國立交通大學

資訊科學與工程研究所

碩士論文

SIP 建構於非結構性點對點網路之研究

A Study on SIP over Unstructured P2P Network

研究生：周金璋

指導教授：曹孝櫟　教授

中華民國 九十五 年十月

SIP建構於非結構性點對點網路之研究

A Study on SIP over Unstructured P2P Network

研究生：周金璋　　Student：Jin-Chang Chou

指導教授：曹孝櫟　　Advisor：Shiao-Li Tsao

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

October 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年十月

SIP 建構於非結構性點對點網路之研究

學生：周金璋　　　指導教授：曹孝櫟

國立交通大學資訊工程學系（研究所）碩士班

摘要

SIP（Session Initiation Protocol）已被廣泛使用在建立基於 IP 的多媒體通訊連線。然而客戶-伺服器架構的 SIP 卻使得使用者遭受網路瓶頸和單點錯誤的風險。點對點網路架構的 SIP 隨即被提出，由於點對點的網路取代中央伺服器的設計，故具有延展性及可靠性的優點。其中，基於 DHT 架構的 Chord 方法廣受歡迎而被採納使用於建立邏輯層網路並支援使用者註冊及電話連接。基於點對點的網路設施，使用者可以用 Chord 提供的搜尋機制來連接 SIP 電話連線。

然而，先前的研究並未考慮到行動性的問題，這是在行動網路上不可或缺的重要性質。在具有行動使用者的 DHT 架構點對點網路中，它會使系統產生額外的電話連線延遲，並且還需要耗費嚴重的維繫費用來確保 Chord 網路在行動環境下的正確性。

在這篇論文中，我們建議使用非結構性的點對點網路來作為 SIP 的平台，另外每個節點使用與其具有關係的節點作為鄰近節點，因此節點可以更容易地找到朋友在點對點網路上的所在位置，並可快速開啟電話連線。最後我們比較我們提出的解決方案與建構於 Chord 的 SIP 方案，証明我們的方法耗費更少的維繫費用而且比先前的方法有更高的系統效能。

A Study on SIP over Unstructured P2P Network

Student: Jin-Chang Chou　　　Advisor: Shiao-Li Tsao

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

Abstract

SIP (Session Initiation Protocol) has been widely used for call setup of IP-based multimedia communications. The client-server architecture of SIP, however, exposes users in the risk of bottleneck and single point failure. P2P (Peer to Peer) SIP is consequently proposed with the advantages of scalability and reliability by replacing centralized servers with a self-organizing P2P network. In which, DHT-based Chord approach is attractively used to build the logical overlay and supports registrations and call setup. Based on the P2P infrastructure, users can initiate an SIP call by invoking a process of user look-up in a Chord ring.

However, previous studies did not consider the mobility issue which is an essential property in mobile Internet, and extra call setup latency is introduced while there are some roaming nodes in a DHT-based P2P network. Also, a DHT-based P2P network needs to spend heavy maintenance efforts to ensure the correctness of a Chord ring in a mobile environment. In this thesis, we suggest an unstructured P2P network for P2P SIP in which nodes establish neighbor links based on their relationships. Therefore, nodes can easily locate their friends in a P2P network and quickly initiate an SIP call. In the end, we compare the proposed solution with the P2P SIP over Chord, and demonstrate that our method requires less maintenance effort while better system performance can be achieved than the previous approach.

致謝

首先我要感謝我的指導老師，曹孝櫟老師。在這研究所兩年，我受到他的指導與教誨，不僅學得做研究的方法及專業的學識，更在他身上學到做人處事要誠懇、要積極的態度。曹老師的栽培，養成我獨立思考、動手解決問題的能力；教學上深入淺出、幽默風趣的風格，使我獲益良多；更難得的是身先士卒的求學態度，更是帶給我們前進的動力。我很感謝能夠在我的求學過程中遇到這樣的明師指導，對我未來的人生規劃及前景提供很正面的幫助，謝謝您。

接下來我要感謝實驗室的同仁，博士班一正學長、建明學長、海倫學姊不吝給予我很多研究上的指導與建議；君豪、凱翔、邦翔、宥霖、雅聯及誌謙、中暉等學弟，不僅在修課上甚至生活中皆帶給我很大的支持及鼓勵，使我在學習的過程中，仍保有熱誠與活力，很高興能認識你們。

最後要感謝我的家人對我精神上的鼓勵，讓我能夠專心的在兩年內完成研究工作。謹以此論文，獻給我最摯愛的家人。

目錄

圖目錄

表目錄

I. Introduction

Speeding and spreading of networks are followed with numerous protocols and applications made for multimedia. Especially, SIP [1] of IETF (Internet Engineering Task Force) has been popularly and widely used for a call setup protocol of VoIP services. The client-server based architecture, however, exposes users in the risk of bottleneck and single failure.

On the other hand, P2P technology is growing mature and impressive with the advantages of high scalability, robustness, and fault tolerance. A large number of applications based on the P2P infrastructure are produced and well used such as not only file sharing (e.g. Gnutella [2] and Chord [3]) but also telephony services (e.g. Skype [4]).

Therefore, some researches propose a novel P2P system for Internet telephony using SIP, called P2P SIP. These approaches expect to integrate the pervasively exited SIP-based telephony applications with a resiliently decentralized P2P system in replace of centralized servers. So that each SIP agent in P2P SIP acts both the roles of requesting client and relay server, and it accomplishes a SIP call relying on the cooperation of others rather than asking the formerly centralized servers.

Current P2P SIP is mostly built over Chord [5, 6] which is one particular DHT algorithm famous for its simplicity and convergence properties within P2P community. In these approaches, SIP agents, regarded as nodes, are numbered with node-IDs and structured as a ring-based topology. User accounts of each SIP agent are also numbered with resource-IDs and user registrations including the mapping of the user account and location are published to those nodes whose node-IDs are closest to, but not smaller than, the resource-IDs. Besides, each node in a N-node Chord ring maintains a finger table which records at most $\log(N)$ pointers linked to others exponentially distributed on the overlay. While call initiation, by relaying the request message to the closest hops recorded in finger tables can be proved converging the call setup latency toward O(log *N*) time.

However, previous studies did not consider the mobility which is an essential property in mobile Internet. While users are moving from one subnet to another subnet, they depart the overlay and then presently rejoin with a completely new location in the Chord ring. It results in that finger tables which previously recorded the information of these mobile users are becoming stale and serious maintenance overhead is introduced to ensure the correctness of Chord ring. Furthermore, additional call setup latency is required while some of the users in a Chord ring move. Most users cannot allow such long latency just using for initiating a call.

In this thesis, we propose the approach, UP2P SIP, which is a simple but useful unstructured P2P system for Internet telephony using SIP. In which, nodes establish P2P neighbor links with their online buddies rather than random neighbor selection in Gnutella or DHT-based distribution in Chord. We construct a relation-based network and nodes inside only mastered those connections they are interested such that nodes can easily locate their friends in a P2P network and quickly initiate a phone call. Besides, only relationship links cannot satisfy users which have requirements to call a non-buddy sometimes. We adopt the mechanism in [7] to additionally construct a low-diameter P2P network which guarantees the longest shortest path in the network would be limited at O(log *N*), *N* is the size of the network, so that users can establish a session to somebody through a possibly shortest path in the network, which implies possibly short latency.

