Extensions to an Internet signaling protocol to support telecommunication services

Vijay K. Gurbani[1,2] and Xian-He Sun[1]
vkg@{lucent.com, iit.edu} sun@iit.edu
1. Department of Computer Science, Illinois Institute of Technology, Chicago, Illinois
2. Lucent Technologies, Inc./Bell Laboratories, Naperville, Illinois

Abstract**—** We discuss extensions to the Session Initiation Protocol (SIP) to enable a rich set of services between the cellular Public Switched Telephone Network (PSTN) and the Internet. We consider all manner of cellular endpoints: 2G, 2.5G, and early 3G endpoints. These extensions allow the PSTN endpoints to act as smart SIP user agents and participate in Internet-oriented services such as presence and instant messaging (IM) *without* the endpoint itself being connected to the Internet. For cellular networks, our methodology conserves precious radio spectrum by offloading such data services from the bandwidth normally allocated to voice. Our approach is based on open standards and is not an ad-hoc or incremental solution.

I. Introduction

The service architectures for the PSTN and the Internet are at the opposite end of the spectrum from each other. The PSTN, to provide services, concentrates intelligence in the network core; the Internet does just the opposite -- intelligence is pushed to the edges. While it has long been possible to digitize voice and send it over a packet network, and to use a telecommunication line as an access conduit to the Internet, what has been missing is service integration. Such integration, however, is an important step towards next generation networks, where heterogeneous networks work in concert to provide services to users in a seamless and transparent manner. Market conditions and technical realities dictate that future networks will be integrated and will co-exist with current ones. This integration is required on two planes: the signaling plane (for services provided through signaling) as well as the media plane (for services provided through the media, or bearer, channel; such as voice).

The work in this paper pertains exclusively to the signaling plane; we explore the integration of the Internet and the cellular PSTN at the services layer. We consider 2G endpoints which are themselves not connected to the Internet and 2.5G endpoints which can connect to the Internet (GPRS, 1xRTT), but because the data connection and its signaling is distinct from the voice connection and associated signaling, there isn't any beneficial interaction between the data and voice channels (service isolation). An example illustrates this: when a cellular subscriber turns on her 2.5G Internet-capable phone, it can inform a presence manager which toggles her presence indicator to 'on' and availability status to 'available.' However, when the same cellular user initiates (or receives) a phone call, the presence system is unable to reflect her current availability status (i.e. 'busy'). The reason is that the process of initiating (or receiving) a call uses different signaling protocols and a separate voice channel. Services using the Internet connection do not interact with the services on the voice channel to provide yet more innovative benefits of integrated networks to their users. Thus, it is impossible to derive a complete state of the subscriber based on only using one network and its protocols; more intelligence is required.

Until 3G is ubiquitously deployed and all signaling between the endpoints and the core network is based on the Internet Protocol, there will be a need to interoperate services between the existing cellular PSTN infrastructure and early 3G endpoints. Core services such as presence and availability ought to work across all endpoints: 2G, 2.5G, and early 3G. In fact, a case can be made for 2G and 2.5G endpoints that such services should be provided without the endpoints themselves being connected to the Internet. A 2G endpoint may not be capable of connecting to the Internet, a 2.5G endpoint is, but the subscriber may not want to pay extra money to do so. By providing these services without the endpoints being connected to the Internet, radio bandwidth can be preserved to beused instead for voice communications.

We discuss extensions to SIP [10] which, when applied to a target architecture allow 2G and 2.5G endpoints to participate in such services without being on the Internet. Furthermore, even for current 3G endpoints which are already Internet-capable, our methodology minimizes service isolation by tighter integration across the data and voice channels. Note that even in current 3G endpoints, the packetized voice is not transported over the data channel, a separate circuit voice channel is employed for this.

Our architecture also enables other services, some of which are: the cellular PSTN sending an IM to the cellular subscriber's Internet user agent if she missed a call (assuming, of course, that the cellular subscriber does not have her cell phone with her, or that it ran out of power). A cellular subscriber can also instruct the cellular PSTN to transform incoming Short Message Service (SMS) messages into an IM and deliver it to her Internet user agent. Or consider the following service: when disaster strikes, the PSTN network usually becomes overwhelmed due to increased call volume. A subscriber calling loved ones constantly finds all circuits busy. It should be possible for the subscriber to store preferences with the PSTN such that the PSTN arranges to send an email (or IM) to loved ones when the user attempts to call them but the call fails due to excessive network congestion. As long as the call request arrived at a mobile switching center, the PSTN can undertake such a service.

