Services spanning heterogeneous networks

Vijay K. Gurbani[1,2], Xian-He Sun[1]
vkg@{iit.edu, lucent.com}, sun@cs.iit.edu

1. Department of Computer Science
Illinois Institute of Technology
Chicago, Illinois

2. Lucent Technologies, Inc./Bell Laboratories
Wireless Networks Group
Naperville, Illinois

Abstract: **Computer networks exist to provide services to users. In domains where more then one network dominates and provides useful services, users will want to avail themselves of services on either of the networks. This paper looks at the convergence of two networks in the telecommunication domain: the Internet and the Public Switched Telephone Network and discusses techniques to access services between the two networks, including an architecture for realizing services which would not be possible if either of the networks was operating in isolation.**

Index Terms - **Services, PSTN, Internet, Telephony, XML, SIP, Instant Messaging, Presence**

I. Introduction

The intrinsic value of a computer network is measured by the services it provides to its users. As the number of networks increases, so do the chances that services residing on one network will need to be accessed by users on a different network. Nowhere is this more true than in telecommunication networks; currently, there are two major networks in place for telecommunications: the Public Switched Telephone Network (PSTN) and the Internet. Increasingly, these networks are converging [1], which necessitates access to services[1] residing in one of these networks from the other one.

A. Motivation

Consider, for instance, the following scenario regarding users from one network accessing services in the other one: Alice has an Internet telephony endpoint on her desk. She uses it to call the 1-800 number of her travel agency. The data and procedures associated with translating a 1-800 number to a valid routing number reside on the PSTN, not the Internet. From a service point of view, it would be advantageous to access the PSTN-resident data and procedures transparently from the Internet telephony infrastructure and route the call to the travel agent.

Bob, who works for Alice's travel agency is waiting for a call from Alice so he can give her directions to reach the travel agency. When Alice calls the agency, Bob provides directions to Alice and instructs his Internet-based personal digital assistant (PDA) to send him an instant message when Alice is within 500 meters of the travel agency. The PDA interacts with the PSTN and request Alice's location to be reported to it (Alice's location information is being tracked by the PSTN through her cellular phone). When Alice is within 500 meters of the travel agency, the PSTN sends an instant message to Bob's PDA, which intimates Bob of the impending event and causes Bob to get ready to receive Alice.

This example illustrates the need for services to span heterogeneous networks. In the first part of the example, the Internet call controller servicing Alice's request needs to interface with a service and data that reside in the PSTN. The PSTN, over the years, have evolved to host many user services with well-defined interfaces to access them. There isn't any reason why these services need to be re-created for the Internet; they should be leveraged transparently from the PSTN. Thus far, PSTN switches were the only entities accessing these services; but these services could just as well be accessed by an Internet entity.

The second part of this example demonstrates a service which would not be possible in isolation on either of the network. The PSTN cellular network already tracks its mobile users. This ability, coupled with the services provided by the Internet (like Instant Messaging), can be harnessed to create brand new services which leverage the deployed infrastructure of heterogeneous networks.

B. Contribution

This paper discusses the means and architectures to achieve such heterogeneous services, which term as *crossover* services [2]; i.e. the request for service starts in one network, but crosses over into the other network for service fulfillment. These types of services can be categorized in two: Internet-originated crossover services (where the request for the service originates in the Internet, but the service is executed on the PSTN), and PSTN-originated crossover services (where the request for the service originates in the PSTN, but the service itself is executed in the Internet).

The rest of this paper is organized as follows: sections II and III summarize Internet- and PSTN-originated crossover services, respectively. Section IV discusses related work in this area. Section V looks at some open issues in crossover services, and we provide our conclusion in the last section.

II. Internet-originated crossover services

In order to appreciate the need to access existing PSTN services from Internet endpoints, consider that the majority of services that end users are accustomed to -- Call Waiting, 800-number translation, etc. -- reside on the Intelligent Network (IN) [3], the service platform of the PSTN. Users on an

[1] In the context of the PSTN and the Internet, we define *service* as a value added functionality provided to the users by the network; thus Call Waiting and Caller ID are examples of PSTN services and email and ftp are examples of Internet services.

Internet telephony endpoint should be able to avail themselves of these services in the same transparent manner that they do when using a traditional PSTN handset. Call requests that originate on Internet telephony endpoints will need access to these services; the Internet telephony servers cannot provide these services natively in a uniform and scalable fashion. Thus, these call requests will cross into the PSTN domain for the application of IN services. Internet-originated services, thus, open up the world of existing and deployed IN services to Internet telephony end users.

