P2P/ Grid-based Overlay Architecture to Support VoIP Services in Large Scale IP Networks

Wei Yu[*], Sriram Chellappan[#] and Dong Xuan[#]
[*] Dept. of Computer Science,
Texas A&M University, U.S.A.
{weiyu}@cs.tamu.edu
[#] Dept. of Computer and Information Science,
The Ohio-State University, U.S.A.
{chellapp, xuan}@cis.ohio-state.edu

Abstract: *Communication services such as, Voice over IP (VoIP), over the Internet have gained much attention in recent years, as next generation Internet applications are requiring the integration of voice and data in the single IP infrastructure. In this paper, we propose a Peer-to-Peer (P2P)/ Grid-based architecture to efficiently provide VoIP services in large scale IP networks. Particularly, our technologies include: 1) A Multi-overlay architecture that leverages the resources/ capabilities of individual VoIP components. 2) We design and develop several models and protocols for realizing VoIP services in our architecture. We conduct extensive performance evaluations on different schemes proposed. The evaluation results show that the load-aware weight-based call routing scheme can achieve much better performance than static selection schemes in terms of average call routing delay. The experimental results also demonstrate that the P2P+hierarchy model for conferencing applications can achieve better performance than all other models in terms of minimizing the network bandwidth overhead.*

Keywords: Communication Services, Grid Computing, P2P-based Computing, Overlay, VoIP

I. Introduction

In this paper, we propose a Peer to Peer (P2P)/ Grid-based architecture to provide scalable and efficient Voice over IP (VoIP) communication services in large scale IP networks. We leverage the resources/ capabilities of all components present in the VoIP networks into two grid overlays: a call routing overlay and a user/service endpoint overlay. In this P2P/ Grid-based architecture, we study various issues related to design, analysis and evaluation of VoIP services on these overlays.

Providing scalable and efficient communication services such as VoIP over the Internet has become an active research area in recent years, as the next generation Internet applications are requiring the integration of voice and data in the IP infrastructure. However existing system for VoIP have following drawbacks: 1) *Centralization of call control*. A majority of the existing approaches to deploy communication services in the IP network adopt the *client/ server* model. For example, H.323-based communication system has two types of components: the centralized call control agent as the *server* and endpoint IP phone user agent as the *client*. As the server provides the call control and feature services for all clients, it easily becomes the bottleneck due to the centralization of data access and control. 2) *Centralization of feature delivery.* Similar to the approach for call control, the existing approach to deliver features is also centralized. The drawbacks can be illustrated by the following example: Centralized approaches make the system unable to effectively support some basic features such as, conferencing. Traditionally the conference feature is conducted by the call control server which directs the call to the centralized conference bridge with a large amount of conferencing resources. When conferences have large number of users, the network will easily be congested due to a large amount of traffic directed to certain area of the network. 3) *Manual call routing configuration*. To support the communication service with clients at different locations, the traditional approach requires that each call agent manually configures a 'route-pattern' to each other call agent. In this sense, in a system with *N* locations, each call agent needs to maintain *N-1* 'route-patterns'. This approach cannot is not scalable and will have much maintenance overhead and will be unable to realize global deployment of the service in the near future. In the migration period, a scalable call routing architecture is highly required to support a variable network topology considering call agents joining/ leaving the system dynamically. This will be necessary for transparent and automatic deployment of new call agents without much impact on existing communication systems, call agent failing due to DoS (Denial of service) attacks in the IP network [1].

Following from the above observations, we propose a novel and effective approach to address the problem of VoIP communication services. We propose a scalable and efficient P2P/ Grid-based architecture integrating all the components present in VoIP systems. Our architecture leverages the resources/ capabilities of all components such as, IP phones, service agents etc, that can be shared, managed, coordinated and controlled in an effective way. Our architecture is thus completely distributed, as each entity maintains comparatively small states and increases the capacity of the global system to better conduct the call control and provide features. In recent years, P2P and Grid computing have become effective methodologies to support the large scale services. Examples of such P2P systems are application level overlay systems such as CAN [2], Chord [3] and Tapestry [4] are scalable, decentralized and self organizing systems. Globus toolkits are evolving towards an Open Grid Services Architecture (OGSA) in which a grid provides an extensible set of services for the virtual organizations [5]. Leveraging P2P/ Grid technologies for global communication services bring in two valuable benefits – improved efficiency and better QoS (quality of service). Since the IP phone endpoints in this infrastructure can exchange information directly, rather than through intervening dedicated servers, work and results can be distributed quickly and efficiently.