The remaining paper is organized as follows. We describe the architecture next followed by related work. Extensions to SIP are discussed in Section IV and implementation of the architecture based on the extensions is detailed in Section V. We then present a section on performance and conclude by presenting challenges faced during implementation.

II. Architecture

The architecture is based on separating the network on which the service executes from the network that generates the events that lead to the service execution. The service itself is executed entirely on the Internet, but the events that lead to the execution of the service occur on the cellular network. We first presented a general framework for such services in [5,6] and a specific instance demonstrating use of the framework for wireline networks in [7]. Our focus in this paper is services enabled by the architecture on the cellular network. We use the cellular network as a backdrop to detail the extensions we propose to SIP. Cellular networks present a rich palette of events upon which services can be build: registration, mobility, and SMS arrival are some of the events beyond normal call control that can be used to execute services on the Internet.

Our architecture, depicted in Figure 1, uses the publish/subscribe mechanism that has proved to be well suited for an event-based mobile communication model [8,9]. User agents (software programs) on the Internet subscribe to events on the cellular network. When the event occurs, the cellular network notifies the user agent (UA), which executes the desired service. The centerpiece of the architecture is the Event Manager (EM) that straddles both the networks. It insulates the cellular network entities from Internet protocols and vice-versa. It is also responsible for authenticating UAs and maintaining subscription state so it can transmit notifications when an event subscribed to transpires.

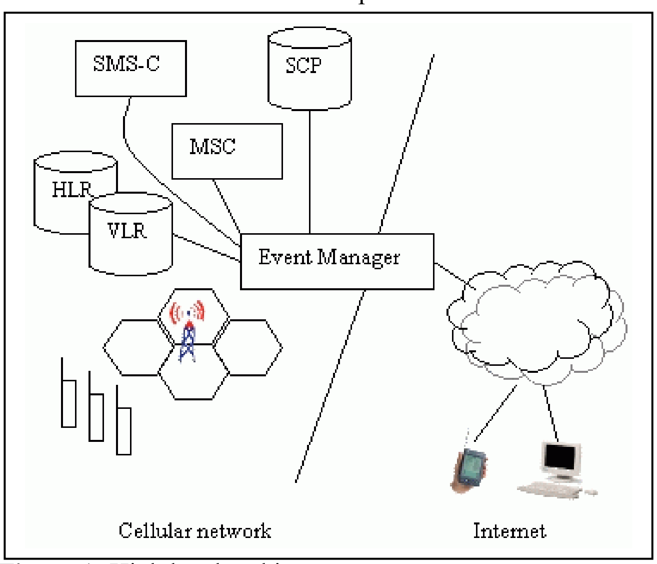

Figure 1: High level architecture.

There are many entities generating events in the cellular network. Cell phones generate call-related, registration and mobility events. Call-related events (making/receiving a call, hanging up) are propagated to the Mobile Switching Center (MSC); location-related events are propagated to the Home Location Register and Visitor Location Registers (HLR/VLR); and yet other application-specific events such as arrival of a SMS are propagated to the Short Message Service Center (SMS-C). Our architecture normalizes and aggregates all these events at the Event Manager (EM).

An early challenge involved quantifying the set of events that may occur in the cellular network. Clearly, these events have to be implemented in a standardized manner across many cellular service providers (CSP) in order to make services ubiquitous. To that extent, we leverage the Wireless Intelligent Network (WIN). WIN is a conceptual architecture that separates the call control from service fulfillment in the PSTN. WIN contains fixed finite state machines that govern the behavior of communicating entities. Transitions between states go through a construct called a "detection point" (DP). A DP may be armed or disarmed. If the DP is armed, the processing switch suspends further call processing and issues a remote procedure call to a service control point (SCP), which is a general-purpose computer on which subscriber services are hosted and executed. The procedure call in the SCP performs the required service and returns a result back to the switch. WIN contains state machines for call processing as well as registration. We leverage these DPs to provide services as follows: Internet hosts will subscribe to select DPs and when these DPs are triggered, the cellular network will inform the Internet host (through the EM) of this occurrence. Included in the notification will be enough information to allow the Internet host to execute a service appropriate to the DP.

