A Systematic Approach for Closer Integration of Cellular and Internet Services

Vijay K. Gurbani and Xian-He Sun
Illinois Institute of Technology, Chicago, Illinois
Lucent Technologies, Inc./Bell Laboratories, Naperville, Illinois

Abstract

The road to 4G runs through 3G, 2G, and 2.5G networks. Until 3G is deployed ubiquitously, 2–2.5G networks and endpoints will dominate. Because of their widespread deployment and adoption, these endpoints can help ease the transition to an all-Internet 3G/4G network. We describe a standards-based service architecture and its implementation that allows 2–2.5G endpoints to participate as active actors in realizing Internet services like presence and instant messaging without the endpoints themselves being connected to the Internet. Our architecture also demonstrates service migration, wherein a successful 2G service such as short messaging service morphs transparently into its Internet equivalent, an instant message. Our methodology conserves precious radio access bandwidth by offloading such data services from the bandwidth normally allocated to voice. Our approach, as embodied in the service architecture, is to leverage the best of the Internet protocols (SIP, XML) and technologies (instant messaging, presence) to provide a general framework for personalized service specification and execution.

The Internet and cellular systems have been designed and implemented by people with different backgrounds in computers and communications, respectively, so their integration will not be an easy task. Such integration, however, can be considered a first step toward next-generation networks, where heterogeneous networks must work together in order to provide differentiated services to users in a seamless and transparent manner [1]. This integration is required on two planes: signaling (for services provided through signaling) as well as the media plane (for services provided through the voice channel). The work in this article pertains exclusively to the signaling plane.

The first-generation (1G) cellular network was an analog circuit-switched system. Mobile handsets were bulky; voice quality was poor and security nonexistent. 2G networks improved on the disadvantages and provided additional data services like short message service (SMS). 2.5G is an intermediate step toward 3G, utilizing Internet protocols and packet switching in portions of the cellular network, but, unlike 3G, the 2.5 network is not "Internet to the core." 4G networks are a step beyond 3G, providing data transmission speed equivalent to a local area network and more personalized services for its users [2].

Market conditions and technical realities dictate that future networks will be integrated and coexist with current ones; thus, 3G/4G networks will have to integrate with the current crop of successfully deployed wireless communication systems. Network operators have made substantial investments in 2–2.5G networks that can be leveraged as the migration continues to all-IP-based 3G/4G networks. It is certainly possible to be connected to the Internet from a 2.5G endpoint; however, currently that Internet connection is used for data services only; voice is not packetized and transmitted over it (in the future it will be possible to do so). Furthermore, services using the Internet connection do not interact with those on the voice channel to provide yet more innovative benefits of integrated networks to their users. Some examples will illustrate the claim in the preceding sentence.

First, consider availability. When a cellular subscriber powers her Internet-capable phone, it can — using the Internet connection — inform a buddy list manager that turns her presence indicator to "on." 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. Thus, it is impossible to derive an aggregate state of the subscriber based on only using one network and its protocols; more intelligence is required.

Second, consider the example of service migration. SMS allows a cellular subscriber to receive short text messages on her cellular phone. However, if she does not have physical access to her cellular phone, or it is rendered inoperable (drained batteries), she cannot receive SMS messages destined to the phone. It should be possible for the subscriber to inform the cellular network to make intelligent decisions on her behalf such that the SMS is converted to an Internet instant message and routed to her Internet-connected PDA in real time.

The remainder of this article is organized as follows. We enumerate the contributions of the article followed by a description of an architecture and use it as a backdrop to list pressing research challenges. We then contrast the architecture with related work. We present the results of our implementation by discussing key services it enables, and end with a conclusion outlining future work.

0890-8044/05/$20.00 © 2005 IEEE

Contributions

We employ a standards-based approach to arrive at an architecture. For the deployed cellular network, we use the wireless intelligent network (WIN) standards. WIN specifies the capabilities and protocols used by all participating entities in the cellular network. For the Internet, we use a new protocol developed in the Internet Engineering Task Force (IETF) called SPIRITS [3], which has been published as a proposed standard by the IETF.

A unique contribution of this work is a framework that allows discrete cellular network events to be percolated from the cellular network to the Internet for service execution. WIN provides an effective service platform for services executing in the cellular network; however, the current trend points toward providing services on the Internet. Our work enables this by transparently leveraging the ubiquitously deployed WIN architecture through SPIRITS.