There are three ways [2] to access IN services for the Internet telephony user: first, the easiest, albeit the most intrusive way would be to re-write all the existing PSTN services for the Internet environment. This is not feasible since it takes anywhere from 6 months to a year to get a PSTN service specified, implemented, tested, and deployed. This already assumes a stable service delivery infrastructure as it exists in the PSTN. Internet telephony being a new medium does not as yet have a well-specified services architecture that can be leveraged to deploy new services. The service architecture for Internet telephony is in the early stages of being proposed [4,5,6,7]. These factors make it extremely difficult, and in fact, undesirable, to replicate existing PSTN services from scratch in the Internet domain.

Another way to enable services in dissimilar networks is to use a language neutral service framework. Much like a Java program can run on multiple architectures simply by porting the Java Virtual Machine to each such architecture, one can envision a service-specific language that is architecture and platform neutral. Thus services programmed in such a language can easily be ported from the PSTN domain to the Internet domain. Unfortunately, such a platform-neutral solution does not exist; although early research proved that such a language was indeed feasible. In 1992, AT&T Bell Labs assigned researchers to study if such a solution could be formed. The result of this research was Application Oriented Parsing Language (AOPL) [8]. AOPL specifies a grammar and methodology that provides the service creators with platform neutral building blocks to create services. Services are written in a platform neutral language and can be compiled into the native language of the platform where the service is to be provided. While AOPL proved that this could indeed be done, industry interest in AOPL was simply not there to push it towards a standard. Thus efforts in it waned [9] as time progressed.

A final, and the preferred option is to devise a technique such that services running on PSTN can be transparently accessed from Internet endpoints. This preserves the (tested and) deployed service infrastructure in the PSTN, while at the same time, allowing transparent and scalable access to the service from the Internet. Service porting or re-writing is not necessary as the service can be accessed in a network agnostic manner.

Reference [10] discusses such a technique we term Call Model Mapping with State Sharing (CMM/SS) in detail. CMM/SS allows transparent access to services residing in one domain from another domain. CMM/SS, in a nutshell, consists of mapping the call model of a foreign domain to that of a local domain so that the foreign domain can access services resident in the local domain.

The CMM/SS technique depends on, and assumes the availability of a call model. A call model is a deterministic finite state machine (FSM). States in the FSM represent how far the call has progressed at any point in time. The current state, plus a set of input stimuli transition the FSM to the next state. In telecommunication signaling, these input stimuli consist of timers firing and arrival/departure of signaling messages resulting in the execution of significant events. Events cause transition into and out of a particular state.

Fortunately, call models are already an intrinsic part of telecommunication signaling protocols. For instance, the PSTN call model consists of 19 states and 35 input stimuli [3,11,12] and the Internet telephony signaling protocol, SIP [13, 14, 15], consists of 8 states and 20 input stimuli (Figures 5 and 7 of reference [13]). Call models, besides providing a uniform view of the call to all involved entities, also serve to synchronize these entities.

The CMM/SS technique has been successfully applied to the telecommunication domain in the context of accessing PSTN services from SIP endpoints [10,12,24]. A call model mapping has been established between the SIP protocol state machine and the PSTN call model. Furthermore, we have authored two pieces of software to demonstrate this mapping concept. First is a PSTN/IN call model which is a software layer written in C/C++, and the second is a proxy SIP server that has hooks to this PSTN/IN call layer.

In a SIP network with access to PSTN services, call requests from SIP endpoints are received by the SIP proxy, which intimates the PSTN/IN call layer of this event through a functional interface. The proxy passes parameters to the PSTN/IN call layer that include the caller's telephone number, the callee's telephone number and other pertinent information. The PSTN/IN call layer then steps through its states and triggers service queries to PSTN elements. These service queries are triggered using native PSTN protocols, thus insulating the SIP proxy from the PSTN details and insulating the PSTN service element from knowing that the call request actually originated on a non-PSTN endpoint. Figure 1 depicts a SIP proxy running an PSTN/IN call model and accessing PSTN/IN services transparently.

Once the call request has been thus serviced, control is returned back to the SIP proxy, which continues processing the call. The PSTN/IN call layer and the SIP proxy have to execute in lockstep since events can occur in either of the state machines to effect the other. For example, if the caller hangs up, the SIP proxy will get notified of this event first. It must now propagate this event to the PSTN/IN call layer so that it (the PSTN/IN call layer) can clean up any state associated with that call. Likewise, if the IN service logic needs to drop a call, it needs to propagate this to the SIP proxy, which will in turn send a SIP final response to the caller and clean state associated with the call.

