

# Ubiquitous Computing Using SIP

Stefan Berger, Henning Schulzrinne, Stylianos Sidiroglou, Xiaotao Wu<sup>\*</sup>  
Department of Computer Science  
Columbia University  
New York, New York 10027  
stefanb, hgs, stelios, xiaotaow@cs.columbia.edu

## ABSTRACT

In the past decade, there have been numerous efforts in ubiquitous computing, making computational resources or communication more widely available. We believe that it is time to move to a global-scale ubiquitous computing system that is securable, administered by multiple independent administrators and integrates off-the-shelf hardware and software. We are developing such a system based on the 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 and can support a range of activities, from home-based settings to collaboration between distant sites.

## Categories and Subject Descriptors

C.2.4 [Distributed Systems]: Distributed applications

## General Terms

Design

## Keywords

Ubiquitous computing, SIP, SLP, Bluetooth, scalability, location based services

## 1. 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.” [40] In the past decade, this goal has been pursued in a large number of prototypes [15], making computational resources or communication more widely available. 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

<sup>\*</sup>Authors listed in alphabetic order

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NOSSDAV'03, June 1–3, 2003, Monterey, California, USA.  
Copyright 2003 ACM 1-58113-694-3/03/0006 ...\$5.00.

administrators and operators and integrate off-the-shelf hardware and software. Thus, scaling requires more than just having a large number of systems.

We are developing such a scalable ubiquitous communications and computing system. 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 can be split across devices, so that a multimedia conferencing session might be controlled by a PDA, video shown on a wall-hanging plasma display and audio acquired through an echo-canceling microphone, each with its own network interface.

**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 [29] as a core component of our system, as it scales to large numbers of users spread across administrative domains.

**Location-aware:** Location is one of the key contexts that determines the types of devices that 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 (Section 3.2). 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.

**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 requiring 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 sipc [38] or Microsoft Windows Messenger or even a basic cell or landline phone. Our system derives its scalability from the underlying SIP architecture. This paper 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 [18].

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 [20] [21] [42], SIP servlets [28], and SIP CGI scripts [22] 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.

The remainder of the paper is organized as follows. Section 2 discusses earlier research in the area. Section 3 shows our system architecture and explain how we determine and handle location information. Following that, we talk about our current prototype implementation in Section 4. Section 5 presents our service scenario and explain the architectural components. Finally, we conclude this paper in Section 6.

## 2. RELATED WORK

The field of ubiquitous computing has provided some very motivating work over the past few years. Some examples are the Intelligent Room (MIT) [5], the Interactive Workspaces Project (Stanford University) [4], the Aura Project (CMU) [12], the Composable ad hoc location-based services [16] and the Easy Living (Microsoft) project [6], to name a few. 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 paper is centered around open protocol standards like SIP, SLP [13] and Bluetooth technology and on-going efforts in the IETF.

Some of the work that has been incorporated into our system has been the work of Roy Want and others [39] on active badge location systems. This work introduced the idea of using a variety of active badges developed for location sensing of people and devices. We use an infra-red and radio variation on our active badges as part of authenticating users with the CINEMA system (Section 3.1).

Location based services have received a lot of attention by wireless providers and are integrated into the 3GPP (UMTS) service architecture and WAP [23].

A related piece of work to the one presented in this paper was proposed by Pospichil et al [27] but deals only with push-based applications and therefore users need to explicitly subscribe to services which take advantage of location information.

The Local Location Assistant (Lol@ [1]) offers a template for location-based services in UMTS networks. Lol presents a reference implementation for a local location assistant based on a UMTS network.

There are a number of efforts currently underway for establishing a standard for positioning techniques and a standard for relaying location information. Proposals are offered by the Location Inter-operability Forum (LIF) under the Mobile Location Protocol [25] and the Open GIS Consortium. The GEOPRIV working group in the IETF deals primarily with 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 for SIP (RPIDS) [35] and as part of the DHCP option for civil location [34].

