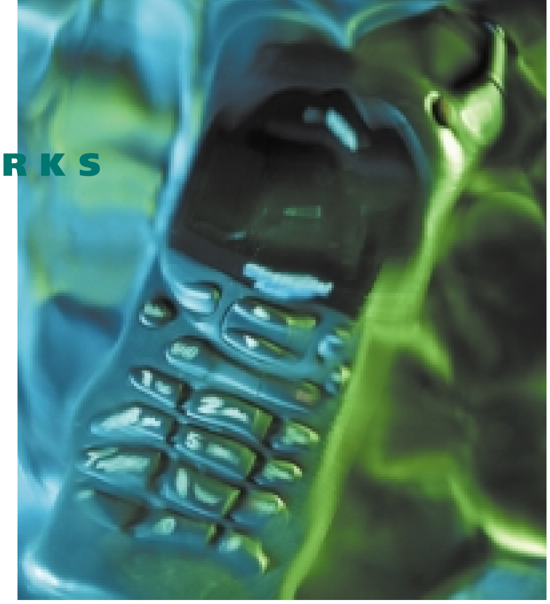


Truly ubiquitous and seamless integration of wireless networks and services will require clear interfaces between IP and telecom.

Bhaskaran Raman, Helen J. Wang, Jimmy Shih, Anthony D. Joseph, and Randy H. Katz

The Iceberg Project: Defining the IP and Telecom Intersection

Like many IT people today, you probably carry both a cell phone and pager. Vendors like Skytel are already offering services that combine information delivery and e-mail access (both receiving and sending), but you usually have to buy yet a third, IP-aware device. But who wants to carry three lumps around? Eventually, users need to be able to access services through any device via any network.

Browsers are becoming the interface of choice for applications like personal information access, e-commerce, network management, and collaboration tools. However, users would like this functionality in a handheld, wireless device.

These examples and the range of services available in third generation cellular networks point to a convergence of IP and telecom in wireless handheld devices, which will provide powerful new capabilities for users. But the revolution inherent in these devices will never be fully realized without a consistent framework for integrating services across networks. For example, how do you integrate services across networks as diverse as the PSTN (Packet-Swtiched Telephone Network) and the Internet? Where do you store a directory that tracks user device IDs? How do you scale services to a large user base? Each device needs a consistent framework to handle these tasks.

Inside

Today, the intersection of IP and telecom are uncharted waters, ungoverned by any standards, de facto or otherwise. Before conflicting standards and technologies arise, the Iceberg project is exploring the research issues in establishing the interfaces—and it aims to answer some hard questions along the way.

ICEBERG

We are developing Iceberg (Internet-based core beyond the third generation) to go beyond the third-generation cellular and personal communication services (PCS) models. Iceberg offers an Internet-based integration of telephony and data services across different access networks. Its prime design goals are extensibility, scalability, and personalized communication. We leverage the Internet's low cost of entry for easy service creation, deployment, and integration.

Ideally, we want to use communications devices like in the personalized, integrated communication scenario depicted in Figure 1. In this scenario, users can have calls redirected to different endpoints based on parameters like time of day and caller ID. They can also receive e-mail headers as paging messages or have a cell phone read the e-mail, transforming the message into audio.

However, the current communication architecture can't support this scenario because it doesn't have a way to integrate services from heteroge-

1520-9202/99/$10.00 © 1999 IEEE

neous networks. The Iceberg project aims to enable Potentially Any Network Service (PANS), meaning that any service can be accessed from any device via any network, for example, accessing e-mail via a cell phone.

The Internet Protocol (IP) provides a simple, packet delivery service model, which can serve as a spanning layer across heterogeneous access networks. As Figure 2 shows, Iceberg's architecture integrates services from heterogeneous networks like the PSTN, the Global System for Mobile Communications (GSM), and pager networks to build innovative applications that form a core on top of the Internet.

Iceberg leverages the Internet's client-proxy server model to provide customizable communication services. We treat the Internet as the switch that routes information such as telephone calls or e-mail, integrates diverse networks and devices, and enables user-level service creation and customization. The "Design Goals" and "An Architecture for Integrated, Wireless Services" sidebars describe how Iceberg uses the Internet as the glue for integrating various networks.

