Terminating Telephony Services on the Internet

Vijay K. Gurbani and Xian-He Sun, *Senior Member, IEEE*

***Abstract*—We propose a general purpose service architecture for realizing services which start in the Public Switched Telephone Network (PSTN) but terminate and execute on the Internet. We discuss the needs for such services, our early research efforts in this direction which lead to prototyping certain benchmark services, and the current state of work in this area. We demonstrate the feasibility of the architecture by focusing on services which involve wireline PSTN as well as the wireless aspects (2 G, 2.5 G) of the PSTN. Our methodology is attractive since it keeps each of the domains (PSTN and Internet) unaware as to where the service is executing with respect to which domain actually requested the service. Individual entities participating in the service do not have any knowledge that external entities from another domain also contributed in the execution and fulfillment of such services. Our approach, as embodied in the service architecture, is to leverage the best of the Internet protocols (SIP, XML, HTTP) and technologies (instant messaging, presence) to provide a general framework for personalized service specification and execution.**

***Index Terms*—HTTP, Internet, Public Switched Telephone Network (PSTN), services, SIP, wireless, wireline.**

I. Introduction

THERE ARE currently two ubiquitous networks in use: the Public Switched Telephone Network (PSTN) and the Internet. Whereas the former is a specialized network whose main objective is to transport voice-based media with low delay and a guaranteed quality of service, the latter is a general-purpose network which can transport arbitrary media, including voice, video, and data. Increasingly, these networks are merging [28] and the union of these networks has lead to open research questions on at least two planes: the transport plane (i.e., the protocols and procedures for digitizing and transporting voice as packets over an inherently best-effort delivery network) and the service plane (i.e., the protocols and procedures for enabling new services and accessing existing services between the networks).

The work presented here deals exclusively in the service[1] plane, and is part of an overall research approach for enabling what we call crossover services; i.e., services where the intelligence to execute them is distributed in multiple network domains. Furthermore, a request to start a crossover service originates in one domain (PSTN or Internet) but terminates in another domain [1]. There are two types of crossover services: those that originate on the Internet and terminate on the PSTN (termed as Internet-originated crossover services), and those that originate on the PSTN and terminate on the Internet (termed as PSTN-originated crossover services). The work presented here pertains to the latter category, i.e., the PSTN is the network on which the request for a call is initially received, but the services for the session are provided on the Internet.

A. Motivation

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, the following scenario: Bob arrives to work only to discover that his cellular phone's battery is close to losing power. Bob is expecting an important call from his wife Alice, and he is not planning to be at his desk all the time. He would really like to be notified on his portable personal digital assistant (PDA) when Alice attempts to call his cell phone so that he can break out of whatever he is doing and return Alice's call. Furthermore, Bob is also expecting to have an important meeting with the vice-president of his company, who is flying in to meet him. In order to prepare for his meeting, Bob would like the network to send him an Instant Message (IM) when the vice-president of the company arrives at the airport.

Clearly, the services Bob expects are not simple; the complexity arises because they do not reside in the same network or use homogeneous protocols. What Bob would like to do when Alice calls him is to have the PSTN notify him on his PDA, which may be on the wireless Internet. Also, when the vice-president of the company arrives at the airport, Bob expects the cellular network to determine this event (presumably through the registration information of the vice-president's cellular phone) and send an IM to him. As can be observed, there is a strong need to tie services across the two networks—PSTN and the Internet—in a transparent and standardized manner.

This example demonstrates the potential for an architecture that would be general enough to provide this and other more complex crossover 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.

B. Overview of PSTN-Originated Crossover Services

PSTN-originated crossover services originate in the PSTN, but at a later time, cross over into the Internet for subsequent service fulfillment. In such services, both the networks—PSTN and Internet—are involved as follows: an Internet host informs

Manuscript received May 20, 2002; revised September 27, 2002, and July 16, 2003; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor H. Schulzrinne.

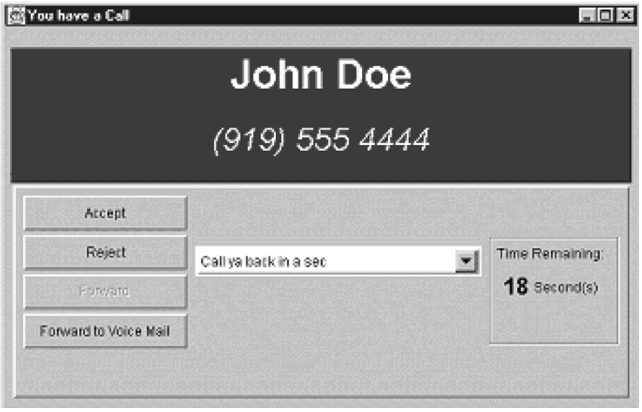
V. K. Gurbani is with the Illinois Institute of Technology, Chicago, IL 60616 USA, and also with Lucent Technologies, Inc., Naperville, IL 60566 USA (e-mail: vkg@iit.edu; vkg@lucent.com).

X.-H. Sun is with the Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: sun@iit.edu).

Digital Object Identifier 10.1109/TNET.2004.833145

[1] We define the term *service* as a value-added functionality provided by a network to its users. Thus, transporting voice and Call-Waiting are services provided by the PSTN, just as email and Web are services provided by the Internet.

1063-6692/04$20.00 © 2004 IEEE


You have a Call

John Doe

(919) 555 4444

Call ya back in a sec

Time Remaining: 18 Second(s)

Fig. 1. ICW screen popup.

the PSTN that it is interested in the occurrence of certain events, for instance, the event might be an attempt to call a certain PSTN number. When the said event occurs, the PSTN takes a snapshot of the state of the call and transfers this to the Internet host. The latter entity can execute arbitrary services upon the receipt of the notification. Thus, the state of the service is distributed across the two domains and some form of synchronization and a protocol is required to transfer the state of the service from the PSTN to the Internet for execution. We will revisit this topic in much detail in Section III.

The idea for PSTN-originated crossover services was first suggested as an outgrowth of a service which actually predates the idea itself. Internet Call Waiting (ICW) was the prototypical PSTN-originated crossover service and is described in [2], [35]. ICW 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. When the PSTN received a call destined to the phone line that was thus busy, it 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 (see Fig. 1).

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.

C. Genealogy and Early Work

The idea for PSTN-originated crossover services was first articulated in [3] and a preliminary architecture based on it was subsequently presented at the 44th IETF [4]. The architecture was ratified [36] and influenced by the ongoing ICW work; however, since many of the protocols that would be used in PSTN-originated crossover services were in mid-to-late stages of specification and development, none of the ICW implementations interoperated across vendor boundaries [35]. In 1999, the IETF sanctioned an official working group called SPIRITS [30] to enquire how services supported in the Internet can be started from the PSTN in a standardized manner. The resulting SPIRITS architecture is slightly different than ours, which faithfully follows and builds upon the architecture first presented in [4].