## 3. SYSTEM ARCHITECTURE

Our system consists of three core components: location sensing, service discovery, and call control. The call control part receives events from and sends control messages to the other two parts. The system is built on our CINEMA architecture. We describe each one in detail below.

### 3.1 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, or information that can be translated into location information, 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 [24]. However, GPS does not work well indoors and few devices have GPS capabilities. Similarly, deriving location from 802.11 signal strength requires that each location is covered by two, preferably more, base stations. The accuracy often depends on user orientation and may require extensive site mapping. In many cases, knowing the identity and location of the base station suffices to locate a user to within one floor of a building. While we have not yet implemented GPS-based and 802.11-based location information, it fits naturally into our framework.

The second mode uses location beacons that announce their current location. 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. Thus, traditional accuracy measurement are not all that helpful since a small error that places a user in a neighboring room is far more annoying than a geometrically larger error that simply puts the target in the wrong corner of the room. We use three different location beacons in our prototype: Bluetooth “beacons”, 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 create a LOC service in the SDP server where the context aware information can be extracted in a number of ways (e.g., SDP text field, OBEX exchange, sockets etc.).

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. Similar to [4], the badge can also act as a simple universal remote control, changing the behavior of devices in a particular location.

We have also enhanced a common open-source DHCP server, ISC’s dhcpd, with our location information [34]. The server can operate in two modes, providing either server or jack location. Without knowledge of the local Ethernet wiring plant, 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 MAC address provided by the DHCP client, 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 that they require little user interaction, but they do require that the user carry a PDA, 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 student ID or credit card, an iButton hardware token [37] or a smart card. This mode may also be more acceptable for users who can readily maintain tight control as to when and where they update their location information.

### 3.2 Location Information

Most location systems deliver geographic (longitude, latitude, altitude, heading) coordinates or sometimes civil location information such as a street addresses. However, this information often reveals too much information, yet offers too little context. In many cases, others do not really need to know the precise place that somebody is located at, but rather they want to know what kind of communication is appropriate for that 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 [35] a number of place-types, such as “home”, “office”, “driving”, “public-transport”, and “public” as well as a privacy classification into “public”, “private” and “quiet” to indicate whether communication is likely to be overheard or audio communication is considered undesirable.

We are also considering distance-based relative location information that allows others to gauge whether somebody is within walking or driving distance, without revealing their current whereabouts. For example, the system may indicate that a person is within 10 miles of her office, regardless of whether the user is driving into the office parking lot or still on the road.

### 3.3 Publishing Location Information

Our system uses location information in three ways: it triggers automated behavior, it becomes part of presence information revealed to selected watchers [31], and it governs communication behavior. All three uses are derived from location-enhanced presence information.

SIP devices can upload their location information, including civil (country, street and community), categorical (such as ‘move theatre’) and behavioral (such as ‘phone prohibited’) descriptions, via the SIP REGISTER and PUBLISH [7] mechanisms. The information is kept at the user’s presence agent and becomes part of the user’s presence information. This presence information also includes other context, such as how long ago a device was last used and what kind of activity the user is involved in. Presence information is also provided by external sources such as the user’s calendar manager.

Other entities, so-called watchers, subscribe to this information. Watchers can be friends and colleagues, but also software applications. Currently, we envision two kinds of applications: The first, run by the user, takes presence information and translates it into call filtering behavior, expressed as a dynamically-created SIP cgi or CPL script[20] [21] [42] The script itself does not use location information directly. Alternatively, an extended version of our call filtering language, may directly access the callee’s location and other context information and use it in rejecting or forwarding communication attempts.

The third type of watchers are software agents that control devices in rooms. We have made a number of devices controllable, such as a CD player and radio, a lava lamp and the 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 con-

troller.