A canonical list of all DPs we quantified is presented in [5,11]; Table I contains a subset of the DPs to aid the reader:

Event	Comments
LUSV	Location update in same VLR area
LUDV	Location update in different VLR area
REG	Registration of mobile station
UNREGMS	Mobile station initiated de-registration
UNREGNTWK	Network initiated de-registration
OA	Mobile station in a call (originated a call)
TA	Mobile station in a call (received a call)
ORSF	Network congestion (unavailable routes to make a call)
OD	Mobile station disconnected a call that it originated
TD	Mobile station disconnected a call that it received
TB	Mobile station is busy; cannot accept a call

Table I: Partial list of cellular events

Besides the call-related and location-related DPs, the cellular network generates many other events. For instance, an arrival of a SMS generates events that can also be leveraged for Internet services. A pre-paid account balance reaching a certain threshold is another event that can be potentially used to provide a service.

III. Related Work

Several projects and products are related to our work, demonstrating the vibrant research in this area. While these efforts share some aspects of our work, they do not provide the framework to transport discrete events from the cellular network to the Internet and use these events as pre-cursors to advanced services. Our work does not by itself specify a set of services; rather, it provides building blocks to construct many services from cellular network events.

To the extent that our architecture enables presence-based services, it can be contrasted with similar systems which exhibit "awareness", such as Sun Microsystem's Awarenex [1] and Milewski's Live Address Book [2]. Awarenex does indeed designate if a cell phone is in a conversation, but it does so in an incremental and ad-hoc solution that mandates that a call request must be placed through an Awarenex server for real-time status updates. It is easy to bypass the server completely. Our architecture arranges for presence-related events to be detected by the deployed and pervasive cellular network. Furthermore, in order to allow richer Internet services, our architecture provides many more events beyond those required for real-time status updates. The Live Address Book also permits its users to provide real-time updates of the status of their phones. But it does so manually. As Milewski's research indicates, users will not consciously remember to always update their status. Our architecture, by contrast, updates the status automatically.

Berkeley's ICEBERG project [3] integrates telephony and data services spanning diverse access networks. Their approach is complex since their architecture maintains an overlay network consisting of different geographic ICEBERG points of presence (iPOP) and many ICEBERG access points to isolate the access network from the overlay network. The iPOPs are coordinated by a centralized clearing-house which serves as a bandwidth broker and accountant. Our approach, by contrast, is extremely lightweight and follows the service mantra of the Internet whereby the core network is used simply as a transport and services are provided at the edges. In a sense, the entire cellular network is abstracted as a UA generating and sending events to another UA which then executes the services.

Stanford's Mobile People Architecture (MPA) is another effort to bridge the wireless and Internet networks [4]. However, its main goal is to route communications to a mobile person, independent of the person's location communication device being used. MPA's goal differs from our work, which aims to provide discrete events to user agents on the Internet for service execution.

The Parlay Group (www.parlay.org) is an industry consortium that specifies application programming interfaces (API) to integrate telecom network capabilities with arbitrary applications. It is paramount to note that Parlay specifies a programming interface only, not a communication protocol. The work described in this paper can easily serve as an "on the wire" protocol underneath the Parlay APIs.

Commercial enterprises like Yahoo! allow a cell phone to become a 'buddy' in a presence list. However, this feature is only provided for phones that are connected to the Internet and is not integrated with call processing, leading to service isolation. Yahoo! also makes it possible for a computer user to send a text message to a cellular user; the text message is converted to a SMS and delivered to the cellular user. However, the reverse (turning a SMS into an IM) is not supported. Our architecture mitigates both of these shortcomings.

Today, presence is viewed as a key service. There are many commercial products including unified messaging products and presence-based call routing products that use presence as a key enabler to route calls. It should be noted that our architecture provides a far richer set of events then those associated with detecting presence only. Furthermore, we do so in a standards-based framework [11], as discussed next, instead of relying on proprietary interfaces.

IV. Extensions to the Protocol

We extend the SIP protocol across two axes: first, we specify two SIP *event packages*, and second, we introduce a new (Multi-part Internet Mail Extension) MIME type used to describe a payload transported by SIP. It is assumed that the reader has a working knowledge of SIP; [10,12] are good resources for more information on the protocol.