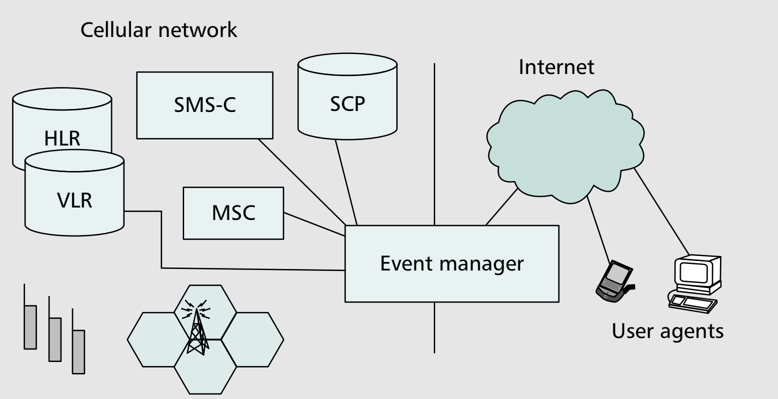
Accordingly, we present the architecture and its implementation that enables services detailed previously. It leverages the deployed 2–2.5G network infrastructure (cellular phones; mobile switching center, MSC; home location register, HLR; visitor location register, VLR; etc., as depicted in Fig. 1) to provide services that further integrate the Internet and the cellular network. The architecture makes possible presence, availability, location information, and service migration, even for those cellular endpoints that are themselves not connected to the Internet (some cellular subscribers may not have Internet-ready phones; others may have the phones but balk at paying higher fees for Internet access). Doing so has distinct advantages. For one, the cellular endpoint does not use precious radio bandwidth for such services. Second, the architecture introduces such services to subscribers who will readily use them if they do not have to upgrade their phone or pay extra money to connect to the Internet. Finally, it allows the network operator to recoup investments made in existing cellular networks while moving to 3G/4G networks.

Architecture

The architecture is based on separating the network on which the service executes from the one that provides events required for 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. Such networks present a rich palette of events on which Internet services can be built: registration, mobility, and text messaging are some of the events beyond normal call control that can influence Internet services.

Our architecture, depicted in Fig. 1, uses the publish/subscribe mechanism that has proved to be well suited for an event-based mobile communication model [4, 5]. User agents (software programs) on the Internet subscribe to events on the cellular network. When an event occurs, the cellular network notifies the user agent, which subsequently executes the desired service. The centerpiece of the architecture is the event manager, which straddles both networks. It insulates the cellular network entities from Internet protocols and vice versa. It is also responsible for authenticating user agents and maintaining subscription state so it can transmit notifications when an event subscribed to transpires.

The high-level description of the architecture is extremely simple; however, as we discuss next, this simplicity belies the complexity that becomes apparent upon closer inspection.



HLR:	Home location register. Primary database repository of subscriber information.
VLR:	Visitor location register. Maintains temporary records of subscribers.
MSC:	Mobile switching center. Provides cellular radio telephony switching functions.
SCP:	Service control point. Stores program logic for a service subscibed to by a cellular subscriber.
SMS-C:	Short message service center. Provides SMS service to subscribers.

Figure 1. *Architecture.*

Challenges

There are numerous research issues that must be addressed for widespread deployment. Below, we enumerate these areas and how they impact our understanding of the problem

The Business Model

Our architecture spans two different networks owned by potentially two different operators, the cellular service provider (CSP) and the Internet service provider. Getting them to work together is not as daunting as it initially appears. We envision that the CSP will distribute — through surface mail or by making them available for download on their Web site — specialized Internet user agents that encapsulate the services the CSP wishes to provide. For instance, to provide real-time presence and availability of cellular subscribers to users on the Internet, the CSP will distribute a user agent that sends a subscription containing presence and availability events to the event manager. The event manager interfaces with other entities owned by the CSP (HLR, MSC, etc.) such that it is notified when the cellular subscriber attempts to make or receive a call or powers up/down the cell phone. Upon notification, the event manager in turn notifies the Internet user agent.

There are two issues with the business model: roaming and billing. If a cellular subscriber roams into the network of another CSP, how does the new CSP notify the user agent? Due to the Internet security model (discussed later), the user agent does not have a trust relationship with the new CSP. There are two ways to address this: one, the new CSP sends the events to the origin CSP, which in turn, delivers them to the user agent (this assumes that the new CSP and origin CSP have an existing trust relationship; not an uncommon occurrence in current cellular networks). The second option is for the origin CSP to install the public key of the new CSP in the user agent, thus effectively acting as a trusted intermediary to rendezvous the new CSP and the user agent.

Another problem with roaming is that a new CSP may not support all required events, leading to the service subscribed to going unfulfilled. A way to combat this is to standardize an atomic set of events that will be supported by all CSPs (we have done this in SPIRITS), thus guaranteeing that, at the very least, the most common services work across providers.

An ISP recoups its investment by billing the owner of the

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 deregistration
UNREGNTWK	Network initiated deregistration
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 1. *Partial list of events in the cellular network.*

Internet user agent for access to the Internet. A CSP could provide bundled services at a flat rate, or it may choose to charge a minimal amount per service invocation. The question is, who does the CSP charge: the subscriber of a cell phone generating the events or the recipient of these events? To make things even more complex, for certain services — SMS to instant message, for example — the subscriber and recipient may be the same billable entity. Furthermore, the concept of billing in telephony is predicated on the notion of call duration. We feel that this model will become outdated as other services impact the duration of a call. For instance, if a user agent subscribes to the presence state of a cellular subscriber before making a call, it can be argued that subscription to the presence state itself is a billable item, besides the actual call, which may last for far less time than the subscription for presence.