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., cs.columbia.edu). If the user is reported to be in a new location, the service script queries the SLP server for that domain [43] for the devices located in the room, selecting those types of devices that the user has defined preferences for. For example, it might determine that a particular location has a device of type “radio”, with address sip:radio-712@columbia.edu. The script, written in the LESS end-system service creation language [41], can send a SIP instant message (MESSAGE) to that device, instructing it to tune to a particular radio station. (Alternatively, UPnP [11] can also be used, but is more suited for direct human interaction via a web page rather than automation.)

In device-centric mode, devices subscribe to user presence and store user preferences. They may add a notification filter [19] to limit notifications to those events that affect their room. The device then acts on the notification and changes the device state. Multiple devices, e.g., within a building, can all share a single preference database and use SIP MESSAGEs to send control commands to the controllers or devices. This architecture is a refinement of the SIP control architecture described in [33]. A presence policy language can determine which watchers are allowed to access the presence information.

### 3.4 Service Discovery using SLP

Service discovery is an essential step in mobile computing if a user owning 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 [13] as a service discovery protocol for a number of different reasons. First, SLP is an open standard. Second, the query language of the Service Location Protocol is fairly capable. It does not only allow simple matching for equality or prefixes of names, but also allows comparisons with mathematical operators such as “ $\leq$ ”, “ $\geq$ ”, which is particularly interesting when used with location based services. We find this to be one of the most important requirements for query language features employed in service discovery. By using this query language we can easily search for services within a given area, if for example, services are stored with geodesic coordinates representing their exact location. Wild card searches for substrings using the “\*”-operator, allow us to query for services located in a certain area, described for example by the street name they are located on. Another handy set of features that the query language supports are logical disjunctions and conjunctions, allowing an effective filtering query to be executed.

### 3.5 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 in user’s end systems. The call control module of the network servers or end systems executes the action scripts and sends control messages to related network components, such as an echo-canceling microphone in the user’s environment.

#### 3.5.1 Events

Events is one of the core abstraction in our ubiquitous computing architecture. We use it to propagate information about the pres-

ence 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 himself with the environment using another passive device like a swipe card or a simple iButton where neither one of the devices has the capability of actively sending a message.

We use the SIP event framework [31] for event transmission. The framework defines mechanisms for transferring detailed status information the kind of place the user is in (e.g., home, office, etc..) 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 that we use to dynamically enforce policies depending on the current local restrictions of the user. When the call control part receives the events, it will perform CPL, SIP CGI or SIP servlet scripts accordingly.

### 3.5.2 Control Messaging using SIP, HTTP and SOAP

We propose that devices be controlled using existing control protocols. There are at least three choices: SIP MESSAGE, HTTP and SOAP [8].

While the SIP MESSAGE method was originally designed for text-based instant messaging between human users [23], sending a control request to, say, turn on the lights is not fundamentally different except for the message content type. The message content could be an XML document for complicated devices such as stereos, PTZ video cameras and projectors, or a simple command such as 'on' and 'off' for basic devices such as 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 that preferred by the first person entering the room.

## 3.6 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 video conference room camera into a surveillance camera or to 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, just because somebody visited a room, that person should not be granted access to the equipment when returning home.

A second mechanism employs cross-domain AAA. When the 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 that is physically in the visited domain can probably already manipulate the equipment, so it does not add much vulnerability to grant 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.

## 3.7 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 [9]. For example, a user might restrict delivery of location information by location ("on campus only"), information content ("time zone only") or time-of-day ("business hours only"). Some forms of mutuality might also be factored in, but it is hard to determine whether the watcher actually reveals the information they promise.

## 4. PROTOTYPE IMPLEMENTATION

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

### 4.1 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 of the users' registration and location information. It can also perform authentication for access to the 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.

