Ubiquitous Computing in Home Networks

Henning Schulzrinne, Xiaotao Wu, and Stylianos Sidiroglou, Columbia University
Stefan Berger, IBM Research, Columbia University

ABSTRACT

In the past decade, there have been numerous efforts in ubiquitous computing. For home networks, we believe that ubiquitous computing requires a global-scale system that is securable, administered by multiple independent nonspecialist administrators, and integrates off-the-shelf hardware and software. In this system every home owner acts as an administrator of the network in the home. We are developing such a system based on Session Initiation Protocol (SIP), with Bluetooth devices for location sensing and Service Location Protocol (SLP) for service discovery. We also introduce context-aware location information to augment device discovery and user communication. The system builds on our CINEMA infrastructure.

INTRODUCTION

Ubiquitous computing aims to "enhance computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user" [1]. In the past decade this goal has been pursued in a large number of prototypes. We believe that it is time to move from special-purpose one-of-a-kind systems to more widely deployable systems that scale to the global Internet. Such global-scale ubiquitous computing systems need to be securable, administered by multiple independent nonspecialist administrators and operators, and integrate off-the-shelf hardware and software. Thus, it is a desirable system for home networks where every home owner acts as an administrator. An example usage of such a system in a home environment is to automatically change communication devices when the talker roams from one room to another. The talker always uses the wired devices of the room he/she is in. Compared with using wireless devices for roaming in home, using wired devices can get better conversation quality since they usually have higher bandwidth and bigger displays. Another example is to allow a traveler to easily use the devices in a hotel room. We will describe example scenarios in detail later.

We are developing such a scalable ubiquitous communications and computing system in our laboratory. The system incorporates core tenets of ubiquitous, pervasive, and context-aware computing, in particular:

Multimedia: We believe that communications incorporates all types of media, from continuous media to application sharing, and we consider multimedia support a core component of a ubiquitous computing environment.

Device integration: Our system integrates mobile devices such as programmable active badges, PDAs, and laptops with resources embedded into the environment, such as large displays, video projectors, high-resolution video cameras, loudspeakers, stereos, and lights. Active multimedia sessions can be moved from one device to another and be split across devices. In a home environment, a multimedia conversation might be controlled by a PDA, video shown on a TV, and audio played through a stereo system. A user can use his/her PDA as a universal communication and control agent, not only for communication, but also as a controller of network-connected appliances.

Event-based: We consider that events offer a useful abstraction for tying together diverse systems while requiring modest knowledge about their properties. We chose the SIP event model [2] as a core component of our system, as it can scale to large numbers of users spread across administrative domains and is also suitable for small-scale networks.

Location-aware: Location is one of the key contexts that determines which types of devices are available and how communication should be conducted to minimize disruption to the user. Rather than just providing geographic location information, we add higher-level information that describes the category of a place, such as "theater" or "public transport," and its properties. Other user context, such as the number of people in a room, active conversations, or how recently a device has been used, also influence system behavior. In a home environment, when a