Choosing Target Events

There are three sets of target events generated by the cellular endpoints that can be harnessed for Internet services. The first set consists of events that occur during call setup and teardown, such as an attempt to make a call, receiving a call, dialing digits, and far-end ringing, to name a few; the second set includes events unrelated to call setup, like cellular endpoint registration, deregistration, and location update. The final set consists of application-specific events, such as the arrival of an SMS message.

For call-related events we leverage WIN and the basic call state model for capability set 2 (CS-2) [6]. WIN is an architectural concept that includes many physical entities; the two most important for this discussion are the MSC and SCP (Fig. 1). CS-2 contains a set of events to control call processing in the MSC. The SCP is notified by the MSC whenever an event occurs. The SCP percolates the notification to the event manager, which in turn notifies the user agent. WIN also contains events for registration and location events associated with a cellular endpoint.

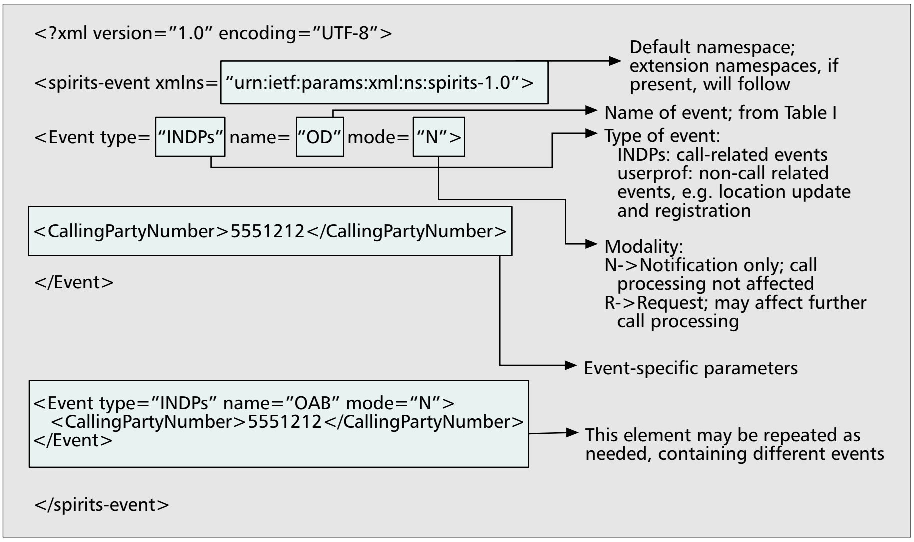
Since the number and type of events generated will vary among different CSPs, we propose a minimum set of events that is general enough to be supported by all CSPs. This set is culled from the WIN events related to the first two sets, call setup/teardown and location/registration updates. Application-specific events, which are difficult to quantify in a standard manner, will be handled separately. CSPs wishing to provide such services must use proprietary events that will depend on the specific service.

Table 1 contains a partial catalog of standard cellular events to provide a perspective to the reader on the richness of information available in the cellular network; a complete set is provided in [3, 7].

Choosing a Protocol

We evaluated three protocols: the Session Initiation Protocol (SIP), H.323, and Bearer Independent Call Control (BICC). BICC essentially allows existing telephone switches to route traffic over an asynchronous transfer mode (ATM) or IP network instead of traditional time-division multiplexed circuits. It has limited support for fostering Internet-type services. H.323, while more amenable as an Internet signaling protocol, lacks some key characteristics we were looking for in a protocol, such as support for asynchronous event notification, extensibility, and support for arbitrary payloads in signaling messages. SIP [8] is another signaling protocol from the IETF. It possesses all the important features we required and, in addition, has strong industry backing from wireline as well as wireless (3G) service providers. While SIP does not play a part in the signaling or voice transport for 2G and 2.5G endpoints, nevertheless some backend services (presence and availability) for these networks can still be made SIP-aware at an early stage to ease the eventual transition to an all-IP network. Thus, of the three we chose SIP.

Two important factors in favor of SIP were its support for arbitrary payload types and for asynchronous event notifications. SIP supports primitives to allow Internet hosts to subscribe to and receive subsequent notifications of changes in a monitored resource. In a publish/subscribe system, event filters provide a means for consumers to subscribe to the exact set of events they is interested in receiving. We represent the event filters as an Extensible Markup Language (XML) object and transport it in a SIP request. A subscription from a user agent is encapsulated as an XML object and routed using SIP to the event manager. The notifications from the event manager are also encapsulated as XML objects and routed to the Internet host over the SIP mesh.



```
<?xml version=”1.0” encoding=”UTF-8”>
<spirits-event xmlns=”urn:ietf:params:xml:ns:spirits-1.0”>
<Event type=”INDPs” name=”OD” mode=”N”>
<CallingPartyNumber>5551212</CallingPartyNumber>
</Event>
<Event type=”INDPs” name=”OAB” mode=”N”>
  <CallingPartyNumber>5551212</CallingPartyNumber>
</Event>
</spirits-event>
```

Default namespace; extension namespaces, if present, will follow

Name of event; from Table I

Type of event:
INDPs: call-related events
userprof: non-call related events, e.g. location update and registration

Modality:
N->Notification only; call processing not affected
R->Request; may affect further call processing

Event-specific parameters

This element may be repeated as needed, containing different events

■ Figure 2. *Understanding the XML payload.*

We have specified an XML schema, including a namespace used to describe the events. The schema is extensible to allow application-specific events. It is defined in its entirety in [3]; for illustration, Fig. 2 contains an XML document corresponding to this schema.

