Small World VoIP

Xiaohui Yang[1], Angelos Stavrou[1], Ram Dantu[2], and Duminda Wijesekera[1]

[1] George Mason University, Fairfax VA 22030, USA
{xyang3,astavrou,dwijesek}@gmu.edu
[2] University of North Texas, Denton TX 76203, USA
rdantu@unt.edu

Abstract. We present the analysis and design of a *Small World VoIP system (SW-VoIP)* which is geared towards customers that are communicating with their *Small World* of social contacts. We use the term *Small World* to refer to the Peer-to-Peer (P2P) network of a client and his contacts both incoming and outbound. We reconstruct the *small world* of a user by collecting calling patterns over a configurable period of time. We enable user mobility by using a stepwise *social identity* to an IP address binding propagation model. We propose an efficient algorithm to locate users by electing popular users and leveraging the users closeness. We also introduce a self-stabilized load balancing mechanism to optimize the system performance under heavy network traffic. We evaluate our SW-VoIP system performance by simulating the user's lookup process using real-world telephone logs. Our experimental results show that our SW-VoIP system offers a better performance in optimizing the required routing path and reducing the average lookup delay when compared to traditional, non small-world P2P VoIP systems.

Keywords: small world, VoIP, Peer-to-peer, mobility, electing, popular, closeness, optimize.

1 Introduction

In popular P2P VoIP systems, to make a call, the caller's agent has to first query for the network location information of the called party using the P2P routing. Only after this query is completed, the caller can continue the launch of the SMS, voice, or video stream either using P2P routing directly or other protocols to transmit content. In contrast to traditional client-server VoIP model, the P2P VoIP systems are designed to provide a reliable and cost efficient telecommunication environment, that avoid heavy infrastructure investments, centralized bottlenecks and single points of failures. Unfortunately, due to P2P's decentralized nature of routing and the lack of explicit or implicit trust among peers, P2P VoIP systems are vulnerable to attacks such as Spam, Sybil, Phishing, to name a few. A malicious user with a spurious or unconfirmed network identity can mis-route, eavesdrop, or even resume the identity of a trusted party, leading to call dropping, wiretapping, or unsolicited calls. To make matters worse, even normal user behavior that involves continuous joining and leaving of users and

M. Griss and G. Yang (Eds.): MOBICASE 2010, LNICST 76, pp. 137–155, 2012.

their changing of physical locations lead to higher maintenance cost and lead to communication degradation. Beyond the nuisance that can cause, all of the aforementioned problems can inhibit the widespread adoption of VoIP systems.

To alleviate some these problems, we harness the social nature of the telephone service to design a system that uses social information to optimize the quality of the underlying P2P VoIP network. To that end, we observe that user telephone call graphs exhibit characteristics of a *small-world network* [24]. In such networks most nodes, in our case users, can be reached from every other node by traversing a small number of hops. Inspired by this scale-free property of the telephone call graph, we propose to develop a *small world* call graph based VoIP system. Therefore, we leverage the structural properties of the small world network of the telephone call graph for each user to efficiently route call connection requests, we call our approach SW-VoIP. In SW-VoIP, we cluster users that communicate frequently together. Furthermore, we designate the users that are called by many others as *leaders* for locating users, creating a *hub-and-spoke* type of network. We derive this telephone call graph information from real-world call logs and we developed a prototype system to evaluate the performance benefits of SW-VoIP.

For our performance evaluation, we employed a testbed of 1000+ nodes on 300+ PlanetLab [2] machines using OpenVoIP [1]. We measured our algorithms by comparing user lookup performance on SW-VoIP to OpenVoIP using Kademlia [11]. Based on the call log data set, we explored the performance test on 97 nodes of the SW-VoIP and the OpenVoIP system. We measured the average hop count and system performance by varying the number of and size of user clusters. Our experimental results show that our proposed small world P2P VoIP system offers reduced average routing hop count and minimized average user lookup delays when compared to normal P2P VoIP system.

The rest of this paper is structured as follows: section 2 details the related work. Section 3 explains our design goals and choices. Section 4 describes how we construct the small world overlay using call graphs. Section 5 provides details on the workings of our routing algorithm. Section 6 illustrates our experimental setup and evaluation. We conclude with Section 7.

2 Background and the Related Works

2.1 P2P VoIP System

P2P VoIP traffic differs from P2P file-sharing network traffic and from traditional telecommunication network due to its interactive nature and characteristic resource consumption. Currently, there are several proposals for P2P VoIP system design, and some of them achieved pervasive commercial deployment.

As the most popular commercial P2P VoIP application for Internet Telephony, Skype [3] is deployed to offer free computer-to-computer calls or paid telecommunication services through VoIP-to-PSTN gateways for distributed network users. Skype uses centralized authentication server to establish initial identities, and offers a super node mechanism to manage contacts, relay data flow when necessary, and enable NAT traversal. Skype gives VoIP users advantages by lowering operating costs and facilitating worldwide calls. But deficiencies exist due to the proprietary nature of protocol and the inherent vulnerabilities of P2P system. As a result, lots of general telecommunication Use Cases are unavailable in Skype, and difficulties exist in adding new features as third party solutions to this P2P VoIP system.

On the other hand, work was in progress on developing open standards based P2P VoIP systems for Internet telecommunication. In particular, Session Initiation Protocol (SIP) [17] based P2P VoIP have emerged as a mainstream in system design. In P2P-SIP system, the selection of P2P algorithms and the overlay organizing mode are the two main factors affecting system design. Most current proposals prefer using structured P2P overlay such as Distributed Hash Tables (DHTs) to identify peers and store resources. DHT routing algorithms such as CAN [15], Chord [23], etc, are chosen according to network and user requirements such as tradeoff efficiencies, failure resilience, routing hot spots, node heterogeneity, etc. In addition to intrinsic P2P properties, methods used to combine P2P and SIP protocols are crucial to designing an efficient P2P VoIP system.