 0163-6804/03/$17.00

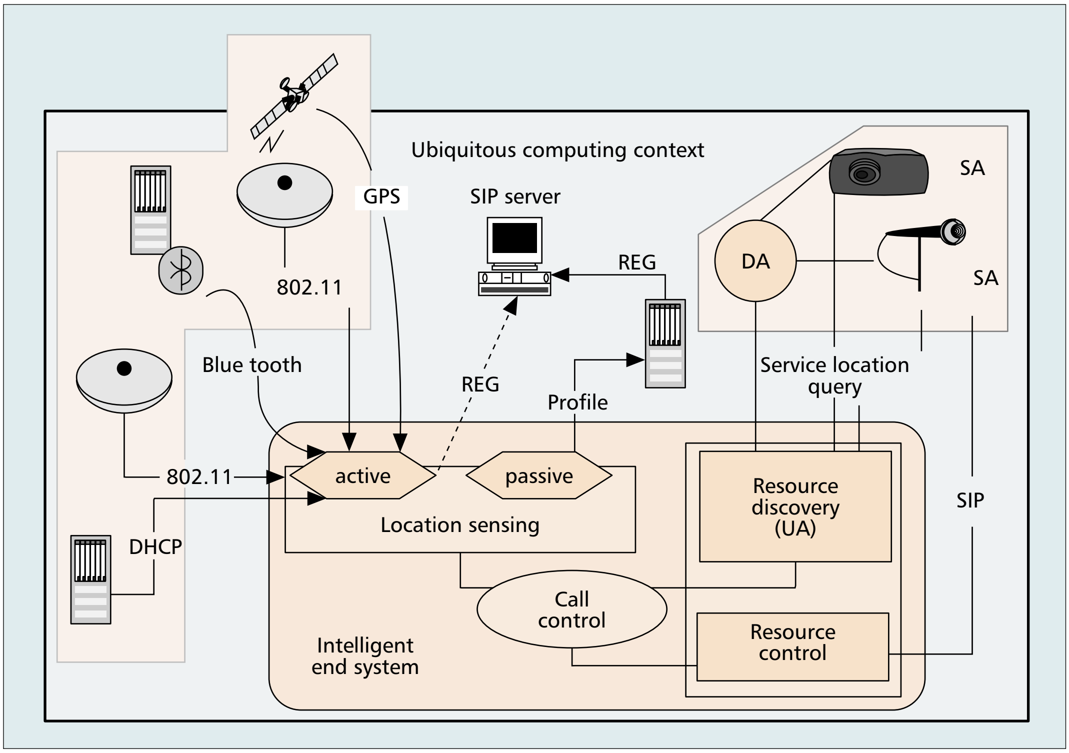

Figure 1. *Ubiquitous computing system architecture.*

While these projects have successfully built systems that effectively interact with the user and the environment, they use proprietary systems and are primarily based on non-standard protocols and are generally limited to a single organization or building.

user is roaming from one room to another, the system will be aware of the user's location and enable the devices in the current room for conversation.

Privacy-conscious: We aim to give users maximum control over their incoming communication and the amount of information about their context that is revealed to others.

Invisible to user: Wherever possible, we delegate system behavior to user-defined policies rather than require direct user interaction. Policies can be generated dynamically by presence and location information.

The system is designed to support a range of activities, from home-based settings to collaboration between distant sites. It is designed so that participation only requires standard SIP speaking tools such as Microsoft Windows Messenger or even a basic cell or landline phone. Our system derives its scalability from the underlying SIP architecture. This article describes work in progress; we have implemented some of the core ideas and outline others below. It builds on our Columbia InterNet Extensible Multimedia Architecture (CINEMA) infrastructure for multimedia collaboration [3].

CINEMA is a set of SIP-based Internet multimedia servers for creating an enterprise Internet telephony and multimedia system. The core part of CINEMA is a SIP proxy, redirect, presence, and registration server named sipd, which can execute service scripts, such as Call Processing Language (CPL) scripts, SIP servlets, and SIP CGI scripts to perform call control services, such as follow-me and call screening. CINEMA also contains a SIP conference server and a SIP voicemail/unified messaging server. All the components of CINEMA are software-based and can run on a home PC.

The remainder of the article is organized as follows. We discuss earlier research in the area. We show our system architecture and explain how we determine and handle location information. Following that, we talk about our current prototype implementation. We then present our service scenario and explain the architectural components. Finally, we conclude this article.

Related Work

A number of projects in ubiquitous computing have motivated our work. Some examples are the Intelligent Room (MIT), the Interactive Workspaces Project (Stanford University), the Aura Project (CMU), and the Easy Living project (Microsoft). While these projects have successfully built systems that effectively interact with the user and the environment, they use proprietary systems and are primarily based on non-standard protocols and are generally limited to a single organization or building. The work presented in this article is centered around open protocol standards like SIP, SLP [4], Bluetooth technology, and ongoing efforts in the Internet Engineering Task Force (IETF).

Location-based services have received much attention from wireless providers and are integrated into the Third Generation Partnership Project (3GPP) Universal Mobile Telecommunications System (UMTS) service architecture and Wireless Access Protocol (WAP) [5]. There are several efforts underway to establish a standard for positioning techniques and a standard for relaying location information. Proposals are offered by the Location Interoperability Forum (LIF) under the Mobile Location Protocol [6] and the Open GIS Consortium. The GEOPRIV working group in the IETF deals primarily with

When Bluetooth-enabled devices connect to our access point, they can pinpoint their current location by querying information from our location profile. We created a LOC service in the SDP server where context aware information can be extracted in a number of ways.

location-dependent privacy issues. Civil and categorical presence information has also been proposed by one of the authors as part of the Rich Presence Information Data Format (RPID) [7] and the DHCP option for civil location [8].

System Architecture

As shown in Fig. 1, the system involves the context around the user and the intelligent end system the user holds. The ubiquitous computing context provides location information and multimedia resources to users. The intelligent end system interacts with the context, and retrieves the location and resource information, transfers the information to the SIP server, and controls the resources in the context. The intelligent end system consists of three core components: location sensing, resource discovery and management, and call control. The call control part receives events from and sends control commands to the other two parts. We describe each component in detail below.

Location Sensing and Location-Based Services

Determining User Location

We support a wide variety of location techniques, depending on what is available in the local environment.