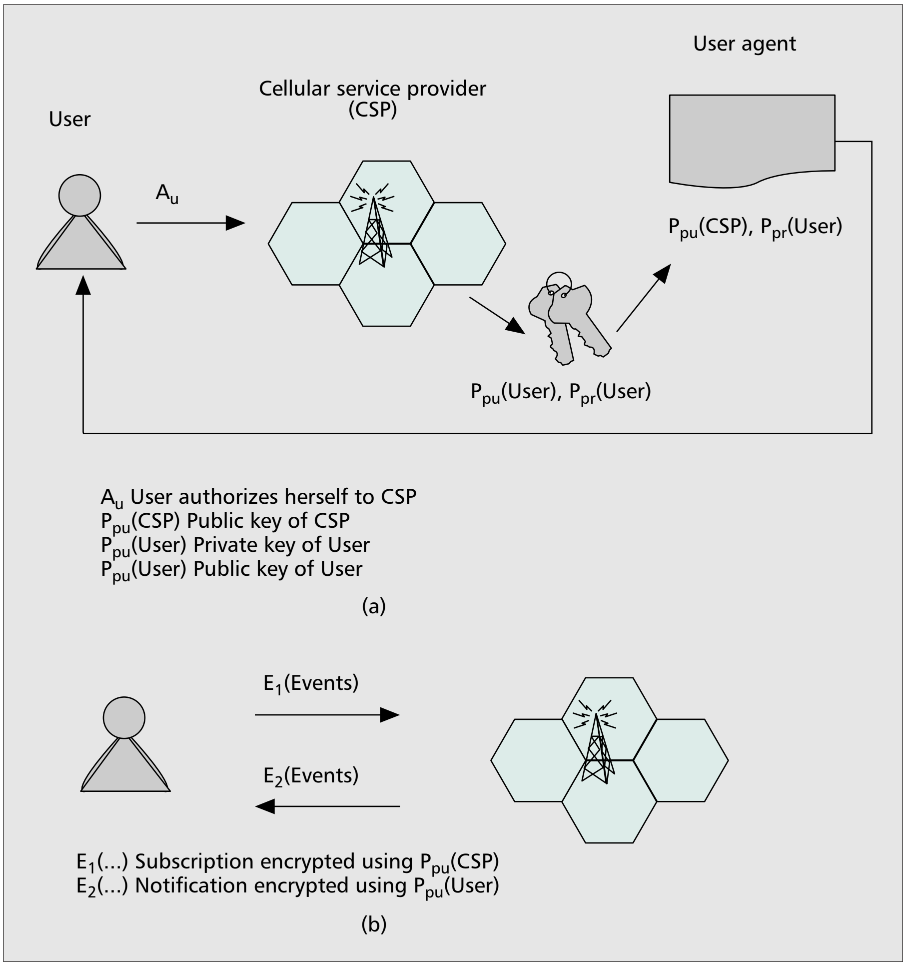


Figure 3. *The authentication process.*

Privacy and Trust

The events one subscribes to and the subsequent event notifications may relate to quite private information. The notifications have the potential to reveal sensitive location information or other damaging information (e.g., an SMS message from a broker to a client containing an account number). Maintaining privacy of this information is of paramount importance. Besides privacy, another important aspect is trust: the event manager must be sure that the subscriptions are coming from an authenticated user agent. Also, the user agent must ascertain that the notifications are coming from an authenticated event manager instead of a malicious hijacker acting as an event manager.

In order to authenticate and encrypt communications between two previously unknown parties on the Internet, public key cryptography is the best option. Two known problems with it are key distribution and the lack of a well-known and universally trusted certificate authority. We outline a method that mitigates both of these.

We assume the worst case scenario: all parties are unknown to each other. An Internet user authenticates herself to the CSP using a credit card, driver's license, or a pre-existing customer relationship with the CSP. The CSP's service management system creates a pair of keys — P_{pr}(User) and P_{pu}(User) — corresponding to private and public keys, respectively (Fig. 3a). The SMS then "burns" the public key of the CSP, P_{pu}(CSP), and the private and public key pair of the user in the user agent; P_{pu}(CSP) is also escrowed at the CSP. The user agent arrives to the user through a download link or mailed on a disk. When the user subscribes to certain events, the data is encrypted using P_{pu}(CSP). When the CSP sends a notification, P_{pu}(User) is used to encrypt the contents (Fig. 3b). With the respective key pairs available to both parties, privacy and trust concerns are adequately addressed. If a well-known certificate authority is not available, the CSP can act as one itself to issue the keys.

