Internet service execution for telephony events

Vijay K. Gurbani[1,2], Alec Brusilovsky[3], Igor Faynberg[1], Hui-Lan Lu[1], Xian-He Sun[2], and Musa Unmehopa[1]
{vkg, faynberg, huilanlu, unmehopa}@lucent.com sun@cs.iit.edu abrusilovsky@ieee.org

1. Lucent Technologies, Inc Naperville, Illinois (USA)
2. Illinois Institue of Technology Chicago, Illinois (USA)
3. ArtCare Naperville, Illinois (USA)

Abstract
The intrinsic value of a network is measured by the services which it provides to its users. Currently, there are two principal networks for communications: the public switched telephone network (PSTN) and the Internet. As these networks converge, new service[1] ideas become apparent that are not possible in isolation on either of the networks. This paper discusses how the merging of the Internet and the (wireline- and wireless-) PSTN provides for an opportunity to transport the state of a PSTN phone call to the Internet for service execution. 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.

1. Introduction

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 receive an instant message (IM) 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 to be notified 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 and use homogeneous protocols. What Bob would like to do when Alice calls him is to have the PSTN send him an IM to 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 notify him of it. The notification may consist of sending Bob an IM on his Internet-based PDA. 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.

Our approach for doing this is to recognize that the events of interest to Bob are occurring in the public switched telephone network (PSTN) and are being propagated to the Internet for the execution of services. Thus, an architecture and protocol are required which will allow the PSTN to transfer information of the events occurring in its domain to an Internet endpoint. The Internet host receives this information, analyses it and starts executing services. The information passed from the PSTN could be a simple notification, or it could be a more complex dialog, in which the Internet host may further engage the PSTN until the service has been completed. The architecture should allow for both types of interactions to occur.

The Internet Engineering Task Force (IETF) has chartered a working group called SPIRITS (Services in the PSTN/IN Requesting Internet Services) [1] to investigate this topic. SPIRITS addresses how services supported by the Internet network entities can be initiated from the PSTN/IN entities. Since the currently deployed worldwide wireless networks (2G/2.5G) are based on circuit switching, they are considered PSTN networks for SPIRITS purposes. Adding SPIRITS type of services to wireless networks can allow new services to be developed (for example geo-location information obtained on the wireless network can be exploited in the IP network for SPIRITS services).

Certain aspects of SPIRITS have been standardized; for

[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.

instance, the SPIRITS architecture [2] and requirements [3] have already been approved by the IETF. In this paper, our focus will not be on topics that have been standardized (besides providing a short summary to afford all the readers the same level of understanding), but rather, we will take a look at issues that we are currently working on and report on a preliminary implementation of a SPIRITS architecture to provide services on a wireline switch (wireless SPIRITS services are to be implemented next).

The rest of this paper is organized as follows: Section 2 provides a quick overview of services in the PSTN and the SPIRITS architecture; section 3 discusses the SPIRITS protocol and how it enables new services. Section 4 covers security aspects of SPIRITS and section 5 outlines our implementation of a SPIRITS architecture to provide a presence-based service. The final two sections provides related work and some conclusions and a summary, respectively.

2. SPIRITS in a nutshell

Put simply, SPIRITS aims to export the state of a PSTN call into the Internet for service execution. A distinguishing factor of SPIRITS is that the entire PSTN can be abstracted as a specialized (and powerful) user agent.

We assume that the reader is familiar with PSTN and the Intelligent Network (IN) architectural concept which decouples call processing from service logic. Uninitiated readers are urged to consult [17] for more information; here we assume familiarity of IN concepts.

We use the term PSTN here represents both the wireline and wireless aspects of the switched network. Specifically, the current wireless services infrastructure (2G, 2.5G) is heavily influenced by the concepts of IN and is well integrated in the PSTN [4,5]. 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 Service Control Point (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.

In the PSTN, the SCP has the capability to detect all manner of events, call-related and non-call related, that are generated by the PSTN switches. We exploit this relationship between the SCP and the PSTN switches for SPIRITS.