We are active contributors to the work progressing in the SPIRITS working group [5]–[7] and leverage our contributions in the working group to further refine and implement our architecture. For the remaining of this paper, we will assume that SPIRITS is synonymous with PSTN-originated crossover services, and the work we are doing in SPIRITS is equally applicable to PSTN-originated crossover services. The places where the SPIRITS architecture and our architecture diverge will be pointed out, but functionally, this diversion has no bearing on the services architecture.

D. Contributions

A key requirement of PSTN-originated crossover services will be third-party programmability of such services. Arguably, the service creation framework for the world wide web (WWW) infrastructure has thrived since it enables third parties to provide value-added services over a common transport, namely IP. The most important factor for the success of WWW services has been a common lingua franca (HTTP/HTML) and an extensive service creation toolset (Web CGI, Active Server Pages, Java scripts, servlets, SOAP, WDSL, UDDI, etc.).

Telephony, on the other hand, has traditionally been an environment where the inner workings of the protocols and services, while not entirely secret, were not subject to as much public access and scrutiny as Internet protocols have been. We believe that the web model of allowing open, well-defined protocols needs to be replicated for PSTN-originated crossover services. To that extent, one of the contributions of this work is an open, extensible architecture for crossover services based on standard protocols to help third parties in developing such services.

Ref. [1] establishes a taxonomy of PSTN-originated services. Taxonomizing such services is important so that implementers can quickly identify various techniques for rapid implementation. Thus, a second contribution of this work is the taxonomy of PSTN-originated services, which will build upon the work in [1].

Finally, PSTN-originated crossover service architecture resembles a *distributed software architecture*, as described in [8]. Such architectures employ distributed middleware (CORBA, RMI) to design systems. However, we eschew these middleware technologies in favor of industry standard signaling protocols for call control and data/state transfer. Services are best executed when the service execution platform has unfettered access to the signaling information; APIs tend shield the programmer from the details of the signaling protocol. Thus, a final contribution of this work is to establish our use of Session Initiation Protocol (SIP) [10] as a distributed middleware component for PSTN-originated crossover services.

The rest of the paper is organized as follows: Section II provides some background on service execution in the PSTN as well as the Internet domain and introduces the relevant actors for those domains. Section III describes our PSTN-originated crossover service architecture. Section IV shows examples of such services, including call flows and message layouts. Section V establishes a taxonomy of PSTN-originated crossover

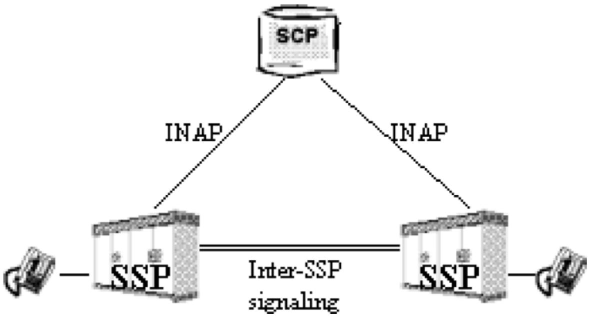


Fig. 2. Simplified PSTN/IN architecture.

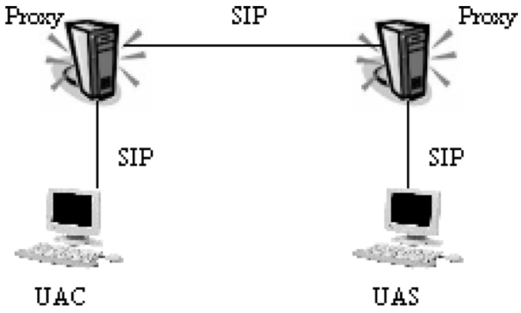


Fig. 3. Simplified SIP network.

services. Section VI discusses an important aspects of crossover services: security. Section VII compares our work to the existing literature on service architectures, and finally, we provide concluding remarks on our approach.

II. Background on Service Execution

A. Service Execution in the PSTN

The Intelligent Network (IN) is used in the PSTN to provide services such as 800-number translation, pre-paid calling, etc. It is described in detail in Faynberg *et al.* [15]; here we provide a very rudimentary overview to introduce the main entities and to aid the reader who may not be familiar with the concepts.

Fig. 2 shows a simplified IN architecture in which the telephone switches called Service Switching Points (SSP) are connected via a packet network called Signaling System 7 (SS7) to a general purpose computer called a Service Control Points (SCP). This leads to a clean separation of components since call control and service switching functions are performed at the SSP, while the service control (and data) functions are hosted by the SCP.

An SCP, along with other peripherals (like a media server for prompting and digit collection or voice recognition) provide services to PSTN subscribers. SSPs run a call model called Basic Call State Model (BCSM) when handling a call. A call model is basically a directed graph which accurately and concisely reflects the current state of a call at any given point in time (it is used to synchronize the many distributed entities that may participate in a call). Call models consist of states called Points In Call (PIC) and transitions between states. Inter-state transitions pass through elements called Detection Points (DP). DPs house one or more triggers. Every trigger has a firing criteria associated with it. When a trigger is armed (made active), and its associated firing criteria are satisfied, it fires. When a trigger fires, a message is formatted with call state information and transmitted by the SSP to the SCP. Further call processing may be suspended at the SSP until the SCP returns a response. When the SCP gets a request for instructions, it can reply with a single response, such as simple number translation augmented by criteria like time of day or day of week, or, in turn, get into a complex dialog with the SSP which may involve playing or recording voice announcements and collecting digits. The resulting protocol as well as the BCSM is standardized by the ITU-T and is known as the Intelligent Network Application Protocol (INAP) [15].

The term PSTN here represents both the wireline and wireless aspects of the switched network. Specifically, the current wireless services infrastructure (2 G, 2.5 G) is heavily influenced by the concepts of IN discussed above and is well integrated in the PSTN [21], [22]. Much like IN, Wireless IN (WIN) is based on an architecture that separates call processing from enhanced feature functionality. The wireless services infrastructure uses the same set of IN components used by their wireline counterparts, including the SCP. An important difference in wireless networks is that there are many events generated outside the context of establishing a call; for instance, turning on a mobile phone results in a registration event at the network and roaming in a wireless network generates location update events.

B. Service Execution in the Internet

The service architecture for Internet telephony is still evolving [8], [14], [16], [18]. The architecture of the PSTN has been characterized by a centralized control; the network core (in the form of the SSP, the SCP, etc.) asserts control over the signaling, media, and services being provided to the end points. Architectures for the Internet, on the other hand, tend to follow the opposite path; i.e., the network core is fairly simple while the intelligence is distributed to the end points. A service architecture for Internet telephony is therefore no exception, and the lack of central control makes it a complex problem to tackle.