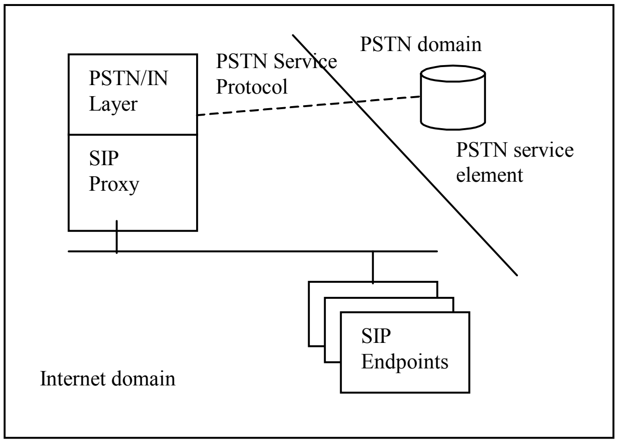

Figure 1: Internet-originated crossover services

Certain IN services can be readily accessed with the technique of CMM/SS from Internet endpoints. When Alice picks up her Internet-telephony phone and dials an 1-800 number, her service proxy will trigger a request to the PSTN service element to translate the 1-800 number to a valid routing number and route it by sending the call setup request to a gateway. All this happens transparently to Alice, as it should. On the PSTN side, the PSTN service element receives a request over a native protocol and simply services the request. It is oblivious to the fact that the request for service was issued by a SIP proxy and not a PSTN switch.

It should be noted that the services we have demonstrated through our implementation are confined to those executed during call setup and teardown. Most importantly, services which use the media are not directly applicable to a proxy-centric approach described in our implementation. This drawback is not due to any inherent weakness in CMM/SS; instead it is due mainly to the fact that Internet telephony disaggregates the media from signaling: a proxy only has access to the signaling information, not the media associated with the signaling. While this is generally a benefit, it proves to be a hindrance for executing services that depend on tones or utterances carried in the media stream. Future work may thus look into such services by focusing on platforms where a call controller also has access to the media stream. Also services which occur in the *middle* of a session in the PSTN (like pressing a flash hook to receive a new call while already engaged in a previous call) are for further study. It is not entirely clear that replicating the PSTN manner of dealing with such services is applicable in the Internet domain, where the endpoints are far more powerful than simple PSTN phones.

III. PSTN-ORIGINATED CROSSOVER SERVICES

The Internet has already become a ubiquitous part of our daily life; the telephone has been for an even longer time. The convergence of these two networks leads to innovative service ideas that are not possible in isolation on any one network. Consider for instance a basic Internet service, instant messaging. Also consider a basic PSTN service: making a voice call. An instant message is an asynchronous delivery mechanism; if the receiver is not present, the message is queued at the receiver's side until it is delivered. A phone call, on the other hand, is a synchronous event; if the receiver is not present when his or her phone rings, the service goes unfulfilled.

Now, consider how the convergence of the PSTN and the Internet engenders a brand-new service: sending an instant message to a phone subscriber informing him or her of the missed phone call and providing information on the caller (this assumes, of course, that the phone subscriber is physically at a different place than his/her phone is at). If the phone call was important enough, the receiver of the instant message can pro-actively contact the caller identified in the instant message.

This is but a simple example of PSTN-originated crossover services. However, it does demonstrate the potential for an architecture that would be general enough to provide this and other more complex services. Also note that in isolation, instant messaging or completing a phone call are just atomic services; but when combined as crossover services, their utility increases manifold than if they were simply operating alone.

PSTN-originated crossover services aim to export the states of a call occurring in the PSTN to an Internet endpoint for service execution. The hallmark of a PSTN-originated crossover service is that the service itself is executed on the Internet, just that the events to trigger that service occur on the PSTN.

Internet Call Waiting (ICW) [16,17] was the first attempt at a PSTN-originated crossover service. In this service, the PSTN kept track of the fact that a phone subscriber was utilizing the line to get on the Internet. If the subscriber subscribed to the PSTN Call-Waiting service and received a phone call while he or she was online, the resulting Call-Waiting tone would interrupt the Internet session. Thus most subscribers turned the Call-Waiting feature off before they went online, and would thus be unreachable for the duration of the Internet session.