NINJA: ICEBERG'S EXECUTION ENVIRONMENT

Iceberg's primary goal is to make services easily deployable and accessible. The Ninja project at UC Berkeley is looking at the research issues in making Internet services scalable (able to support many thousands of concurrent users), fault-tolerant (able to mask faults in the underlying server hardware or software), and highly available (resilient to network and hardware outages). A *base* is a service-execution environment on a cluster of workstations located in the network infrastructure. Examples of current and future Ninja services include

- a jukebox service that plays streaming audio over the Internet,
- an instant messaging service, and
- a persistent mail store that supports a large number of users.

Figure 1. Integrated communication requires accessing the same services from various devices and locations.

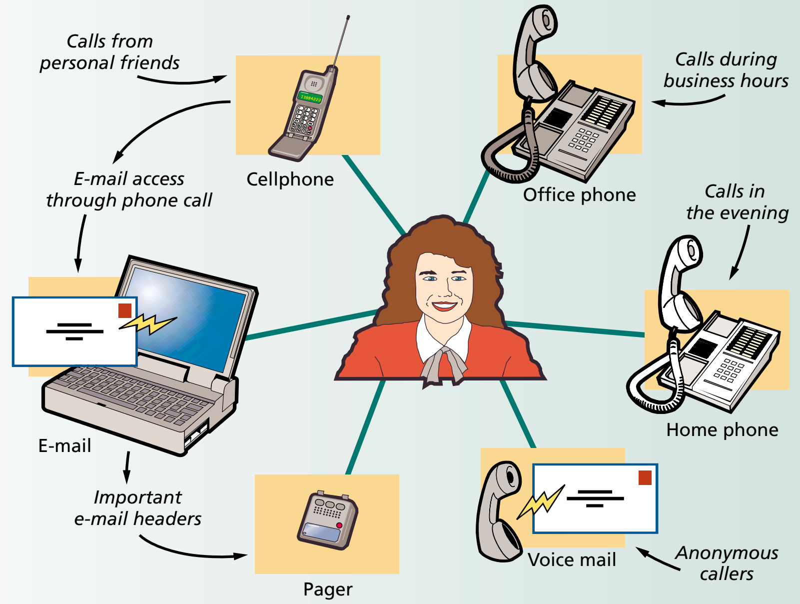
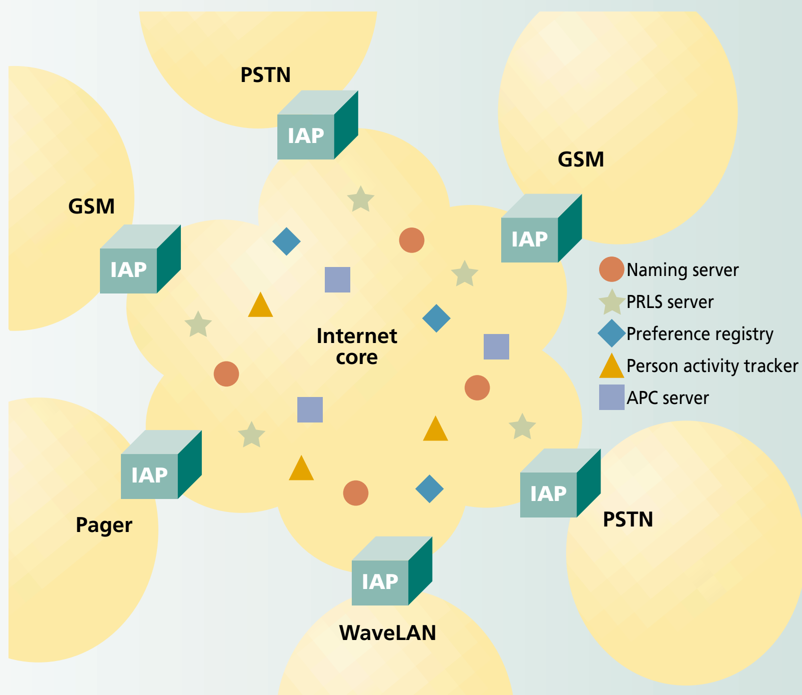


Figure 2. Iceberg's access points (IAPs) provide an interface between different communication networks and the Internet core.



Iceberg uses Ninja's service execution environment for its components as well as for additional, enhanced data and telephony services.