Most of current P2P VoIP system design propose to transmit signaling in P2P mode whereas direct media flows end-to-end. For example, the earlier SIPpeer project [19] and SOSIMPLE project [5] use SIP messages to create DHT overlay and convey routing information. Meanwhile, an alternative research [20] at Columbia University uses an external DHT to establish communication sessions among users. Additionally, a few companies claim to offer their own P2P systems for VoIP application, such as Avaya's one-x and Popular Telephony's Peerio.

As commercial P2P VoIP deployments began to merge, the Internet Engineering Task Force (IETF) showed great interest in P2P signaling protocol development. Recently, the REsource Location And Discovery (RELOAD) base protocol [8], a peer-to-peer signaling protocol was proposed to provide clients an abstract storage and messaging service between a set of cooperating peers in the overlay network. RELOAD supports a P2P-SIP network and functions in environments where many nodes are behind NATs or firewalls. The lightweight load on participating peers ensures a high performance system routing.

2.2 Small World Utilization

Watts and Strogatz [24] characterized a small world network as a collection of loosely connected subgraphs with a high clustering coefficient and a low average shortest path. In small world networks, short chains of acquaintances link arbitrary pairs of strangers, connects any user to their *local* and *remote* contacts using a small number of long edge and a large number of short edges, where the length refers to the communication overhead. Additionally, a hub node with a high number of connections serve as a trusted common connection to mediate the short path length between edges of interconnected nodes.

Many researches have investigated properties of small world networks to improve network performance and solve application level complexities. One typical utilization of social network, as shown by Marti [10] is to improve routing performance and to use social links as *paths of trust*. The algorithm proposed by Marti adds a 2-level lookahead *friends links* to Chord [22], and uses the minimum hop distance to efficiently integrate a social network to with a P2P network. Although SPROUT proved to be robust under a large fraction of malicious users, it neglect the differences between a users list of friends and uses a greedy routing algorithm to work with the underlying DHT algorithm without taking trust into consideration during routing.

Hui [6] developed the SWOP protocol, a representative application of small world properties to construct and maintain an structured P2P overlay network. Taking the advantages of high clustering coefficient and low average hop distance, the small world overlay P2P network in SWOP can efficiently lookup resource objects and perform well under heavy traffic. SWOP introduces the concept of head nodes and long links from small world network, but security may be compromised due to choosing inappropriate head nodes during cluster formation.

In order to achieve the shortest routing path in a small world overlay, nodes can be grouped using many criteria. For example, clusters can be constructed using nodes' hashed key value distance to other nodes. Iamnitchi [7] proposes an algorithm to dynamically identify groups with similar interests by using information about data consumed by P2P users. Other criteria such as transmission delay, common interests, etc, are also the popular ways of grouping in current small world P2P networks.

3 Problem Setting and Design Goals

A P2P network to be useful for VoIP has to satisfy some extra conditions in order to accommodate operational realities of telecommunications. Firstly, VoIP customers frequently change their physical location, resulting in changing IP addresses but retaining a fixed telephone number - requiring updating the IP address attributes of the small world social graph. Secondly, telephony requires some centralized control in order to provide services such as E-911 [16], Lawful Interception (CALEA) [18] etc. Thirdly, telephone customers use many customized service packages, and in face of user mobility, these may raise difficulties. Fourthly, telephony communications are delay sensitive - an unwise routing decision or an unexpected node congestion may result in failed calls or inexecutable services. Based on the properties of a small world network and general telephone user's calling behavior, we provide the following three mechanisms as a solution:

– *Optimizing Update:* User mobility makes social links and stored object references obsolete, making many extra transmissions necessary, resulting in call setup delays, and perhaps missing calls. As a solution, we propose continuous IP to telephone number binding updates in a step-wise manner. We propose this mechanism for normal system operations rather than to be used in emergency situations requiring fast connections.

- *Distributed Control:* Administration, services deployment and membership management in telecommunications requires centralized functionality. We propose to distribute these functions to users with high social connections and credibility. These users will play some pivotal roles in controlling message flow (e.g. prioritizing E-911 signal or choosing call dissemination targets) and act as regional auditing center (e.g. checking points for CALEA). They may use different routing policies to send connection requests between friends or unfamiliar users. Messages may also be filtered or differentiated at each distributed center in order to be processed under different services.

- *Delay Protection:* We group users based on the strength of their social relationships and locate called parties using popularly accessed nodes, thereby enhancing message propagation. To avoid single point congestion and reduce message delays, system is optimized to encourage the second best routing choices.

4 Small World Overlay Construction

Telephone call graphs correctly reflect users' calling habits that can be used to measure the *closeness* of their calling behavior, which can be leveraged to enforce identity assurance, reduce call delay, and improve network performance in P2P VoIP systems. The first step is to obtain and update the call graph in order to create the P2P network.

4.1 Distributed PCrawler Design

To acquire current P2P VoIP call graph, we intermittently collect users' call logs, and extract related information such as user ID, number of friends, call frequency, call duration, etc. Because collecting call logs from a large number of peers at one time will lead to a low network performance, using multiple queries and parallel responses are infeasible. Using the idea of web crawlers [9], we introduce a *Peer Crawler (PCrawler)* that crawls a subset of P2P VoIP network to acquire up-to-date copies of call logs. The collected copies of call logs will be downloaded into the cache of the dispatcher with an expiration timer. Meanwhile, important identity information will be extracted and indexed by call log analyzer to provide and validate user identification and accelerate preliminary statistical evaluation.