The PSTN stored the knowledge that the subscriber was in an Internet session over the phone line. When the subscriber received a call to the phone line that was busy, the PSTN would use the Internet to route a session setup request to the subscriber's PC. A specialized server, running on the subscriber's PC would cause a popup to appear on the screen detailing the name and number of the caller as well as disposition options.

The subscriber could choose to "Accept" the incoming call, thus disrupting the Internet session. In this case, the specialized server running on the subscriber's PC would send a message to the PSTN to transfer the call to the subscriber's line, and immediately disconnect the modem connection thus causing the line to ring. Alternatively, the subscriber could choose to "Reject" the call or "Forward" it to an alternate number.

There are three criteria for a service to be considered a PSTN-originated crossover service:

1. *Subscription*: An Internet host subscribes to an

event of interest, *e*, in the PSTN,

2. *Action*: The PSTN, during its normal course of operations, undertakes certain actions which lead to the occurrence of the event *e*,
3. *Notification*: The PSTN notifies the Internet host of event *e*, and service itself is executed on the Internet. The service may be completely executed on the Internet, or the service execution may be shared between the two networks.

Based on the above conditions, a target architecture must support Internet hosts subscribing to events of interest occurring in the PSTN and the subsequent notification of the said event of interest by the PSTN to the concerned Internet host.

The architecture to realize PSTN-originated crossover services is deceptively simple, and in keeping with the Internet tradition, it distributes the intelligence to the edges. In fact, the entire PSTN is simply viewed as a SIP[2] endpoint to provide crossover services. Figure 2 presents the architecture and shows the PSTN domain on the left hand side of the diagram and the Internet domain on the right hand side. Note that the PSTN domain consists of both wireless and wireline components; thus cellular PSTN-originated crossover services are possible for cellular endpoints as well. The most important components of the architecture in Figure 2 is the PSTN extension. It is this component that allows us to abstract the PSTN entirely and view it as a SIP endpoint in the Internet domain. The PSTN extension, which we assume will be co-located with the PSTN service element for reasons of security, accepts subscriptions from the Internet hosts containing the event they are interested in getting a notification for (*subscription*). When the event of interest occurs in the PSTN (*action*), the PSTN notifies the Internet host of the event (*notification*). The Internet host, upon receiving the notification from the PSTN, can run any arbitrary service that is possible within the realm of crossover services. The service might be as simple as logging the event reported in the notification, or as complex as instructing the PSTN to take further action with the call (as in ICW).

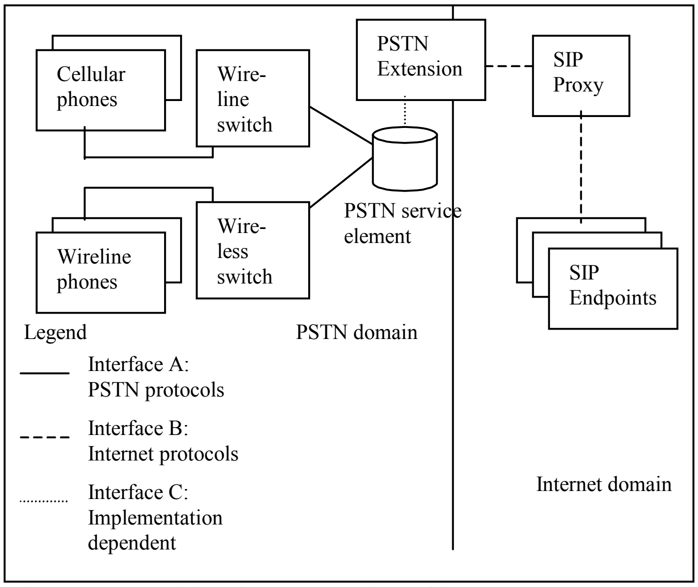

Figure 2: PSTN-originated crossover service architecture.

The PSTN extension straddles the two networks and acts as a protocol translator for each of the network thus insulating one from the other. It accepts subscription requests from Internet hosts and interfaces with the PSTN infrastructure to arrange for the PSTN switches to notify it when the event of interest occurs. Upon the receipt of such an event, the PSTN extension arranges for the event to be transported to the appropriate Internet host for service execution. The entities in each of the network are unaware that portions of the service are being fulfilled on a different network. The other entity of interest in Figure 2 is the proxy server. This is an access proxy belonging to the same autonomous system that owns the PSTN infrastructure. The proxy acts as a gatekeeper for the PSTN resources by authenticating and authorizing the subscription requests arriving from the Internet hosts.