A portion of the SPIRITS architecture from [2] is reproduced in figure 1. As can be observed, a SPIRITS capable PSTN system consists of a SPIRITS client resident on the SCP which interfaces with a SPIRITS server in the Internet domain. The SPIRITS client accepts subscriptions from the SPIRITS server (which is an IP host) and interfaces with the SCP to set the DPs and other non-call related events. When these events occur, it will send a notification to the SPIRITS server and enable a set of converged services. Figure 1 also contains an additional entity called the SPIRITS gateway which interfaces with the SPIRITS client and SPIRITS server. The SPIRITS gateway can act as a gatekeeper to the SPIRITS client, authenticating requests arriving at the SPIRITS client, or it can act as a proxy between the SPIRITS entities.

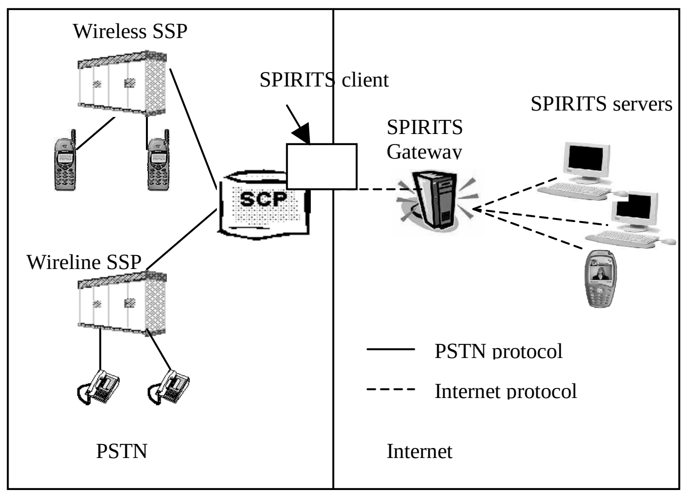

Figure 1: SPIRITS architecture

3. The SPIRITS protocol

As is the case with distributed systems, a protocol is required to synchronize the attendant entities for deterministic behavior. We list the properties that are desirable in such a protocol.

A SPIRITS service occurs when the PSTN performs an event of interest to an Internet host. When the event of interest occurs, the PSTN takes a snapshot of the call and transfers it to the Internet host for service execution; and depending on the service, the PSTN may actually await further instructions from the Internet host. Since the service executes on the Internet, 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.

There were three candidate protocols – SIP [6], H.323 [7], and BICC [8]. Based on the needs outlined in the previous paragraph and the fact that many pre-SPIRITS implementations [9,10] already used SIP, we felt that SIP was a good protocol choice for SPIRITS. 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 towards multi-media communications. SIP also possesses built-in support for asynchronous event notification [11] and enables services like presence [12] and instant messaging [13] that we view as vital components of SPIRITS services. Finally, SIP provided the MIME primitives and necessary headers to carry arbitrary payload in a structured fashion between the PSTN and the Internet.

SPIRITS requires the Internet host to inform the PSTN of the events it is interested in receiving a notification for. Conversely, SPIRITS requires the PSTN to notify the Internet host when an event of interest occurs. Implicit in this give and take between the Internet and the PSTN is the need for transporting structured information elements representing the events and their associated parameters between the networks. We have decided to use XML as the meta-language for this representation. Thus, an Internet host interested in receiving the notification of events in the PSTN encodes the events and their associated parameters in an XML document which becomes part of a SIP request. Likewise, the PSTN informs the Internet host of the occurrence of an event by encapsulating an XML document in a SIP request and sending it to the Internet host. The mechanics to do this in detail are described in [14,15]. The core components of the SPIRITS protocol are the use of SIP event packages (SPIRITS defines two new ones and one MIME type) and the representation of PSTN events and their notification into XML. This representation is carried as a *session description* in SIP requests as described in [14,15].

4. SPIRITS security

In order to address trust and security in the architecture, it is instructive to keep in mind the call flow of a typical service and focus on the interfaces where the information is exchanged between the PSTN and the Internet. Clearly, in the SPIRITS architecture, the Internet host sends a subscription to express an interest in certain events in the PSTN. This message must, at the very least, be authenticated by the PSTN to ensure that the Internet host is a known entity (HTTP digest authentication is preferred over HTTP basic), and encrypted to afford privacy. When the PSTN notifies the Internet host of the event, some form of reverse authentication is needed; i.e. the PSTN must ensure that the notification is indeed going to the right Internet host, and not a spoofed, or hi-jacked host (note that the Internet host authenticating the PSTN will not help here). This can be somewhat mitigated if the PSTN and Internet host use public-key cryptography to sign and encrypt the requests going back and forth. The biggest problem with public-key cryptosystems appears to be the lack of a central authority to distribute public keys and vouch for their authenticity.