In addition, the PCrawler automatically maintains the small world overlay. As a user logs in and actively participates in P2P VoIP communication, his/her identity information is stored in a tuple <*SID, IP, Port*> and also harvested by his/her friends using social links. When the user changes the location, his/her telephone number to IP binding will be changed and the PCrawler will find the new binding.

The PCrawler architecture is shown in Figure 1. PCrawler is composed of two components: a call log collector and a consistency checker. PCrawlers are generated by a *Dispatcher*, that is in charge of PCrawlers authentication, call log analysis, and small world network ID generation for the whole P2P VoIP system. PCrawlers are disseminated in each cluster, and the number of PCrawlers is decided by the Dispatcher depending on the size of network. Dispatcher acquires the size of network from enrollment server according to the number of users enrolled. Starting with an list of unexplored peers assigned by the Dispatcher, PCrawlers collect call logs, pick a list of unexplored friends from peers, and crawl those friend's entries recursively while extracting all useful information from the collected call logs and reporting them to the Dispatcher.

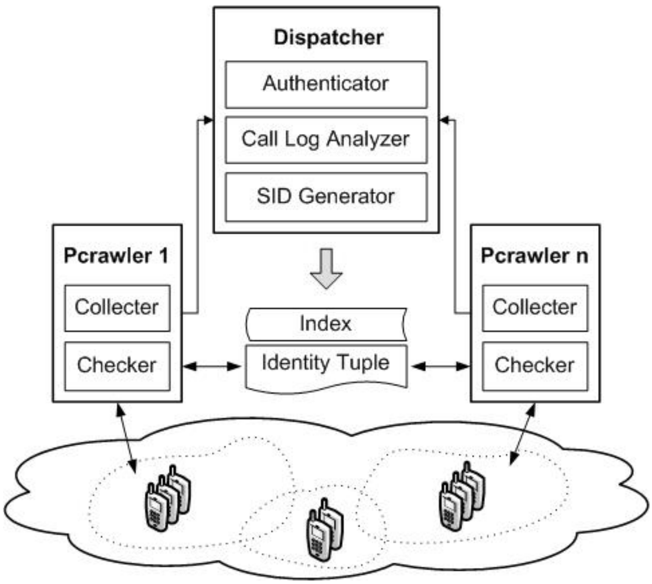

Fig. 1. PCrawler Architecture

Dispatchers are responsible for authenticating PCrawlers before they start a session with P2P peers and request for call logs. Authentication is enforced in order to avoid illegal utilization of PCrawlers to launch malicious attacks. On receiving calling data from PCrawlers, a *Call Log Analyzer* clusters users and generate a small world network ID for each user based on some rules, which are described shortly. *SID* in the identity tuple will thereupon be filled up in the order of call log index.

We implement the Dispatcher in C++ using *Tulip* [4] libraries to construct or update call graphs. We adopt a clustering technique provided by Tulip based on the calculation of the strength of the edges of a small world graph $G = (V, E)$, with a set of vertices V edges E. The PCrawler uses a map $\phi : E \to R$ assigning a real number $\phi(e) \geq 0$ to each edge $e \in E$, and call graphs can be clustered based on removing the edge with the metric ϕ lower than threshold value of $t \in [a, b]$.

In addition to collecting call logs, PCrawlers periodically check the status of users with the help of a Breadth-first search algorithm. If the user's *SID* does not match the *SID* stored in the indexed cache, or if the user's address data is different than that of the identity tuple under the same *SID*, the PCrawler will trigger an update program in the peers to replace the old tuple with the new one. If a user has changed its locality for a sufficient time or have more contact with a different group of users (as measured by the function ϕ on the edges), can be assigned a new *SID* belonging to a new cluster.

4.2 Utilizing Structural Properties of Call Graph

We maintain and leverage the *closeness* of callers in the call graph cluster to enrich cluster our P2P VoIP system. The variable number of clusters m is determined by the number of user n and the closeness of the whole small world network. In each cluster, users are tightly connected, and able to reach others by a small number of hops, which we call *intra-cluster* communications. Clusters are loosely connected, and the users who have connections to users in other clusters provide links for *inter-cluster* communications. Nodes providing interconnections between clusters are called *hub nodes*. The i-th cluster has h_i hubs, where $1 \leq h_i < n$, and every *normal user* in a cluster is directly connected to at least one hub, as shown in Figure 2. These hubs are endowed with three important properties: (1) highly trusted in the whole small world networks, (2) bridge clusters and provide routing path for inter-cluster communications, (3) offers *check points* for the centralized control, making them supervise inter-cluster activities.