In this paper, we employ the following technologies for VoIP services. 1) *Multi-overlay architecture*: We propose an intelligent P2P/ Grid based architecture with two function grid overlays that leverages the resources of all components in the VoIP systems. The two overlays are: call routing grid overlay and user/ service endpoint overlay. The call routing grid overlay is composed of call agents, which are dedicated to perform the call routing. The endpoint grid overlay is composed of all the user endpoint agents and service agents, which are dedicated for the call control and feature delivery. Our architecture is highly distributed and is also compatible with IETF standard – SIP (Session Initial Protocol) [6]. 2) *P2P-based call routing grid overlay*: We propose a P2P based call routing grid overlay. We also study protocols for overlay construction and maintenance under dynamics, and for forwarding the received call routing request messages by designing efficient multi-path routing algorithms. We propose two schemes, namely a Static routing scheme and Weight random selection scheme for path selection. Our protocols use easily available system information to achieve better QoS performance. 3) *P2P-based user/service endpoint overlay*: This overlay takes care of call control and feature delivery. We develop four overlay models, namely centralized, hierarchical, P2P and P2P+hierarchy models to provide such services.

We conduct extensive performance evaluations on different models proposed. The rest of paper is organized as follows: In Section II, we introduce the network model and some terminologies. In Section III, we present our architecture with two grid overlays: call routing overlay and endpoint overlay. In Section IV, we discuss the call routing overlay along with network dynamics. In Section V, we discuss the endpoint overlay for large conferences. In Section VI, simulation and evaluation results are presented. In Section VII, we give the survey of related work. The summary of this paper and some future work are given in Section VIII.

II. Network Models

In the communication system, we consider a world-wide voice signaling network with a large number of region call agents. The call agent is a signaling middleware-box and is capable of handling call related signaling messages for the IP phone endpoints. The region call agent can support different application layer signaling protocols in the IP network, such as SIP, H.323 and MGCP (media gateway control protocol). Each region call agent is assigned with a route-pattern for global call routing purpose. Assuming that each region call agent can support 10,000 phones, 979432 is the corresponding route-pattern for the particular region call agent with dialing number – 979432XXXX.

Assume that each region control agent manages a given number of registered IP phones. When the IP phone makes the outgoing call by dialing the destination digits, the region agent finds the destination region call agent through the call routing infrastructure. The destination region call agent will negotiate with the source region agent and will setup the call. We restrict our work to a single domain or multiple domains within our jurisdiction. In this sense, we can deploy call agents anywhere in domains with known network topology.

We assume that endpoint IP phone can contribute to the communication service, its available CPU and memory resource. For example, it may have limited DSP (digital signal processing) resources to support ad-hoc conferencing features. In other words, besides the normal DSP resources to play the tone and ring, we also assume the each IP phone has extra DSP resources to conduct conferencing with limited number of participants.

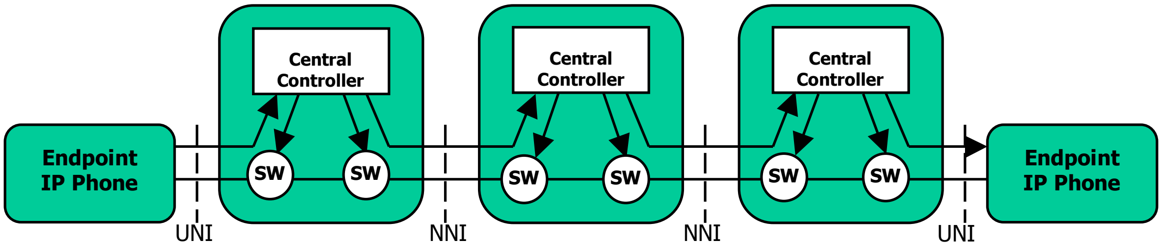


Figure 1: Traditional Centralized Call Model

III. Intelligent P2P/Grid-based IP Communication Architecture

In the traditional VoIP system, services are delivered by the *client/server* model shown in Figure 1, where the centralized call control server conducts the basic call signaling and feature interaction for the IP phone endpoint user agents. In this model, the user has little control over the call processing and feature delivery as the user device is considered as a dummy device without any intelligence or resources to deliver features. All the system intelligence is located at the centralized call control agent, which becomes a very complex system. The services and features are implemented on the top of the central call control server by using feature primitives. With more features required in the communication services, software design becomes the bottleneck due to complexity in handling the interactions of different features and call control coupled with fast delivery demanded by next generation communication services.

In order to develop a scalable, lightweight and easily deployable architecture to efficiently provide communication services, we propose an intelligent P2P/ Grid-based architecture that leverages the resources/ capabilities of system components as shown in Figure 2. The highlights of our architecture are: 1) The workloads of both call control and feature service are distributed to all components including the user endpoint IP phone. Thus call control and feature delivery becomes scalable, as there is no centralization in maintaining large number of states for the call control and feature delivery. 2) Call routing is conducted by the call routing grid, which is composed by a number of call routing agents which are dedicated to perform the call routing. In the call routing grid, the P2P-based approach is adopted to efficiently support large scale systems. It can also handle network dynamics under call agents joining/ leaving the system. 3) Feature delivery is handled by the endpoint overlay grid, which is composed by all endpoint user agents and service agents in the system. In this architecture, components contribute their available resources/ capabilities and make the system effectively support features such as, large conferencing etc.