We provide a brief overview of executing services in the Internet using the Session Initiation Protocol. SIP is described more completely in [10], [19], [20]; here we simply provide a very rudimentary overview to introduce the actors and to aid the reader who may not be familiar with the concepts.

SIP is an ASCII-based protocol used to initiate, maintain, modify, and terminate multimedia sessions. It shares its ancestry with other ASCII-based protocols from the IETF, including SMTP and HTTP, on which it is largely based. A SIP network is depicted in Fig. 3 and includes User Agents (UA) that resides on the periphery of the network. There are 2 types of UAs: User Agent Clients (UAC) which make requests for establishing sessions, and User Agent Servers (UAS) which accept these requests and issue responses. SIP, like HTTP, is a transaction-based protocol, where a transaction consists of a request and some responses. In the core of a SIP network reside network-based servers, the most important one for our discussion being a SIP proxy server. The main task of a SIP proxy server is to route requests from UAC to UAS based on many factors, including local registration information, DNS, SIP CGI, and SIP CPL [17]. Additionally, SIP proxies can also be used to provide other services, such as authentication of incoming requests and authorization of protected resources identified in the request.

Services in the SIP network can be provided at multiple places. Since SIP endpoints are far more capable in terms of functionality than their PSTN counterparts, they can actively participate in a service. Thus, originating side services can be provided by the UAC and the terminating side ones by the UAS. Network-resident SIP entities can also provide services which involve media as well as services based purely on signaling.

In direct contrast with the PSTN, which uses INAP for service execution and another SS7 protocol for inter-switch signaling, and yet another protocol for switch to endpoint signaling, a distinguishing factor of SIP is that the protocol used for signaling and service execution is the same. In our work, we leverage this distinction by abstracting the entire PSTN as a SIP endpoint and using SIP signaling to deliver service requests from the PSTN to the Internet.

III. PSTN-Originated Crossover Service Architecture

A. Choosing a Protocol

As is the case with any distributed system, a protocol is required to synchronize the attendant entities for deterministic behavior. We list the properties that are desirable in such a protocol and analyze three signaling protocols—Bearer Independent Call Control (BICC), H.323, and SIP—that can serve as candidate protocols for our architecture.

A PSTN-originated crossover service occurs when the PSTN performs an event of interest to an Internet host. When the event of interest occurs, the PSTN must take a snapshot of the call and transfer it to the Internet host for service execution; and depending on the service, the PSTN may actually await further instructions from the Internet host. Thus, the first property of our target protocol is a simple transactional, request-response driven signaling that has proved durable on the Internet (witness the success of HTTP, FTP, etc.). A request-response property in the target protocol will aid in synchronizing the entities on the PSTN and IP network by allowing the PSTN to temporarily suspend call processing until the Internet host returned further instructions. The second property of a target protocol should include the ability to carry arbitrary descriptive elements between the two networks. This will enable the Internet host to inform the PSTN of events of interest, and conversely, allow the PSTN to take a snapshot of a call in progress and intimate the Internet host of it. The third, and final property of a target protocol is support of a flexible naming scheme. Resources in the PSTN are generally identified by numbers, but in the IP network, resources can be identified using a much richer vocabulary which includes names, numbers, domains, etc. We now apply these three properties to the set of signaling protocols we chose.

1) BICC: ITU-T's BICC [31] is an adaptation of an existing circuit telephony protocol for the support of telephony services independent of bearer technology and signaling message transport technology. BICC can be used as a network-to-network signaling protocol for the circuit-based PSTN as well as ATM or IP core networks. The advantage of BICC is its rich telephony heritage and the ensuing ease of delivering existing PSTN services to endpoints in a different network. However, BICC's PSTN legacy does not render it as a good fit for simple request-response transactions; nor is BICC an end user protocol, thus criteria such as identifying resources using descriptive names and domains does not apply. And finally, while BICC has good capability negotiation procedures, it does not have support for carrying arbitrary descriptive elements as part of its signaling.

2) H.323: ITU-T's H.323 [32] is an umbrella protocol for establishing multi-media sessions over a local area network. H.323 includes associated protocols such as H.225.0 (basic call control signaling), and H.245 (media stream manipulation, conference control and flow control). Compared to BICC, H.323 does have a much more flexible naming scheme through its support for aliases (including H323 URL) and zones (collection of terminals and gateways managed by a single gatekeeper). However, it does not possess the primitives to transport arbitrary descriptive elements during signaling.

3) SIP: SIP was described in Section II-B, here we simply observe how it weighs in against our established criterion for a target protocol. SIP, following other IETF protocols, is a transactional protocol with a simple request-reply nature. SIP has built-in support for carrying arbitrary descriptive elements during signaling using the IETF Multipurpose Internet Mail Extensions (MIME). MIME allows communicating entities to exchange any arbitrary data on the Internet; inter-operability is provided by registering new MIME types in a global registry. And finally, like H.323, SIP has extensive support for a flexible naming scheme in the form of a SIP URL.

Based on the desired properties of a target protocol, we chose SIP as the protocol of choice. In a sense, Internet telephony protocols like SIP provide a richer set of tools to work from in our problem domain since they are already better tuned toward multi-media communications. SIP also possesses built-in support for asynchronous event notification [23] and enables services like presence [24] and instant messaging [25] that we view as vital components of crossover services.

B. Architecture

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

1) *Subscription*: An Internet host subscribes to an event of interest in the PSTN,
2) *Action*: The PSTN, during its normal course of operations, undertakes certain actions which lead to the occurrence of the event,
3) *Notification*: The PSTN notifies the Internet host of event and service itself is executed on the Internet. Depending on the taxonomy of the service, it may be completely executed on the Internet, or the service execution may be shared between the two networks.

A target architecture must thus 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. Given the background in the last two sections, we now propose our architecture for realizing PSTN-originated crossover services that meets the three conditions outlined above. The architecture is deceptively simple,

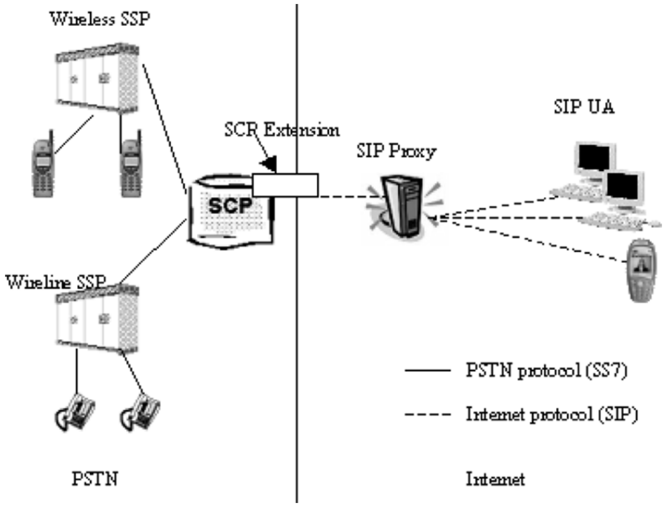


Fig. 4. PSTN-originated crossover services architecture.