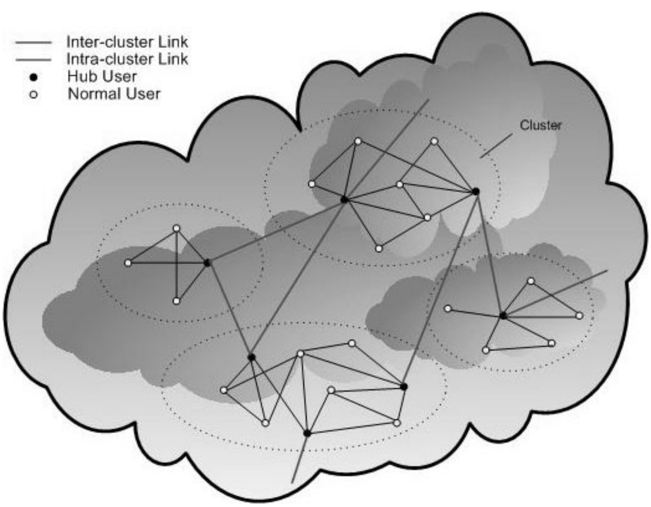

Fig. 2. Clustered Small World P2P VoIP Network

Electing a hub in each cluster relies on many factors such as node degree d, number of inter-cluster links l, calling frequency f, Round Trip Time rtt, etc. We measure a user's popularity based on the assumption that the more friends a user has and the more calls a user made/received over a period of time, the more popular the user is. Popularity p_i is proportional to the average calling frequency $ave(F) = \frac{1}{d}\sum_{k=1}^{d} f_k$ and the contact degree $d^{l/d}$, where the ratio of l to d is calculated to weigh up the importance of a hub for inter-cluster communication. Meanwhile, telecommunications low latency property requires a low average rtt of a hub to its neighborhoods, which is calculated by $ave(RTT) = \frac{1}{d}\sum_{j=1}^{d} rtt_j$ and is reversely proportional to p_i. Given a cluster with y users, hub user election is defined by comparing the popularity p of each user i based on the parameters of d, l, f, and rtt.

$$p_i = \frac{d^{l/d} * ave(F)}{ave(RTT)} \quad (i = 1, 2, \ldots, y) \tag{1}$$

The user with the highest popularity p_{max} will announce its parameter values to all the other users in the same cluster for the normalized popularity $\boldsymbol{p_i}$ calculation.

$$\boldsymbol{p_i} = \frac{p_i}{p_{max}} \quad (i = 1, 2, \ldots, y) \tag{2}$$

Each user will select from its friends list and put forward a hub with $Max(\boldsymbol{p_i})$. As a result, each user is connected to at least on hub which is regarded as the most popular user in its neighborhoods.

5 Algorithms

The incorporation of structural properties [12] of small world network makes user oriented P2P VoIP system more efficiently organized and routing path more intelligently designed. Equipped with the functionalities of quick locating and dynamic updating, our intimacy oriented P2P VoIP system is applicable to any structured or unstructured P2P overlays, and also compatible with any normal P2P system for telecommunications.

5.1 Closeness-Based Routing

We use a numeric *Social Identifier (SID)* chosen from $[0, 2^{160})$ for each social user in the small world call graph. We divide the SID space by the number of social clusters m, and users in the same cluster share the same x-digit prefix, where $m = 2^x$. We further divide the SID space of each cluster according to the number of hubs and users and express the *entities* in the SID space as a binary tree, where users are the leaves of the tree, and the height of the tree is determined by the number of digits in the identifier. We illustrate the SID space binary tree of the above small world network in Figure 3. Thus clusters are differentiated by the identifier sequence along the binary tree, where length of the sequence is determined by x. For clusters with only one hub user, we assign the lowest *SID* in the cluster to the hub, and assign the subsequent *SIDs* to normal users. For multi-hub cluster, SID space is divided by the bundles of hub and normal users, and the bundles with higher popularity are assigned lower *SIDs*.

In our setup, a cluster Hub is in charge of collecting and disseminating information and is considered a highly credible routing pivot. In order for a caller's agent to generate a call, it needs to get the address information of the callee. As in any DHT system, all the users information is stored in the small world overlay in the form of $< key, value >$ pairs. We use a hash of the mapping $< IP, Port >$ as the key, and use the same procedure to locate an user object as to locate a closest user to the key.

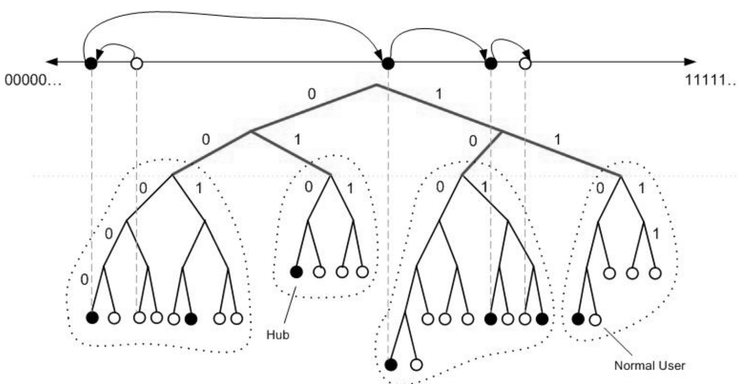

Fig. 3. Routing on the Clustered Small World Network

We illustrate the user lookup process in Figure 3 when *user_a* calls *user_b* in our structured small world network in Algorithm 1.

The intra-cluster routing process contacts only $O(\log \frac{n}{m})$ users, where n is the total number of user and m is the number of clusters. Whereas the overall routing process has time complexity of $O(m + \log \frac{n}{m})$, where the value of m is the guarantee for efficient routing.

In addition, hubs act as virtual authorities and supervises activities in and between clusters. Their high popularity makes a hub highly trusted by its connected normal users, and gives a hub particular rights for supervising and auditing. By analyzing telephone call graph, each user will be given an initial credibility based on the node degree d and the call frequency f. Each user can then choose its hub from the connected friends and confer full trust based on the provided credibility. Users may prefer to trust its most intimate friends rather than the publicly known virtual authorities, and their communication path will also vary correspondingly. Our ongoing work incorporate adaptive trust computing into the routing algorithm.