SIP supports asynchronous events notification of network resources [13]. The framework considers that there are entities (called subscribers) that issue a subscription to monitor the state of a resource. The subscription is accepted by another entity (called a notifier) that represents the resource itself or a suitable stand-in that knows how to interact with the resource and send out notifications when the state of the resource changes. RFC3265 [13] specifies the behavior of subscribers and notifiers. It includes details on how long subscriptions last, when they should be refreshed, the rate of notifications, whether forking (replication of a request across multiple search branches) is permitted or not. It expects -- since the universe of events is potentially unlimited -- that each usage of the SIP events framework be rigorously defined in an *event package*. As an analogy, RFC3265 can be considered an abstract base class while each event package is a concrete instantiation of the abstract class.

We have defined two SIP event packages -- spirits.INDPs and spirits.user-prof. The former package corresponds to all DPs associated with originating/receiving a call while the second event package corresponds to DPs associated with non-call events (registration, de-registration, location updates, etc.). The behavior of subscribers and notifiers corresponding to these packages is rigorously defined in [11]. Unfortunately, space does not permit us to list the detailed behavior in this paper.

The second extension deals with the payload carried by the subscription and notifications. An aspect that RFC3265 leaves open is how a subscriber chooses to represent the events of interest in a subscription. Likewise, it is also left open as to the payload format a notifier uses to inform a subscriber that certain events have occurred. In [11], we have defined a new MIME type -- application/spirits-events+xml. This MIME type corresponds to a URN (urn:ietf:params:xml:ns:spirits) which defines an XML schema used to represent the DPs and associated parameters (calling party, called party, cell ID, digits dialed, etc.) When a subscriber subscribes to events of interest, it creates a payload which contains an XML corresponding to the new MIME type. When a notifier transmits a notify, it also creates a payload which corresponds to the new MIME type. Figure 2 outlines a representative call flow. The XML schema corresponding to the new MIME type is presented in [11].

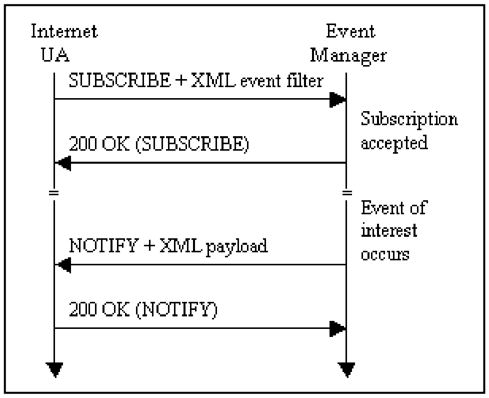

Figure 2: Protocol flow.

A major advantage of using XML is that it is extensible. Thus, events that cannot be characterized by a standard WIN call model can still be represented by XML extensions as long as the CSP and the user agent agree to their representation.

The extensions described in this section have been approved by the Internet Engineering Task Force (IETF) as a Proposed Standard and are fully described in [11]. As of this writing, the draft is awaiting publication as an IETF Request For Comment (RFC).

V. IMPLEMENTATION

We implemented a stand-alone, multi-threaded EM that executes on a general purpose Sun Microsystems platform. All events in our prototype implementation were propagated to the MSC by the cellular endpoints. We employed a simple TCP-based message passing protocol between the EM and the MSC. Our laboratory also consisted of simulated cells, base stations and 2G/2.5G cellular endpoints that received signals from the base stations. Using simulation tools, we are able to simulate signal attenuation that reproduces the movement of cellular subscribers between cells. As a proof of concept, we implemented three benchmark services: presence for cellular subscribers based on events generated by their phone (registration, de-registration); availability of cellular subscribers based on events leading to initiating or receiving a phone call, and finally, SMS to IM service migration.

For cellular presence, an Internet host (user agent) issues a subscribe which contained a set of two DPs: REG and UNREGMS (note the MIME type in Content-Type):

```
SUBSCRIBE sips:em.csp.net SIP/2.0
Content-Type: application/spirits-event+xml
…
<?xml version="1.0" encoding="UTF-8"?>
<spirits-event xmlns="urn:ietf:params:xml:ns:spirits">
  <Event type="user-prof" name="REG">
    <CalledPartyNumber>8475551212</CalledPartyNumber>
  </Event>
  <Event type="user-prof" name="UNREGMS">
    <CalledPartyNumber>8475551212</CalledPartyNumber>
  </Event>
</spirits-event>
```

Here, a user agent is interested in receiving notifications when these two events occur for the cellular number 847-555-1212.