With reference to figure 1, information flows across two interfaces: between the SPIRITS client and the SIP proxy, and between the SIP proxy and the SPIRITS servers. There is information exchange in the PSTN as well; between the SPIRITS client and the SCP, and between the SCP and the switches. However, we assume that the PSTN infrastructure is secure by the virtue of it being in a controlled environment. It is the Internet we have to be more concerned with since providing end-to-end security in an open network without trusted intermediaries is a daunting task.

To secure communications between the SPIRITS client and the SPIRITS gateway, and between the SPIRITS gateway and the UAs, public-key cryptography appears to be the means. The problem of key distribution and vouching for authenticity can be somewhat diminished if we assume that the PSTN operator will act as a trusted intermediary. However, this places the burden of on every UA to get a public key from the PSTN operator in order to use the system.

Privacy, trust and security are hard issues to address in a uniform manner. TLS appears to be the answer for providing encryption and S/MIME appears to afford privacy of the SPIRITS payload. We do not claim to have all answers yet. We are researching this issue [16] to find a happy medium between protecting privacy and enabling the service execution.

5. SPIRITS implementation

We have implemented a SPIRITS framework and have used it to demonstrate two services: a presence-based service and an Instant Messaging service. We used a specialized load on a Lucent Technologies 5ESS ® switch that generated call processing events. We authored two additional pieces of software: first is an SCP simulator that captured raw events from the switch and turned them into DPs. The SCP simulator also accepted subscriptions from Internet hosts and maintained a subscription database. When an event of interest occurred for a line, the SCP simulator sent out a notification to the Internet host for service execution. For all practical purposes, the SCP simulator emulates IN behavior for IP hosts. The second piece of software is a SPIRITS UA that we have implemented according to the latest SPIRITS protocol I-D [14]. Besides understanding the SPIRITS protocol, this UA also supports SIP-based instant messaging and presence extensions.

We now describe the services enabled by the SPIRITS framework implementation.

5.1 PSTN presence

Presence is a well-known concept on the Internet; however, thus far, it has not been exploited on the PSTN. This is despite the fact that the PSTN is a virtual storehouse of presence- (and availability-)related information. Every time a PSTN user receives a call or makes a call, presence is implied. It is imperative to point out that this is coarse-grained presence indeed, and nothing can be authoritatively said about availability. However, for many situations, coarse-grained presence suffices, as does a small window of availability. It is equally as imperative to point out that the manner to realize this service is an exercise in validating the SPIRITS architecture. No representation is made of the social, cultural, privacy and legal aspects of providing this service.

A PSTN user is presumably present at a certain line if she receives a call or makes a call. Once the said user hangs up, she is presumably available for another call for a short period of time. We exploit this facet of behavior to implement a PSTN presence-based service.

A sample call flow for this service is presented in figure 2. A SPIRITS UA running on an Internet host issues a subscription to 4 events: TA (Terminating Answer), OAA (Origination Attempt Authorized), TD (Terminating disconnect) and OD (Origination disconnect). These events comprise the presence status of a PSTN line. The subscription arrives at the SCP simulator where a database entry is updated for the PSTN line.