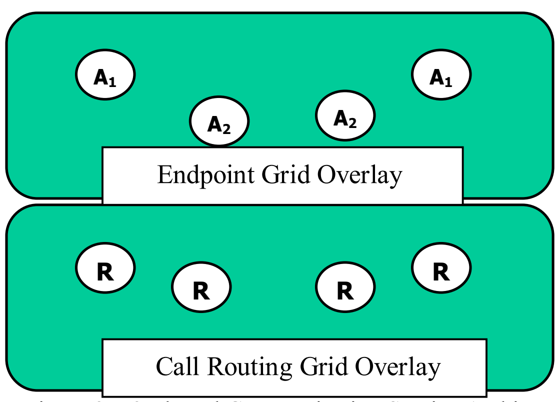


Figure 2: P2P-based Communication Service Architecture
(A_1 – User Endpoint Agent, A_2 – Service Endpoint Agent, R – Call Routing Agent)

Generally, our communication service architecture provides an intelligent environment, which can enable all components in the system to effectively manage their resource, complement each other, hence making the system scalable to support features for communication services. In the following sections, we will discuss these two overlays in details.

IV. Call Routing Overlay

A. Overview

Our call routing problem is defined as follows: Given a signaling network with call agents and destination route-pattern, we need to find the corresponding call agent for the given destination route-pattern. In other words, we need to find the mapping between the dialing number and the corresponding call agent. Accordingly, we have following different approaches: 1) *Centralized approach:* Just like Napster [7] and the DNS system for the Internet file sharing and naming service respectively, this approach has a centralized directory server or a hierarchical tree with different directory servers. All call agents register with the directory server. When the call agent receives call setup requests, it checks the dialed number. If the dialed number is out of its region, it sends the request to the directory server for the destination lookup and the directory sever responds with the corresponding call agent for the dialed number. The main advantage of this scheme is its simplicity and easy implementation. But this server could be a bottleneck and a single point of failure. 2) *Purely distributed approach:* In this approach, each call agent maintains the information of all other call agents by means a fully connected topology. Similar to the centralized approach, the call routing path for each source/destination pair will be just one hop. With increase in system size, this approach is not scalable as each call agent needs to maintain a large number of states for other call agents. 3) *P2P-based overlay approach:* To alleviate the above problems, we propose a P2P-based overlay approach. This approach requires an overlay signaling network consisting of call agents and each call agent in the system has only a limited number of neighbors. Thus, each call agent only needs to maintain local information and call agents cooperate to achieve the global call routing objective. The benefit of this approach is that it is highly scalable and fault tolerant. The P2P overlay approach is naturally scalable under large system sizes.

In our P2P-based overlay, the following two tasks need to be conducted: 1) Overlay construction and maintenance: This task deals with how the overlay topology is constructed and its maintenance under network dynamics such as node joins/ leaves. 2) Call routing forwarding: This task deals with how to forward the call routing request in the overlay to achieve the call routing objective. In the following sections, we will discuss the above two components in detail.

B. Overlay Construction/ Maintenance Protocol

The objective of overlay construction and maintenance protocol is to establish call routing table for the routing message. This is achieved by, 1) generating candidate entries for constructing and updating call routing tables, 2) determining eligibility of candidate entries, 3) constructing or updating call routing tables with the eligible entries, which will be used by the call routing forwarding protocol, discussed later.

We can use existing approaches to collect the network information, e.g., Landmark approach proposed in [8] can be used for monitoring the node distance. Once the information on network status is collected, the routing table for each node can be easily constructed, by say, selecting the set of nodes with shortest distance. For the overlay construction and maintenance, the main task is to handle dynamic behavior of nodes in the network (node joins/ leaves). Protocol 1 lists two sub-protocols to handle the cases of node joins and node leaves. During the system initial time, the overlay can be automatically constructed by each node running the node join sub-protocol. During the run-time with new node joining/leaving, these two sub-protocols will be executed correspondently, the new updated call routing table will be automatically generated.

Protocol 1: Overlay Construction and Maintenance Protocols (OCMP)

H: the set of neighbors of node N (node leaving case), d: the number of neighbors, N: the route pattern of node joining/ leaving the system, where N[i] represents i-th digit, S: the set of neighborhoods for node N (node joining case), R: root node

```
// Node Join Sub-protocol (for node N)
We assume that new node N can always find the network root node R in its local region
    Sends message(R, N) to the root node R and sets S empty
    R broadcasts the message to its neighbors
    When node L receives the message
        Calculate the difference of route-pattern (L, N) with mask_LN
```

The $mask_{LN}[l]$($l \in [1, T]$) is calculated as

$mask_{LN}[l] = 1$, if $L[l] < N[l]$
$mask_{LN}[l] = 2$, if $L[l] > N[l]$
$mask_{LN}[l] = 0$, if $L[l] = N[l]$