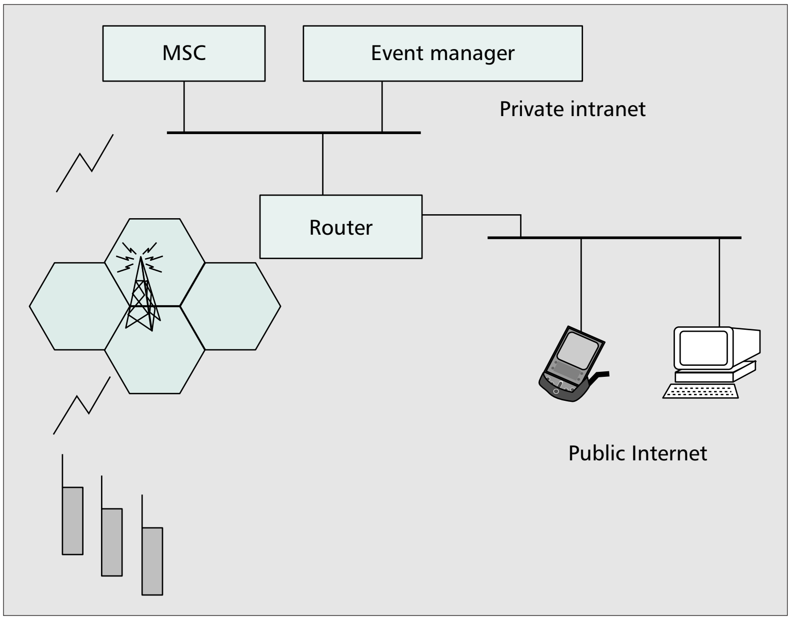
A final step in the privacy equation is a policy directive driven by the cellular subscriber allowing access to selected events for certain users (example: "Allow user Vijay K. Gurbani access to my location between 5:00 p.m.–7:00 p.m. every day"). The cellular subscriber can set such policies through a Web interface or by calling the CSP's customer service center. Since a relationship already exists between the CSP and the subscriber, the subscriber can use it to authenticate himself to the CSP and install specific policies.

Authentication, authorization, and accounting (AAA) is expected to be a primary focal point of services spanning multiple networks. Our privacy model does not require a user agent to have an a priori trust relationship with the CSP; however, in other systems, such as the integration of 802.11 and 3G, a 3G user roaming into an 802.11 hotspot may leverage the existing trust relationship between it and the 3G service provider by allowing the 802.11 provider to access AAA servers of the 3G carrier. Buddhikot *et al.* [9] discuss this issue in the context of allowing the 802.11 operator access to the user profile and policies of a 3G user stored in the 3G carrier's network.

Event Aggregation

The wireless communication network has myriad entities that contribute to events. An important question for an implementation is how to best aggregate these events. For instance, location-based events update the HLR while call-related events are generated in the MSC. Application-specific events (like SMS) will update specialized application servers. Instead of requiring each entity to send notifications based on the events, it is best to aggregate the events in one location. Figure 1 demonstrates a logical entity, the event manager, where events can be aggregated.

To provide a specific service, the event manager can be collocated with the cellular network entity responsible for events leading to the service (for call-related events it could be co-resident on a SCP; for providing location information it can be co-resident on an HLR). However, to provide a wide set of services, a standalone event manager with a simple message-passing interface to all relevant cellular network entities will be required. We do not make any assumptions of this interface since it occurs with entities entirely in the CSP's environment. In our implementation, all events were aggregated at the event manager.



■ Figure 4. *Laboratory setup.*

Scalability of the Event Manager and Performance

It is a complex task to gather events in the cellular network. The event manager has to react with a number of entities that are generating events, as discussed above. While in our implementation we used one event manager, it is anticipated that during deployment a CSP will use a federation of event managers, possibly locating each event manager in close proximity to an MSC.

In addition to event aggregation, the event manager handles SIP transactions, parses XML documents, authenticates user agents, updates subscription databases, and transmits notifications when an event of interest occurs. It is a crucial part of our architecture.

A related challenge will be quantifying the performance of cellular entities such as the SCP, MSC, HLR, and VLR, as they will need to notify the event manager of the requisite events.

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 precursors 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 that exhibit “awareness,” such as Sun Microsystem’s Awarenex [10] and Milewski’s Live Address Book [11]. 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 the work in [11] indicates, users will not consciously remember to always update their status. Our architecture, by contrast, updates the status automatically.

Berkeley’s ICEBERG project [12] 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 (iPOPs) and many ICEBERG access points to isolate the access network from the overlay network. The iPOPs are coordinated by a centralized clearinghouse that 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 user agent generating and sending events to another user agent that then executes the services.