### 4.2 Customized Lab Environment

We have implemented a service prototype that allows our lab members to customize their working environment automatically. When a lab member enters the room and puts his iButton into a reader or swipes his identification card, the lab 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 infra-red or X10 control signals. The SIP MESSAGE request contains command such as 'on', 'off' or 'AM 1010KHZ', to turn on or off the stereo or tune the channel to AM 1010KHZ, respectively. In our implementation, we use a Slink-e device [17] to send infra-red signals to control a stereo with CD player. This allows us to change the device according to the user's preferences. Similarly we make use of an X10 device controller to control a lamp in the lab.

We are also using the CINEMA system to set preferences for devices. We have designed web pages for each one of the devices and a user who has been granted access to a certain room, may configure public devices to behave according to his preferences, once he enters that room.

## 5. SERVICE EXAMPLES

Figure 1 shows an example of a SIP-based ubiquitous computing configuration. In this configuration, we name the network environment that does not hold the visitor's profile the *visited domain*, and the environment containing the visitor's profile the *home domain*. In the visited domain, the SLP server and Bluetooth access point provide information about available services to the visitor's device. The SIP server performs authentication and authorization. The SIP-enabled camera enhances the visitor's communication capabilities. In the home domain, the SIP server can make location-based call routing decision and the AAA server provides authentication information. We explain each component in detail below.

### 5.1 Bluetooth Access Point

We have implemented a simple location profile on the Bluetooth access point side that allows a mobile device to retrieve its current location description from the Bluetooth access point by using SDP, the Service Discovery Protocol component of Bluetooth. We have extended the location element definitions in the Mobile Location Protocol XML format [25] to include context aware location information and transmit this information as one attribute of our location service.

### 5.2 Service Location Protocol

We are using SLP to discover SIP-enabled local services like cameras, lights, and stereos. In SLP, services are described with a list of text attributes and values. Since we want to provide location-based services, we added some location information to the regular service description. One of the current descriptions is the room number the service is located in, other attributes describe the service location with approximate geodesic coordinates.

The service discovery query is formulated with location information from DHCP or a Bluetooth access point. We use a set of attribute names such as 'LOC', 'LON' and 'LAT' which describe the services' location information in different formats.

If true ubiquitous computing is to become reality, all implementers of services will have to agree on how to describe their services in the discovery databases, since otherwise a service discovery client will not be able to interpret the information it received from the discovery database. The following shows an example of how we currently describe the camera service in our room. Other services are described in a similar way.

```
access=sip:camera-712@cs.columbia.edu
NENA.LON=-739610 // scaled by factor 1000
NENA.LAT=+408100 // scaled by factor 1000
NENA.STN=120th
NENA.STS=Street
NENA.HNO=530 W
```

```
NENA.LMK=Schapiro building
NENA.NAM=IRT-LAB
NENA.ZIP=10027
```

We have taken the NENA standard labels [2] for describing location information. The DHCP options for civil location are also based on this standard and thus make it easy for a machine to query for services once it has received location information through these options.

### 5.3 Simple Authentication Devices

If a user does not carry a capable device like an iPAQ, he can use some of his other credentials to authenticate himself with the environment. Examples of devices that can perform this are credit or student identification cards, iButtons, or one of our infra-red or radio frequency wearable devices, also known as "badge". All of them contains a unique identifier that is mapped to a user identity.

After a user has been granted access to the CINEMA system, a user can then add his own credential using our recently added ubiquitous computing subsystem, which allows the association of the credential with the user.