There are two modes for dealing with location. In the first, a mobile device determines its own location and announces it to other system components that need the information. For example, the device can use GPS or measure the field strength of wireless access points [9]. GPS does not work well indoors, and few devices have GPS capabilities. To determine location information by measuring field strength requires at least two wireless access points; however, a residential setting usually has only one access point. Thus,the method is not applicable for home networks.

For home networks we consider it a better approach to use location beacons to announce the current location information. This works well as long as the coverage region of a beacon corresponds roughly to the desired location accuracy. For example, indoors it is usually sufficient to know which room a person is in. We use three different location beacons in our prototype: Bluetooth beacons, infrared/radio frequency (IR/RF) programmable badges, and DHCP extended with location information.

For Bluetooth-enabled devices, we have implemented a simple location profile using the service discovery protocol (SDP) of Bluetooth. When Bluetooth-enabled devices connect to our access point, they can pinpoint their current location by querying information from our location profile. We created a LOC service in the SDP server where context-aware information can be extracted in a number of ways (e.g., SDP text field, OBEX exchange, and sockets).

We have built small IR/RF programmable badges that are capable of sending a unique identifier to an access point. Once the access point recognizes that a user has entered a room, it can forward this event into our system, along with the location information indicating where the user was seen. The event can then trigger some services, such as turning on the lights when a user enters a room.

We have also enhanced a common open-source DHCP server, ISC's dhcpd, with location information [8]. The server can operate in two modes, providing either server or jack location. Without knowledge of the local Ethernet wiring plan, the server simply indicates the region it serves, such as a campus or building. For more precise information we are adding a backtracking mechanism, where the server takes the medium access control (MAC) address provided by the DHCP client and traces it to a particular Ethernet switch port via the Cisco Discovery Protocol (CDP) and SNMP bridge table walks. It then consults an administrative port-to-room database that records the location of the Ethernet jack for each switch port.

These techniques have the advantage of requiring little user interaction, but they do require that the user carry a PDA, a laptop, or our programmable badge. We complement these techniques with token-based systems where users explicitly log into a room with a magnetic swipe card such as a library ID or credit card, an iButton hardware token, or a smart card. This mode may also be more acceptable for users who can readily maintain tight control of when and where they update their location information.

Location Information

Most location systems deliver geographic (longitude, latitude, altitude, heading) coordinates or, sometimes, civil location information such as street addresses. However, this often reveals too much information, but offers too little context. In many cases, others do not really need to know the precise place where somebody is located, but rather want to know what kind of communication is appropriate for the location. Thus, we have our location beacons deliver location attributes that mobile devices can then pass on as part of their presence information. We distinguish [7] a number of place types, such as *home*, *office*, and *public*, as well as a privacy classification into *public*, *private*, or *quiet* to indicate whether communication is likely to be overheard or audio communication is considered undesirable.

In a home environment, it is desirable to deliver room information by the type of room, such as *bedroom*, *living room*, or *kitchen*, or the names assigned by the home owner. For example, a home owner can use his/her kids' names for their rooms. This enables services such as being *quiet* near a nursery after a certain time.

Location-Based Services

Our system uses location information in three ways: it triggers automated behavior, becomes part of presence information revealed to selected watchers, and governs communication behavior. All three uses are derived from location-enhanced presence information.

SIP devices can upload their location information, including room (name or function information), civil (country, street, and community), categorical (e.g., movie theatre), and behavioral (e.g., quiet) descriptions, via the SIP REGISTER and PUBLISH mechanisms. The information is kept at the user's presence agent and becomes part of the user's presence information.

Other entities, so-called *watchers*, subscribe to this information. Watchers can be family members, friends, and colleagues, but also software applications that take presence information and translate it into call filtering behavior, such as rejecting or forwarding communication attempts.

The third type of watchers are software agents that control devices in rooms [10]. We have made a number of devices controllable, such as a CD player and radio, a lava lamp, and room lights. Devices either have their own network interface or are controlled through a PC in the room, typically via a serial port or X10 controller. With device control, the lights in a room can automatically be turned on or off based on whether people are present. The control behavior is freely programmable using one of the aforementioned scripting languages.

We are exploring two control modes: user-centric and device-centric. In user-centric mode, each user maintains a service script that listens for location presence updates for itself. The location updates contain the civil address as well as the location's service domain (e.g., example-hotel.com). If the user is reported to be in a new location, the service script queries the SLP server for that domain for the devices located in the room, selecting the types of devices for which the user has defined preferences. For example, it might determine that a particular location has a device of type radio with address sip:radio-712@example-hotel.com. The script, written in the LESS end-system service creation language [11], can send a SIP instant message (MESSAGE [12] or DO [13]) to that device, instructing it to tune to a particular radio station.