The EM accepts this subscription and after successfully authenticating the user agent, creates a subscription for these events. When they occur, the EM is notified by the MSC and it (EM) sends out a NOTIFY message with the event that occurred. The user agent, upon receiving the NOTIFY toggles the state of the cellular subscriber in a presence application. One of the attributes returned in a NOTIFY when a cellular subscriber registered is a cell ID. We used this cell ID to display a map of the approximate location of the cellular subscriber.

It is entirely possible to track the progress of a cellular subscriber in real-time (privacy issues not-withstanding). For location, the user agent would subscribe to two events: LUSV and LUDV (see list of events in Table I) to receive timely updates of a cellular subscriber's current location.

For availability, a user agent can subscribe to two events: OA and TA. When the cellular subscriber makes or receives a call, the user agent is notified and it can display a real-time availability status of the cellular subscriber as depicted in Figure 3.

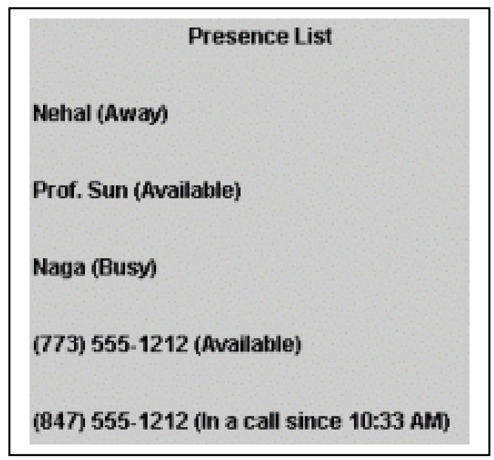

Figure 3: Presence and Availability.

And finally, for converting an SMS into an IM, an user agent issues a subscription with the XML filter shown below. The event filter directs the cellular network to send an IM consisting of the SMS message to the SIP address identified in the <im> element if the network cannot successfully deliver (delivery type="failure") the SMS message to the destination cellular phone (6302240216). The EM, upon being notified by the MSC of the SMS, converts it to a SIP IM and delivers it to the SIP address.

```
<Event type="application" name="SMS">
<CalledPartyNumber>6302240216</CalledPartyNumber>
    <delivery type="failure"/>
    <im>sips:bob@isp.example.com</im>
</Event>
```

VI. PERFORMANCE

The performance of a system described in this paper depends on many variables. Propagation of signaling between a user agent and the EM across the Internet is one factor; the number of cellular subscribers and the lookups/updates in the subscription database at the EM is another. To derive a performance estimate, we made one simplifying assumption:

we consider the propagation delay of the subscription and notification messages across the Internet to be constant. Given this assumption, we focused on the arrival rate of events at the EM and how long it takes to process them. Processing consists of performing a lookup in a user database (of 1 million users), creating and sending out a notification.

We executed our experimental model on two machines: a Sun Microsystems 4-CPU Netra T 1400/1405 (each CPU was an UltraSPARC-II 440 MHz) with 4096 Mbytes RAM and a Sun Microsystem 1-CPU Netra 60 (UltraSPARC-II 296 MHz) with 128 Mbytes RAM. On both, we created an in-memory database of 1 million users. We assumed that events arrive at the EM following a Poisson distribution with a mean we varied across different runs. Figure 4 contains the performance graph for the model executing on the two machines with an arrival rate mean varying from 200 events/sec to 1000 events/sec.

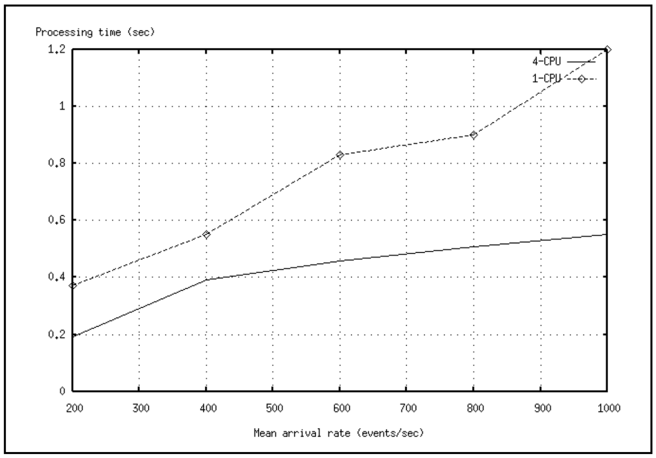

Figure 4: Performance.