If only one of $mask_{LN}[l]$ for $l \in [1, T]$) is not equal to 0 and there is no neighbor, which is nearer to N than L
Send message response back to R
Both L and N update their call routing tables
Else
Forwarding the message based on protocol 1
With receiving d responses, the node N finalizes the call routing table

// Node Leaving Sub-protocol (for node N)

Node N sends message(X) to all local neighbors H[1,...,d] and each neighbor H[i] just deletes the entry N from its call routing table
For entry H[i], find the symmetric neighborhood H[x] satisfying following condition
Assume only i-th digit is different between H[i] and N
If (N > H[i])
Find the neighbor X with i-th digit larger than N[i]
H[x] = X
If(N < H[i])
Find the neighbor X with i-th digit less than N[i]
H[x] = X
H[i] and H[x] set up the neighbor relationship.

C. Overlay Call Routing Forwarding Protocol

The objective of the call routing request forwarding protocol is to forward the received call routing request messages. This is achieved by performing the following three tasks in sequence: 1) locate the call routing entry (entries) for the incoming call request, 2) select one entry among the located entries and 3) forward the message according to the fields in the selected entry.

To perform these tasks, we propose the call routing request forwarding protocol below.

Protocol 2: Call Routing Request Forwarding Protocol (CRRFP)

T: the length of route pattern for the dialing digits, S: the route pattern of the calling party number, D: the route pattern of the called party number, P^i: the route pattern at call agent node i, M(S, D): routing request message with source S and destination D

For call agent i, it receives the call routing request message M(S, D)
Calculate the $mask_{SD}$ to identify the difference between route pattern P^i and D
The $mask_{P^iD}[l]$ ($l \in [1, T]$) is calculated as

$mask_{P^iD}[l] = 1$, if $P^i[l] < D[l]$
$mask_{P^iD}[l] = 2$, if $P^i[l] > D[l]$
$mask_{P^iD}[l] = 0$, if $P^i[l] = D[l]$

If $mask_{P^iD}[l]$ are 0 for all $l \in [1, T]$
The call agent i is the destination
The call agent i finds the corresponding IP phone as the destination and forwards the request to the IP phone user agent
Else
Assume there are d non-zero entries in $mask_{P^iD}[1, .. , T]$
Select 1 from d non-zero entries (recorded at the $mask_{SD}$) set with selection schemes by formula (2)
For example, j-th entry is selected
Find the next hop, i.e., N
Forward message M(S, D) to the next hop N

In connection with Protocol 2, we have following theorem.

Theorem 1 (Correctness of the protocol 2): If the network has no faults, then with our call routing protocol, the call requesting message from a source will eventually be delivered to the destination.

Proof: Let us say an agent *R'* receives a message from node *R* (*R* can be source node). From the call routing algorithms mentioned above, we can easily achieve the following conclusion: given any path taken by our protocol, we can prove that *R'* is always one hop near to the destination than *R* (recall that we choose one of *d*-non-zero entities in the mask calculated in Protocol 2). In this sense, the call routing is loop free, which guarantees that the call routing request will eventually arrive at the destination. *Q.E.D.*

With QoS considerations, our motivation for the call routing is to minimize call routing message delay. As there are multiple paths from the source to the destination, we adopt multi-path routing selection approach. As we know, the throughput of routing messages depends on the network topology and load of network node. Hence, a multi-path selection decision should take all available information into account.

The path selection information represents the system information which can be used for path selection. We consider two important factors: the distance of the path *P* and load of network node *L*, where *P* can be obtained by call routing algorithms and *L* can be obtained by exchanging the node load information among network neighbors periodically. From here, we will regard the available information for the path selection as metrics. A single type of information corresponds to one metric. In the case where there is only one type of the information (in other words, only one metric), say, the distance of the path, it is straightforward to figure out that the path with shortest distance is better than one with longer distance. In the case of multiple metrics, say, the distance of the path, and load on a node, it is difficult to judge which one is better among the following two paths: one is with shorter distance and heavy load, or one with a longer distance and lighter load.

Hence, we need to combine different individual metrics together to get a composite one, which can be done in many different ways. The basic idea is to design a function, taking these metrics as input parameters. The output of this functions in the composite metric *C*, i.e., $C = f(P, L)$. Note that this function should be monotonically increasing or decreasing for each metrics. A simple function can be

$$C = L / P . \tag{1}$$

Once the composite metric is ready, the following issue is how to use it to select a path. We propose following weighted random selection approach.

The basic idea of the weighted random selection approach is that each path is assigned to a weight, and a path is chosen randomly based on weight assignment. The path with higher weight value has a higher probability of being selected than one with lower weight values. This approach may effectively distribute the traffic over different routes and potentially improve the throughput performance.

Weight assignments for multi-path routing have been studied in the literature. For example, an optimal weight assignment was proposed in [9] in which the average delay of message can be minimized. We will take a heuristic approach for weight random selection.