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 UA to provide crossover services.

Fig. 4 shows the PSTN domain on the left hand side of the diagram and the Internet domain on the right hand side. As noted in Section II-A, the PSTN domain consists of both wireless and wireline components. Certain events in the wireless network—like mobile phone registration and mobile phone movements—do not have a counterpart a wireline network. But these distinctions can, in fact, be harnessed to provide powerful crossover services. For example, when a wireless user turns his or her phone on, a registration event is generated which can be propagated to an Internet host for executing presence based services. Likewise, when a mobile user enters a pre-defined geographic zone, a location event is generated which can also be propagated to an Internet host to deliver specific geo-location services. Our proposed architecture is thus transparently able to capture the actions that happen in wireless networks as well and exploit these for subsequent crossover services.

Fig. 4 shows the PSTN domain on the left hand side of the diagram and the Internet domain on the right hand side. As noted in Section II-A, the PSTN domain consists of both wireless and wireline components. Certain events in the wireless network—like mobile phone registration and mobile phone movements—do not have a counterpart a wireline network. But these distinctions can, in fact, be harnessed to provide powerful crossover services. For example, when a wireless user turns his or her phone on, a registration event is generated which can be propagated to an Internet host for executing presence based services. Likewise, when a mobile user enters a pre-defined geographic zone, a location event is generated which can also be propagated to an Internet host to deliver specific geo-location services. Our proposed architecture is thus transparently able to capture the actions that happen in wireless networks as well and exploit these for subsequent crossover services.

TABLE I
SET OF VALUES FOR E_v RELATED TO A PHONE CALL (WIRELINE OR CELLULAR)

Detection Point	Comment
OAA	Caller is authorized to make a call
AI	Dialed digits have been analyzed
CI	All dialed digits have been collected
OA	Callee has answered the call
OTS	Callee is being alerted
TANS	Callee has picked up the phone
ONA	Caller's BCSM timed out waiting for the callee to answer
RSF	Caller's BCSM is unable to send signaling messages on selected route
TNA	Callee was alerted, but did not answer within a predefined time interval
OCB	Callee is busy
OMC	Caller's BCSM detected mid-call activity ('hook flash', for instance)
TMC	Callee's BCSM detected mid-call activity ('hook flash', for instance)
OAD	Caller abandoned the call (i.e. hung up) before signaling was complete
OD	Caller disconnected the phone (i.e. normal hang up)
TAB	Callee abandoned the call (i.e. hung up) before signaling was complete
TD	Callee disconnected the phone (i.e. normal hang up)
TAA	Callee can receive an incoming call
FSA	Callee is not busy, and a line or trunk is available to reach the callee
TB	Callee is busy

Not all subscriptions will involve a DP; specifically, wireless events such updating the location of a mobile phone or registering a mobile phone do not involve establishing a call. When a subscription for a noncall related event presents itself to the PSTN, the address of record is extracted from the subscription, and a mark is set in the user profile of the subscriber identified by the address of record. Subsequent actions that correspond to the mark set in the subscriber's user profile will elicit a notification from the PSTN to the Internet host.

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 behaving like an SCP and instructing the PSTN to take further action with the call. The other entity of interest in Fig. 4 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.

C. Encapsulating Information Elements

In order to send subscriptions from the Internet host (and notifications from the PSTN) in a standardized manner, we use XML to carry tuples S and N from the Internet to the PSTN, and from the PSTN to the Internet, respectively. An Internet host subscribes to an event of interest represented by a finite tuple $S = (e_v, e_m, e_v^1, e_v^2, \ldots, e_v^n)$, with $\mathrm{n} \geq 1$, where:

e_v: The event that is being subscribed to. For events generated as a result of a phone call on the wireline or cellular network, the set of valid values for e_v are given in Table I. The set of events in the cellular network not related to a phone call are depicted in Table II.

TABLE II
SET OF VALUES FOR E_v RELATED CELLULAR NONCALL EVENTS

Event	Comment
LUS	Location update in home area
LUA	Location update while roaming outside the home area
IA	Mobile registration
MSID	Mobile de-registration (mobile initiated)
NID	Mobile de-registration (network initiated)
SUP	Supplementary services

e_m: The mode of the event; $= e_m = \{notify,\ request\}$. A mode of $notify$ requires the PSTN to simply notify an Internet host of the event. A mode of $request$ requires that the PSTN temporarily suspend its processing and await instructions from the Internet host on how to proceed further.

$e_v^1, \ldots, e_v^n$: Additional parameters relevant to e_v. For example, in most cases, one of the parameters sent during subscription will be a phone number about which the Internet host seeks notifications for. Any PSTN action that leads to the execution of e_v on that phone number will be of interest to the Internet host. The notification tuple is represented by $N = (e_v, e_v^1, e_v^2, \ldots, e_v^n)$, with $\mathrm{n} \geq 0$. Note that N does not contain the component e_m, and any additional information besides e_v is optional.

Our decision to use SIP as the protocol of choice pays off here since SIP can carry arbitrary bodies (defined by a MIME type in a SIP header). Delivering tuples S and N as XML-encapsulated SIP payloads yields a descriptive, extensible and standards based codification scheme. Considerable work has been conducted to identify and categorize the DPs (and their relevant parameters), noncall related events, and defining XML schemas for S and N [5], [7], [26], [27].

D. Usage Model

With a protocol chosen and an architecture defined, we now present a usage model to help understand the various players involved in the execution of a crossover service. There are three parties of interest in a PSTN-originated crossover service: the PSTN service provider, the Internet service provider, and the end user of the service. The PSTN service provider owns and/or operates the PSTN network on which events are generated. The end user is the party in the IP domain which requests the PSTN service provider to send it events of interest for service execution. Finally, the Internet service provider is the party that provides the IP transport to the end user. The PSTN service provider and Internet service provider can belong to the same organization, but they do not have to. As a general rule, we will assume that they are not part of the same organization.

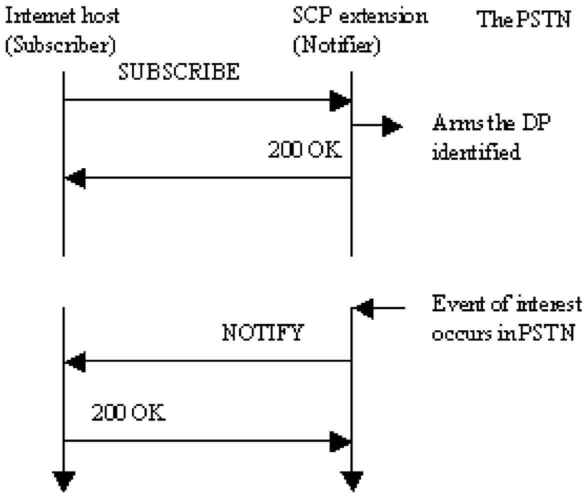


Fig. 5. Notification of incoming call.