In device-centric mode, devices subscribe to user presence and store user preferences. The device then acts on the notification and changes the device state. Multiple devices (e.g., within a building) can share a single preference database and use SIP MESSAGE or DO commands to send control commands to the controllers or devices.

Resource Discovery and Control

Resource Discovery Using SLP

Service discovery is an essential step in mobile computing if a user with a wirelessly connected device enters a new environment and wants to use services in the surrounding area. In our scenario we decided to use SLP [4] as a service discovery protocol for a number of different reasons. First, SLP is an open standard. Second, the query language of SLP is fairly capable. It allows not only simple matching for equality or prefixes of names, but also comparisons with mathematical operators such as $\leq$ and $\geq$, particularly interesting when used with location-based services. We find this one of the most important requirements for query language features employed in service discovery. Using this query language, we can easily search for services within a given area.

Back-to-Back User Agent for Resource Control

With available resources discovered, there must be a way to utilize the resources. For example, a user should be able to use his/her PDA to instruct an echo-canceling microphone in the context to start recording and send packets to a certain address, and to enable a wall hanging plasma display to accept packets and play video streams. In a SIP environment, a secure and flexible way to utilize the resources in an environment is to have the visitor's device work as a SIP back-to-back user agent (B2BUA) and follow the third-party call control architecture [14] to control the devices. "A back-to-back user agent is a logical entity that receives a request and processes it as a user agent server (UAS). In order to determine how the request should be answered, it acts as a user agent client (UAC) and generates requests [15]." For an incoming call, the visitor's PDA generates calls to the resources with necessary information (e.g., the remote party's SDP [16] information). All calls to the resources must go through the visited domain SIP server for authentication and authorization. The resources in the context automatically accept the authenticated and authorized calls, send their own information (e.g., IP address and port number for receiving packets) to the B2BUA, and start communication with the remote party. On receiving the resources' information, the B2BUA uses the information in its response to the remote party.

Call Control

Event-Triggered Actions

Actions can be triggered by the events received from the location sensing and the service discovery parts, or the inbound or outbound calls, or the presence information. The actions can be described by CPL scripts, SIP servlets, or SIP CGI scripts. The actions can be executed in network servers, such as SIP proxy servers, or users' end systems. The call control module of the network server or end system executes the action scripts and sends control messages to related network components, such as an echo-canceling microphone in a user's environment.

Events

Events are one of the core abstractions in our ubiquitous computing architecture. We use them to propagate information about the presence of people and devices. When a user enters a room with a device containing the user's identification, such as one of our badges, an access point located in the room will sense the presence of the badge, read the serial number from the device, map it to the user's identifier, and propagate a presence detection message to a presence server located in the network. Similar behavior can be expected if a user authenticates him/herself with the environment using another passive device like a swipe card or a simple iButton where neither of the devices has the capability to actively send a message.

We use the SIP event framework [2] for event transmission. The framework defines mecha-

When a user enters a room with a device containing the user's identification, an access point located in the room will sense the presence of the badge, read the serial number from the device, map it to the user's identifier, and propagate a presence detection message to a presence server located in the network.

Already, SIP for instant messaging and presence offers mechanisms to restrict subscribing to presence information, but this is a binary decision that is too coarse-grained for location information.

nisms for transferring detailed status information, the kind of place the user is in (e.g., home, office), and the location-dependent privacy model the user is holding (e.g., whether a person is allowed to make outgoing calls). Generally it contains additional information we use to dynamically enforce policies depending on the user's current local restrictions. When the call control part receives an event, it will perform CPL, SIP CGI, or SIP servlet scripts accordingly.

CONTROL MESSAGING USING SIP, HTTP, AND SOAP

We propose that devices be controlled using existing control protocols. There are at least three choices: SIP, HTTP, and SOAP [17].

Both SIP `DO` and `MESSAGE` methods can be used for device control. The SIP `MESSAGE` method was originally designed for text-based instant messaging between human users; however, we consider sending a control message, say, "turn on the lights," to a device not fundamentally different from talking to a people, except for the message content type. The message content could be an XML document for complicated devices such as stereos, pan-tilt-zoom video cameras, and projectors, or a simple command such as on and off for basic devices like lights and blinds. One major advantage of the SIP approach is that the SIP proxy infrastructure can map a generic long-term-stable identifier such as sip:lamp@cs.columbia.edu into a current IP address and port.

HTTP offers a second approach, with the request encoded as URI query parameters, as in http://www.example.com/light?op=on. Normal HTTP user authentication can then be used to restrict access to devices.

SOAP, the third approach, is the most powerful, but also adds the most implementation complexity.