Without loss of generality, we assume that *K* eligible entries are in the routing table for a given destination route pattern. We let these entries be indexed by *1, 2,…, K*; and their associated weights are denoted as $W_1, W_2, .., W_k$, respectively. We can assign the weight $W_i \sim C_i$ (C_i is calculated by (1)) for each neighbor node.

To normalize W_i to be a probability such that $\sum_{i=1}^{K} W_i = 1$, we should assign W_i as follows: for $i = 1, \ldots, K$,

$$W_i = \frac{L_i / P_i}{\sum_{j=1}^{K} L_j / P_j} . \tag{2}$$

V. Endpoint Overlay

In the endpoint overlay, the most important component is the endpoint user agent, i.e., IP phone. In this overlay, the P2P-based approach is adopted to make all components contribute their available capability/ resource and make the system effectively support features such as, large conferencing and so on. For large conferences (a large number of endpoint users are joining in a conference), depending on sequence of each node joining in the conference, all the conference members construct an overlay dedicated to the conference. In this overlay, each endpoint node can contribute its local DSP resources to conduct the conference media mixing. Thus, more efficient conferencing can be supported by this approach. We discuss this in detail below. Note that our analysis is generic and can be used for providing other features too.

Assuming that *N* endpoint user agents are in a large conference, we consider following models for the conference feature:

1) *Centralized Model:* Each endpoint user agent keeps two media sessions connecting to the centralized media service agent. From the endpoint user agent perspective, the outgoing media session sends the *RTP* packets from the endpoint user and the incoming media session receives the *RTP* packets from the media service agent.

2) *Hierarchical Model:* The system has multiple media conference service agents, which are hierarchically organized. For example, the two layer hierarchical network can be the system with *R* regions. Each region has the *N/R* entities and *1* centralized media service agent is deployed. All endpoint user agents just connect to the local region media service agent just like the centralized model. All region media service agents can be reconnected to support the large conference.

3) *P2P-based Model:* In this system, we assume that end point user agents have limited capability to mix the media connected to it. For example, consider user *A* and *B* having an active call. Then, *A* decides to conduct a conference by inviting user *C* by making a consultation call to *C*. There is no call set up directly between *B* and *C*. *A* receives media stream from both *B* and *C*, and mixes them. *A* sends a stream combining *A*'s and *B*'s streams to C. In this sense, the endpoint user agent within the small area can be grouped as a cluster with a leader (*A* is the cluster leader in this case) which takes the responsibility to conduct the media mixing for the local cluster. Depending on sequence of all nodes joining in the conference, all the conference members construct an overlay dedicated to the conference. In this overlay, each endpoint node can conduct the conference media mixing locally to achieve efficient conferencing.

4) *P2P+Hierarchy Model:* In the system, the Hierarchical model and P2P-based model are combined. In this sense, the network has *R* regions and each region adopts the P2P-based model.

Based on above models, we define following performance metrics: Bandwidth Overhead (*BO*) defines the bandwidth consumption for different models. Based on the media topologies for each model, we can easily calculate the bandwidth overhead as following:

1) *Centralized Model:*

$$BO_c = K + 2K^2 + ... + (\lceil \log_K N \rceil - 1) * K^{\lceil \log_K N \rceil - 1} \quad (3)$$

2) *Hierarchical Model:*

$$BO_h = R + R * (K + 2K^2 + ... + (\lceil \log_K (N/R) \rceil - 1) * K^{\lceil \log_K (N/R) \rceil - 1} \quad (4)$$

Similarly, we can find *R* to minimize the overhead for the network with size *N*.

3) *P2P-based Model:* Each intelligent endpoint user agent can handle a part of media session locally. For example, several nearby IP phone user agents can construct a P2P topology to mix the media. Suppose each endpoint user has the capability to support *M* media sessions, the bandwidth overhead is,

$$BO_p = K + 2K^2 + ... + (\lceil \log_K N/M \rceil - 1) * K^{\lceil \log_K N/M \rceil - 1} \quad (5)$$

4) *P2P+Hierarchy Model:* The Bandwidth overhead is given by,

$$BO_{ph} = R + R * (K + 2K^2 + ... + (\lceil \log_K (N/(RM)) \rceil - 1) * K^{\lceil \log_K N/(RM) \rceil - 1} \quad (6)$$

VI. Simulation Results and Analysis

In this section, we evaluate the performance of the system that uses our proposed algorithms and protocols. We will first describe the experimental model and then report performance results.