In addition to its execution environment, Ninja provides easy service composability. Operators are units of computation that have well-defined interfaces. For example, software that converts pulse code modulate audio into ASCII text through a Java remote message invocation interface is an operator. A path is a set of operators strung together using connectors—an abstraction of a data flow. For example, a series of coder-decoder operators followed by a speech-to-text operator is a path.

Automatic path creation (APC) is a process that sets up a data path of operators given any two end points. Iceberg uses this concept to abstract the data flow between devices in a call session, which makes the goal of any-to-any data transformation and transport easy to achieve.

FUNCTIONAL ARCHITECTURE

Iceberg's architecture has several unique components that distinguish it from other services that offer personalized integrated communication and call handling.

Iceberg access points

Iceberg access points are software agents within the Iceberg network that offer call establishment and control services on behalf of communication end devices. Iceberg's design separates the functionality of IAPs from gateways. Network-specific gateway hardware interconnects an access network with an IAP. This decoupling of functionality not only makes the design and implementation of the IAP access-network independent, it also yields better fault tolerance. When a gateway fails, a call is usually lost. In contrast, when an IAP fails, another IAP can take over and continue serving the ongoing phone call.

Call control

The call control protocol alters the call state in a call session. Call state includes calling party ID, current call status (active or on hold), dataflow information (IP address and port number of the data stream end-points, and location of transformation agents, if any), and the ID of serving IAPs. The call setup process establishes the call states on the serving IAPs.

The complexity in call control is usually in the reliable propagation of call state changes to all the IAPs in a call session. The dynamic characteristic of a call session makes this more difficult because IAPs may restart or new IAPs may join an existing call-session to serve new call parties. This requires a call session that is independent of the participating IAP locations, yet responsible for reliably delivering call state changes to all involved IAPs.

A multicast channel can serve as a call session mechanism because it allows the IAP location to be independent of call sessions. In this mechanism, each IAP periodically announces its portion of the call state to the multicast channel. The IAP also "listens" to the multicast channel and gathers the other call states from other IAPs. Each IAP maintains the complete call state for a call

Design Goals

Iceberg is a communication integration project that offers these features as part of its design.

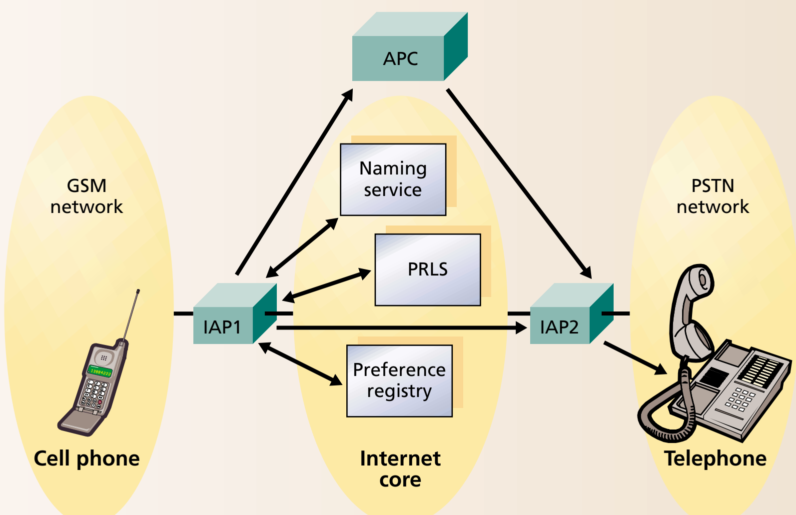
- *Potentially any network services (PANS):* **The goal of PANS is to permit any service to be transparently accessed through any end device via any access network.**
- *Personal mobility:* **The person (not the communication device) is the communication end point. We can use a single identity for an individual to forward information to the desired communication end point.**
- *Service mobility:* **Seamless mobility from one device to another during a service session is, for example, switching from a cell phone to an IP-phone in the middle of a conversation. This requires separating control (signaling, semantic information, and service metadata) and data information for each network and propagating this information across each network boundary.**
- *Giving control to the callee:* **This means shifting the locus of control from the caller to the person being called. In current communication networks, a caller chooses how (and when) to reach a callee. To shift control to the callee, users must be able to customize communication services such as call forwarding, call waiting, and service transformation.**
- *Scalability:* **Systems must incrementally scale to operate in large geographic regions and support hundreds of thousands of simultaneous calls.**
- *High availability and fault tolerance:* **Network components must be available 24 hours a day, seven days a week; the architecture must tolerate failures gracefully and hide failures from users.**
- *Security, authentication, and privacy:* **A carefully designed trust model and security mechanism is necessary because many of Iceberg's components are part of the untrusted Internet.**

An Architecture for Integrated, Wireless Services

Iceberg's architecture consists of an Internet core that includes these unique components.