The rest of the paper is organized as follows. In Section II, we introduce the background technologies of P2P system, IETF's SIP, and the related work about P2P SIP over Chord. Section III presents the design goal and architecture of UP2P SIP. Section IV discusses the performance evaluation and simulation results, while the last section concludes the works.

II. Background and Related Work

A. Peer-To-Peer Technology

P2P system depends on the cooperation of nodes acting the roles of both server and client in replace of centralized servers. We introduce unstructured P2P system and structured P2P system using a typical case, respectively Gnutella and Chord, below.

1. Unstructured P2P System: Gnutella

Gnutella is the representatively famous file sharing application in unstructured P2P system that is known for its simplicity, flexibility, and robustness. Each node in Gnutella network maintains several randomly selected neighbors which are the pointers linked to others and relied on to accomplish search procedures.

In Gnutella, flooding is applied on the search mechanism. Simply say, nodes fan out messages with different requests to surrounding neighbors whose locations are stored in neighbor table. Any node who receives messages checks if the request could be satisfied and then deals with it. If satisfied, the receiving node sends a response back toward the message originator along the previous path. Else the receiving node fans out the messages to his neighbors and the same process goes on. Such a process above is called flooding.

Messages applied contain several requests with different purposes and showed in table 1 [8]. Besides, usually not to stuff the network with useless messages, Gnutella limits the flooding scope to a time-to-live (TTL) which specifies the fixed length of radius messages can be fanned out (it is typically 7 in Gnutella).

Type	Description	Contained information
Ping	Announce availability and probe for other servents	None
Pong	Response to a Ping	IP address and port number of the responding servent; number and total kB of files shared
Query	Search request	Minimum network bandwidth of responding servent; search criteria
QueryHit	Returned by servents that have the requested file	IP address, port number, and network bandwidth of responding servent; number of results and result set
Push	File download requests for servents behind firewalls	Servent identifier; index of requested file; IP address and port to send file to

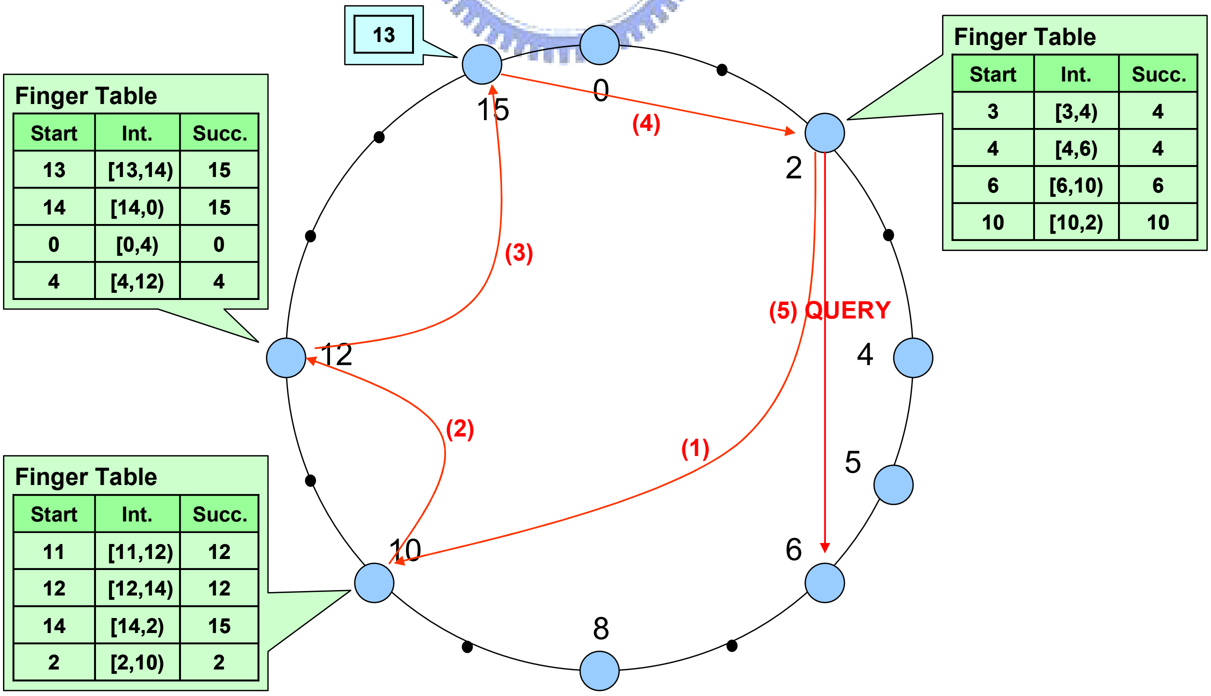
Table 1: Messages types in Gnutella protocol [8]

2. *Structured P2P System: Chord*

Chord is typically a DHT-based structured P2P system known for scalability and convergence property. Nodes in Chord are assigned with a node-ID and structured in a ring-based topology. Any file liked to be published is represented with an “index” assigned with a resource-ID (use the same naming space with node-ID). An “index” is similar to a pointer which specifies where the node owning the file is and each “index” is distributed in an “index node” whose node-ID is closest to, but not smaller than, the resource-ID of index on the overlay. It is necessary to look up the representative “index” in advance for a node who wants to request the file.

To locate a file, each node in base-n Chord ring keeps a finger table with pointers linked to at most n other nodes whose node-IDs are exponentially increasing away from the node. Take a node with node-ID, x, for example, the node-ID of the ith finger in finger table is closest to, but not smaller than, $x+2^\wedge(i-1)$. When invoking a file search, if a node wants to lookup a file with resource-ID, y, it sends a query message to the finger whose node-ID is closest to, but smaller than, *y* (if the finger is not existed, send the message to successor). Any node who receives a query message checks if the requested index is stored in itself. If matching, the node sends query response with the information of index back toward the originator. Else the node similarly refers to his finger table and continues forwarding the query message to next hop. Figure 1 shows a simple example about lookup process.

Nodes in Chord have to achieve the stabilization to improve the decreasing efficiency caused by stale finger tables. Because P2 P network is usually dynamic due to node joining, departure, or failure and that will cause the staleness of finger tables containing pointers linked to these dynamic nodes. And it will cost additional latency for a node to select a suitable finger from a stale finger table when routing messages. For this reason, nodes in Chord follow the rules below to stabilize the correctness of the finger table. First, nodes must inform their successors if they leave gracefully. Second, nodes update and check their fingers periodically if they are alive to avoid unexpected departure.