In order to use PSTN-originated crossover services, we envision a specialized UA will be made available to end users by the PSTN service provider or a third-party working with the service provider. The specialized UA, in addition to supporting the base SIP functionality [10], will also support the SIP extension for asynchronous event notification [23], the SIP extensions for instant messaging and presence [24], and the extensions to enable it to understand and interpret tuples S and N discussed in Section III-C. The specialized UA will also be pre-configured with the address of a SIP proxy in the domain of the PSTN service provider which will be contacted for all PSTN-originated crossover services. Furthermore, it is not expected that the end user will be conversant with XML in order to formulate event of interest S or interpret the notification N. Rather, the PSTN service provider will codify the events it supports in a GUI to make it easier for the end user to choose events of interest. The specialized UA will construct the appropriate XML based on the user selection and send it to the pre-configured SIP proxy.

IV. EXAMPLES

We now present two crossover services which demonstrate the technique we have described thus far. The first example entails a crossover service that involves notifications of the selected event occurring on a certain PSTN line. The second example demonstrates the PSTN Short Message Service (SMS) as a crossover service as it migrates to and is executed on the Internet.

A. Crossover Service: Incoming Call Announcement

The service scenario is thus: a user at work wishes to get intimated whenever someone calls her home phone. She is possibly expecting an important call, the arrival of which she would like to know instantaneously, or may be simply generating a real-time log of calls to her home phone. From her Internet host, she subscribes to the PSTN for a specific DP which will get fired whenever an incoming call is destined to her home line. When the event of interest occurs in the PSTN, the SCP sends a notification to her Internet host.

The call flow is reproduced in Fig. 5, and the pertinent SIP messages are reproduced after the figure. Note that only the relevant entities are depicted in Fig. 5; for instance, while the SIP access proxy from Fig. 4 is employed to authenticate the user and proxy the messages, we have dispensed with it for reasons of terseness. Of interest to us in the SUBSCRIBE request that is sent from the Internet host is the body of the request (relevant headers shown, many omitted):

```
SUBSCRIBE sip:myprovider.com SIP/2.0
From: <sip:vkg@lucent.com>;tag = 8177 –
afd – 991
To: <sip:16302240216@myprovider.com>
Event: spirits.INDPs
Allow-Events: spirits.INDPs,
spirits.user-prof
Accept: application/spirits-event
Content-Type: application/spirits-
event
Content-Length: 221
<spirits-event>
<DP INDPs=TAA/>
  <Mode>Notify</Mode>
  <Termination_Attempt_Authorized>
    <CallingPartySubaddress>6302240216
    </CallingPartySubaddress>
  </Termination_Attempt_Authorized>
</spirits-event>
```

The body of the SUBSCRIBE request contains an XML formatted payload, which in this particular case, identifies the DP that it wants to subscribe to (TAA—Termination Attempt Authorization DP; this DP is triggered in the T_BCSM on attempts to complete a call on a particular phone line) and parameters associated with the DP. The TAA DP is defined with one mandatory parameter—calling party's phone number, encoded by the XML element *CallingPartySubaddress*. Based on the information in the SUBSCRIBE request, the PSTN arms the DP, and when a phone call attempts to complete on the line identified by the *CallingPartySubaddress*, a notification is sent to the Internet host. The notification request travels from the PSTN to the Internet host:

```
NOTIFY sip:vkg@host.lucent.com SIP/2.0
From:
<sip:16302240216@myprovider.com>;tag =
SA – 6302240216
To: <sip:vkg@lucent.com>;tag = 8177 – afd –
991
Subscription-State: termi-
nated;reason=fired
  Accept: application/spirits-event
Event: spirits.INDPs
Allow-Events: spirits.INDPs,
spirits.user-prof
Content-Type: application/spirits-
event
Content-Length: 263
<spirits-event>
<DP INDPs=TAA/>
  <Termination_Attempt_Authorized>
    <CallingPartySubaddress>16302240216
    </CallingPartySubaddress>
    <CalledPartySubaddress>13125675000
    </CalledPartySubaddress>
  </Termination_Attempt_Authorized>
</spirits-event>
```

The body of this NOTIFY request contains the DP that was fired (TAA) and any associated parameters. In this example, two parameters are passed from the PSTN to the Internet host: the line number that was being monitored for events (identified by the *CallingPartySubaddress* element) and the number of the party that attempted to place a call to that line (identified by the *CalledPartySubaddress* element).

The notification that goes from the PSTN to the Internet host has all the elements required for a call announcement service. The Internet host can subsequently alert the user by popping up a window with the relevant information. It is also entirely possible to send an instant message with more detailed information to the Internet host. This can be accomplished by the PSTN sending a special request in SIP (the MESSAGE extension [25]) used specifically for instant messages.

Finally, note that the SUBSCRIBE and NOTIFY contain crossover service related MIME types and several SIP-specific headers. Interested readers are directed to [7] for crossover service relevant headers and MIME types; and to [10], [23] for SIP relevant headers and event notification headers. As a quick summary, [7] globally registers the tokens "spirits.INDPs" and "spirits.user-prof" with IANA (Internet Assigned Numbers Authority). These tokens aid in subscribing to call-related and noncall related events, respectively. It also registers with IANA a new MIME type "application/spirits-event" to describe the contents of XML-encapsulated SIP requests.

B. Crossover Service: Turning Short Message Service to an Instant Message

Short Message Service (SMS) is a wireless service for sending short text messages to mobile phones. SMS messages do not require mobile phones to be powered on; if a recipient's phone is not powered on, the message can be buffered in the PSTN until such time that it can be delivered to the recipient.

We can leverage SMS as a crossover service by electing to receive a SMS as an instant message in cases where the user's (the recipient) wireless phone is powered off, or is otherwise unavailable or not in close physical proximity to the user. In such a scenario, the user will start a UA on his Internet host. The UA subsequently subscribes to the wireless network infrastructure to get notified when a SMS arrives for the mobile user:

```
SUBSCRIBE sip:myprovider.com SIP/2.0
From: <sip:vkg@lucent.com>;tag = 8177 –
afd – 991
To: <sip:16 302 240 216@myprovider.com>
Event: spirits.user-prof
Allow-Events: spirits.INDPs,
spirits.user-prof
Accept: application/spirits-event
Content-Type: application/spirits-
event
Content-Length: 139
```