- ***Iceberg access points*: The interface between an access network and the Internet core network, an IAP acts as the service transformation agent, allowing service access across networks.**
- ***Preference registry*: A mechanism for storing and processing user preferences. Services can query this registry to get the user's current preferences, such as the current preferred end point for receiving calls.**
- ***Preference registry location service (PRLS):* PRLS provides a location-independent means for locating the user's preference registry over a wide area.**
- ***Person activity tracker (PAT):* This component interoperates with the preference registry to track dynamic information such as the user's current location or the call state at a particular end point. PAT would report, for example, when the user's office phone is busy or when the user moves out of GSM cellular coverage range. The preference registry uses information from the PAT to process the user profile.**
- ***Naming service:* This component facilitates personal mobility by mapping the user's different end points onto the user's identity. Name servers in Iceberg communication network maintain this mapping.**
- ***Data path creation:* This mechanism performs any-to-any data transformation and transport functions such as converting from GSM-encoded audio to ASCII text. We use the Ninja project's paths and automatic path creation (APC) concepts to integrate services across heterogeneous end devices. The APC service creates paths that handle the data flow and conversion between any two end points.**

Figure A. Iceberg's components work together to route a call from a cell phone to an office PSTN phone.



IAP1 intercepts a cell-phone call, uses the naming service to find the callee's identity, uses the PRLS to locate the callee's preference registry, and finds the callee's current preferred end point in the preference registry. In the remaining steps, IAP2 directs the call to the callee's preferred end point—his office phone. The two IAPs have to contact the APC service before completing the call to set up an appropriate path to abstract the data flow.

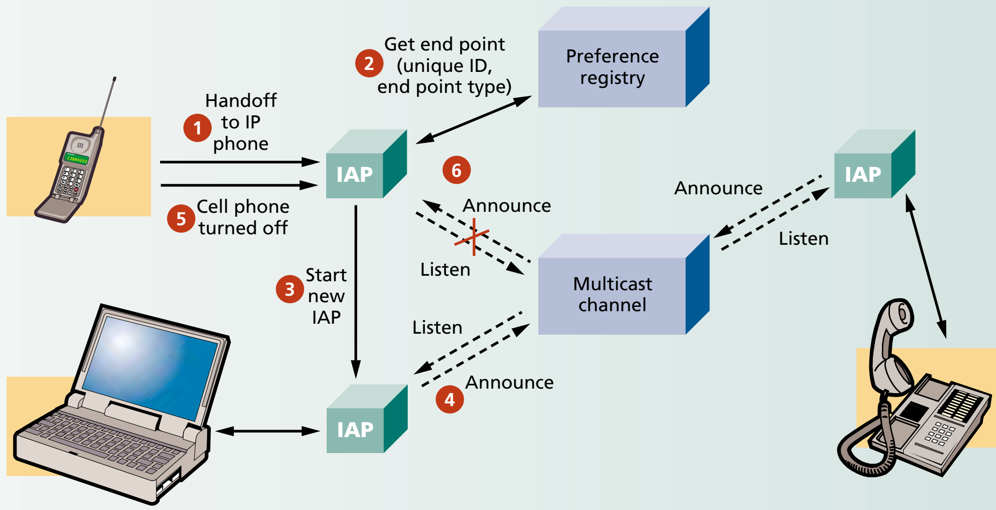
session. In this announce-listen model, periodic retransmission over the call's lifetime ensures the reliability of new call state propagation.

Using a single multicast address per call session poses a scalability problem because of the flat IPv4 multicast address space. The advent of IPv6 or Simple Multicast would eliminate this problem. If multicast doesn't succeed in the future, Iceberg will require an application-level call session maintenance agent that is independent of IAP locations and maintains the group membership of the participating IAPs. Other unresolved issues in our design include allocating multicast addresses and choosing the appropriate call state time-out values and announcement periods.