Finger Table

Start	Int.	Succ.
3	[3,4)	4
4	[4,6)	4
6	[6,10)	6
10	[10,2)	10

Finger Table

Start	Int.	Succ.
13	[13,14)	15
14	[14,0)	15
0	[0,4)	0
4	[4,12)	4

Finger Table

Start	Int.	Succ.
11	[11,12)	12
12	[12,14)	12
14	[14,2)	15
2	[2,10)	2

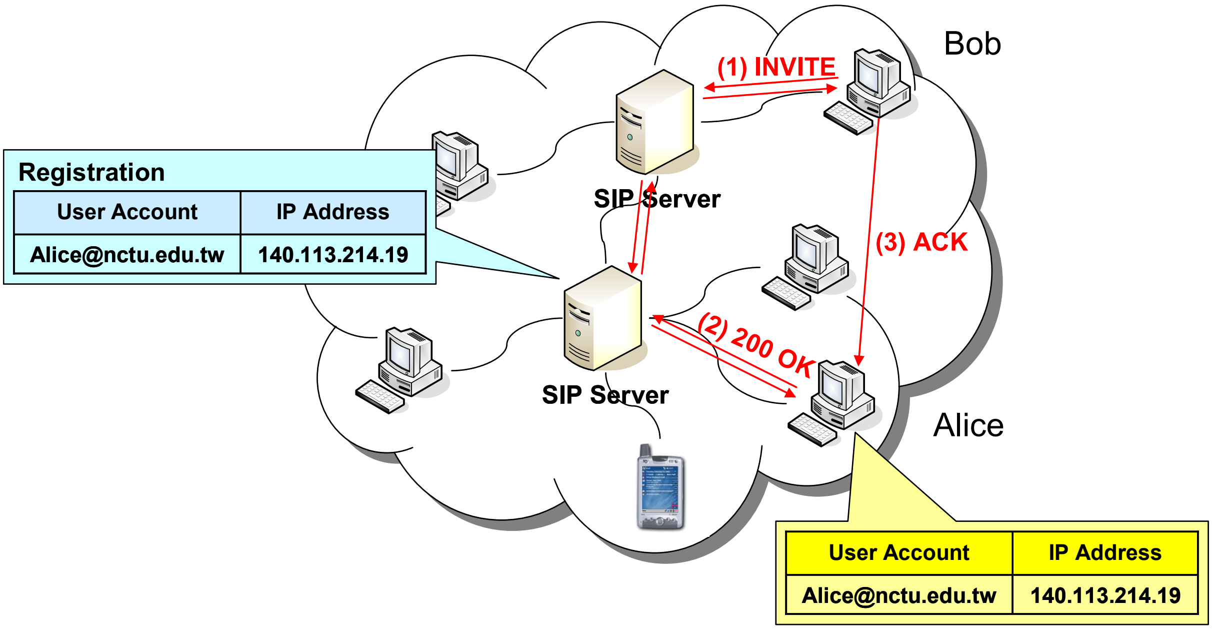
Figure 1: The process of file searching in Chord algorithm

B. Session Initiation Protocol

Session Initiation Protocol (SIP) was developed by the SIP working group within IETF. It is a signaling protocol for initiating, managing, and terminating multimedia sessions across the network. There are four types of entities participating in SIP: user agents, registrars, proxy servers, and redirect servers.

- **User agents** are usually the end devices of users for initiating and terminating the sessions by exchanging request and response messages.
- **Registrar** is a server that keeps the nodes' contact information within their assigned network domain.
- **Proxy servers** are the intermediate entities responsible to forward the requests and responses to user agents or next proxy server.
- **Redirect servers** receive requests and response location information to caller rather than forwarding the message.

It is common to see that proxy server, registrar, and redirect server are implemented together in the same applications. Figure 2 shows the process of call setup in SIP in which Bob sends an INVITE message to Alice and SIP servers in the middle of the path help to relay the message.



Registration

User Account	IP Address
Alice@nctu.edu.tw	140.113.214.19

User Account	IP Address
Alice@nctu.edu.tw	140.113.214.19

Figure 2: The process of call setup in conventional client-server SIP

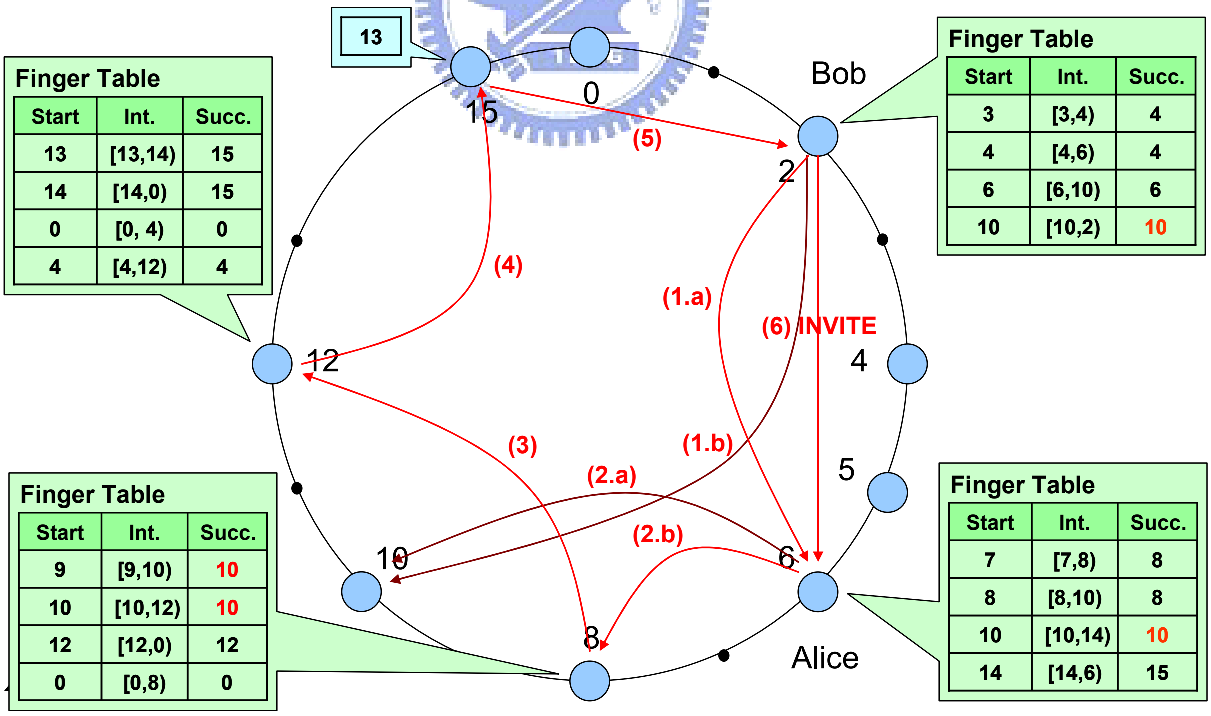
C. P2P SIP over Chord

Some researches propose the system, P2P SIP over Chord [5, 6], using in telephony with the purposes not only to raise the scalability and reliability but also to be integrated with the exited components in SIP.

P2P SIP over Chord follows the lookup mechanism using finger tables in Chord approach. And similarly, a node applying to join the system firstly invokes the node registration. That is, the joining node contacts a bootstrap node to help it to methodically lookup the admitting node which is responsible to allow the joining node to obtain the information about his neighbors and resources had to be maintained. Secondly, the node proceeds to invoke resource registration. In which, the node registers the owning user to the node whose node-ID is closest to, but not smaller than, the resource-ID belonged to the user on the overlay. Periodically nodes explore the fitted fingers in finger table and update the states of them to maintain the quality of searching performance.