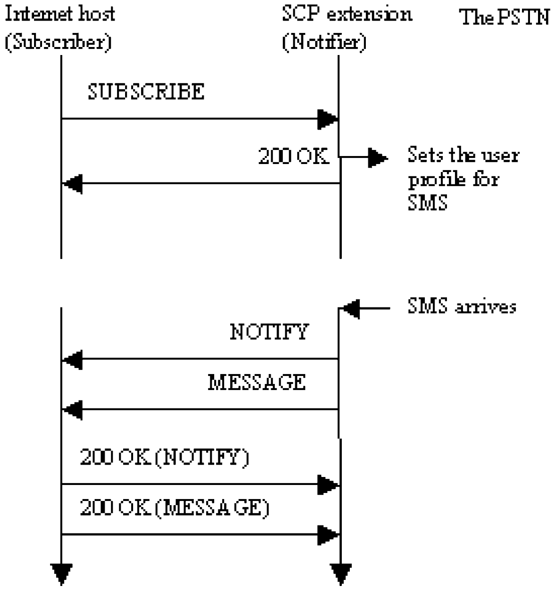


Fig. 6. SMS and instant messaging.

```
<spirits-event>
  <SUP app="SMS"/>
  <Mode>Notify<Mode>
  <CallingPartySubaddress>16 302 240
216</CallingPartySubaddress>
</spirits-event>
```

Note that the body of the SUBSCRIBE request contains an element called SUP (for supplementary services) and an attribute called "app" to further specify the supplementary service of interest. Here, the value of "app" is the string "SMS" to indicate the Short Messaging Service.

When a SMS arrives for the address of record identified by the number 16302240216, a NOTIFY followed by a MESSAGE request will be sent from the PSTN to the Internet host. A call flow is shown in Fig. 6. The NOTIFY itself does not contain the instant message; that arrives in a separate SIP request used specifically for instant messages (MESSAGE). Both SIP requests—NOTIFY and MESSAGE—belong to the same SIP dialog established by the SUBSCRIBE; thus, while Fig. 6 depicts a NOTIFY followed by a MESSAGE, the SIP protocol does not impose any temporal order on sending messages. Either one could have gone first; they both belong to the same dialog.

It is within the realm of possibility that a series of SMS messages arrive to an Internet host (to which the Internet host sends yet more instant messages in reply) to constitute a chat session. While this scenario is not covered in the example above, it is indeed feasible. The instant messages from the Internet host will be delivered to the PSTN using SIP MESSAGE requests. The PSTN will convert them to wireless SMS messages and deliver them to the recipient.

V. Taxonomy of PSTN-Originated Crossover Services

In order to impose some organization on PSTN-originated crossover services as well as help implementers in characterizing such services for rapid implementation, we attempt to taxonomize PSTN-originated crossover services. By and large, the taxonomy is suggested by the e_m element of tuple S. In other words, PSTN-originated crossover services can be categorized in two classes: *notification* and *dialog-oriented*. The latter automatically implies the former, the reverse is not true.

Notification services are the simpler of the two. The PSTN simply notifies the Internet host of the occurrence of the event of interest. Once the notification is send out, call processing continues normally in the PSTN without further aid of the Internet host. It should be pointed out that the notification may not be the result of call processing at all. For instance, in wireless networks, a notification may be sent to an Internet host when a user turned on his or her cell phone, thus registering with the network; or a wireline PSTN user may simply lift and set down the receiver on the cradle. A notification can be sent to Internet host which executes an appropriate service such as toggling the presence and availability state of a wireless or wireline PSTN user.

Dialog-oriented services are executed when the Internet host receives an INVITE request from the PSTN (the ICW service discussed in Section I-B is a good example). The Internet host acts as an extended SCP to the SSP as the latter has temporarily suspend call processing until it gets further instructions from the Internet host. Services under this classification scheme may exhibit long delays and mandate strict timing behavior on part of the Internet host. If the Internet host expects a fair amount of time (in the order of seconds) to generate further instructions, it should periodically send messages (provisional responses in SIP) to the SSP to reset any relevant timers in the PSTN. The PSTN, on the other hand, should start tearing the call down and re-claiming resources if it does not get any response from an Internet host (the Internet host may have crashed, or it could be misbehaving).

Dialog-oriented services can be further sub-classified as:

> **Static dialog**: Under this classification, the Internet host establishes a relationship with the SSP, thereby effectively controlling the SSP until the service is executed. Using ICW as an example again, call processing is suspended at the SSP until the Internet host makes a final determination on the disposition of the call. This disposition is sent to the PSTN in form of a final response to the INVITE request. There are two distinguishing facets for this classification: first, the DPs are armed *a priori* on the SSP; in other words, a SUBSCRIBE may not be needed (ICW implementation experience [35] confirms this). The second factor is that once the Internet host has sent a final disposition, the relationship between the SSP and the Internet host effectively terminates.
>
> **Dynamic dialog**: The key property of this classification is that the Internet host maintains an ongoing relationship with the SSP even after sending a final disposition to the INVITE request. It can, for instance, choose to get subsequent events from the PSTN by arming successive DPs after a call has been established. For example, the Internet host may subsequently subscribe to the "hang-up" event; i.e., have the PSTN notify it when the call is terminated. A static dialog may transition to a dynamic one based on the service aspects of Internet host.

We believe that this taxonomy will aid in better understanding PSTN-originated crossover services and that the classification outlined here helps form a standard reference template for implementation design issues.

VI. Security Issues With Crossover Services

Securing the signaling data that constitutes crossover services is of paramount importance. Consider that when a user powers on a cellular phone or moves across a cellular coverage area, his or her location can be exploited to provide crossover services. Location is an extremely private piece of information, the misuse of which may have severe consequences [34]. Wireline events resulting in crossover services are just as important a resource to protect from the viewpoint of privacy (who called whom and at what time?).

The security model presented here has been influenced by the ongoing work in the IETF GEOPRIV working group [9]. We model the security aspects by considering three entities and two interfaces: the event subscriber/event receiver employ the event interface to communicate with an event generator. A policy maker employs a policy interface to communicate with the event generator (a policy may consist of the directive "allow user Vijay Gurbani unlimited access to all events on my wireless and wireline phone"). The event subscriber/event receiver corresponds to the Internet hosts or SIP user agents of Fig. 4, and the event generator corresponds to the entire PSTN infrastructure in the same figure. The policy maker (not depicted in Fig. 4) is the physical user who owns the PSTN resource (phone line, wireless phone) for which events are generated. Good security implies securing the two interfaces: the policy interface and the event interface.

We do not make any assumptions of the policy interface. It could consist of the policy maker picking up a wireline (and thus secure) phone and directing the PSTN customer service representative to set policies on the line. Alternatively, it could consist of the policy maker using a secure web interface provided by the PSTN operator to set policies on the line. The policy maker would have to authenticate himself to the customer service representative or the web server before policies can be set on the line.

Securing the event interface is more complex. The event subscriber should authenticate the event generator (prevents malicious entities masquerading as event generators). The event generator, in turn, needs to authenticate the event subscriber before accepting the event (allows only valid subscribers to access the events based on an established policy). The event generator must also ensure that notifications go only to authenticated event receivers. Once the authenticity of the entities is established, the data transmitted over the event interface should be encrypted.