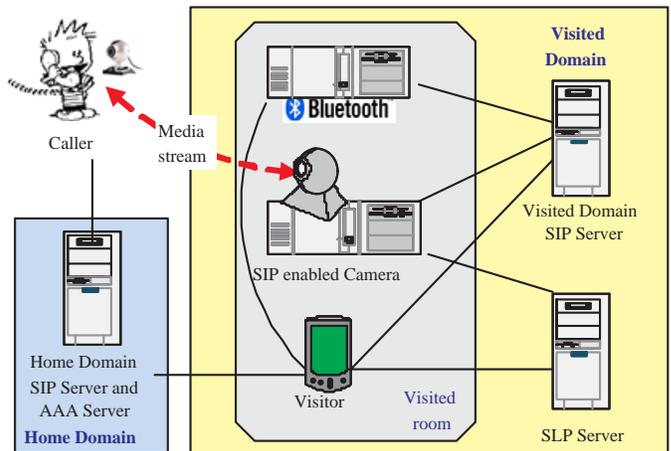


Figure 1: SIP-based ubiquitous computing environment

### 5.4 SIP Server

When the user's end system gets the new IP address from the visited domain, it will send a SIP REGISTER request to its home domain SIP server. Usually, the REGISTER request only contains the end system's new contact address. In our architecture, the REGISTER request also contains location information acquired from Bluetooth devices or the DHCP server. In addition, by using SLP, the visitor's end system can get the information about what devices are available in the new environment and the capabilities of these devices, such as the codec the devices can support. Thus, the communication capabilities of the visitor are more than what his own end system can provide. For example, a visitor can use the video camera in the new environment for a video call.

```
.....
<location-switch type="categorical"
    field="destination">
  <location is="theater">
    <reject status="busy" />
  </location>
</location-switch>
.....
```

As the home domain SIP server receives the REGISTER request from the visitor's end system, it can make call control decisions based on the location and device capabilities [36]. For example, by using the SIP PUBLISH method, the visitor can put a service script as above in his home domain SIP server. When the visitor is in a theater, his home domain SIP server will automatically reject incoming calls.

## 5.5 Visitor's Device

The visitor's device can search for Bluetooth location beacons, search for services as an SLP client and act as a SIP back-to-back user agent.

### 5.5.1 Bluetooth

When a device connects to one of our access points, it can determine its location through our location profile.

### 5.5.2 Connectivity

The visitor may have connectivity to the Internet via a number of mobile access methods (802.11, UMTS, GPRS, and CDPD). If the visitor cannot access the Internet directly, the Bluetooth access point will act as a gateway for the visitor's device through the use of the PAN profile [3] and provide Internet connectivity.

### 5.5.3 SLP

An SLP client will be utilized in order to discover available resources for a particular location. The same location information can be used by the resources when they REGISTER with the SLP server so that the wireless device can search by particular attribute such as room number. Currently, there is ongoing work in the SLP(svrloc) working group about the mechanics of calculating the distance between resource and location of the query so that the resource that is in physical proximity can be selected but as a first stage an exact match (e.g., room number) can be used to select devices.

### 5.5.4 SIP Back-to-back User Agent

There are several ways for the visitor's device to use the Media I/O devices in the new environment, such as Media Gateway Control Protocol (Megaco) [10] or the Message Bus (MBUS) [26]. With Megaco, media gateway and media gateway controller are tightly bound to each other. Usually, a visitor's device cannot work as a media gateway controller in a new environment. The MBUS relies on multicast and so may be difficult to get to work in the case of wireless controller and wired device. In addition, there is no existing authentication method defined for MBUS when there is no shared secret between the visitor's device and the devices in the new environment. In a SIP environment, a secure and flexible way 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 [32] 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 [30]." For an incoming call, the visitor's device generates calls to the necessary devices with the remote party's SDP [14] information. All the calls to the devices must go through the visited domain SIP server for authentication and authorization as described in Section 5.4. For incoming calls, the devices will automatically accept the call and send media streams to the remote party based on the SDP information it receives. The devices used for receiving and playing media streams, such as a video projector, will send the SDP information in the acceptance response to the visitor's device. The