A single device can easily be accessible by all three mechanisms. The SLP entry enumerates all service interfaces. In the user-centric approach, the device interface needs to ensure that actions initiated by different users do not interfere with each other. For a device-centric approach, the device controller can more readily devise a priority algorithm that, for example, keeps the setting to the one preferred by the first person entering the room.

ACCESS CONTROL

Remote control of devices and access to services can expose the infrastructure to significant risk. For example, we do not want to allow random strangers to turn the home video camera into a surveillance camera or turn the lights off.

Three security models are plausible. In the first one, users and visitors are explicitly registered with the local SIP server, obtaining a suitable shared secret or, if a public key infrastructure exists, simply verifying their identity against a local access list with an expiration date. Unfortunately, manually enrolling and removing users is tedious, but can be simplified by automating enrollment with physical tokens such as swipe or smart cards, as described earlier. Also, someone should not be granted access to the equipment when returning home just because this person visited a room.

A second mechanism employs cross-domain authentication, authorization, and accounting (AAA). When user `alice@example.com` visits `visited.com`, the visited domain queries the AAA server, using RADIUS or DIAMETER, in the `example.com` domain and ascertains that Alice is a valid user in her home domain. This approach requires some kind of roaming agreement between domains.

A third approach makes use of location information. A user who is physically in the visited domain can probably already manipulate the equipment, so it does not add much vulnerability to grant control protocol access to those in the room. For example, the Bluetooth location server can tell the user a secret that is tied to the visitor's temporary network address or public key identity, somewhat similar to a Kerberos ticket. The ticket can then be used in call requests or control messages.

PRIVACY

Location information is highly sensitive. Thus, users will be reluctant to allow a system to acquire such information unless they can tightly control who obtains this information under what circumstances. Already, SIP for instant messaging and presence offers mechanisms to restrict subscribing to presence information, but this is a binary decision that is too coarse-grained for location information.

We are currently designing privacy extensions to the call processing language that make it easy for a user to construct privacy profiles that satisfy the IETF GEOPRIV requirements [18]. For example, a user might restrict delivery of location information by location ("in home only"), information content ("time zone only"), time of day ("night only"), or subscriber ("family member only"). Some forms of mutuality might also be factored in, but it is hard to determine whether the watcher actually reveals the information he/she promises.

PROTOTYPE IMPLEMENTATION

We base our implementation on the CINEMA system and have implemented a service that can customize the laboratory environment based on a person's preferences in the laboratory.

THE CINEMA SYSTEM

The Columbia InterNet Extensible Multimedia Architecture (CINEMA) has been developed with the intention to replace the local PBX network with an IP telephony infrastructure. It contains SIP components such as SIP proxy, redirect, presence and registration server (sipd), SIP voicemail server, and SIP conference server. In addition, it provides a service logic execution environment (SLEE), allowing users to program the services they need. Users can upload CPL, SIP CGI, and SIP servlet scripts through a Web page for the services. The capabilities of CINEMA make it a suitable platform for our ubiquitous computing architecture. Sipd can keep track