To be mentioned is that, after right joining procedures, each user starts a lookup for his buddies specified in buddy list one by one and caches the position if the buddy is online. Such that when a user would like to make a phone call with an online buddy, he could directly and rapidly connect the session to the node which the buddy is located on. Besides, each time when a user would like to contact a stranger (it means not being specified in buddy list), the user has to follow the lookup algorithm of Chord hop by hop to locate the registration of the stranger and then establish the call setup.

However, in mobile environment, users moving form one subnet to another may churn the DHT-based overlay frequently. Such a behavior brings a great influence not only staling finger tables in the Chord ring but also being followed with maintenance overhead and long call setup latency. In VoIP services, users usually cannot allow long latency waiting for connecting the session. In Figure 3, Bob want to initiate a session with Alice, however, it cost a penalty of additional two hops latency to establish the session due to one node moved out the network without notifying in the middle of routing path.



Finger Table

Start	Int.	Succ.
13	[13,14)	15
14	[14,0)	15
0	[0, 4)	0
4	[4,12)	4

Finger Table

Start	Int.	Succ.
3	[3,4)	4
4	[4,6)	4
6	[6,10)	6
10	[10,2)	10

Finger Table

Start	Int.	Succ.
9	[9,10)	10
10	[10,12)	10
12	[12,0)	12
0	[0,8)	0

Finger Table

Start	Int.	Succ.
7	[7,8)	8
8	[8,10)	8
10	[10,14)	10
14	[14,6)	15

Figure 3: The process of call setup in P2P SIP over Chord in mobile environment

III. Architecture

A. Node Search

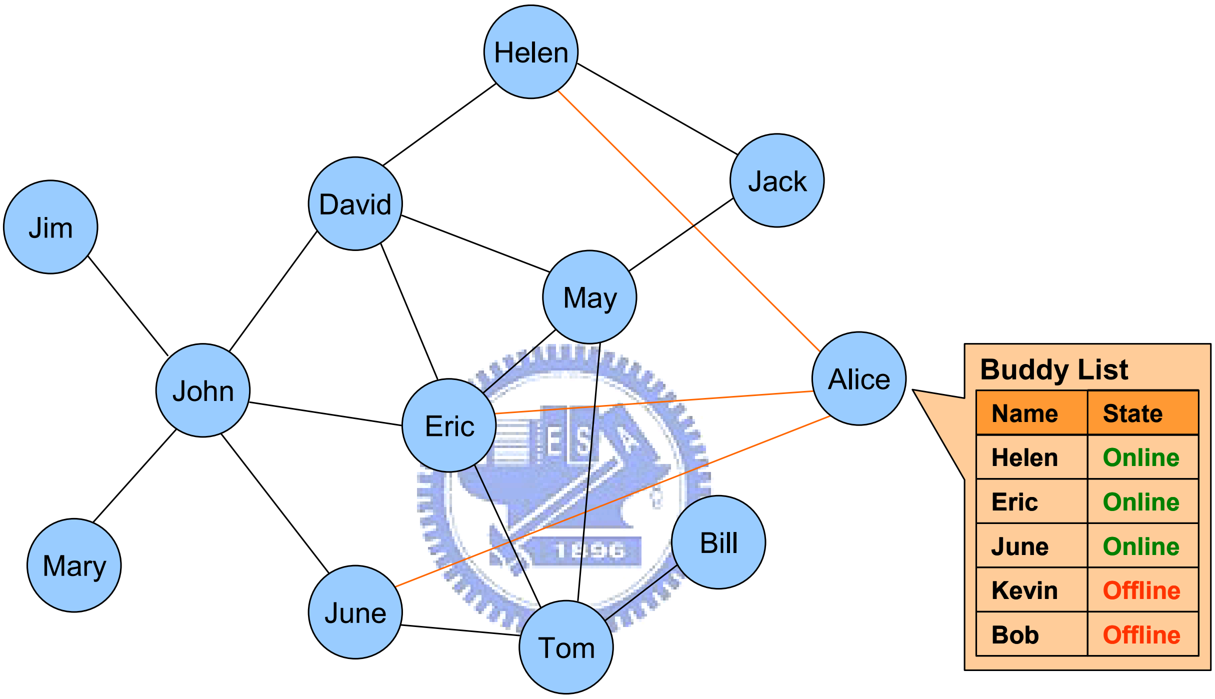
UP2P SIP adopts flooding as the search mechanism like Gnutella. Each node who wants to search somebody in UP2P SIP invokes a search by sending query message to nodes which are stored separately in the two tables, buddy list and neighbor table. Any node receiving a flooded message checks if it matches the request and has it necessary to respond. Then it goes on forwarding this query message to the nodes specified in his buddy list and neighbor table. After flooding to the end of network or to the TTL limit, it means a successful search if the request originator receives responses from others before timeout. Due to the flooding mechanism it can effectively disperse the risk of mobility effect.

B. Node Joining

UP2P SIP approach separately constructs a social network and a low-diameter network which help to decrease maintenance overhead and call setup latency. We describe how to build these two networks in following sections.

1. Relationship Links Discovery

A node who wants to join the system firstly contacts a bootstrap node, which may be a well-known node or a cached node kept last participation. The joining node then sends the bootstrap node a query message including his buddy list to ask for searching his online buddies. Any node receiving the query checks if the held user matches one of the specified buddies. If matching, the receiving node, i.e. one of the joining node's buddies, sends a response to the request originator and goes on fanning out the message. At the same time, he sets the displaying field representing user states in buddy list as "online" and so is query originator when he receives the response. If not matching, the receiving node follows the search mechanism going on forwarding the request out. At the end of flooding, it is expected to see that the joining node has directly connected to those nodes with friendship relation and also these buddies are aware of the node’s joining. Figure 4 shows the connectivity after relationship links discovery.



Buddy List

Name	State
Helen	Online
Eric	Online
June	Online
Kevin	Offline
Bob	Offline

Figure 4: The connectivity of network. In which, Alice connects with buddies who are online.

2. *Build Low-Diameter P2P Network*

In addition, only relationship links cannot satisfy the requirement of efficiency for calling to a non-buddy, furthermore, there may be nodes with no relationship links. We observe that the problem how to make the length of routing path minimal in the network can be seen as the traditional graph problem that how to construct a network with a low diameter. The problem was discussed for a long time in graph theory and many outstanding solutions are proposed. So we adopt the mechanism proposed in [7] which constructs a low-diameter unstructured P2P network and satisfies our requirements. It guarantees that with great probability the topology can remain connected with a continuous network dynamics and the diameter of the network would be limited at O(log *N*). The main idea is to let each node maintain enough number of links to reach a connected topology and with a low diameter.

C. Node Leaving and Failure Recovery

If a user is going to depart the network, he must invoke a notification to all his online buddies in buddies list and all his neighbors in neighbor table. Nodes receiving a NOTIFY message with leaving information then set the displaying field “offline” in buddy list. With such a strategy for a node can also easily detect a failure node by periodically SUBSCRIBE message.