As stated before, PSTN-originated crossover services aim to export the states of a call occurring in the PSTN to an Internet endpoint for service execution. Thus a means is needed to capture the state of the call on the PSTN and transfer it out to the Internet. The choice of using SIP as the protocol of choice judiciously pays off since a SIP request can contain an arbitrary payload; in case of PSTN-originated crossover services, the payload will consist of a XML encoding of the call state. The PSTN extension encodes the event of interest on the PSTN into a XML payload and using the SIP events extension [18], sends it out to the Internet host. Likewise, the Internet hosts also encode the event of interest in XML and use the SIP events extension to request the PSTN to monitor it. Work is underway [19] in the Internet Engineering Task Force (IETF) to standardize the XML encoding of the events and transporting them in a secure manner between the PSTN and the Internet hosts.

The architecture of Figure 2 makes possible a wide range of Internet-based services which can be triggered from PSTN events. These services include sending instant messages from the PSTN to Internet hosts, arranging for Internet hosts to subscribe to presence information of telephone users, and triggering services based on the location of cellular telephone users. We have implemented wireline portions of the architecture in figure 2; the implementation and the services realized through it are described in [23].

IV. Related Work

The IETF has standardized PINT [20] which involves IP hosts invoking certain telephony services. But the crucial difference is that in all cases of PINT services, a telephone session is established between two entities, both of which are on a homogeneous network, namely the PSTN. Our work, by

[2] We have chosen to use SIP as the Internet telephony protocol of choice here as well; it is powerful, extensible, and in a way, provides a far richer tool set in our problem domain since it is better tuned towards multi-media communications.

contrast, does not necessarily involve in a telephone session being established and thus it does not mandate that parties involved in a service be on a homogeneous network.

Gbaguidi et. al. [21] describe a platform for what they term as *hybrid* services; i.e. services that span the PSTN and Internet. However, the proposed hybrid service architecture is well suited for interactive forms of communications that require two or more end users at the same time. It is ill-suited for exporting the states of the PSTN towards IP hosts for richer services in the latter domain.

Rosenberg et. al. [4,22] discuss a component-based architecture for telephony services. In their architecture, a central entity (a controller) co-ordinates all these individual components to create a service. Their interaction with the PSTN is limited to the use of a telephony gateway; thus they do not consider PSTN events as a service stimulus. Like Gbaguidi et. al. [21], their architecture is well suited for services that require interactive communications.

V. Open Issues

Providing security in a closed system is a formidable task; providing it in a distributed system where interfaces cross administrative domains is harder still. In both of the crossover services, security is a concern on inter-domain interfaces. In Figure 1, the interface between the SIP proxy and the PSTN service element needs to be secure. Likewise, in Figure 2, interface B is of interest. It is at these interfaces that security can be compromised. Intra-domain interfaces can be assumed to be secure; for example, the interface A in Figure 2 lies completely in the rather closed and controlled PSTN domain and is secured by that very fact. But interfaces that cross over into the public Internet need to be encrypted and rendered secure least they compromise the overall security of the system by allowing crackers to mount denial of service attacks or allow them access to the subscriber-related private data stored in the PSTN.

VI. Conclusion

In this paper we have presented techniques and architectures to enable crossover services; i.e. services which span heterogeneous networks. As communications networks merge and become indispensable to our daily lives, the need to access services which span them will also increase. This paper presented a high level overview of the work we are conducting in the telecommunication domain to realize such services.

References

[1] I. Faynberg, L. Gabuzda, and H-L. Lu, "Converged Network and Services: Interworking IP and the PSTN," *John Wiley and Sons Publishers,* January, 2000.

[2] V. Gurbani, "Enabling Services through Protocol Interworking", *Ph.D. Research Proposal,* unpublished, Department of Computer Science, Illinois Institute of Technology, Chicago, Illinois, April 2002.

[3] I. Faynberg, L. Gabuzda, and N. Shah, "*The Intelligent Network Standards: Their Applications to Services,"* McGraw Hill Publishing Company, 1997.

[4] J. Rosenberg, "Distributed Algorithms and Protocols for Scalable Internet Telephony," *Ph.D. Thesis*, Graduate School of Arts and Sciences, Columbia University, New York, New York, 2001.

[5] J. Rosenberg, P. Mataga, and H. Schulzrinne, "An Application Server Component Architecture for SIP," *IETF Internet-Draft*, Expired September 2001.