An in-depth treatment of security is not possible in the remaining space; however, use of public-key cryptography to authenticate identities and encrypt the information appears promising. The difficult problem of distributing keys can be mitigated by hard-coding the public key of the event generator in specialized user agents distributed to event subscribers/event receivers. A primary aspect of the authentication is manifested by the event generator associating a public key with the event subscriber/event receiver; to that extent, the policy maker can act as a trusted intermediary between the event generator and event subscriber/event receiver by vouching for the authenticity of the latter to the former.

VII. Related Work

The work closely related to PSTN-originated crossover services is PINT [11], which involves Internet hosts invoking certain telephone call services. PINT, like PSTN-originated crossover services operates in the services plane, but unlike our work, PINT concerns itself with invoking telephone services *from* the Internet. Furthermore, 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 contrast, does not necessarily involve in a telephone session being established and thus does it mandate that parties involved in a service be on a homogeneous network.

The SIP-T architecture [33] discusses other means by which a PSTN-originated call enters an IP network (or more specifically, a SIP network) which can in turn provide services to the call. However, SIP-T is concerned mainly with encapsulating PSTN call setup requests in a SIP message and translating PSTN information to SIP headers (and vice-versa) with the aim of preserving feature transparency of existing services. It does not provide a framework of the type discussed in this paper to allow interested Internet hosts access to selected events in the PSTN for service execution.

Gbaguidi *et al.* [12], [13] 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 toward Internet hosts for richer services in the latter domain. Furthermore, their approach effectively requires detecting triggers twice: once at the SCP and then again later in their service component. And finally, their methodology abstracts away the details of the signaling messages by using high-level APIs. We strongly believe that a standardized signaling interface of the kind outlined in this paper is a far richer medium to export signaling-related information in its entirety to interested entities. Telecommunication services require full access to signaling headers in order to make intelligent decisions; APIs add a layer of abstraction which often restricts unfettered access to such headers.

Rosenberg *et al.* [8], [14] discuss a component-based architecture for telephony services. They observe that many such services require common components—dialog servers (to play announcements and collect digits), mixers (for transcoding and mixing media streams), text-to-speech servers, etc. In their architecture, a central entity (a controller) coordinates 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 the BCSM of PSTN as a service stimulus. Like Gbaguidi *et al.* [12], [13], their architecture is well suited for services that require interactive communications. Furthermore, they claim that the dependency of a call model as a trigger for services make it difficult to integrate other Internet-based applications such as instant messaging and presence. We believe this is not necessarily the case, and as this paper has demonstrated, the call models of the PSTN can indeed be used as a rich stimulus for executing services which use instant messaging and presence.

VIII. Conclusion

We have presented the need for crossover services and a specific architecture targeted toward realizing them. This architecture has been validated by implementation experience [37], [38]. The results of our implementation have justified the ease of use and flexibility of the architecture in creating crossover services for the set of events we presented in Tables I and II.

The hardest part in an architecture that includes multiple entities and spans network topologies is identifying a good synchronization and message passing protocol. Our choice of SIP as the protocol of choice is, we believe, a sound one. The entire PSTN is abstracted as a SIP UA for crossover services. The advantages that this abstraction provides us are tremendous. For one, the PSTN entities do not know (nor do they care) that a portion of the service is being executed on a foreign domain, namely the Internet. Furthermore, the usage of SIP enables us to transport call-related data in a standard signaling protocol between different entities, synchronizing them and passing information between them in one attempt. Finally, the architecture presented here further separates the services plane from the call signaling information; services occur on one network, the signaling stimulus for them occurs on another network. It is our belief that this separation will help third party service providers to innovate novel services, some of which have been presented in this paper.

Acknowledgment

The authors are grateful to the anonymous reviewers for improving the quality of the paper vastly. In addition, they acknowledge sincere appreciation to A. Brusilovsky and Prof. H. Schulzrinne for their invaluable time.

References

[1] V. Gurbani, "Enabling services through protocol interworking," Ph.D. Research Proposal, Dept. Comput. Sci., Illinois Inst. Technol., Chicago, IL, 2002.

[2] A. Brusilovsky, V. Gurbani, E. Gausmann, and A. Jain. A proposal for Internet call waiting using SIP: An implementation report. Internet Draft. [Online]. Available: http://www.iit.edu/~gurbvij/I-D/draft-brusilovsky-icw-00.txt

[3] A. Brusilovsky, V. Gurbani, D. Varney, and A. Jain. Need for PSTN Internet Notification (PIN) services: A proposal for a new working group. Internet Draft. [Online]. Available: http://www.iit.edu/~gurbvij/I-D/draft-brusilovsky-pin-00.txt

[4] V. Gurbani, "PSTN Internet notification BOF (pin)," in *Proc. 44th Internet Engineering Task Force*, Minneapolis, MN, Mar. 1999. http://www.ietf.org/proceedings/99mar/slides/pin-services-99mar/sld001.htm.

[5] I. Faynberg, J. Gato, V. Gurbani, and H.-L. Lu. Toward the definition of the SIP events package for SPIRITS. Internet Draft. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-spirits-sip-evt-package-02.txt

[6] I. Faynberg, J. Gato, and H.-L. Lu, "SPIRITS protocol requirements," IETF, RFC 3298, http://www.ietf.org/rfc/rfc3298.txt, Aug. 2002.

[7] V. Gurbani, A. Brusilovsky, I. Faynberg, H.-L. Lu, M. Unmehopa, K. Vemuri, and J. Gato. The SPIRITS protocol. Internet Draft. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-spirits-protocol-08.txt

[8] J. Rosenberg, "Distributed algorithms and protocols for scalable Internet telephony," Ph.D. thesis, Graduate School of Arts and Sciences, Columbia Univ., New York, 2001.

[9] The GEOPRIV Working Group, http://www.ietf.org/html.charters/geopriv-charter.html.

[10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," IETF RFC 3261, http://www.ietf.org/rfc/rfc3261.txt, June 2002.

[11] S. Petrack and L. Conroy, "The PINT service protocol: extensions to SIP and SDP for IP access to telephone call services," IETF RFC 2848, http://www.ietf.org/rfc/rfc2848.txt, June 2000.

[12] C. Gbaguidi, J.-P. Hubaux, G. Pacifici, and A. Tantawi, "Integration of Internet and telecommunications: An architecture for hybrid services," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1563–1579, Sept. 1999.

[13] C. Gbaguidi, J.-P. Hubaux, G. Pacifici, and G. Tantawi, "An architecture for the integration of Internet and telecommunication services," in *Proc. IEEE Open Architectures and Network Programming Conf.*, Mar. 1999, pp. 9–21.