D. Periodical Stabilization

Nodes in UP2P SIP periodically update the states of buddy list and neighbor table. We follow the former SIP approach [8] using messages, SUBSCRIBE and NOTIFY, to reach the goal continuously tracking the state of nodes stored. So that users can be always aware of the presence of their buddies and the latest distributions of their neighbors.

E. Call Setup

Caching the contacts of buddies provides large assistance for call setup because of the fact that people mostly call their friends more often than strangers. In UP2P SIP, if a user wants to establish a session with an online buddy, he can directly connect to the node in which the buddy belongs. It cost only one hop latency for the user to initialize the phone call rather than wasting additional overhead to explore the dynamic topology.

Besides, users have requirements to call out to non-buddies sometimes. It is achieved by neighbor tables which construct a low-diameter unstructured P2P network. That is, users initialize a phone call to a non-buddy by following the search mechanism that flooding to those in neighbor table and buddy list. Such a connection process can be proved in [7] that the routing paths are limited in O(log *N*) hops (*N* is the total numbers of nodes in the network). In other words, all above implies that a user can be possessed of a low-latency connection speed when calling to an uncached non-buddy in the low-diameter unstructured P2P network.

IV. Performance Evaluation and Simulation Results

To be an application of telephony, it is firstly important to take account of the call setup latency. That is, users usually can tolerate a long sustained ring for seconds but not long latency for waiting the session to be connected. So our simulation is emphasized on the comparison of call setup latency. In the same time, to speed up the process of call setup comes with lots of additional overhead. It pays to maintain the correctness of routing tables avoiding frequently staled when nodes churning. These tables are very important factors deciding the length of routing path when call setup, thus we measure the overheads produced by node churning and stabilizing and work out a completely comparison.

A. Methodology

We implement two simulators to separately simulate the behaviors of P2P SIP over Chord and UP2P SIP. To be fair to compare the characters of these two designs, we arrange each node in simulators spend equal size of memory for routing table. In P2P SIP over Chord we simulate on a DHT-base network with name space 2^15, therefore, each node in which owns 15 entries in finger tables. Equally, nodes in UP2P SIP record at most 15 neighbors in routing tables. So we can suppose the comparison are on the square and observe the effect of other parameters.

To the efficiency, the number of online nodes is the first parameter to be concerned. We make a premise that if there is an online node liked to leave the network, there would be another offline node like to join. Such that we steady the total number of nodes online in the network and ensure the conditions in simulations are constant from beginning to the end.

In addition, when to leave and join for a node need to be concerned, too. We allocate nodes with certain amount of session time, the period of time that a node joined the network and leaves after a while. So the median session time of the network would decide the life of nodes and is closely linked with the churn rate of the network [10]. To obviously reveal the effect of mobility, we fix the churn rate of 12/minute which means nodes will see one node churning every 5 seconds in simulations. And the churn rate would partially decide the median session time of the network as well as the total number of online nodes.

Except the session time representing how long one node needs to leave, mobile nodes are additionally allocated with stationary time representing how long one node keeps static. Each time it expired, mobile nodes start the actions of moving (leave the network and rejoin after fetching a new IP address from local Accessing Point). We follow the configuration in study [11] which sets the stationary time uniformly distributed from 5 to 50 minutes. And relying on varying the number of mobile nodes we can clearly observe how serious mobility affects the performance of system.

The frequency of stabilization is one of the most key parameters. It is hard to master well and deeply affects the efficiency. We follow the suggestion in the draft of P2P SIP over Chord [5] in which nodes initiate the stabilization every 300 seconds in the network.

To be close to the real situation, we take the buddy lists of the users on the famous web site [12] as the sample of buddy lists in our simulations. So we can adopt the real relations between users and obtain the most actual result.

B. Results

To be a call setup protocol of VoIP service, latency for session initiation should be the first consideration that decides the quality of the service. However, introduced overheads for maintaining correctness of system are unavoidable if a well quality of service is required to be provided. So the tradeoff between call setup latency and maintenance overhead would the key points for performance of systems. In our simulations, we observe the variation of call setup latency and maintenance overhead while being in different environments.

We do not measure the effect of physical network because of the emphasis of the performance in architectures. Instead, we calculate the hop counts and number of messages needed for call setup and maintenance overhead. Followings are the results and analysis of performance.

1. Comparison with different numbers of online nodes

We vary the number of online nodes from 1000 to 10000 and explore the scalability of each approach.

In Figure 5 and Figure 6, we measure the average of call setup latency in different proportions of buddies to non-buddies such that that we can know the practicability of systems. In which, these line segments go curving following the increasing of numbers of online nodes and with a trend toward an upper bound in both approaches. The diagrams present the convergence properties owned in both approaches and these have been proved in [3, 7]. Besides, we can see that with larger proportion of calls to buddies costs less latency in establishing sessions and the performances of both approaches are similar. That is because there is already cached and tracked information in buddy list by invoking the presence policy [8] such that users can directly connect the sessions to buddies. Else, call setup with larger proportion of calls to non-buddies mostly relies on the performance of search mechanisms in the approaches and essentially costs more latency. It's clear to see that latency needed in

UP2P SIP approach is all the time lower than P2P SIP over Chord approach.

In Figure 7, it shows how much maintenance overhead needed to provide such quality of call setup services above. We can see both curves are arising exponentially following the increasing number of online nodes. But it's obviously that the requirement of maintenance overhead in UP2P SIP approach are all the time slower than in P2P SIP over Chord approach.

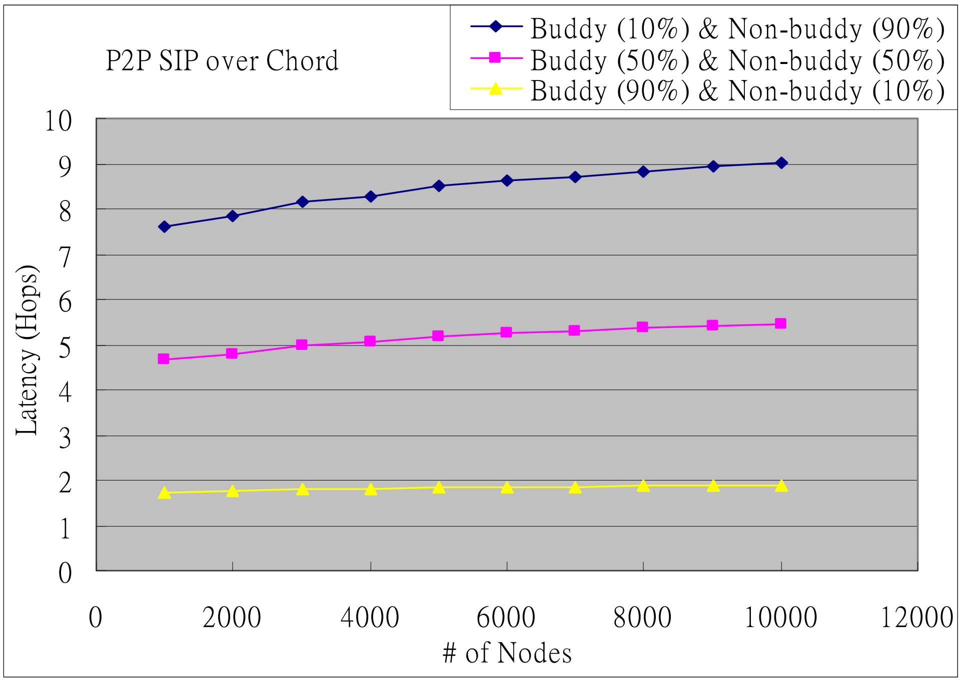


Figure 5: The variation of call setup latency while varying the number of online nodes in P2P SIP over Chord approach.