A. Performance of Call Routing Overlay

1) Experimental Model

- **Network model:** For the performance evaluation of the call routing algorithms, we evaluate two types of network topologies: random and real networks; 1) Random network: we randomly generate a random topology with network size, 400. 2) Real network: we consider the *MCI* ISP backbone network with 19 nodes, which are interconnected by links with 100 *kbps* for voice signaling traffic. We assume each node is deployed with a call agent.
- **Traffic Model:** We assume that call requests form a *Poisson* process with rate [*0, 10*] per second.
- **Baseline System**: For the call routing algorithms, we consider following baseline systems $A \in \{SR, WR\}$, where *SR* denotes the static routing (the next hop is always chosen by shortest path selection) and *WR* denotes the weight random selection scheme (the next hop is calculated by the weight formula (2))

- **Performance Metrics:** For the routing algorithms, we consider the following performance metric - *Average Routing Delay (RD):* it is defined as the average call routing message delay in the system. The higher the *RD* value is, the worse is the performance.

2) Performance Results

In this section, we report results of our experiments along with our observations. Due to space limitations, we only present a limited number of cases here. However, we found that the conclusions we draw are generally held for many other cases we have evaluated.

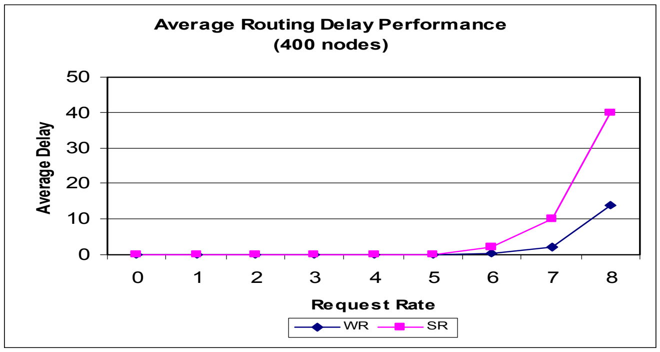


Figure 3: RD Performance of Routing Algorithms (network with 400 nodes)

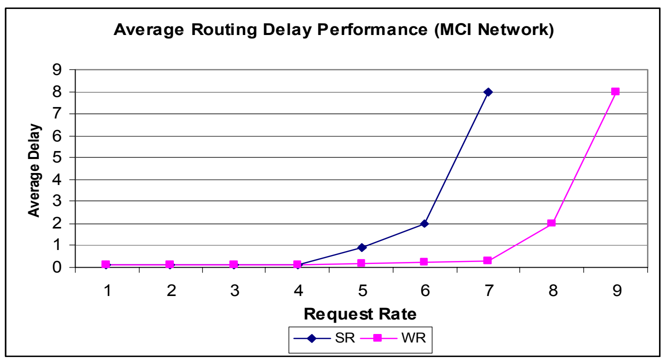


Figure 4: *RD* Performance of Routing Algorithms for Real Network Topology

Figures 3 and 4 show the sensitivity of *RD* in a random network of size 400 and real *MCI* backbone network, respectively. We see that *RD* is indeed sensitive to the different network sizes and call routing schemes. We have following observations.

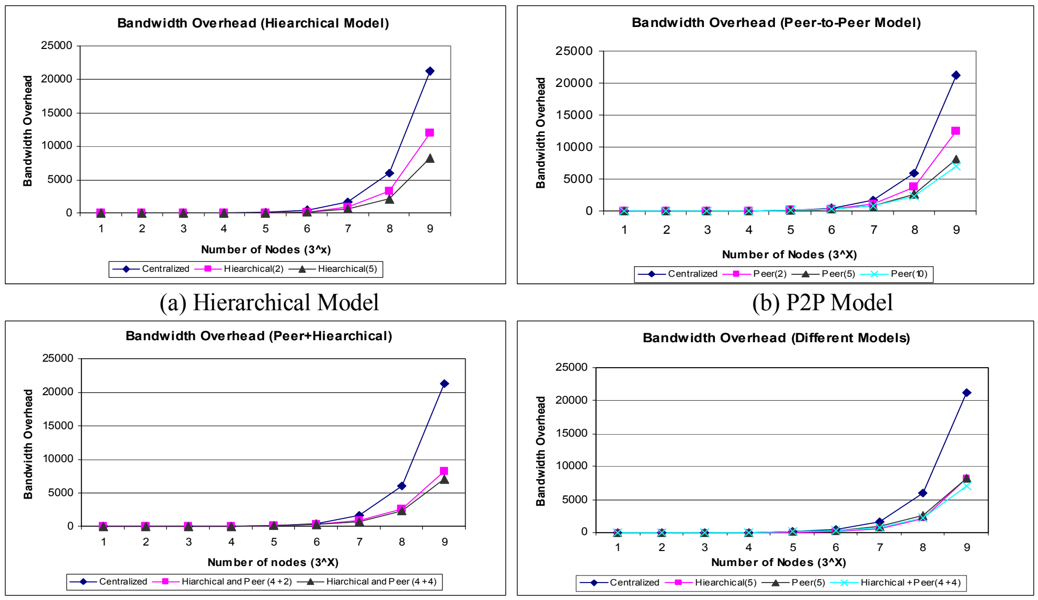
- *RD* of weighted random routing selection scheme performs better than the shortest path routing scheme consistently for different network sizes. The reason is that weighted random selection scheme relies on the fact that the high congestion paths are not being selected.
- *RD* is sensitive to the service request rate *R*. The value of *RD* increases as *R* increases. The reason is obvious: a large *R* implies that a large number of requests are sent to call agents increasing the processing time.