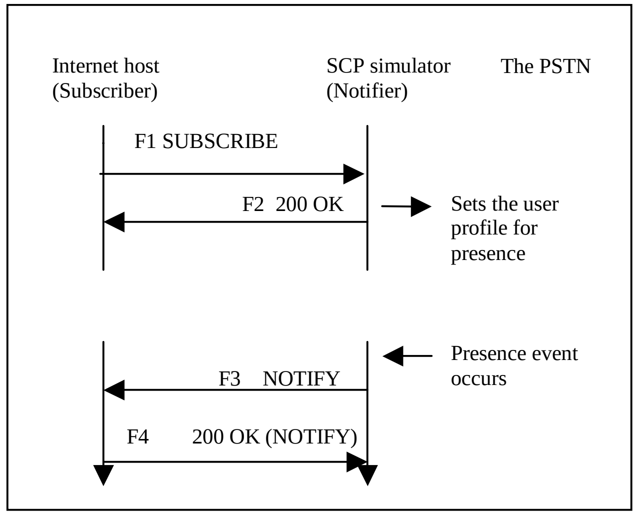

Figure 2: PSTN-presence

```
SUBSCRIBE sip:spirits.ih.lucent.com SIP/2.0
…
Event: spirits-INDPs
Accept: application/spirits-events, application/cpim-pidf+xml
Content-Type: application/spirits-events

<?xml version="1.0"?>
<spirits-event>
  <DP INDPs="OAA" Mode="N">
    <CallingPartySubaddress>16302240216</CallingPartySubaddress>
  </DP>
  <DP INDPs="TA" Mode="N">
    <CalledPartySubaddress>16302240216</CalledPartySubaddress>
   </DP>
   <DP INDPs="TD" Mode="N">
    <CalledPartySubaddress>16302240216</CalledPartySubaddress>
  </DP>
  <DP INDPs="OD" Mode="N">
    <CallingPartySubaddress>16302240216</CallingPartySubaddress>
  </DP>
</spirits-event>
```

Figure 3: SUBSCRIBE request

Figure 3 depicts the SIP SUBSCRIBE request that is issued in F1. Not all the SIP headers are shown for brevity; only the most important ones are depicted in figure 3. One of these headers is the “Accept” header. This header informs the receiver of the kinds of MIME types the sender is capable of receiving and decoding. The SPIRITS UA is capable of receiving and decoding two MIME types – *application/spirits-events* and *application/cpim-pidf+xml*. The former MIME type indicates support for SPIRITS and the latter MIME type indicates support for presence documents.

The SPIRITS UA also runs a presence-aware buddy-list application depicted in figure 5. When the PSTN line identified in the SUBSCRIBE request receives a call (or makes a call), the presence status of the PSTN line changes and the SCP simulator sends out a notification based on the events arriving at it from the 5ESS ® switch. There are actually two notifications sent (figure 3 only shows one); the first one is for the SPIRITS event that fired and the second notification is for the current presence and availability state of the PSTN line.

The notification for the presence and availability is depicted in figure 4 and the corresponding change it causes in the SPIRITS presence-aware buddy list is shown in figure 5. Once again, figure 4 only contains a partial list of headers for brevity; of interest is the body of the notification which is of MIME type *application/cpim-pidf+xml*, indicating that this is a presence document. The presence/availability status of the PSTN line is extracted from the <local:category> element of the document.

```
NOTIFY sip:vkg@il0015vkg1.ih.lucent.com SIP/2.0
…
Content-Type: application/cpim-pidf+xml

<?xml version="1.0"?>
<presence xmlns="urn:ietf:params:xml:ns:cpim-pidf"
          xmlns:local="urn:lucent-com:pidf-category"
          entity="pres:16302240216@lucent.com">
  <tuple id="981suu9">
    <status>
      <basic>closed</basic>
      <local:category>Busy since 13:20:32</local:category>
    </status>
    <contact>sip:16302240216@lucent.com</contact>
  </tuple>
</presence>
```

Figure 4: NOTIFY request

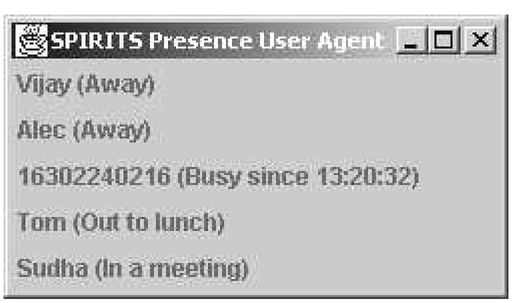

Figure 5: SPIRITS UA buddy-list: change in state

5.2 PSTN Instant Messaging

In this service, an Internet host subscribes to the TAA (Terminating Attempt Authorized) event of a PSTN number. As soon as the TAA detection point is fired, an instant message (using SIP MESSAGE extension [13]) is delivered to the Internet host detailing the callee's information and the time of call. This service is useful for situations where a user at work wants to monitor her home line to see who called, and if the call is important enough, it can be returned immediately.