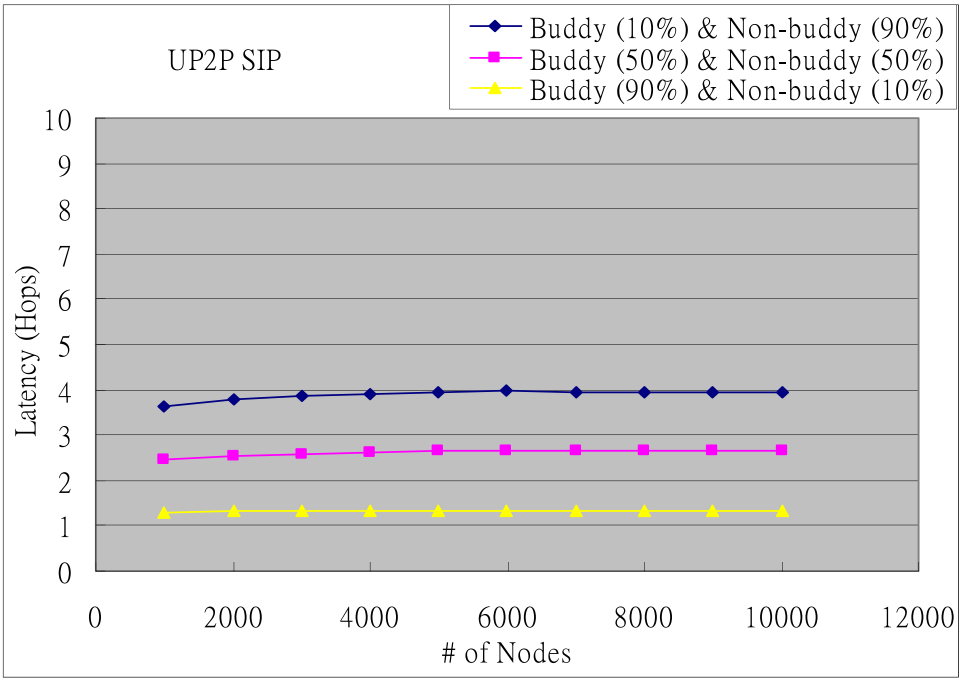


Figure 6: The variation of call setup latency while varying the number of online nodes in UP2P SIP approach.

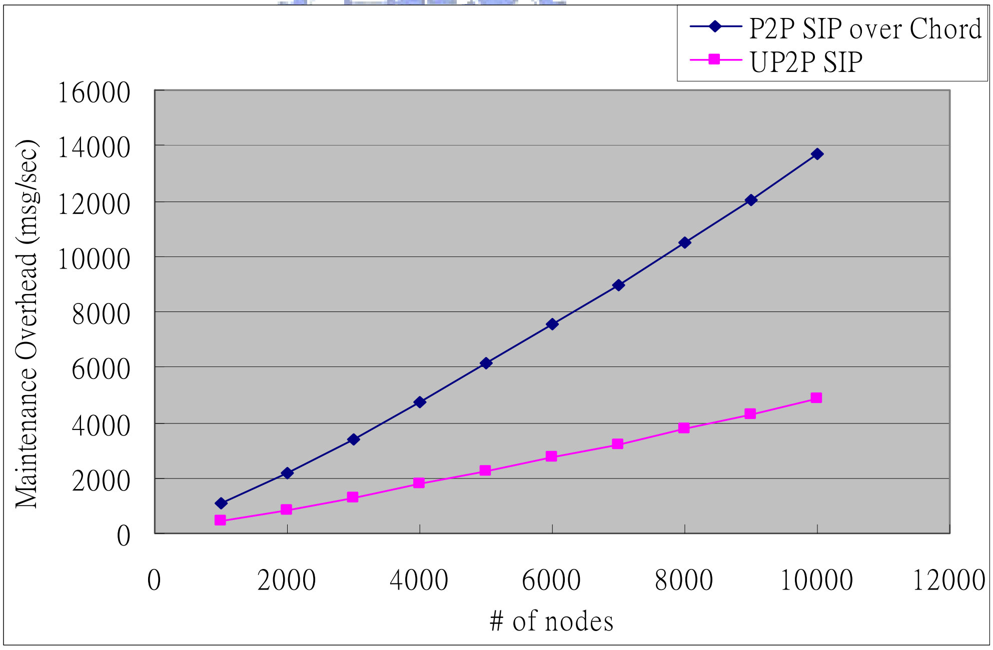


Figure 7: The variation of maintenance overhead needed to provide such quality of call setup service above.

2. Comparison with different numbers of mobile nodes

In the section, we try out to know the effect of mobility by changing the number of mobile nodes in the network, i.e. mobile rate.

In figure 8, it illustrates that the increasing of mobile rate follows the longer latency with no exception of different proportions of callee when call initiation in P2P SIP over Chord approach. This is because with more mobile users comes with larger opportunity to churn the network. This phenomenon results in more staled finger tables lengthening routing paths while call initiation.

On the contrary, we can see the latency remains steady in UP2P SIP approach in Figure 9. This is because the unstructured P2P network we adopt suffers less from the effect of mobility. Even several nodes moving out the network simultaneously, the relaying nodes can still flood the request to other existed neighbors and buddies. Besides, a low-diameter P2P network is constructed and in which each node seems to be arranged in the center of the network such that the call setup messages can be relayed along near shortest path.

In the case of maintenance overhead, Figure 10 shows the results that both systems pay additional penalty to handle the dynamics of network with increasing proportion of mobile users inside. But the costs paid in both approaches are scarcely comparable. Maintenance overhead in P2P SIP over Chord approach is a great quantity more than that in UP2P SIP approach. The reason is that nodes in P2P SIP over Chord has to maintain a logical overlay which is unstable due to frequent churns, especially caused by nodes moving. Nodes in which spend huge effort updating the staled finger tables periodically. In contrast, nodes in UP2P SIP approach master plenty connections for routing simultaneously and have no inflexible necessary correcting few entries which are staled in neighbor table.