Stanford’s Mobile People Architecture (MPA) is another effort to bridge the wireless and Internet networks [13]. However, its main goal is to route communications to a mobile person, independent of the 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 (APIs) 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 article could serve as an “on the wire” protocol behind 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. As discussed earlier, our architecture mitigates both of these shortcomings.

Commercially 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 than those associated with detecting presence only. The next section introduces other services besides presence that are enabled by our architecture.

Implementation Issues and Results

During implementation of the architecture, we addressed the various research challenges articulated earlier. Some challenges were successfully overcome while others had to be worked around in order to realize a functional system.

Even though the call-related events in our architecture are based on WIN trigger points, we were unable to use a WIN-capable MSC. Instead, we modified a generic MSC such that at appropriate points in call processing, it would act as a WIN-capable MSC and notify the event manager when an event of interest occurs; the event manager played the part of an SCP. It should be noted that our inability to use WIN-capable cellular entities was a necessary shortcut; the events subscribed to are standard WIN triggers, and the manner in which the MSC notifies the event manager is no different than if WIN-capable cellular entities were used in the first place.

We implemented a standalone multithreaded event manager that ran on a general purpose computer using the Solaris

5.8 operating system. The event manager employed a simple TCP-based message passing protocol to the MSC. The event manager received subscriptions from user agents and maintained a database. Events from the MSC were matched against the database to determine if a notification should be sent out. In our implementation, the MSC sent all events to the event manager without filtering (which would cause an unnecessary burden on the MSC). The filtering criteria (which event needed a notification to be sent out) were performed by the event manager.

```
SUBSCRIBE sips:em.csp.net SIP/2.0
…
<Event type=”userprof” name=”REG”>
   <CalledPartyNumber>7735551212</CalledPartyNumber>
</Event>
<Event type=”userprof” name=”UNREG”>
   <CalledPartyNumber>7735551212</CalledPartyNumber>
</Event>
<Event type=”INDPs” name=”OA”>
   <CallingPartyNumber>7735551212</CallingPartyNumber>
</Event>
<Event type=”INDPs” name=”TA”>
   <CalledPartyNumber>7735551212</CalledPartyNumber>
</Event>
```

■ Box 1. *XML payload.*

```
<Event type=”userprof” name=”LUSV”>
   <CalledPartyNumber>8475551212</CalledPartyNumber>
</Event>
<Event type=”userprof” name=”LUDV”>
   <CalledPartyNumber>8475551212</CalledPartyNumber>
</Event>
```

■ Box 2. *Second subscription.*

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

■ Box 3. *Delivering an SMS message as an instant message.*

To simulate real-world conditions, we distinguish between a protected intranet and the public Internet; the event manager and MSC reside on a protected intranet, and the user agents were assumed to be on the public Internet. Entities on the protected intranet are assumed to belong to a single CSP, so security is not a concern for communication between them. But all communication between the intranet and the Internet is encrypted. Encryption was provided using OpenSSL v0.9.7b. We acted as a self-signed CA and issued certificates for the CSP and user agents.

Our laboratory also consists of simulated cells, base stations, and cellular endpoints that received signals from the base stations. Using simulation tools, we are able to simulate signal attenuation, which reproduces the movement of cellular subscribers between cells. Figure 4 depicts the laboratory setup.

We implemented three benchmark services: presence for cellular subscribers based on events generated by their phone (registration, deregistration); availability of cellular subscribers based on events leading to initiating or receiving a phone call; and finally, SMS to instant message service migration. We discuss these in more detail next by illustrating a scenario that uses these services.

Bob comes into work only to discover that he has left his cellular phone (630-224-0216) at home. He uses SMS to keep in touch with his colleagues, so he would like to continue receiving SMS messages in real time on his laptop or PDA. Bob is expecting a colleague, John, to drop by his office for a meeting. Thus, he would like to keep track of John's location through John's cellular phone (847-555-1212) so that he can prepare to meet John upon his arrival. Finally, Bob has subscribed to the presence and availability status of Alice, his wife, through her cellular phone (773-555-1212). John and Alice have already installed policies with their CSP's network that allow Bob access to the events generated by their cellular phones.

Bob has downloaded a specialized user agent from his CSP that allows him access to these services. He starts his user agent. The user agent authenticates Bob to the CSP's event manager and issues a subscription for the events in which Bob is interested. In order to receive presence and availability notifications for Alice, Bob's user agent issues a SIP SUBSCRIBE request with the XML payload shown in Box 1.

Note that the request is destined to the CSP's domain (em.csp.net) through a secure transport (sips). Using the keys embedded in the user agent, the request is encrypted to protect from eavesdroppers, and the CSP can rest assured that it is indeed from Bob. The body of the request contains an event filter that consists of four events: registration (REG), deregistration (UNREG), making a call (OA), and receiving a call (TA). These first two events are sufficient enough to provide a near approximation to a presence service for users using cellular phones, and the latter two events provide an availability ("on a call," "available") indicator.