Service handoff: enabling service mobility

Our integration architecture enables service handoff, a novel feature that allows users to switch between their communication devices during a phone call. Service mobility is a powerful new concept that makes service hand-off possible. In telecommunication, the term "service mobil-

Figure 3. A service handoff scenario.



A call session has been established between a cell phone and a PSTN phone, and the two serving IAPs are periodically announcing and listening to the multicast channel. Now, the person using the cell phone walks into her office, and wants to use an IP phone (laptop) to continue her conversation. Step 1: The serving IAP receives an indication of a service handoff (triggered by, say, the user's key-press). Step 2: The IAP looks up the preferred end point (the laptop in this case) in the preference registry. Step 3: With the end point information and the multicast address of the call session, the IAP launches a new IAP serving the IP phone. Step 4: The new IAP creates a new data path between the IP phone and PSTN phone (perhaps using multicast for data transport). This IAP starts participating in the multicast session for the call session management. Steps 5 and 6: The cell phone is eventually turned off, and the original serving IAP tears down the original data path and stops announcing and listening to the multicast session. This completes the service handoff.

ity" usually means the ability to share user service profiles across different networks—it does not refer to the ability to cross a network boundary in the middle of a service.

Service hand-off is also a demonstration of personal mobility—our architecture identifies a call session by the people involved in the call, not by the devices they are using. Therefore, switching to a new device doesn't imply switching to a new call session. Figure 3 illustrates a typical service handoff scenario.

Preference registry

The preference registry captures user-specific personalization parameters. There are two separate but related issues with user-preference specification:

- how the system represents, stores, and processes user preferences; and
- building a reasonable user-interface for the user to specify personal preferences.

Although we focus on the first issue, the user interface is also important, especially for complicated services. For example, feature interaction—where users don't understand the implications of subscribing to multiple related services—has proven difficult to handle even for limited service models

Figure 4 shows the inputs to the preference registry and the output. User preference is essentially a function of inputs such as caller-ID, time of day, or user location. The function's output is the preferred end point—the user's cell phone or e-mail ID. We classify the inputs to the preference registry into two main categories:

- per-call information such as caller-ID or the caller end-point type, and
- dynamic information such as the user's location or call state (who the callee is currently talking to).

The person activity tracker (PAT) collects the dynamic-information inputs; the preference registry uses this in deciding its output. The preference registry is a user-specified function. In our experience thus far, we have found that a rule-based or procedural scripting language is useful for modeling a wide range of user preferences. In the example shown in Figure 5, the user wants to redirect communication to her office (9:00 a.m.-5:00 p.m.), home (5:00 p.m.-12:00 midnight), or voice-mail (off hours) based on the time of day.

The preference registry is an Iceberg component that runs on a Ninja base—it has persistent state (the user profiles), and it needs to be scalable and highly available.

Naming service

The naming service maps the Iceberg user's multiple devices onto the same logical entity. In this context, the term "user" could refer to a person or any abstract entity in real life—for example, an airline company's customer service number. Each of the user's service end points (or devices) is associated with a *service-specific ID*. For example, a user could have a telephone number, a pager number, and an e-mail ID. We call these service-specific IDs and not network-specific IDs because a user could have multiple IDs for different services in the same network, for example, an e-mail ID, a voice-over-IP end point, and an Internet ICQ-ID. To achieve personal mobility, our architecture maps a user's service-specific IDs to a *unique ID* that distinctively identifies the user in the Iceberg network.

Name mapping refers to mapping from a service-specific ID to the user's unique-ID. To satisfy our wide-area requirement, name mapping should be distributed, similar to the way the Domain Naming System (DNS) handles name mapping. Name mapping is required only when dealing with devices that don't know about Iceberg's unique IDs, for example, when a caller uses a cell phone to dial a callee's home phone number. If the calling device can provide the callee's unique ID directly, this mapping is not required.

We use a hierarchical approach to distribute the name-mapping functionality in a way similar to the DNS. This approach includes

- a distributed hierarchical tree of the name space,
- multiple name servers that control different portions (subtrees) of the name tree, and
- distributed tree traversal across name servers for name mapping.

However, we can't adopt a DNS-like solution directly because, unlike DNS, Iceberg has multiple name spaces, including telephone number, e-mail address, and pager number. In addition, Iceberg must handle new name spaces as new services are introduced. For example, adding a service like ICQ adds a new name space for the ICQ-ID.