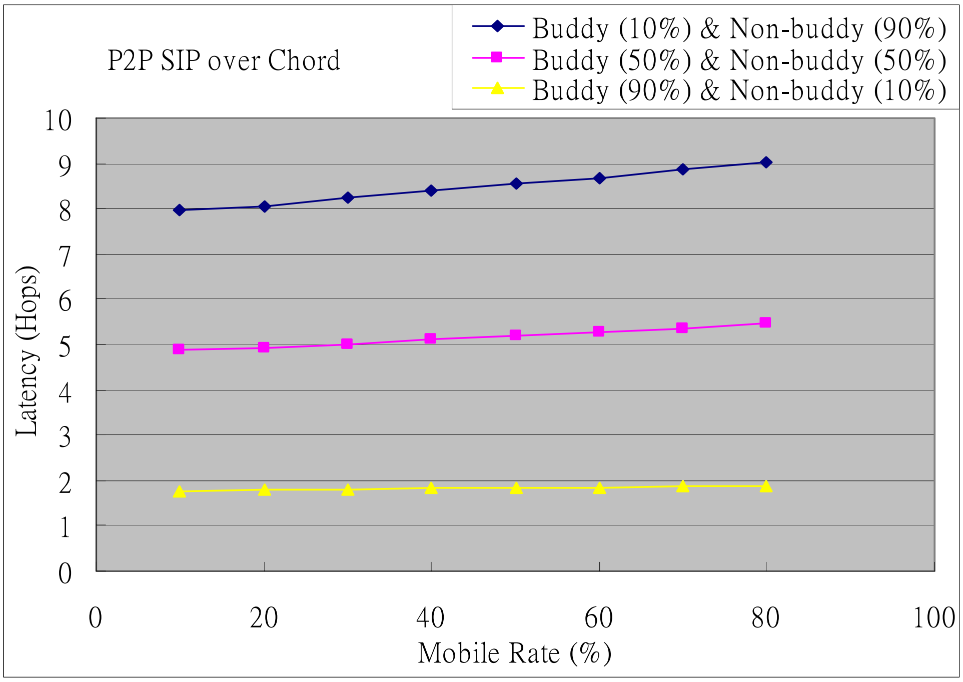


Figure 8: The variation of call setup latency while varying the proportion of mobile users in P2P SIP over Chord approach.

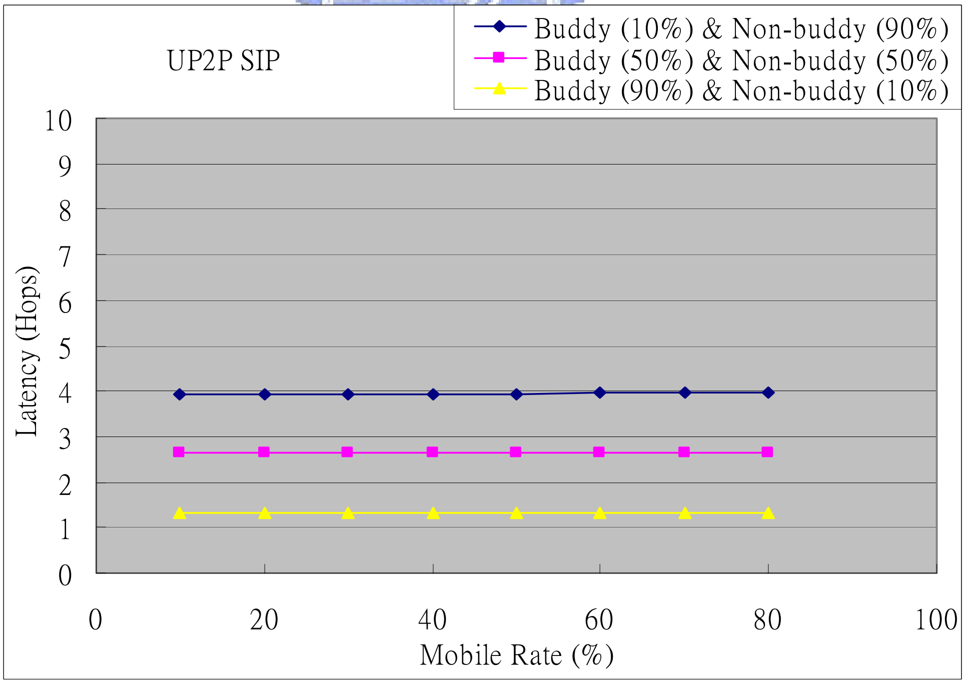


Figure 9: The variation of call setup latency while varying the proportion of mobile users in UP2P SIP approach.

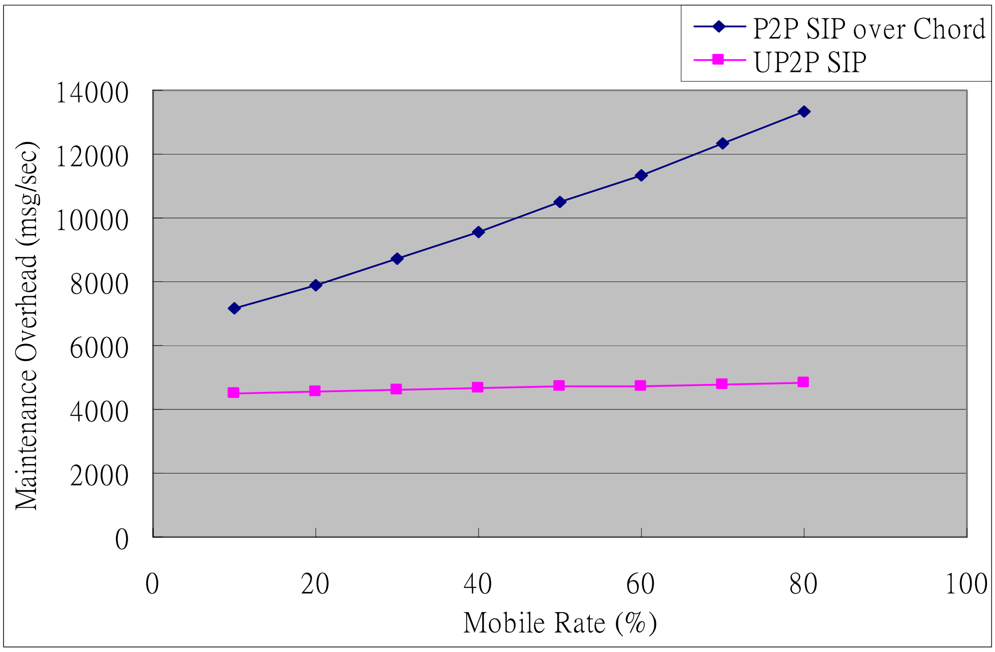


Figure 10: The variation of maintenance overhead while varying the proportion of mobile users in both approaches.

3. Comparison with different period of stabilization

Nodes in P2P network achieve gradual search relying on distributed routing tables (e.g. finger tables and neighbor tables), unfortunately, while are usually staled by network dynamics. Therefore, how long one node updates its routing table is a critical factor affecting performance of system. In this session, we wish to realize the difference of both performances by decreasing the period of stabilization in P2P SIP over Chord approach from 5 minutes to 1 minute. In Figure 11, among the curves represented P2P SIP over Chord approach, we can see any with shorter period cost less latency than the others with longer period. The more frequent stabilization comes with less latency while initiating a SIP call. However, the range of advance is finite that the performance is limited by the routing mechanism so that it can not cost latency as less as in our approach.