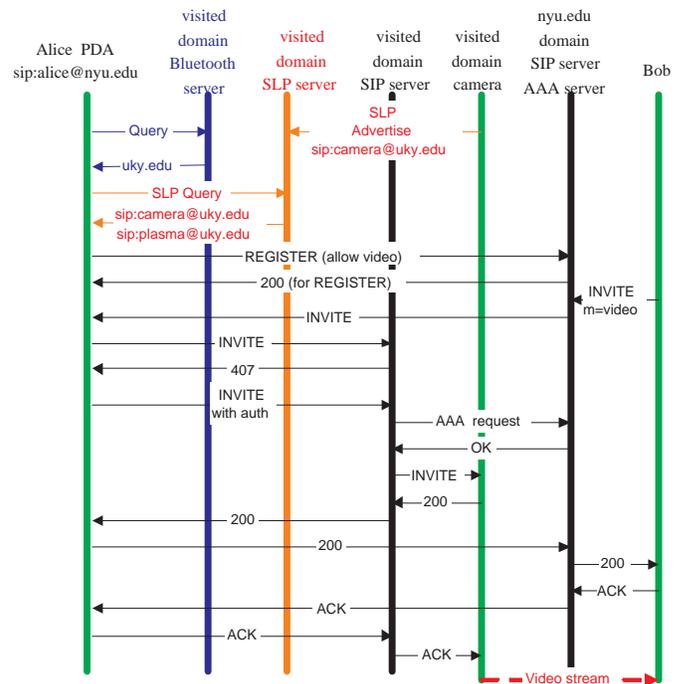


Figure 2: Session setup with using the visited domain camera

visitor's device then uses the device's SDP information in its acceptance response to the calling party so the calling party can send multimedia streams to the receiving devices in the environment.

## 5.6 Example

Figure 2 shows the protocol messages exchanged when Alice, sip:alice@nyu.edu, visits a conference room at the University of Kentucky.

When Alice enters the room in the visited domain, her PDA first acquires location information from the local Bluetooth server. (It may also use the Bluetooth access point as a way to access the Internet.) The Bluetooth message indicates the room number and the service domain, here uky.edu. Alice sends out an SLP query to the SLP server for that domain and finds out that the room has a network-attached camera, with the SIP address camera@uky.edu and a video display, plasma@uky.edu.

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 proxy forwards the call to her PDA. Using third-party call control, Alice sends an INVITE to the camera and display, with Bob's address in the session description. The INVITE request traverses the local outbound proxy server in the uky.edu domain. Since that server does not have Alice's credentials, it consults Alice's home AAA server, using her SIP address to determine the appropriate domain. In our example, the uky.edu domain has a roaming agreement with the nyu.edu domain and thus permits Alice to use the local resources. (Earlier, we had described an alternate mechanism that uses physical proximity to grant permissions.) With the AAA authorization in hand, the proxy server will permit the INVITE to reach the camera and display, which will automatically join the call in progress.

## 6. CONCLUSION AND FUTURE WORK

We have presented a global scale ubiquitous computing architecture based on open standards like SIP and SLP. We also showed 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 on location-aware automated services by enhancing service-creation languages such as CPL and LESS with location-awareness.

## 7. ACKNOWLEDGEMENTS

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.