[6] J. Dianda, V. Gurbani, and M. Jones, "SIP Services Architecture, "*Bell Labs Technical Journal*, Volume 7, Number 1, Jan-June 2002, in press, to be published by Wiley Periodicals, Inc.

[7] Ubiquity Corporation, "SIP Service Architecture", *White Paper*, <www.ubiquity.net/pdf/SIP-Service-Arch-WP1_0.pdf>, May 2001.

[8] L. Slutsman, H-L. Lu, M. Kaplan, and I. Faynberg, "The Application Oriented Parsing Language (AOPL) as a way to achieve platform-independent service creation environment," *Proceedings of the Third International Conference on Intelligence in Networks (ICIN94)*, October 11-13, 1994.

[9] H-L. Lu, unpublished, email between Hui-Lan Lu and Vijay Gurbani, October 2001.

[10] V. Gurbani, X-H. Sun, "Internet-originated crossover services", submitted for publication consideration to the *2nd IEEE International Symposium on Network Computing and Applications (NCA-03)*.

[11] ITU-T Q.1204: "Intelligent Network Distributed Functional Plane Architecture", *ITU-T Recommendation Q.1204*, 1993.

[12] V. Gurbani., F. Haerens, and V. Rastogi, "Interworking SIP and Intelligent Network (IN) Applications", *IETF Internet-Draft, Expires December 2002*, Work in Progress, <http://search.ietf.org/internet-drafts/draft-gurbani-sin-02.txt>

[13] J. Rosenberg, H. Schulzrinne, R. Sparks, J. Peterson, A. Johnston, G. Camarillo, M. Handley, and E. Schooler. "SIP: The Session Initiation Protocol," *IETF RFC 3261*, < http://www.ietf.org/rfc/rfc3261.txt?number=3261>, 2002.

[14] H. Sinnreich, and A. Johnston, "Internet Communications using SIP," *John Wiley & Son Publishers*, October 2001.

[15] G. Camarillo, J. Rosenberg, "SIP Demystified," *McGraw Hill Publishing Company*, August 2001.

[16] V. Gurbani, A. Brusilovsky, L. Conroy, J. Buller, and L. Slutsman, "PSTN Internet Notification (PIN) Proposed Architecture, Services, and Protocols," *Proceedings of the 2000 International Conference on Intelligent Networks (ICIN)*, Bordeaux, France.

[17] V. Gurbani, A. Brusilovsky, A. Jain, E. Gaussman, " A Proposal for Internet Call Waiting using SIP: An Implementation Report, *IETF Internet-Draft,* Expired July 1999, Work in Progress. <http://www.iit.edu/~gurbvij/I-D/draft-brusilovsky-icw-00.txt>.

[18] A. Roach, "Session Initiation Protocol (SIP)-specific Event Notification", *IETF RFC 3265*, June 2002, < http://www.ietf.org/rfc/rfc3265.txt?number=3265>

[19] V. Gurbani, A. Brusilovsky, I. Faynberg, H-L. Lu, M. Unmehopa, and K. Vemuri, "The SPIRITS Protocol", *IETF Internet-Draft*, Work in Progress, expires December 2002, <http://www.ietf.org/internet-drafts/draft-ietf-spirits-protocol-02.txt>

[20] S. Petrack, L. Conroy, "The PINT Service Protocol: Extensions to SIP and SDP for IP access to telephone call services", *IETF RFC 2848*, June 2000, <http://www.ietf.org/rfc/rfc2848.txt?number=2848>.

[21] C. Gbaguidi, J-P. Hubaux, G. Pacifici, and A. Tantawi, "Integration of Internet and Telecommunications: An Architecture for Hybrid Services," *IEEE Journal of Selected Areas in Communications*, September, 1999.

[22] J. Rosenberg, P. Mataga, and H. Schulzrinne, "An Application Server Component Architecture for SIP", *IETF Internet-Draft,* Work in Progress, expired September 2001.

[23] V. Gurbani, X-H. Sun., A. Brusilovsky, I.Faynberg, H-L. Lu, and M. Unmehopa, "Internet service execution for telephony events", accepted for publication in *Proceedings of the 8th International Conference on Intelligence in next generation Networks (ICIN 2003)*, March 31-April 3, 2003, Bordeaux, France.

[24] V. Gurbani and X-H. Sun, "Accessing telephony services from the Internet", submitted for publication consideration to the *Journal of Parallel and Distributed Computing Special Issue on Scalable Internet Services and Architecture*, September 2003.