VII. Challenges and Conclusion

In order for services based on propagating discrete events to the Internet to succeed, the CSP must make these events visible to third party providers. Interest of the industry in Parlay/OSA is evidence that this is already happening. We hope to build on that success. We believe that one major hurdle in widespread opening of CSP's networks to third party providers has been security and privacy concerns.

For security and privacy, we used OpenSSL v0.9.7b in our implementation. The user agent, which we assume will be distributed by the CSP or a trusted third party provider, is configured with the public key of the CSP or third party provider. Communications between the user agent and EM are encrypted using this key. In order to get this user agent, a user should be required to have a pre-existing relationship with the CSP or to have in some way authenticated himself (through a credit card, say) before receiving the user agent.

Another challenge has been event aggregation. In the cellular network, events are propagated to many places -- MSC, HLR, etc. In our prototype, all events were propagated to the MSC and aggregated at the EM. In a deployed system, it will be imperative to interface with the different cellular entities so that a central aggregator can be notified of event occurrence. Furthermore, while in our prototype we had one EM, it seems appropriate that in a deployed system, an event aggregator be coupled with a MSC. This will provide scalability and redundancy to the system.

A policy manager that is easy to use will be an advantage. Such a manager can store user-specific policies and apply them dynamically. We are currently investigating the use of a Bell Labs policy framework called Houdini that has been used to demonstrate, among other features, user specified preferences that are filtered through a rules engine before location information is shared.

Finally, the location information released by the cellular network has a large radius. However, it is more ubiquitous than the global positioning system that tends to fade indoors and in urban canyons. The intersection of cellular location with global positioning system and other such mechanisms will be an active area of research in the future.

This work has demonstrated extensions to SIP for harnessing events occurring in the cellular network to execute services on the Internet. Radio spectrum is a precious resource; our system preserves it by offloading certain Internet services from it, thus making the spectrum fully available for voice services. The work described here can also be used by a 3G Internet Multimedia Subsystem (IMS) to provide Internet-type services to legacy endpoints. IMS signaling is based on SIP; the EM, which already uses SIP, can insulate the IMS by providing it the same SIP interface that a native 3G endpoint would.

References

[1] J. Tang, et al., "ConNexus to Awarenex: Extending awareness to mobile users", Proceedings of the ACM SIGCHI, March 31-April 4, 2001, Seattle, WA.

[2] A. Milewski and T. Smith, "Providing presence cues to telephone users", Proceedings of ACM Conference on Computer Supported Co-operative work (CSCW), Dec 2-6, 2000, Philadephia, PA.

[3] H.J. Wang, et al., "ICEBERG: An Internet-core network architecture for integrated communications", IEEE Personal Communications Special Issue on IP-based Mobile Telecommunications Networks, Vol. 7, No. 4, August 2000.

[4] P. Maniatis, et al., "The Mobile People Architecture", ACM Mobile Computing and Communication Review, Volume 3, Issue 3, pp. 36-42, July 1999.

[5] V. Gurbani and X-H. Sun, "Terminating Telephony Services on the Internet", IEEE/ACM Transactions on Networking, Vol. 12, No. 4, August 2004.

[6] V. Gurbani and X-H. Sun, "Services spanning heterogeneous networks", Proceedings of the IEEE International Conference on Communications (ICC) 2003, pp. 764-768, May 2003.

[7] V. Gurbani, X-H. Sun, et al., "Internet service execution for telephony events", Proceedings of the 8th International Conference on Intelligence in Next Generation Networks (ICIN 2003), April 2003.

[8] R. Meier, "Communications paradigms for mobile computing", ACM Mobile Computing and Communications Review, Volume 6, Issue 4, pp. 56-58, October 2002.

[9] G. Cugola and H.-A. Jacobsen, "Using Publish/Subscribe middleware for mobile systems", ACM Mobile Computing and Communications Review, Volume 6, No. 4, pp. 25-33.

[10] J. Rosenberg, et. al. "The Session Initiation Protocol (SIP)", IETF RFC 3261, <http://www.ietf.org/rfc/rfc3261.txt>, June 2002.

[11] V. Gurbani (Ed.), et al., "The SPIRITS (Services in PSTN requesting Internet services) Protocol", IETF Internet- Draft, Work in Progress, <http://www.ietf.org/internet-drafts/draft-ietf-spirits-protocol-08.txt>, February 2004.

[12] G. Camarillo, "SIP Demystified", McGraw-Hill publishing company, August 2001.

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