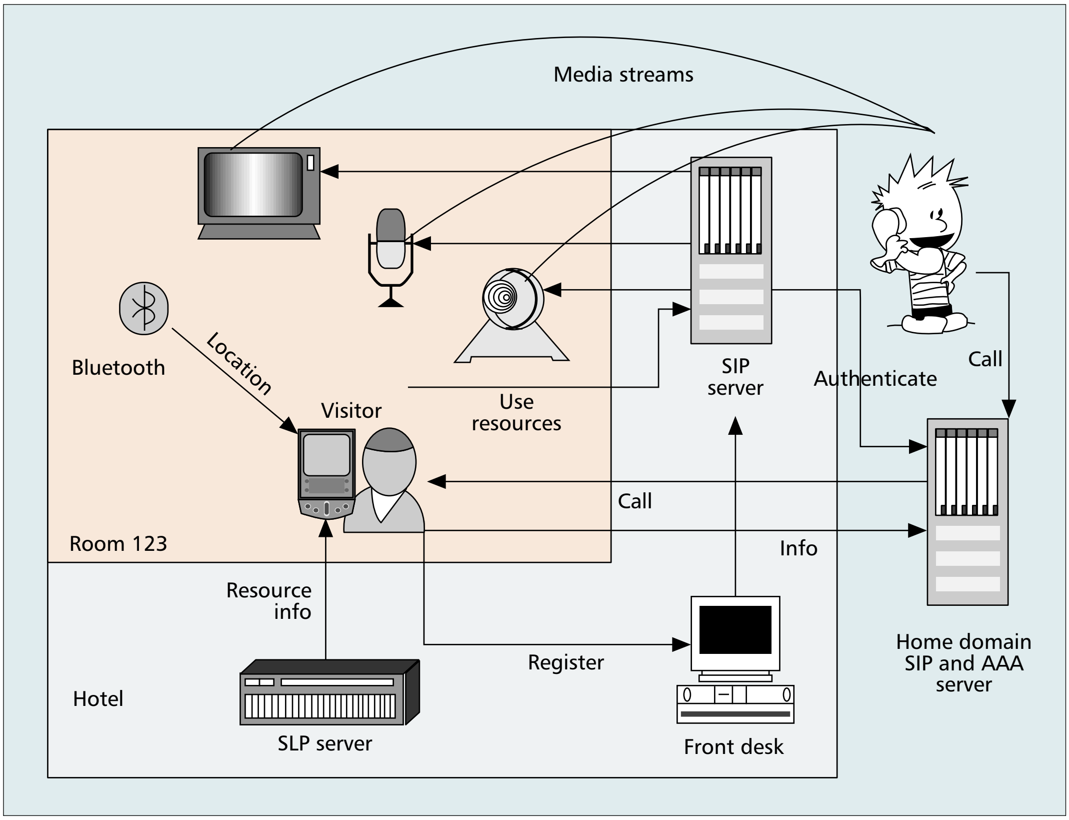

Figure 2. *SIP-based ubiquitous computing in a hotel.*

We have implemented a service prototype that allows our laboratory members to customize their working environment automatically. When a member enters the room and puts his/her iButton into a reader or swipes his/her identification card, the member's identity is sent to sipd, the registrar in the CINEMA system.

of the users' registration and location information. It can also perform authentication for access to available services. The programmability of services allows users to easily perform location-based call control services.

Using the Web interface of CINEMA, users can customize their services. We have added configuration pages for administrators to manage rooms and access to rooms for individuals, and pages for users to add their own credentials to the system.

Customized Laboratory Environment

We have implemented a service prototype that allows our laboratory members to customize their working environment automatically. When a member enters the room and puts his/her iButton into a reader or swipes his/her identification card, the member's identity is sent to sipd, the registrar in the CINEMA system. The CINEMA system will then notify a device controller, which has subscribed to the user's presence information. The device controller can then send a SIP `MESSAGE` request to a device gateway, which translates SIP messages to IR or X10 control signals. The SIP MESSAGE request contains command such as on, off, or AM 1010KHZ to turn the stereo on or off or tune the channel, respectively. Users can customize their preferred working environment from a Web page. In our implementation we use a Slink-e device to send IR signals to control a stereo with CD player, and use an X10 device controller to control a lamp in the laboratory.

In addition to a customized laboratory environment, we also developed a client application that can automatically detect its current location by either querying the DHCP server for location information or using the Bluetooth location profile. The location information is then applied to the service discovery step and used as a query parameter to find nearby resources. Our SIP user agent can then use the available resources for multimedia conversation.

Service Examples

Figure 2 shows an example of a hotel environment with a SIP-based ubiquitous computing configuration. In this configuration, the network environment of the hotel is the *visited domain*, and the environment containing the visitor's profile is the *home domain*. In the visited domain, the Bluetooth access point sends location information, such as the hotel room number, to the visitor's device. The SLP server provides information about available services, such as the SIP addresses of the audio and video devices in the visitor's room, to the visitor's device. The SIP server in the visited domain can contact the home domain AAA server to authenticate the visitor and perform authorization to use available services in the context. For example, the SIP server may allow the visitor to use the devices in his/her own room, but not the devices in the hotel conference room. The home domain SIP server may host call control service scripts. The visitor's device should send its location information to the home domain SIP server so that the server can make a location-based call routing decision.

Figure 3 shows the protocol messages

In our example, the hotel.com domain has a roaming agreement with the nyu.edu domain and thus authenticates Alice. With the AAA authorization in hand, the hotel SIP server will permit the INVITE to reach the camera and display, which will automatically join the call in progress.