Algorithm 1: User Lookup

```
for (i = 1; i ≤ Number_of_Friends; i++) do
    if (SID_b = SID_i) then
        Return u_i;
    else
        if (CID_b = CID_a) then
            FIntra = 1;
        else
            FIntra = 0;
if (FIntra = 1) then
    for (i = 1; i ≤ Number_of_Friends; i++) do
        Dist_ib = |SID_i − SID_b|;
        Find a hub user h_i who has the minimum Dist_ib;
        Return h_i as the next hop;
else
    if (u_a.hub = 1) then
        for (i = 1; i ≤ Number_of_Friends; i++) do
            Dist_ib = |SID_i − SID_b|;
            Find a hub user h_i who has the minimum Dist_ib and CID_i = CID_b;
            Return h_i as the next hop;
    else
        Send UserLookup(Key_b, SID_b, CID_b) to u_a's hub;
```

5.2 Join and Leave

SW-VoIP nodes participate in a P2P VoIP network using normal P2P protocols. However, the construction and maintenance of a small world P2P VoIP system is a user-interactive process which requires timely role reset and user awareness. The process of join and leave in SW-VoIP has its specific operations, which we illustrate in Algorithm 2.

To assure identities and prevent malicious attacks such as Spam, DoS, etc, a user u_a is not allowed to join the small world network until it has communicated with and been recommended by a given number n_r of SW-VoIP members. The criteria that a SW-VoIP member used to judge external users are based on the number of calls made between them ec, and the average duration time those calls a_ed. If $ec \geq 3$ and $a\_ed \geq 10m$, the external user can get that member's recommendation, and if there are recommendations from at least 3 members, the external user can be recommended to join SW-VoIP network.

Receiving the signal of SW-VoIP members' recommendations, a PCrawler acquires a new user u_a's information and call logs, and reports them to the Dispatcher for issuing a SID. The SID can be used by u_a to infer the cluster it belongs to and set its own *cluster Id (CID)* to the appropriate cluster number. u_a then retrieves the corresponding SID and CID of its connected friends, and calculates their popularities, based on which it elects a hub user for future routing. The user thus successfully becomes a member of small world network by setting *Status Flag (SF)* to 1, where $u_a \in Cluster_a$ and $Cluster_a \subseteq SW\!-\!VoIP$, and can launch calls using mechanisms provided by SW-VoIP.

When a user leaves the P2P VoIP network, it checks if it is a member of small world network and if so it is a hub user in the SW-VoIP. A normal SW-VoIP user leaves the small world network by simply informing its connected friends and passing the stored keys on to its closest friend under the same hub. When a hub user leaves, all the connected normal users are informed to recalculate the popularity p of their friends and elect the next best alternatives in the same cluster as their hub. In many cases, hub users leave temporarily and come back to the network in a short period of time t where $t \leq T_{temp}$. Informed with the quick return of the former hub user, normal users will resume the previous connections stored in the cache and push back the current ones. If a hub user returns after a long time t where $t > T_{temp}$, it will be regarded as a new user but with a high initial credibility, and its information stored in the cache will be removed. Meanwhile, the PCrawler responsible for that area will report the new hub to the Dispatcher.

5.3 Load Balancing

Hub users with high popularities may receive a large amount of simultaneous messages due to their pivotal roles in the SW-VoIP. Bottleneck caused by multiple TCP connection requests will result in a big delay for users lookup, which becomes problematic due to the timeliness issue of connection establishment. To optimize lookup performance, SW-VoIP is designed to provide a self-stabilizing and priority-based load balancing mechanism at the application layer.

Before joining the SW-VoIP, a user calculates the throughput Th of its host node based on a well-known TCP throughput model proposed by Padhye [13], where MSS is the maximum segment size, rtt is the average round trip time measured by TCP, and plr is the package loss rate. We assume that MSS and plr are constant across the overall P2P VoIP network, and the value of rtt is based on the average waiting time in the tasks queue of connection request and the propagation delay is determined by the network complexity. The throughput Th should be no larger than $(MSS/rtt) * f(plr)$ to ensure successful message delivery.

Suppose that a nodes had queued n_c many tasks when it receives a user lookup message m_c such as $REQUEST$, $RESPONSE$, ACK, etc, and generates n_c TCP connections to its next hops. We calculate n_c as $n_c = \min_{m_c, NC_{max}}$, where NC_{max} is the maximum number of connections that can be made by each node. NC_{max} is determined by the maximum throughput Th_{max} allowed, user's popularity p, ratio r_p of tasks with high priorities, and two self-controlled parameters α and β $[\alpha, \beta \in (0, 1)]$ using the following formulae.

$$NC_{max} = \alpha * Th_{max} + \beta * p * r_p \tag{3}$$

Algorithm 2: Join and Leave

OnJoin()
Join exiting P2P VoIP network;
if $(Num\_of\_Recommenders \geq n_r)$ **then**
  Acquire small world identity SID and CID;
  for $(i = 1; i \leq Number\_of\_Friends; i{+}{+})$ **do**
    Retrieve SID_i and CID_i from u_i;
    Calculate popularity p_i of u_i;
    Elect a hub user h with highest p;
  $SFlag = 1$;
else
  $SFlag = 0$;

OnLeave()
if $(SFlag = 1)$ **then**
  if $(u_a.hub = 1)$ **then**
    for $(i = 1; i \leq Number\_of\_Friends; i{+}{+})$ **do**
      Inform u_i its leave;
      Ask u_i to cache its information and elect a new hub with the next highest p;
      Set $Timer(t_i)$ on u_i;
      Close connection;
  else
    Inform all the connected friends its leave and close connection;
else
  Leave existing P2P VoIP network;

If a user is popular or a node has a queue of tasks with high priorities, the maximum number of TCP connections will be increased by increasing the parameter α and decreasing β. Otherwise, decrease the maximum number of TCP connection to reduce expense. The maximum number of connections NC_{max} acts as a signal to inform users about its popularity and tasks status.