B. Performance of Endpoint Overlay

Figure 5 shows the performance of conference bandwidth overhead for different models. In this figure, Peer(#) represents the system where each cluster has # nodes, Hierarchical($) represents the system with $ regions, and Hierarchical+Peer($+#) represents the system with $ regions with each cluster having # nodes. We have following observations:

- The bandwidth overhead becomes larger with increase in network size. The reason is simple. The large network size increases the overall network path length, which causes the large bandwidth overhead.
- Given the network size, the bandwidth overhead for the centralized model gives the worst performance compared to other models. The P2P+hierarchy model achieves the best performance.

The reason can be explained. The P2P+hierarchy model makes network media traffic with shorter network paths. As local endpoint user agents contribute the possible media streaming capability, the P2P+hierarchy approach is a good solution to extend the system scalability for large conference feature.



(a) Hierarchical Model

(b) P2P Model

(c) P2P + Hierarchy Model

(d) All Models

Figure 5: Bandwidth Overhead for Conference Feature

VII. Related Work

In this section, we review research work in the area of P2P/ Grid computing and communication systems in the IP network related to our study.

Recent application level overlay networks, e.g., CAN [2], Chord [3] and Tapestry [4] are scalable, decentralized and self organizing systems. With distributed hash table (DHT) [10], nodes in these systems collectively contribute towards administration-free storage space and achieve efficient content lookup. Our approach is different from these systems due to the salient features of call routing. Moreover, as the communication signaling overlay is comparatively stable and our call routing protocols being much simpler, more efficient mechanisms can be realized for call routing. From the application perspective, Grid-based computing has drawn much attention in recent years [11]. Future network systems need to have certain machine-understandable semantics to make the intelligent peer actively and efficiently provide on-demand computing [12]. Building an intelligent social grid, semantic resource grid and knowledge grid are also becoming important research areas [13] [14]. In this sense, our work can be interesting as an intelligent grid application.

There is a lot of research work aiming to improve the availability of voice over IP services. H.323 suite of protocols is developed by ITU, which has been supported by several products [15]. Because of its inability to readily provide new services, SIP is becoming popular and currently standardized by IETF in RFC 2543 [6] [16]. As SIP takes the distributed control approach and supports MIME and URI, it can easily support combined services in the Internet and make the service delivery fast. In both SIP-based and H.323-based call control models, the call routing feature is performed by the controller (H.323 world) or SIP proxy (SIP world), which are just examples of the generic call agent in this paper, which mainly conducts the call routing task. The call routing problems have also been discussed by adopting inter-domain routing protocol (BGP) [17] [18]. Our approach is different from these studies by adopting the P2P-based overlay approach to construct intelligent grids at the application layer to efficiently support the communication service. In this sense, our approach is an extension of previous work from the framework of supporting large scale communication systems with enough intelligence to easily support existing features and extend it to new features.

Much work related to improving the voice quality from the network data plane is done. For example, Karam *et al.* [19] studied the delay and jitter analysis of voice traffic in the Internet by analyzing different scheduling algorithms. Chuah *et al.* [20] studied the statistical approach to make the bandwidth resource management more efficient. Our work differs from the above by mainly focusing on the network control plane.

VIII. Conclusions and Future Work

We have studied different issues regarding the provisioning of VoIP services in large-scale IP systems by adopting intelligent P2P/ Grid computing technologies. To the best of our knowledge, this is the first study that addresses issues in this field. In summary, our technologies include the following: 1) An intelligent P2P/ Grid based architecture, 2) P2P-based call routing overlay, 3) P2P-based endpoint overlay to efficiently support features. We conduct extensive performance evaluations on different architectures and algorithms. The evaluation results show that the load-aware weight-based call routing scheme achieves much better performance than the static selection scheme in terms of average call routing message delay. The experimental results also demonstrate that P2P+hierarchy model can efficiently support large conferences in terms of minimizing bandwidth overhead.

Our work has broad impacts. With a tremendous spurt in communication services demanded by Internet applications, traditional approaches to deliver communication services find it increasingly difficult to do so. Such services can be easily deployed with our approach presented in this paper. There are several directions to extend our study: 1) as the endpoint (IP phone) is a generic device, it can implement a broader spectrum of other services such as, personal computer virus detection and firewalls; 2) our Grid-based overlay design approach is generic to provide different services; our design approach can easily be adopted in other areas like, high performance instant messaging system etc, which will be part of our future work too.