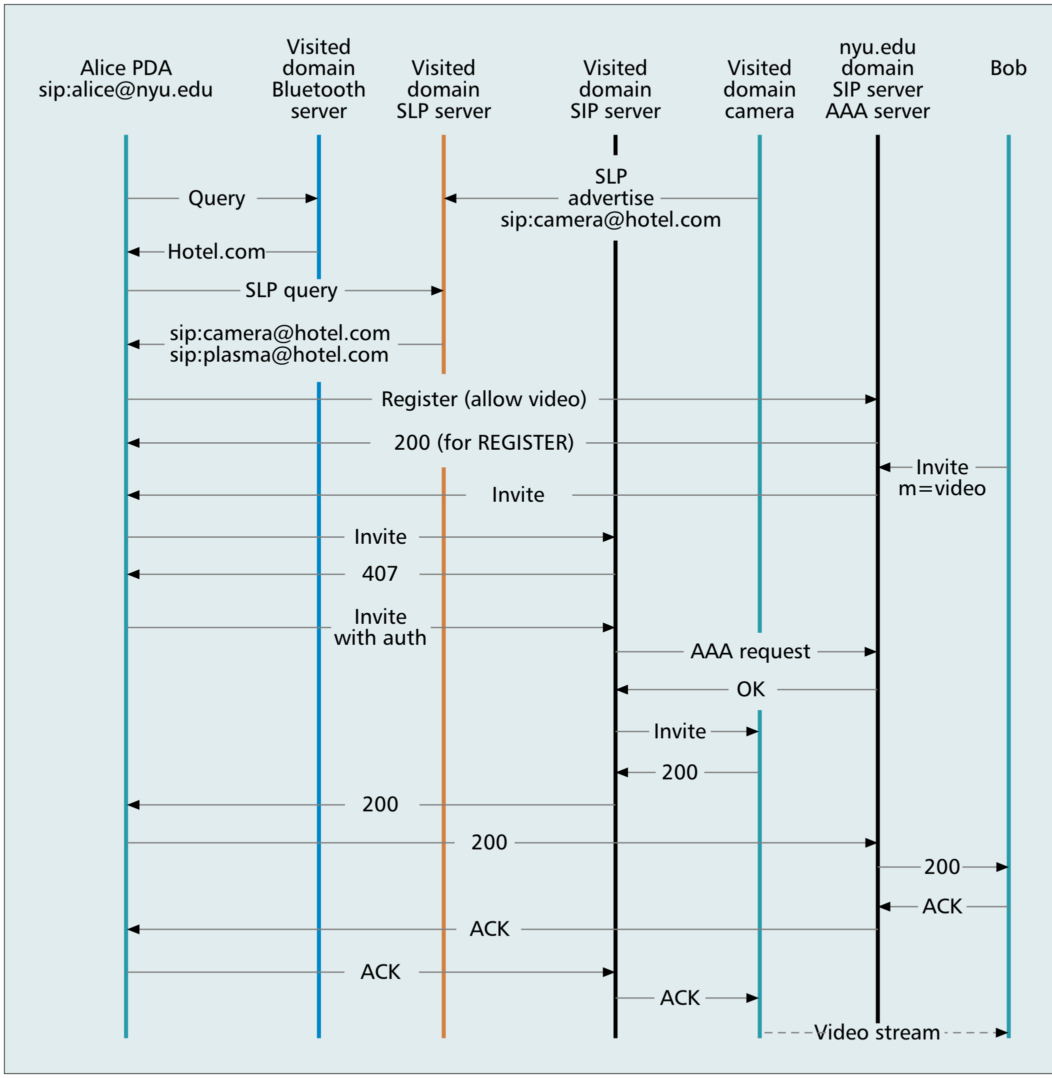

Figure 3. *Session setup using the visited domain camera.*

exchanged when Alice, `sip:alice@nyu.edu`, is in a hotel room and accepts an incoming call.

When Alice enters the hotel, she first registers with the front desk. The hotel SIP server now has her information (e.g., her SIP address and room number). The SIP server will allow her to access the resources in her room. When she enters her room, her PDA first acquires location information from the local Bluetooth server, which can also work as the access point to the Internet. The Bluetooth message indicates the room number and service domain, here hotel.com. Alice sends out an SLP query to the SLP server for this domain and finds out that the room has a network-attached camera, with SIP address camera@hotel.com, and a video display, video@hotel.com.

Alice conveys her new location to her home SIP server via `REGISTER` and indicates via the caller preferences mechanism that she is now capable of sending and receiving video.

When a remote caller, Bob, tries to invite Alice to a video session, using a SIP `INVITE` request, Alice's home domain SIP server forwards the call to her PDA. Using third-party call control, Alice sends an `INVITE` to the network-attached camera and display, with Bob's address in the session description. The `INVITE` request traverses the hotel SIP server in the `hotel.com` domain. Since Alice has registered at the front desk, she is authorized to use the resources in her room. However, the SIP server in the hotel.com domain does not have Alice's credentials for authentication; it consults Alice's home domain AAA server. In our example, the hotel.com domain has a roaming agreement with the nyu.edu domain and thus authenticates Alice. With the AAA authorization in hand, the hotel SIP server will permit the INVITE to reach the camera and display, which will automatically join the call in progress.

Conclusion and Future Work

We have presented a global-scale ubiquitous computing architecture based on open standards like SIP and SLP. We also show how location information can be acquired in our system and how the information is used to augment end system capabilities and change device behavior using location-aware user preferences. We have validated parts of our architecture through a prototype implementation using the CINEMA system. We will propose our extended location information for standardization as part of MLP and further our development of location-aware automated services by enhancing service creation languages such as CPL and LESS with location awareness.