As stated, a user in SW-VoIP maintains a list of friends from whom it can elect a hub user with high popularity, and decide the next routing hop based on the closeness of its friends. We optimize SW-VoIP by giving message senders or forwarders the ability to avoid congestion, which we call the *next best choice*. If the tasks waiting in the queue reaches the maximum number of connections NC_{max}, a signaling flag $FConn$ is used to inform others that this node is congested. If $FConn$ is set to 1 in node a, and any user b who plans to connect to a will give up and choose a next best alternative c from the friends list as its hub user and connect to it. The user b will continue checking $FConn$ of its next hubs until it finds a task queue without congestion. The process of load balancing and choosing the new hub is described in Algorithm 3.

Sometimes users may receive high priority calls such as Reverse 911 [21]. For the time critical lookup messages, SW-VoIP gives the emergency calls the highest priority without waiting in the queue or a relatively high priority to be queued and processed quickly. Requests originating from friends can also be given a priority to be queued and processed, and the priority of a task can be increased or decreased according to the credibility of the message source.

Algorithm 3: Load balancing

```
OnReceive()
if (n_c < NC_max) then
    FConn = 0;
    Enqueue the task of TCP connection;
    n_c++;
else
    FConn = 1;

OnSend()
if (FConn = 0) then
    Generate TCP connection request;
else
    for (i = 1; i ≤ Numb_of_Friends; i++) do
        Find the next hub h with highest popularity p;
    Repeat OnSend to h;
```

6 Experimental Evaluation

In this section, we describe the experiemnt we did in order to compare the object lookup performance of our small world P2P VoIP system with other P2P VoIP systems using normal P2P algorithms. We also conduct repeated trial to find the best performance through adjusting corresponding metrics. The results show that our small world P2P VoIP system can use proposed mechanisms with optimized performance.

6.1 Simulation

Our simulations are conducted using real-life call logs of 97 individual mobile phone users over the course of nine months which were collected by the Reality Mining project group at the Massachusetts Institute of Technology (MIT) [14]. Call logs were collected using Nokia 6600 smart phones loaded with software written to record phone information including incoming/outgoing calls, users in proximity, locations, etc.

We deploy a P2P VoIP system testbed of 1000+ nodes on 300+ PlanetLab [2] machines using OpenVoIP [1], which is an open source P2P VoIP system developed by Columbia University supporting well-known DHTs and unstructured P2P protocols. Based on the assumption of steady user call behavior, we generate P2P VoIP traffic in our system by using these real-life call logs. We set up the six time periods as parameters, which is weekday, weekend, school open, winter break, daytime, and nighttime, and generate traffic with reference to these scenarios.

We report the behavior of node degree distribution in Figure 4, which resembles a power-law distribution with a small portion of high-degree users. Because we have the call log collection of only 97 users, the node degree distribution (e.g. maximum 139 friends) shows that we do not cover all the friends of a user in our small world overlay. We also present the node degree distribution of the SW-VoIP overlay user in Figure 4. We simulated traffic using these two call graph and evaluated our algorithms.

We randomly pick 97 nodes on the testbed and launched about 10,000 real-time P2P VoIP user lookup messages among those nodes, according to who is making or receiving calls at one time, how many calls are made during certain period of time, and how much duration time was used for each call. The small world overlay is constructed based on the information extracted from the call logs on these nodes through distributed PCrawler system, and in turn used to facilitate communications in the P2P overlay.

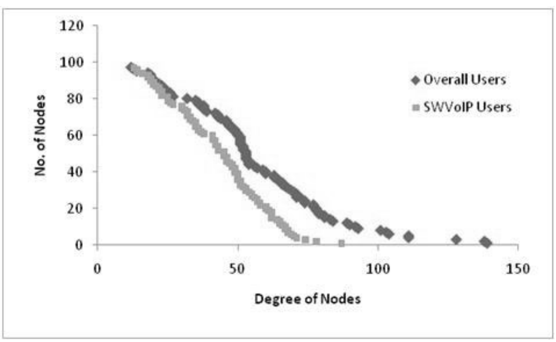

Fig. 4. SW-VoIP Users Call Degree Distribution

Each user in the small world overlay maintains a list of friends, who have called or have been called. If no such friend exists, a normal P2P overlay is used for VoIP communication and the subsequent discovery of new friends. On receiving a user lookup request and confirming itself, the recipient user updates its friends list and stores the call initiator's information carried by the user lookup object. We illustrate this meta-data information in user lookup object in Figure 5, which is a prerequisite for successful routing in small world overlay.

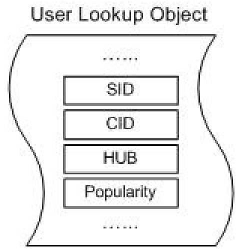

Fig. 5. Small World Meta-data User Info

6.2 Algorithm Evaluation

We evaluated our algorithms by comparing user lookup performance on SW-VoIP to OpenVoIP using Kademlia [11]. Based on the call log data set, we explored the performance test on 97 nodes of the SW-VoIP and the OpenVoIP system. We measured the average hop count and system performance by varying the number of clusters m = 1, 2, 4, and 8 respectively, as the number of hub users varies in the order of 3, 20, 30, 38, correspondingly. This evaluation helps us estimate the optimal number of clusters and reduce user lookup time. The analysis of MIT data set shows that on average 10 percent of the users are on the phone at any time. Therefore we randomly choose 10 nodes and performed a list of users lookups on them at the same time. We illustrate the probability density function of hops for the users lookup under SW-VoIP and OpenVoIP systems in Figure 6, and summarize the average lookup delay in Table 1, which show that our SW-VoIP has a lower average number of hops and lower average delay in user lookup than the OpenVoIP system.