References

[1] J. Xu, W.Y. Lee, "Sustaining Availability of Web Services under Distributed Denial of Service Attacks", accepted to *IEEE Transaction on Computers, special issue on Reliable Distributed Systems,* 52(2) (2003) 195-208.

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker "A Scalable Content Addressable Network*", Proceeding of ACM SigComm*, 31(4) (2001) 161-172.

[3] I. Stoica, "A Scalable P2P Lookup Service for Internet Applications", *Proceeding of ACM SigComm,* 31(4) (2001) 149-160.

[4] B. Y. Zhao, J. D. Kubiatowicz, "Tapestry: An Infrastructure for Fault-resilient Wide-are Location and Routing", *Technical Report* UCB/CSD-01-1141, University of Berkeley, 2001.

[5] I. Foster, C. Kesselman, "Grid Services for Distributed System Integration", *IEEE Computer*, 35(6) (2002) 37-46.

[6] M. Handley, H. Schulzrinne, "SIP: Session Initiation Protocol", *RFC 2543*, 1999.

[7] M. Macedonia**, "**Distributed file sharing: barbarians at the gates", *IEEE Computer*, 33(8) (2000) 99-101.

[8] S. Ratnasamy, M. Handley, R. Karp, S. Shenker, "Topologically Aware Overlay Construction and Server Selection", *Proceeding of IEEE INFOCOM*, 3(23-27) (2002) 1190-1199.

[9] R. G. Gallager, "Minimum Delay Routing Algorithms Using Distributed Computation", *IEEE Transaction on Communications*, 25(1) (1997) 73-85.

[10] M. Naor, U. Wieder, "A Simple Fault Tolerant Distributed Hash Table", *IEEE 2-th International Workshop on Peer-to-Peer Systems*, 2003, 88-97.

[11] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets**",** *Journal of Network and Computer Applications*, 23(1) (2001) 187-200.

[12] J. Kephart and D. Chess, "The Vision of Autonomic Computing", *IEEE Computer Magazine*, 36(1) (2003) 41-50.

[13] H. Zhuge, "Semantics, Resource and Grid", *editorial of the special issue of Future Generation Computer Systems*, 20(1) (2004) 1-5.

[14] H. Zhuge, "China's E-Science Knowledge Grid Environment", *IEEE Intelligent Systems*, 19(1) (2004) 13-17.

[15] ITU, "Registration, Admission and Status Signaling (Recommendation H.225/RAS)", *International Telecommunication Union (ITU)*, 1998.

[16] J. Rosenberg, "Distributed Algorithms and Protocols for Scalable Internet Telephony", *Ph.D Dissertation*, Dept. of Electrical Engineering, Columbia University, 2001.

[17] J. Glasmann, W. Kellerer, H. Muller, **"**Service development and deployment in H.323 and SIP**",** *6-th IEEE Symposium on Computer and Communication*, 3(5) (2001) 378 – 385.

[18] D. Hampton, D. Oran, "The IP Telephony Border Gateway Protocol (TBGP)", *Internet Draft*, draft-ietf-iptel-glp-tbgp-01.txt, work in progress.

[19] M. J. Karam, F. A. Tobagi, "Analysis of the Delay and Jitter of Voice Traffic Over the Internet", *Proceeding of IEEE INFOCOM*, 2(22) (2001) 824-833.

[20] C. N. Chuah, R. H. Katz, "Network Provisioning and Resource Management for IP Telephony", *Computer Science Dept., Report No. UCB/CSD-99-1061*, *UC at Berkeley*, August 99.

Wei Yu's bio: Wei Yu received his B.S (1992) from Nan Jing Technology University, M.S. (1995) from Tong Ji University, and Ph.D degree (1998) from Shanghai Jiao Tong University. All are in Electrical Engineering. Since 1999, he is a Ph.D candidate in the Dept. of Computer Science at Texas A&M University. Currently, he is working for Cisco Systems, Inc. His research interests include network security and distributed systems.

Sriram Chellappan's bio: Sriram Chellappan is a graduate student in the Dept. of Computer and Information Science at The Ohio-State University. His current research interests are in network security, distributed systems and wireless networks. He holds a Masters Degree in Electrical Engineering also from The Ohio-State University and a Bachelors degree in Instrumentation and Control Engineering from the University of Madras.

Dong Xuan's bio: Dong Xuan received his B.S. and M.S. degrees in Electronic Engineering from Shanghai Jiao Tong University (SJTU), China, in 1990 and 1993, and Ph.D degree in Computer Engineering from Texas A&M University in 2001. Currently, he is an assistant professor in the Department of Computer and Information Science, The Ohio State University. He was on the faculty of Electronic Engineering at SJTU from 1993 to 1997. In 1997, he worked as a visiting research scholar in the Department of Computer Science, City University of Hong Kong. From 1998 to 2001, he was a research assistant/associate in Real-Time Systems Group of the Department of Computer Science, Texas A&M University. His research interests include real-time computing and communications, network security and distributed systems.