ACKNOWLEDGMENTS

The authors would like to acknowledge Lucas Dudkowski for his work on our IR/RF location badges and Ron Shacham for enhancing the ISC dhcpd server with our location objects. The work is supported by grants from SIPquest, NSF grant ANI-00-99184, and the NSF CISE infrastructure award EIA-02-02063. Stefan Berger's work is supported by IBM Research. Ron Shacham's work is supported by DoCoMo Eurolab.

REFERENCES

[1] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Commun. ACM*, vol. 36, no. 7, July 1993, pp. 75–84.

[2] A. B. Roach, "Session Initiation Protocol (sip)-specific Event Notification," IETF RFC 3265, June 2002.

[3] W. Jiang *et al.*, "Integrating Internet Telephony Services," *IEEE Internet Comp.*, vol. 6, no. 3, May 2002, pp. 64–72.

[4] E. Guttman *et al.*, "Service Location Protocol," v. 2, IETF RFC 2608, June 1999.

[5] Wireless Application Protocol Forum Ltd., "Wireless Application Protocol — Architecture Specification," Internal doc., p. 20, 1998.

[6] Location Inter operability Forum (LIF). Mobile location protocol v. 3.0.0, http://www.cs.columbia.edu/sip/drafts/LIF_TS_101_v3.0.0.pdf

[7] H. Schulzrinne *et al.*, "RPID — Rich Presence Information Data Format," IETF Internet draft, Jan. 2004, work in progress.

[8] H. Schulzrinne, "DHCP Option for Civil Location," IETF Internet draft, Feb. 2003. work in progress.

[9] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," *GLOBECOM*, 2001, pp. 2926–31.

[10] S. Moyer *et al.*, "Service Portability of Networked Appliances," *IEEE Commun. Mag.*, vol. 40, no. 1, Jan. 2002, pp. 116–21.

[11] X. Wu and H. Schulzrinne, "Programmable End System Services Using SIP," *ICC*, May 2003.

[12] "Session Initiation Protocol (SIP) Extension for Instant Messaging," IETF RFC 3428, Dec. 2002.

[13] S. Moyer, D. Maples, and S. Tsang, "A Protocol for Wide-Area Secure Networked Appliance Communication," *IEEE Commun. Mag.*, vol. 39, no. 10, Oct. 2001, pp. 52–59.

[14] J. Rosenberg *et al.*, "Best Current Practices for Third Party Call Control in the Session Initiation Protocol," IETF Internet draft, Mar. 2003, work in progress.

[15] J. Rosenberg *et al.*, "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002.

[16] M. Handley and V. Jacobson, "SDP: Session Description Protocol," IETF RFC 2327, Apr. 1998.

[17] W3 Consortium, Simple Object Access Protocol (SOAP) 1.1, http://www.w3.org/TR/SOAP/

[18] J. Cuellar, Joel Morris, and D. Mulligan, "Geopriv Requirements," IETF Internet draft, Mar. 2003, work in progress.

BIOGRAPHIES

HENNING SCHULZRINNE (hgs@cs.columbia.edu) is an associate professor in the Departments of Electrical Engineering and Computer Science at Columbia University, New York. He received his Ph.D. from the University of Massachusetts, and worked at Bell Laboratories, Murray Hill and GMD Fokus, Berlin. Research interests include Internet multimedia and telephony services, signaling, network quality of service, scheduling, multicast, and performance evaluation. He is a co-author of the Internet standards-track protocols RTP, RTSP, and SIP.

XIAOTAO WU (xiaotaow@cs.columbia.edu) is a Ph.D. candidate in the Department of Computer Science at Columbia University, New York. He received his M.S. from the Department of Computer Science at Columbia University. His research interests include Internet multimedia and telephony signaling and services.

STELIOS SIDIROGLOU-DOUSKOS (stelios@cs.columbia.edu) is a Ph.D. candidate in the Department of Computer Science at Columbia University, New York. He received his M.S. from the Department of Electrical Engineering at Columbia University. His research interests include mobile ad hoc networks, pervasive networking, ubiquitous computing, and network security.

STEFAN BERGER (stefanb@us.ibm.com) is a Ph.D. student in the Department of Electrical Engineering at Columbia University, New York. He received his diploma in electrical engineering from the University of Erlangen-Nurnberg, Germany. He is working for IBM Research, Hawthorne, New York. His research interests include pervasive networking, ubiquitous computing, service discovery architectures, and Web service technologies.

We will propose our extended location information for standardization as part of MLP and further our development on location-aware automated services by enhancing service creation languages such as CPL and LESS with location awareness.