Bob's user agent also sends out a second subscription to John's events. The event filter in this subscription contains the same events used for presence and availability and, in addition, the events for location (LUSV and LUDV) (Box 2).

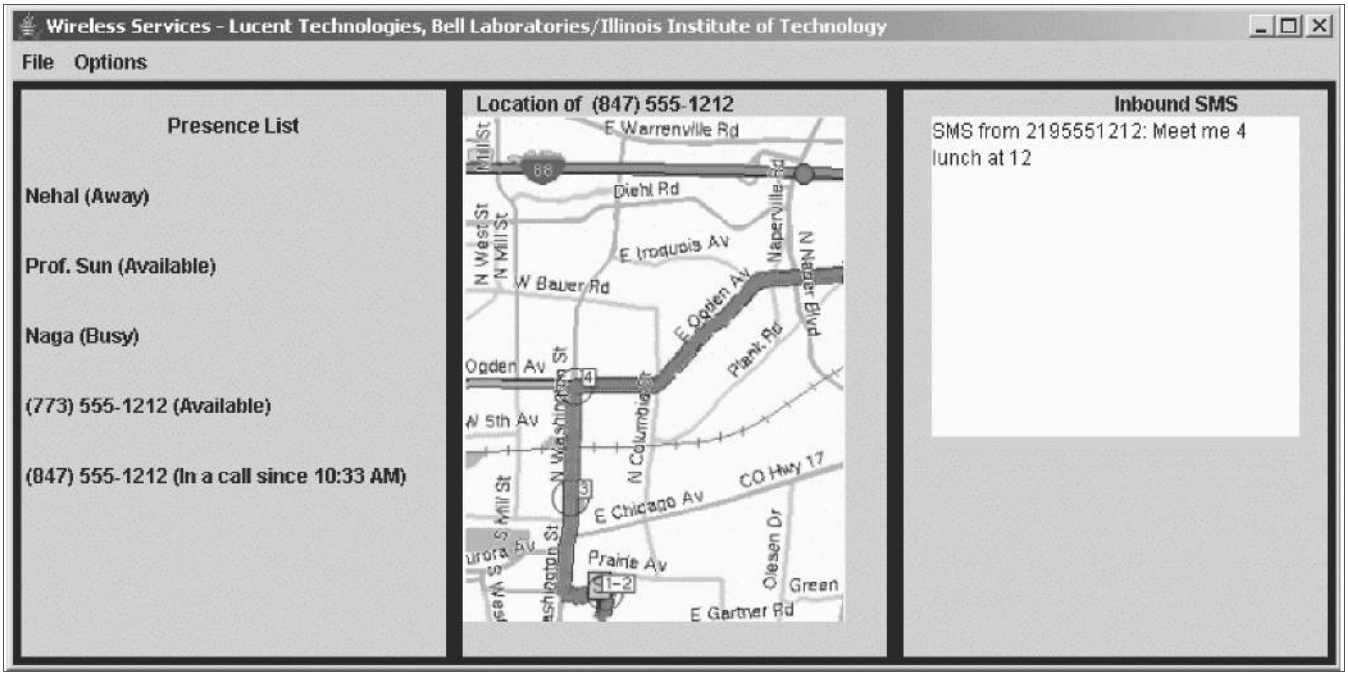
Finally, the third subscription transmitted by Bob's user agent contains an XML object that contains directives that allow the cellular network to transform an SMS message destined to Bob's phone into an instant message and deliver it to Bob's user agent (Box 3).

The event filter directs the cellular network to send an instant message consisting of the SMS message to Bob (at the address sips:bob@isp.example.com) if the network cannot successfully deliver the SMS message to Bob's cellular phone (delivery type = failure).

Figure 5 contains the graphical user interface Bob sees when his user agent executes. It contains three panels; the leftmost one consists of a presence list, the center one displays real-time location of a cellular subscriber, and the rightmost panel is used to display incoming instant messages to Bob.

When Alice powers her cellular phone and it registers with the cellular network, the event manager traps this event and sends a SIP NOTIFY request to Bob's user agent. As a result of the NOTIFY request, Bob's user agent updates the GUI such that the current status of Alice's cellular phone is displayed in the presence list manager (leftmost panel in Fig. 5). Note that besides depicting John's presence, the system also displays his availability ("in a call since 10:33 a.m.").

As John travels to rendezvous with Bob, the cellular network keeps track of his location. Whenever John's travels take him to a new cell site, the cellular network updates John's location. The event manager traps this event and sends a SIP



Nehal (Away)

Prof. Sun (Available)

Naga (Busy)

(773) 555-1212 (Available)

(847) 555-1212 (In a call since 10:33 AM)

SMS from 2195551212: Meet me 4 lunch at 12

■ Figure 5. *Services prototyped.*

NOTIFY request to Bob's user agent. Included in the NOTIFY request will be a cell ID used to display a map of the geographic area John is in (center panel in Fig. 5). There are a variety of ways to transform a cell ID into a location that can be displayed on a map. Our approach involved the event manager dynamically sending a series of thumbnail images of the surrounding vicinity to the user agent. The user agent used the cell ID as an index to display the correct image. However, how best to resolve a cell ID into a location is an open question, and more work is needed to arrive at an optimal manner.