It is important to observe that SW-VoIP has a higher percentage of 1 or 2 hops but a lower percentage of multi-hops $h(h > 2)$ than OpenVoIP. It was also observed that SW-VoIP of 2 or 4 clusters has a better system performance than others, which shows that the efficiency of user lookup is tightly related with the partition of users and the closeness of overall hubs.

Table 1. Average User Lookup Performance

	OpenVoIP	SWVoIP 1	SWVoIP 2	SWVoIP 3	SWVoIP 4
AVE HOP	2.275	1.75	1.72	1.73	1.85
AVE Delay (ms)	470.6575	454.419	324.907	204.83	491.465

We observed that there is a larger number of intra-cluster calls than inter-cluster calls. We generate these two kinds of user lookup traffic on 4 clusters of the SW-VoIP system respectively, and illustrate the system performance in Figure 7 and Figure 8. For intra-cluster calls, most of the routings are processed in $1 \sim 3$ hops with the delay of $0 \sim 400$ ms, whereas inter-cluster routings have the scattered test results ranging from $1 \sim 6$ hops with $0 \sim 1000$ ms delays.

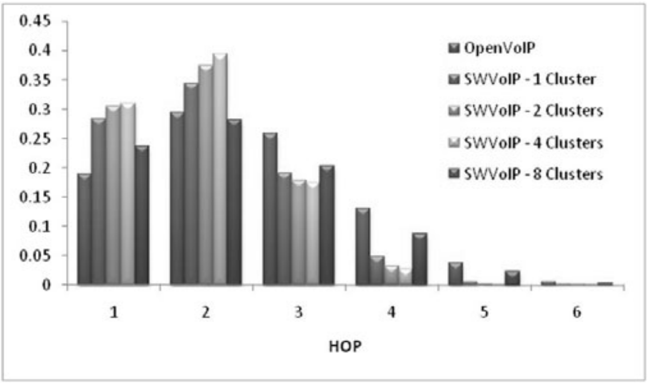

Fig. 6. Probability Density Function of User Lookup Hops

The test results conform that the utility of our SW-VoIP system that message routing are conducted using users' intimated social relationship, and the closer the users are connected, the faster the messages are routed.

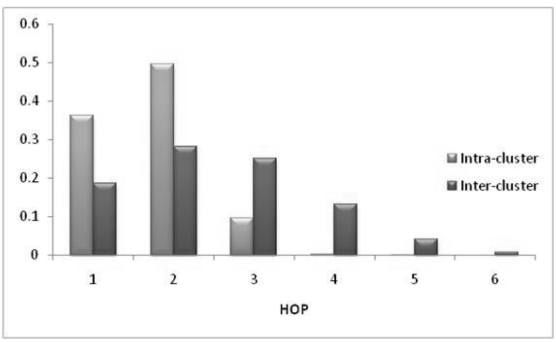

Fig. 7. Intra- and Inter-cluster Hops Comparison

We further explored user lookup performance under heavy network traffic. We attempted to build as many connections as possible by simultaneously launching user lookup traffic on 97 nodes in OpenVoIP and SW-VoIP respectively based on real-life call logs. The traffic generated include messages issued not only from those 97 users but also from additional 52 users in P2P VoIP network, as shown in user call degree distribution of Figure 4. We assume that these 97 users in SW-VoIP are divided into 4 clusters as in earlier described experiments, and load balancing mechanism has little effect on the average link traversal of our SW-VoIP communications. We compared two systems by evaluating user lookup delay under up to 10,000 simultaneous messages. It is observed in Figure 9 and Figure 10 that not only SW-VoIP has less average delay than OpenVoIP, but also

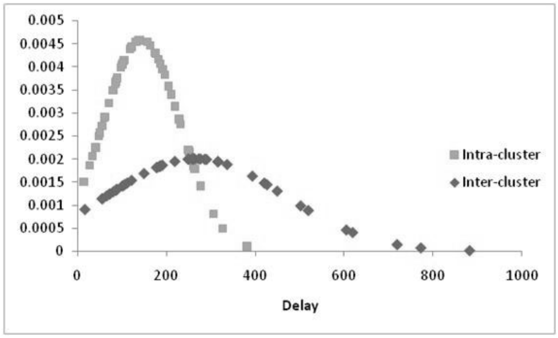

Fig. 8. Intra- and Inter-cluster Delay Comparison

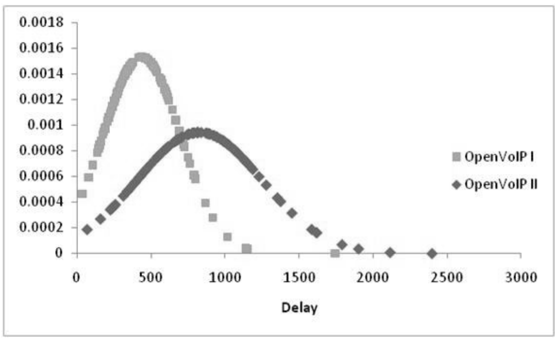

Fig. 9. OpenVoIP Delay under heavy traffic

delays in SW-VoIP increases much less than those in OpenVoIP, which shows that SW-VoIP has a better load balancing capability than OpenVoIP.