To handle these issues, each service-specific-ID name space is a separate tree. Iceberg name servers have a hierarchical arrangement that is independent of any hierarchy within the access network itself. Name servers are persistent and highly available Iceberg components that run on a Ninja base.

Preference registry location service

To establish a call session, Iceberg needs a mechanism for using the callee's unique ID to locate the callee's preference registry. If the calling device can only specify a service-specific ID, the IAP at the calling end first uses the naming service to get the callee's unique ID. Locating the preference registry given the unique ID is similar to using

Figure 4. The preference registry uses information from the person activity tracker to decide its outputs.

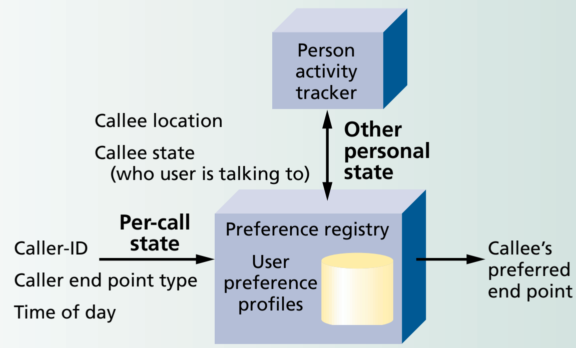


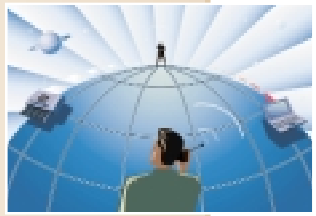
Figure 5. A simplified preference script redirects communication to a preferred end point depending on time of day.

```
IF (9AM < hour < 5PM) THEN Preferred-End-Point
   = Office-Phone; // At Office

IF (5PM < hour < 11PM) THEN Preferred-End-Point
   = Home-Phone;  // At Home

IF (11PM < hour < 9AM) THEN Preferred-End-Point
   = Voice-Mail;  // Off Hours
```

Supporting Anywhere, Anytime Services

As technology improvements introduce new communication networks and devices into the commercial world, providing rapid and easy service creation environments is becoming more important. Services are what make a device or a network worthy.

What does this mean for providers of new devices or access networks?



When deploying a new device (like the Palm VII) or a new network (say, a wireless access network), making existing services accessible is crucial. With Iceberg, providing a new device or network could be as simple as

- **deploying an IAP to interface with the Iceberg core network, and**
- **deploying data transformation agents to do the appropriate data format conversions.**

This done, users can access existing services anywhere in the Iceberg network through their new device.

What does this mean for existing telecommunication providers?

Iceberg's preference registry component provides mechanisms for service personalization. Personalized integration greatly enhances call-management services—such as call forwarding, filtering, and redirection—that telecommunication service providers make available. Iceberg integrates a telecommunication network with other networks to enable personalized communication management and enhanced features like service handoff.

the naming service. The differences are that the lookup is based on the unique-ID rather than a service-specific ID, and the lookup returns the preference registry's location rather than the unique ID. Distributing this service is similar to distributing the naming service: The leaves of the unique ID space's hierarchical tree store the location of the user's preference registry.

INTEGRATED COMMUNICATION SERVICES

Numerous commercial services are available that provide personalized integrated communication, including e-mail-to-fax, voice/e-mail/fax integration, and enhanced telephony services. This emphasizes the importance of communication integration. The Telephony Over Packet Networks (TOPS) architecture uses a directory service component for personalization. Some of the Mobile People Architecture project's goals such as any-to-any integration and personalized call handling are similar to Iceberg's. The telecommunication industry's Intelligent Network service architecture provides easy service creation and customization.

Ongoing efforts in the telecommunications community seek to provide network and service integration. Although service creation is easier than in traditional switch-based networks, it is still tied to the telephony service model: Only network operators can provide services, not end users. In addition, these architectures provide limited extensibility to new networks.

In contrast, adding a new network in Iceberg only involves building an IAP to interface the new network to the Internet core. Iceberg's unique features include service handoff, seamless service integration, and scalable and incremental service deployment model (a feature that we leverage from Ninja). Recent research in hybrid services has focused on generic service components in networks. Iceberg has a stronger notion of personal mobility, customization, and personal-activity tracking. It is also unique because it uses the path abstraction concept to achieve extensible any-to-any service integration. As our architecture evolves, we expect to build upon the SIP and H.323 Internet telephony signaling protocols to provide features such as service handoff.