6. Related Work

The work closely related to SPIRITS is PINT [18], 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 [19] 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.

7. Summary and conclusion

In this paper, we have presented the need for SPIRITS services, and a specific architecture targeted towards realizing them. The hardest part in an architecture that includes multiple entities and spans network topologies is identifying a good synchronization and message passing protocol. Our use of SIP as the protocol of choice is, we believe, a sound one. The entire PSTN is abstracted as a SIP UA for SPIRITS 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 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.

We also presented a SPIRITS implementation of a presence based service. This service is novel in that it introduces the PSTN to a well known Internet service – presence. Presence is a well-established concept on the Internet, however, thus far, it has not been exploited on the PSTN. SPIRITS easily enables such services.

References

[1] The IETF SPIRITS WG Home Page < http://www.ietf.org/html.charters/spirits-charter.html>

[2] L. Slutsman, I. Faynberg, H. Lu, M. Weissman, *The SPIRITS Architecture*, IETF RFC 3136, June 2001, < http://www.ietf.org/rfc/rfc3136.txt>

[3] I. Faynberg, J. Gato, H. Lu, and L. Slutsman, *The SPIRITS Protocol Requirements*, IETF RFC 3298, August 2002, <http://www.ietf.org/rfc/rfc3298.txt>

[4] Faynberg, I., Gabuzda, L., Jacobson T., and Lu, H-L. *The development of the Wireless Intelligent Network (WIN) and its relation to the International Intelligent Network Standards*, Bell Labs Technical Journal, Volume 2, Number 3, Summer 1997.

[5] Bergeren, M., Bollinger, B., Earl, D., Grossman, D., Ho, B-W., and Thompson, R. *Wireless and Wireline Convergence*, Bell Labs Technical Journal, Volume 2, Number 3, Summer 1997.

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

[7] International Telecommunication Union Recommendation H.323, Packet-based multimedia communication systems, November 2000.

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

[9] Brusilovsky, A., Gurbani, V., Gausmann, E., and Jain, A. 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.

[10] Lu, H-L., Faynberg, I., et. al. Pre-SPIRITS implementation of PSTN-initiated services, *IETF RFC 2995*, November 2000, <http://www.ietf.org/rfc/rfc2995.txt>.

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

[12] Rosenberg, J., Willis, D., Sparks, R., Campbell, B., Schulzrinne, H., Lennox, J., Huitema, C., Aboba, B., Gurle, D., and Oran, D. Session Initiation Protocol (SIP) Extensions for Presence, *IETF Internet-Draft*, Expires November, 2002, Work in Progress. <http://www.ietf.org/internet-drafts/draft-ietf-simple-presence-07.txt>

[13] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and Gurle, D., Session Initiation Protocol (SIP) extensions for instant messaging, *IETF RFC 3428*, December 2002, <http://www.ietf.org/rfc/rfc3428.txt>.

[14] Gurbani, V., Brusilovsky, A., Faynberg, I., Lu, H-L., Unmehopa, M., Vemuri, K., and Gato, J. The SPIRITS Protocol, *IETF Internet-Draft*, Expires April 2003, Work in Progress. < http://search.ietf.org/internet-drafts/draft-gurbani-spirits-protocol-03.txt>

[15] Unmehopa, M., Vemuri, K., et. al., On selection of IN parameters to be carried by the SPIRITS protocol, *IETF Internet-Draft*, Work in Progress, expires January 2003, <http://www.ietf.org/internet-drafts/draft-ietf-spirits-in-03.txt>.

[16] Gurbani, V., "Services in the PSTN requesting Internet services (SPIRITS) protocol security", IETF Internet-Draft, expires April 2003, work in progress. < http://www.ietf.org/internet-drafts/draft-ietf-spirits-security-00.txt>

[17] Faynberg, I., Gabuzda, M., and Shah, N., *Intelligent Network Standards: Their application to services*, McGraw Hill Publishing Company, 1997.

[18] Petrack, S., Conroy, L. 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?number=2848>, June 2000.

[19] Peterson, J., Vemuri, A., Session Initiation Protocol for Telephones (SIP-T): Context and Architectures, *IETF RFC 3372*, <http://www.ietf.org/rfc/rfc3372.txt>, September 2002.