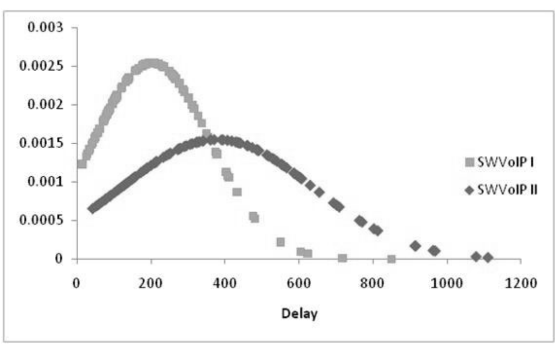

Fig. 10. SWVoIP Delay under heavy traffic

We noticed that system performance may be affected to some extent by parameters setting of PCrawlers such as the frequency of information updating, the rate of simultaneous peer crawling, the number of PCrawers in SW-VoIP, etc. For example, user lookup delay will be increased by up to 10,000 ms if we deploy a PCrawler for each user and concurrently update address information in every 10 ms. Our ongoing experiments are addressing these quantitative issues.

7 Conclusion

We have devised a novel model to leverage the small world properties for Peer-to-Peer VoIP communications based on the telephone users specific social behavior. In order to experimentally validate the utility of our model, we constructed a user oriented small world overlay on top of a P2P VoIP system, and use self-optimization methods to collect data, analyze user behavior, and utilize small world structural properties to facilitate telecommunication. Moreover, the design of SW-VoIP system aims to provide distributed control, epidemic updating, and adaptive trust computing to P2P VoIP users, which are important for the deployment of telecommunication services. Experiments were carried out to optimize clustering users in small world overlay, and compare the performance of SW-VoIP with other P2P VoIP system using normal P2P routing mechanisms such as Kademlia. Initial experiential results reported in this paper show that our model improves system performance by reducing average routing hop and user lookup delays.

Acknowledgments. This work is partially supported by the National Science Foundation under grants CNS-0751205 and CNS-0821736.

References

1. Openvoip: an open peer-to-peer voip and im system, `http://www1.cs.columbia.edu/~salman/peer/`
2. Planetlab: an open platform for developing, deploying, and accessing planetary-scale services, `http://www.planet-lab.org/`
3. Skype, `http://www.skype.com/`
4. Auber, D.: Tulip: A huge graph visualisation framework. In: Mathematics and Visualization (2003)
5. Bryan, D.A., Lowekamp, B.B., Jennings, C.: Sosimple: A serverless, standards-based, p2p sip communication system. In: IEEE, AAA-IDEA (2005)
6. Hui, K.Y.K., Lui, J.C.S.: Small-world overlay p2p networks: Construction and handling dynamic flash crowd. Computer Networks: The International Journal of Computer and Telecommunications Networking (2006)
7. Iamnitchi, A., Foster, I.: Interest-aware information dissemination in small-world communities. In: HPDC (July 2005)
8. Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., Schulzrinne, H.: Resource location and discovery (reload) base protocol. draft-ietf-P2PSIP-base-10 (August 2010)

9. Khambatti, M., Ryu, K.D., Dasgupta, P.: Structuring Peer-to-Peer Networks Using Interest-Based Communities. In: Aberer, K., Koubarakis, M., Kalogeraki, V. (eds.) VLDB 2003 Ws DBISP2P. LNCS, vol. 2944, pp. 48–63. Springer, Heidelberg (2004)
10. Marti, S., Ganesan, P., Garcia-Molina, H.: SPROUT: P2P Routing with Social Networks. In: Lindner, W., Fischer, F., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 425–435. Springer, Heidelberg (2004)
11. Maymounkov, P., Mazières, D.: Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 53–65. Springer, Heidelberg (2002)
12. Nanavati, A.A., Gurumurthy, S., Das, G., Chakraborty, D., Dasgupta, K., Mukherjea, S., Joshi, A.: On the structural properties of massive telecom call graphs: findings and implications. In: CIKM (2006)
13. Padhye, J., Firoiu, V., Townsley, D., Kurose, J.: Modelling tcp throughput: A simple model and its empirical validation. In: Communications Architectures and Protocols (August 1998)
14. Phithakkitnukoon, S., Dantu, R.: Inferring Social Groups Using Call Logs. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 200–210. Springer, Heidelberg (2008)
15. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: SIGCOMM (2001)
16. Rosen, B., Schulzrinne, H., Polk, J., Newton, A.: Framework for emergency calling using internet multimedia. draft-ietf-ecrit-framework-10 (July 2009)
17. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.R., Peterson, J., Sparks, R., Handley, M., Schooler, E.: Sip: session initiation protocol. RFC 3261 (June 2002)
18. Seedorf, J.: Lawful interception in p2p-based voip systems. In: Principles, Systems and Applications of IP Telecommunications, IPTComm (July 2008)
19. Singh, K., Schulzrinne, H.: Peer-to-peer internet telephony using sip. In: NOSSDAV (June 2005)
20. Singh, K., Schulzrinne, H.: Using an external dht as a sip location service. Columbia University Technical Report CUCS-007-06 (February 2006)
21. Sorensen, J.H., Sorensen, B.V., Smith, A., Williams, Z.: Results of an investigation of the effectiveness of using reverse telephone emergency waning systems in the october 2007 san diego wildfires. In: Oak Ridge National Laboratory/TM 2009 (June 2009)
22. Stoica, I., Morris, R., Nowell, D.L., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup protocol for internet applications. ACM Transactions on Network 11(1) (2003)
23. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: SIGCOMM (August 2001)
24. Watts, D.J.: Collective dynamics of small-world networks. Nature (1998)