## 8. REFERENCES

- [1] H. Anegg, H. Kunczler, E. Michlmayr, G. Pospischil, and M. Umlauf. Lol@: Designing a location based UMTS application. *VE-Verbandszeitschrift*, Springer, Heidelberg, Germany, Feb. 2002.
- [2] National Emergency Number Association. NENA recommended formats & protocols for ALI data exchange, ALI response & GIS mapping, January 2002. [www.nena.org/9-1-1TechStandards/Standards\\_PDF/NENA](http://www.nena.org/9-1-1TechStandards/Standards_PDF/NENA)
- [3] Bluetooth. Personal area networking profile, June 2001. [http://www.bluetooth.com/pdf/PAN\\_Profile\\_0\\_95a.pdf](http://www.bluetooth.com/pdf/PAN_Profile_0_95a.pdf).
- [4] Jan Borchers, Meredith Ringel, Joshua Tyler, and Armando Fox. Stanford interactive workspaces: A framework for physical and graphical user interface prototyping. *IEEE Wireless Communications*, pages 64–69, December 2002.
- [5] R. Brooks. The intelligent room project. In *Proceedings of the Second International Cognitive Technology Conference (CT'97)*, Aizu, Japan, Aug. 1997.
- [6] Barry Brumitt, John Krumm, Brian Meyers, and Steven Shafer. Ubiquitous computing and the role of geometry. *IEEE Personal Communications Magazine*, 7(5), October 2000.
- [7] B. Campbell et al. SIMPLE presence publication mechanism. Internet draft, Internet Engineering Task Force, March 2003. Work in progress.
- [8] World Wide Web Consortium. Simple object access protocol (soap) 1.1. <http://www.w3.org/TR/SOAP/>.
- [9] J. Cuellar, Joel Morris, and D. Mulligan. Geopriv requirements. Internet draft, Internet Engineering Task Force, March 2003. Work in progress.
- [10] F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, and J. Segers. Megaco protocol version 1.0. RFC 3015, Internet Engineering Task Force, November 2000.
- [11] UPnP Forum. Universal plug and play specification 1.0. <http://www.upnp.org>.
- [12] David Garlan, Dan Siewiorek, Asim Smailagic, and Peter Steenkiste. *IEEE Pervasive Computing, special issue on "Integrated Pervasive Computing Environments"*, 1(2):22–31, 2002.
- [13] E. Guttman, C. E. Perkins, J. Veizades, and M. Day. Service location protocol, version 2. RFC 2608, Internet Engineering Task Force, June 1999.
- [14] M. Handley and V. Jacobson. SDP: session description protocol. RFC 2327, Internet Engineering Task Force, April 1998.
- [15] Jeffrey Hightower and Gaetano Borriello. A survey and taxonomy of location sensing systems for ubiquitous computing. UW CSE 01-08-03, University of Washington, Department of Computer Science and Engineering, Seattle, WA, August 2001.
- [16] Todd D. Hodes, R. H. Katz, Edouard Servan-Schreiber, and Lawrence A. Rowe. Composable ad hoc mobile services for universal interaction. In *3rd ACM/IEEE International Conference on Mobile Computing*, September 1997.
- [17] Nirvis Inc. Slink-e. <http://www.nirvis.com/slink-e.htm>.
- [18] Wenyu Jiang, Jonathan Lennox, Sankaran Narayanan, Henning Schulzrinne, Kundan Singh, and Xiaotao Wu. Integrating Internet telephony services. *IEEE Internet Computing*, 6(3):64–72, May 2002.
- [19] H. Khartabil et al. Event notification filtering for presence. Internet draft, Internet Engineering Task Force, January 2003. Work in progress.
- [20] J. Lennox and Henning Schulzrinne. Call processing language framework and requirements. RFC 2824, Internet Engineering Task Force, May 2000.
- [21] J. Lennox and Henning Schulzrinne. CPL: a language for user control of Internet telephony services. Internet draft, Internet Engineering Task Force, November 2001. Work in progress.
- [22] J. Lennox, Henning Schulzrinne, and J. Rosenberg. Common gateway interface for SIP. RFC 3050, Internet Engineering Task Force, January 2001.
- [23] Wireless Application Protocol Forum Ltd. Wireless Application Protocol - architecture specification. *Internal document*, page 20, 1998.
- [24] Dragos Niculescu and Badri Nath. Ad hoc positioning system (APS). In *GLOBECOM (1)*, pages 2926–2931, 2001.
- [25] Location Inter operability Forum (LIF). Mobile location protocol v3.0.0. [www.cs.columbia.edu/sip/drafts/LIF\\_TS\\_101\\_v3.0.0.pdf](http://www.cs.columbia.edu/sip/drafts/LIF_TS_101_v3.0.0.pdf).
- [26] J. Ott, C. E. Perkins, and D. Kutscher. A message bus for local coordination. RFC 3259, Internet Engineering Task Force, April 2002.
- [27] G. Pospischil, J. Stadler, and I. Miladinovic. A location-based push architecture using SIP. 4th International Symposium on Wireless Personal Multimedia Communications (WPMC 2001), Aalborg, Denmark, September, 2001.
- [28] Java Community Process. SIP servlet API. Java Specification Requests JSR 116, Java Community Process, May 2002.
- [29] A. B. Roach. Session initiation protocol (sip)-specific event notification. RFC 3265, Internet Engineering Task Force, June 2002.
- [30] J. Rosenberg, Henning Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force, June 2002.
- [31] Jonathan Rosenberg. A presence event package for the session initiation protocol (SIP). Internet draft, Internet Engineering Task Force, January 2003. Work in progress.