When someone sends an SMS to Bob, the cellular network attempts to locate Bob's cellular phone. Upon failing to locate it (remember, it is at home and turned off), the event manager is informed along with the SMS itself. The event manager transforms the SMS message into a text message and delivers it — in real time — to Bob's user agent using SIP extensions for instant messaging.

Conclusions and Future Work

This work has demonstrated the role of harnessing the events occurring in the cellular network to execute services on the Internet. Radio spectrum is a precious resource; our system preserves bandwidth by offloading Internet services from it, thus making it fully available for voice services. We have presented the work in the context of utilizing currently deployed cellular infrastructure. Service providers have invested a tremendous amount of capital in 2G and 2.5G networks. As the move toward 3G/4G continues, the existing network can be leveraged to preserve capital investment.

The standards-based approach of this work also makes it possible for third-party service providers to benefit. They can participate in the ecosystem by using the events to create value-added services executed on a CSP's network.

Going forward, we intend to extend this work by quantifying the performance and scalability of the event manager and other cellular entities. 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 policy framework developed at Bell Laboratories, called Houdini [14], which has been used to demonstrate, among other features, user specified preferences that are filtered through a rules engine before location information is shared.

References

[1] A. Jamalipour and P. Lorenz, "Merging IP and Wireless Networks," *IEEE Wireless Commun.*, Oct. 2003, pp. 6–7.

[2] B. G. Evans *et al.*, "Visions of 4G," *Elect. and Commun. Eng. J.*, vol. 12, no. 6, Dec. 2000, pp. 293–303.

[3] V. Gurbani *et al.*, "The SPIRITS (Services in PSTN Requesting Internet Services) Protocol," IETF RFC 3910, http://www.ietf.org/rfc/rfc3910.txt, Oct. 2004.

[4] R. Meier, "Communications Paradigms for Mobile Computing," *ACM Mobile Comp. Commun. Rev.*, vol. 6, no. 4, Oct. 2002, pp. 56–58.

[5] G. Cugola and H-A. Jacobsen, "Using Publish/Subscribe Middleware for Mobile Systems," *ACM Mobile Comp. Commun. Rev.*, vol. 6, no. 4, Oct. 2002, pp. 25–33.

[6] I. Faynberg *et al.*, "The Development of the Wireless Intelligent Network (WIN) and Its Relation to the International Intelligent Network Standards," *Bell Labs Tech. J.*, 1997, pp. 57–80.

[7] V. Gurbani and X.-H. Sun, "Terminating Telephony Services on the Internet," *IEEE/ACM Trans. Net.*, vol. 12, no. 4, Aug. 2004, pp. 571–81.

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

[9] M. Buddhikot *et al.*, "Integration of 802.11 and Third-Generation Wireless Data Networks," *Proc. 2003 IEEE INFOCOM*, Mar.–Apr. 2003, pp. 503–12.

[10] J. Tang *et al.*, "ConNexus to Awarenex: Extending Awareness to Mobile Users," *Proc. ACM SIGCHI*, Mar. 31–Apr. 4, 2001, Seattle, WA.

[11] A. Milewski and T. Smith, "Providing Presence Cues to Telephone Users," *Proc. ACM Conf. Comp. Supported Cooperative Work (CSCW)*, Dec. 2–6, 2000.

[12] H.J. Wang *et al.*, "ICEBERG: An Internet-Core Network Architecture for Integrated Communications," *IEEE Pers. Commun.*, Special Issue on IP-Based Mobile Telecommunications Networks, vol. 7, no. 4, Aug. 2000.

[13] P. Maniatis *et al.*, "The Mobile People Architecture," *ACM Mobile Comp. Commun. Rev.*, vol. 3, no. 3, July 1999, pp. 36–42.

[14] R. Hull *et al.*, "Enabling Context Aware and Privacy-Conscious User Data Sharing," *Proc. 2004 IEEE Int'l. Conf. Mobile Data Mngmt.*, 2004.

Biographies

VIJAY K. GURBANI (vkg@lucent.com or vkg@iit.edu) works in the Wireless Research and Development unit of Lucent Technologies, Inc. He holds a Ph.D. in computer science from the Illinois Institute of Technology, and M.Sc. and B.Sc. degrees, both in computer science, from Bradley University. His research interests are Internet telephony services, signaling protocols, and pervasive computing in the telecommunications domain. He holds one patent and has six applications pending. He is a member of the ACM and the IEEE Computer Society.

XIAN-HE SUN is a professor of computer science at the Illinois Institute of Technology (IIT), and director of the Scalable Computing Software (SCS) laboratory at IIT. Before joining IIT he was a post-doctoral researcher at Ames Laboratory, a staff scientist at ICASE, NASA Langley Research Center, an ASEE fellow at the Naval Research Laboratory, and a professor at Louisiana State University.