[14] J. Rosenberg, P. Mataga, and H. Schulzrinne. An application server component architecture for SIP. Internet Draft. [Online]. Available: http://www.jdrosen.net/papers/draft-rosenberg-sip-app-components-01.txt

[15] I. Faynberg, L. Gabuzda, M. Kaplan, and N. Shah, *The Intelligent Network Standards: Their Application to Services*. New York: McGraw-Hill, 1997.

[16] J. Dianda, V. Gurbani, and M. Jones, "SIP services architecture," *Bell Labs. Tech. J.*, vol. 7, no. 1, Jan.–June 2002.

[17] V. Gurbani, T.-C. Chiang, and S. Kumar, "SIP: a routing protocol," *Bell Labs. Tech. J.*, vol. 6, no. 2, July–December 2001.

[18] "SIP service architecture. White paper," Ubiquity Corp., http://www.ubiquity.net/pdf/SIP-Service-Arch-WP1_0.pdf, May 2001.

[19] H. Sinnreich and A. Johnston, *Internet Communications Using SIP*. New York: Wiley, 2001.

[20] G. Camarillo, *SIP Demystified*. New York: McGraw-Hill, 2001.

[21] I. Faynberg, L. Gabuzda, T. Jacobson, and H.-L. Lu, "The development of the Wireless Intelligent Network (WIN) and its relation to the international intelligent network standards," *Bell Labs. Tech. J.*, vol. 2, no. 3, Summer 1997.

[22] M. Bergeren, B. Bollinger, D. Earl, D. Grossman, B.-W. Ho, and R. Thompson, "Wireless and wireline convergence," *Bell Labs. Tech. J.*, vol. 2, no. 3, Summer 1997.

[23] A. Roach, "Session Initiation Protocol (SIP)—specific event notification," IETF RFC 3265, http://www.ietf.org/rfc/rfc3265.txt, June 2002.

[24] J. Rosenberg. A presence event package for the Session Initiation Protocol (SIP). Internet Draft. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-simple-presence-10.txt

[25] J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle, "Session Initiation Protocol extension for instant messaging," IETF RFC 3428, http://www.ietf.org/rfc/rfc3428.txt, B. Campbell, Ed., Dec. 2002.

[26] A. Brusilovsky, E. Dacloush, M. Unmehopa, K. Vemuri, A. Zaki, and F. Haerens. On selection of IN parameters to be carried by the SPIRITS protocol. Internet Draft. [Online]. Available: http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-spirits-in-02.txt

[27] D. Moreno. Mobility events management in SPIRITS. Internet Draft. [Online]. Available: http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-spirits-mobility-00.txt

[28] I. Faynberg, L. Gabuzda, and H.-L. Lu, *Converged Network and Services: Interworking IP and the PSTN*. New York: Wiley, 2000.

[29] V. Gurbani, J. Voelker, and D. Moreno. Location services in SPIRITS. Internet Draft. [Online]. Available: http://www.ietf.org/proceedings/02jul/I-D/draft-gurbani-spirits-location-00.txt

[30] "SPIRITS," The SPIRITS (Services in the Internet Requesting PSTN Services) Working Group, http://www.ietf.org/html.charters/spirits-charter.html.

[31] "Bearer Independent Call Control Protocol," International Telecommunication Union, Recommendation Q.1901, June 2000.

[32] "Packet-Based Multimedia Communication Systems," International Telecommunication Union, Recommendation H.323, Nov. 2000.

[33] J. Peterson and A. Vemuri, "Session Initiation Protocol for Telephones (SIP-T): Context and architectures," IETF RFC 3372, http://www.ietf.org/rfc/rfc3372.txt, Sept. 2002.

[34] M. Danley, D. Mulligan, J. Morris, and J. Peterson, "Threat analysis of the geopriv protocol," IETF RFC 3694, http://www.ietf.org/rfc/rfc3694.txt, Feb. 2004.

[35] H.-L. Lu, "Pre-SPIRITS implementation of PSTN-initiated services," IETF RFC 2995, http://www.ietf.org/rfc/rfc2995.txt, Nov. 2000.

[36] A. Brusilovsky, J. Buller, L. Conroy, V. Gurbani, and L. Slutsman, "PSTN Internet Notification (PIN) proposed architecture, services and protocols," in *Proc. 6th Int. Conf. Intelligence in Networks (ICIN)*, Jan. 2000.

[37] V. Gurbani, A. Brusilovsky, I. Faynberg, H.-L. Lu, X.-H. Sun, and M. Unmehopa, "Internet service execution for telephony events," in *Proc. 8th Int. Conf. Intelligence in Next Generation Networks (ICIN)*, Apr. 2003.

[38] V. Gurbani and X.-H. Sun, "Extensions to an Internet signaling protocol to support telecommunication services," IEEE GLOBECOM, Nov. 2004.

Vijay K. Gurbani received the B.Sc. and M.Sc. degrees in computer science from Bradley University, Peoria, IL, in 1988 and 1990, respectively. He is currently working toward the Ph.D. degree in computer science at the Illinois Institute of Technology, Chicago.

He is with Lucent Technologies, Inc., Naperville, IL, in the wireless research and development unit, where he is involved in the specification, prototyping, and implementation of services based on SIP. His research interests are Internet telephony services, Internet telephony signaling protocols, pervasive computing in the telecommunications domain, distributed systems programming, and programming languages. He holds one patent and has four U.S. patent applications pending.

Mr. Gurbani is a member of the ACM and the IEEE Computer Society.

Xian-He Sun (M'90–SM'95) received the B.S. degree in mathematics from Beijing Normal University, Beijing, China, in 1982, the M.S. degree in mathematics, and the M.S. and Ph.D. degrees in computer science from Michigan State University, Lansing, in 1985, 1987, and 1990, respectively.

He was a Staff Scientist at ICASE, NASA Langley Research Center, and was an Associate Professor in the Computer Science Department, Louisiana State University. He has been serving as a faculty member of the Computer Science Department at the Illinois Institute of Technology (IIT), Chicago, since 1999. Currently, he is a Professor and the Director of the Scalable Computing Software Laboratory in the Computer Science Department at IIT, and is a guest faculty at the Argonne National Laboratory. His research interests include grid and cluster computing, software system, pervasive computing, performance evaluation, and scientific computing. He has published intensively in the field and his research has been supported by DoD, DoE, NASA, NSF, and other government agencies.

Dr. Sun is a senior member of IEEE, a member of ACM, New York Academy of Science, Phi Kappa Phi, and has served and is serving as the chairman or on the program committee for a number of international conferences and workshops, including current service as the general co-chair of the Grid and Cooperative Computing (GCC03) workshop, area chair of the technical committee of the IEEE SuperComputing (SC03) conference, vice chair of the programming committee of the International Conference on Parallel Processing (ICPP04), and vice president of the Society of Chinese American Professors and Scientists. He received the ONR and ASEE Certificate of Recognition Award in 1999, and received the Best Paper Award from the International Conference on Parallel Processing (ICPP01) in 2001.