11, the lookup latency is significantly improved when the Hybrid approach is used as compared to other DHT techniques.

Note that, although the hybrid approach could significantly improve the lookup latency as compared to other DHT technologies. It might potentially introduce more control traffic since it has to maintain the full topology information of the upper overlay. On the other hand, for the hierarchy-based DHTs (like PDHT and HDHT) the routing table size could be significantly smaller than a flat DHT (FDHT). This might become an important issue when deploying a large-scale P2P VoIP network. For example, the size of routing table of FDHT in our case is around \( \log(164337) \) while the routing table size for HDHT and PDHT is only half of that (i.e. around \( \log(400) \)). For hybrid approach, the size of routing table is around 400.

Fig. 11. Lookup latency (Centralized vs. FDHT vs. HDHT vs. PDHT vs. Hybrid).
5. QUALITY OF EXPERIENCE

In the previous section we showed that the lookup latency of existing DHT technologies is significantly worse than that of the centralized approach. In this section, we set out to understand if this observation is consistent with the human perception of different technologies, since it might be possible that human users will not be able to notice differences that are in the order of a couple of hundred milliseconds. We used a paired comparison technique to evaluate the user-perceived quality of call setup time when different technologies were employed.

We assume that the lookup latency is the major delay in the call setup time. We wrote a program simulating the behavior of VoIP software. Specifically, we first made a call and then waited for a response. In our simulation, the response/waiting time is derived from the probability distributions shown in Fig. 11. We used a pair comparison technique to evaluate if approach \( i \) is better than approach \( j \) for the five different approaches we discussed. There were \( C(5, 2) = 10 \) pairs in our comparison. Each pair was compared for 10 times by 15 different people. Fig. 12 shows an example of the resulting comparison matrix for one individual. The \( a_{ij} \) indicates the number of times when approach \( i \) was rated better than approach \( j \). For instance, \( a_{12} = 18 \) and \( a_{21} = 2 \), which indicates 90% (i.e., \( 18/(18 + 2) \)) of the time the centralized approach was considered of having a better response time than the FDHT, and 10% of the time FDHT was considered better than the centralized approach. After collecting all 15 people’s measurements, we then used the BTL model to generate average ratio-scale QoE scores, as shown in Fig. 13.

![Comparison matrix](image)

Fig. 12. Comparison matrix.

![QoE score](image)

Fig. 13. QoE score (Centralized, FHDT, HDHT, PDHT, Hybrid).
The score is normalized from 0 to 1, with the score of the centralized approach set to 1. As shown in Fig. 13, our results suggest there is no difference in the perceived quality between FDHT and HDHT. In addition, the perceived quality of experience of all DHT approaches is significantly worse than that of the centralized approach.

6. CONCLUSIONS AND FUTURE WORK

Some prior studies have proposed using P2P approaches for the signaling phase of VoIP communication. However, a P2P approach may increase the call setup time, an important factor when people decide whether or not they will use a VoIP service, given the query information might have to traverse multiple hops in a VoIP network. In this paper, we set out to study the feasibility of using existing P2P approaches for VoIP communication based on real-world Skype traces. We compared the performance of three existing P2P techniques, namely, flat DHT, hierarchical DHT, and proximity DHT, under a trace-based VoIP network topology. In addition, we proposed a new hybrid protocol for infrastructure-less VoIP communication. We also utilized the paired comparison technique to understand the user-perceived quality of experience for the above techniques, as compared to the centralized approach.

Unfortunately, we found that current DHT-based protocols are not satisfactory for VoIP communication when compared to the traditional centralized approach. On the other hand, this suggests that there is still a lot of room to improve the current P2P protocols for VoIP communication. Furthermore, based on our collected traces, we showed that the node density and the duration of user sessions in a VoIP network can be modeled as power-law distributions. In future work, we plan to use this insight to develop a realistic topology generator for VoIP simulations. Finally, the results we obtained in this paper are mostly based on the use of Chord. However, there are other DHT protocols such as CAN, Pastry, etc. We plan to evaluate the performance of using other DHT protocols for infrastructure-less VoIP communication in future research. Note that due to the limitation of our trace collection, we do not have the complete information of the whole Skype network and the detailed node behavior. Although this might introduce some inaccuracy, we’d like to argue that (1) the number of super nodes we collected is, as far as we know, by far the largest Skype traces as compared to the prior work; (2) the topology analysis based on our traces is consistent with the results of the prior work; (3) we only use the collected traces as an approximation to model a VoIP network to understand the feasibility of implementing existing DHT-based P2P technologies for VoIP communication. Intuitively, a more complete Skype network map (as compared to our partial information) will suggest a larger DHT overlay and possibly even higher lookup latency. Therefore, we believe that, although we cannot obtain the detailed behavior of Skype network due to the limitation of our traces, our work is still useful as a first step toward understanding the feasibility of implementing an infrastructure-less VoIP system using DHT-based technologies.

REFERENCES


**Kun-Chan Lan** (藍崑展) received his Ph.D. degree from Computer Science at University of Southern California in 2004. He received his Master degree in Computer Science from SUNY at Stony Brook in 1997. He received his B.S. in Industrial Management Science from National Cheng Kung University in Taiwan. From 1998 to 2004, Kun-Chan joined ISI’s Intelligent Systems Division as a graduate research assistant. During this period, he has been maintaining and contributing codes to the main simulation software ns-2. From 2004 to 2007, Kun-Chan joined the Network and Pervasive Computing Program at NICTA (National ICT Australia) in Sydney as a researcher and lead the effort of building one of the world’s first outdoor wireless testbeds for traffic light communication in the context of STaR (Smart Transport and Roads) project (STaR in press). He is currently an Assistant Professor at the Department of Computer Science and Information Engineering of National Cheng Kung University.

**Tsung-Hsung Wu** (吳宗勳) received his B.S. degree in Computer Information Science from Aletheia University, Taipei, Taiwan, in 2005. He received his M.S. degree in Medical Informatics from National Cheng Kung University, Tainan, Taiwan, in 2009. He is currently working on his Ph.D. degree at the Department of Computer Science and Information Engineering of National Cheng Kung University. His areas of research include peer-to-peer networking, VoIP, and vehicular ad hoc network.