On the other hand, a great amount of maintenance overhead is introduced while increasing the frequency of stabilization. In Figure 12, we observe that the additionally increasing effect to handle staled tables is out of scale with the incoming benefit on call setup.

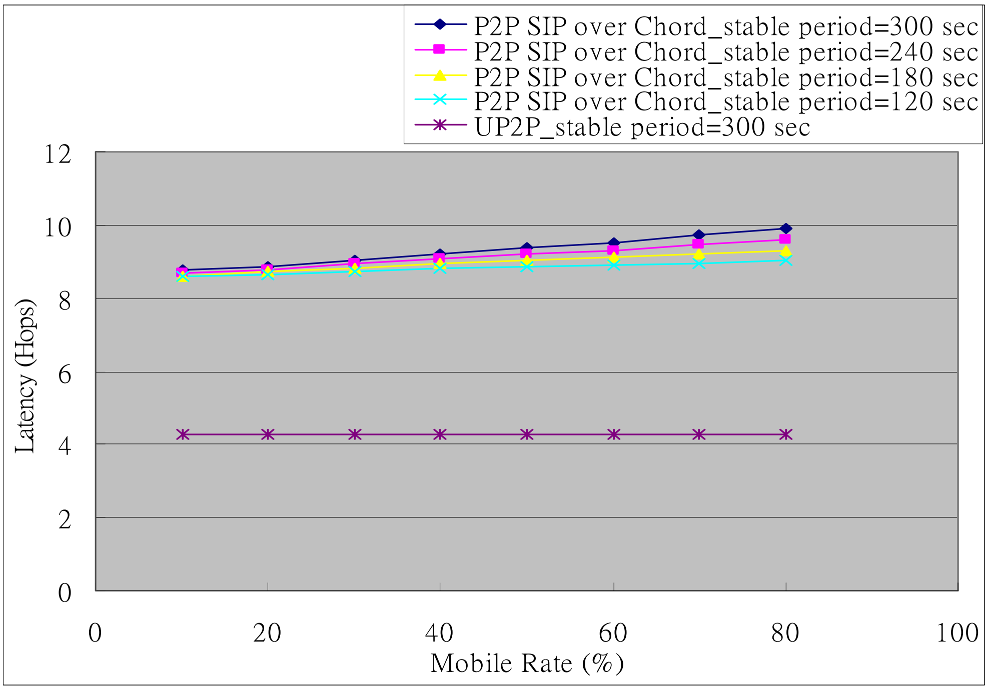


Figure11: The comparison of latency between P2P SIP over Chord approaches with different periods of stabilization and UP2P approach.

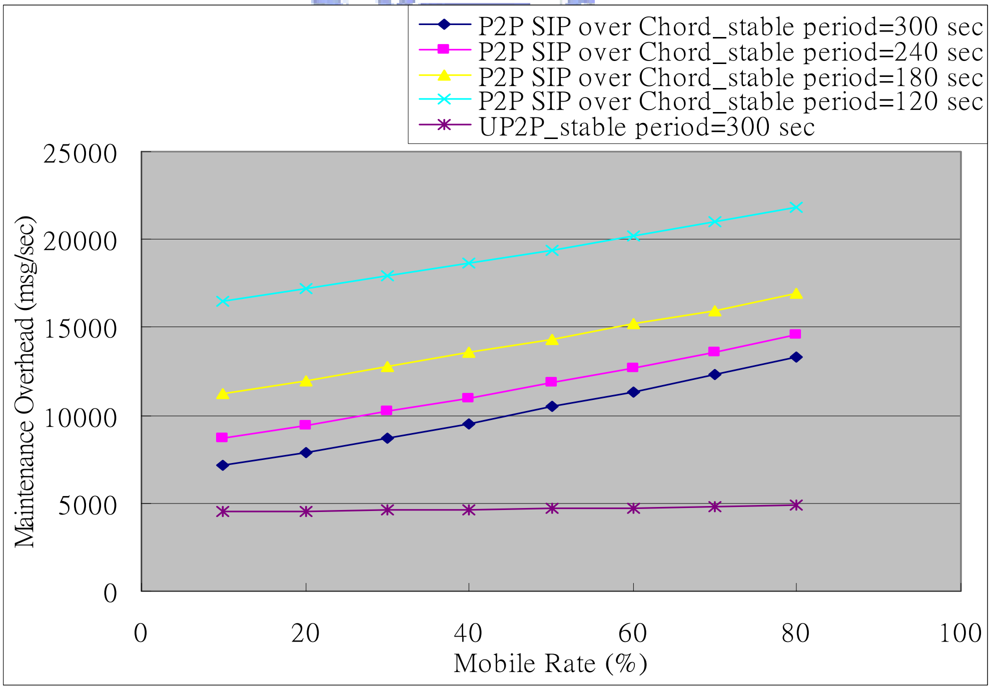


Figure12: The comparison of overhead between P2P SIP over Chord approaches with different periods of stabilization and UP2P approach

V. Conclusions

Recently there are several P2P approaches proposed for telephony using SIP, however, they are not concerned about mobility which are evolved from the popularity of wireless technology. Increasing of mobile users comes with more frequent churns and DHT-base architectures cannot suffer these dynamics emerged in great numbers. As a result, call setup latency and maintenance traffic are first and deep to be affected.

In the study, we propose an approach with unstructured P2P for telephony using SIP. In which, nodes directly connect to those with particular relationships, such as buddies, and not only shortens the latency when call setup but also decreases the overhead spent on maintaining the burdensome loading. Besides, to the non-buddy call initiation, we additionally build a low-diameter network ensuring any session can be rapidly connected. Finally we demonstrate through simulations. The simulation result shows that whether increasing the number of online users or increasing the proportion of mobile users in the network, our approach is more suitable being applied in mobile environment than DHT-base approaches.

References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, June 2002

[2] Gnutella, http://www.gnutella.com

[3] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," In Proceedings of the ACM SIGCOMM conference, Aug. 2001

[4] Skype, http://www.skype.com

[5] David A. Bryan, Bruce B. Lowekamp, and Cullen Jennings, draft-bryan-sipping-p2p-02, Mar. 2006

[6] Kundan Singh and Henning Schulzrinne, "Peer-to-Peer Internet telephony using SIP," 2004

[7] Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal, "Building Low-Diameter Peer-to-Peer Networks," IEEE Journal on Selected Areas in Communication, Aug. 2003

[8] A. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, June 2002

[9] Karl Aberer and Manfred Hauswirth, "An Overview on Peer-to-Peer Information Systems," Workshop on Distributed Data and Structures (WDAS-2002), 2002

[10] Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz, "Handling Churn in a DHT," University of California, Berkeley and Intel Research, Berkeley.

[11] Hung-Chang Hsiao and Chung-Ta King, "Mobility Churn in DHTs,“ In Proceedings of the First International Workshop on Mobility in Peer-to-Peer Systems (MPPS), ICDCSW 2005

[12] 無名小站, http://www.wretch.cc