- [32] Jonathan Rosenberg, James L. Peterson, Henning Schulzrinne, and Gonzalo Camarillo. Best current practices for third party call control in the session initiation protocol. Internet draft, Internet Engineering Task Force, March 2003. Work in progress.
- [33] Arjun Roychowdhury and Stan Moyer. Instant messaging and presence for network appliances using SIP. In *Internet Telephony Workshop*, New York, April 2001.
- [34] Henning Schulzrinne. DHCP option for civil location. Internet draft, Internet Engineering Task Force, February 2003. Work in progress.
- [35] Henning Schulzrinne et al. RPIDS – rich presence information data format for presence based on the session initiation protocol (SIP). Internet draft, Internet Engineering Task Force, February 2003. Work in progress.
- [36] Henning Schulzrinne, Jonathan Rosenberg, and P. Kyzivat. Caller preferences and callee capabilities for the session initiation protocol (SIP). Internet draft, Internet Engineering Task Force, March 2003. Work in progress.
- [37] Dallas Semiconductor. iButton. <http://www.ibutton.com>.
- [38] Columbia University. Columbia sip user agent (sipc). <http://www.cs.columbia.edu/IRT/sipc>.
- [39] Roy Want, Andrew Hopper, Veronica Falcao, and J. A. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992. also Olivetti Research Limited Technical Report ORL 92-1.
- [40] Mark Weiser. Some computer science issues in ubiquitous computing. *Communications ACM*, 36(7):75–84, July 1993.
- [41] Xiaotao Wu and Henning Schulzrinne. Programmable end system services using SIP. In *Conference Record of the International Conference on Communications (ICC)*, May 2003.
- [42] Xiaotao Wu, Henning Schulzrinne, and Jonathan Lennox. An extensible markup language schema for call processing language (CPL). Internet draft, Internet Engineering Task Force, March 2003. Work in progress.
- [43] Weibin Zhao et al. Finding remote directory agents and service agents in the service location protocol via DNS SRV. Internet draft, Internet Engineering Task Force, January 2003. Work in progress.

SLP - Service Location Protocol  
 SOAP - Simple Object Access Protocol  
 UMTS - Universal Mobile Telecommunications System  
 WAP - Wireless Application Protocol  
 XML - Extensible Markup Language

## 9. GLOSSARY

3GPP - Third Generation Partnership Project 2  
 AAA - Authentication, Authorization and Accounting  
 CDPD - Cellular Digital Packet Data  
 CINEMA - Columbia InterNet Extensible Multimedia Architecture  
 CPL - Call Processing Language  
 DHCP- Dynamic Host Configuration Protocol  
 GPRS - General Packet Radio Service  
 GPS - Global Positioning System  
 LESS - Language for End System Services  
 MBUS - Message BUS  
 MLP - Mobile Location Protocol  
 PAN - Personal Area Network (Bluetooth)  
 PDA - Personal Digital Assistant  
 RFC - Request For Comments  
 RPC - Remote Procedure Call  
 SDP - Service Discovery Protocol of Bluetooth  
 SDP - Service Description Protocol  
 SIP - Session Initiation Protocol