Iceberg's main strength—ease of service creation and deployment—comes from using the Internet as its core. It provides a framework for cross-network and cross-device service mobility. A flexible personalization model facilitates building common IN and PCS services such as call redirection, personal numbering, and 800-number service, as well as com- mercial integration services such as voice-to-email and e-mail-to-fax.

Iceberg doesn't replace the current telecommunication infrastructure. On the contrary, its goal is to facilitate the integration of existing and emerging networks and services. Iceberg's design allows the incremental deployment of its components. For example, an Iceberg network can start out just with a few IAPs and other components deployed in a local administrative domain. Service providers can incrementally add more IAPs to accommodate more users and allow wide-area expansion.

To better refine our architecture's design, we have implemented Iceberg's components in a testbed setting. We have a GSM base station representing a GSM cellular network that is connected to the Internet through an IAP. We are building an IAP to interface to the PSTN and to a paging network. The next design iteration will be

Resources

URLs

Faxaway, an Internet faxing service for conducting business on the Web:http://www.faxaway.com/

H.323, an International Telecommunications Union recommendation that sets standards for multimedia communications across IP-based networks, including the Internet: http://www.databeam.com/h323/h323primer.html

ICQ, an Internet tool that lets users chat, transfer URLs, send messages, exchange files, or launch external applications in real time: http://www.icq.com/

Ninja, a UC Berkeley research project looking into the challenges in developing scalable Internet services: http://ninja.cs.berkeley.edu/

Oxford University's system for sending faxes via the Internet: http://info.ox.ac.uk/fax/

Planetary Motion's CoolMail, a service that provides access to e-mail messages via telephone: http://planetarymotion.com/

PocketMail, a service for sending and receiving e-mail messages through a regular phone: http://www.pocketmail.com/

Qualcomm CDMA digital phones: http://www.qualcomm.com/cdma/phones/

ThinkLink, a Web-based communications tool that links access to voice mail, e-mail, faxes, and pager messages: http://www.thinklink.com/

Wildfire, a personal assistant that uses speech recognition to manage phone, fax, and e-mail communications: http://www.wildfire.com/

Articles, reports

E. Amir, S. McCanne, and R.H. Katz, "An Active Service Framework and Its Application to Real-Time Multimedia Transcoding," *Proc. ACM SIGCOMM 98,* ACM Press, New York, Sept. 1998, pp. 178-189.

N. Anerousis et al., "The TOPS Architecture for Signaling, Directory Services, and Transport for Packet Telephony," *Proc. 8th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video*, July 1998, IEEE Press, New York, pp. 41-53.

E. Brewer et al., "A Network Architecture for Heterogeneous Mobile Computing," *IEEE Personal Comm.*, Oct. 1998, pp. 8-24.

N. Faggion and C.T. Hua, "Personal Communications Services through the Evolution of Fixed and Mobile Communications and the Intelligent Network Concept," *IEEE Network*, July/Aug. 1998, pp. 11-18.

S.D. Gribble et al., "The MultiSpace: An Evolutionary Platform for Infrastructural Services," *Proc. Usenix Ann. Tech. Conf.*, Usenix, Berkeley, Calif., June 1999, pp. 157-170.

J-P. Hubaux et al., "The Impact of the Internet on Telecommunication Architectures," *Computer Networks and ISDN Systems*, Special Issue on Internet Telephony, Feb. 1999, pp. 257-273.

H. Schulzrinne, "A Comprehensive Multimedia Control Architecture for the Internet," *Proc. Int'l Workshop Network and Operating Systems Support for Digital Audio and Video*, May 1997, IEEE Press, New York, pp. 65-76.

driven by a performance evaluation of our testbed implementation. We also are looking into many unresolved issues—such as privacy and authentication, service billing, usability, and user interfaces. ■

Bhaskaran Raman, Helen J. Wang, *and* ***Jenny Shih*** *are graduate students in the EECS Department at the University of California, Berkeley.*

Anthony D. Joseph *is an assistant professor in the EECS Department at the University of California, Berkeley.*

Randy H. Katz *is a professor in the EECS Department at the University of California, Berkeley.*

Contact the authors at {bhaskar, helenjw, jshih, adj, randy}@cs.berkeley.edu.