The Virtual Device: Expanding Wireless Communication Services through Service Discovery and Session Mobility

Ron Shacham, Henning Schulzrinne, Srisakul Thakolsri, Wolfgang Kellerer

***Abstract*—We present a location-based, ubiquitous service architecture, based on the Session Initiation Protocol (SIP) and a service discovery protocol that enables users to enhance the multimedia communications services available on their mobile devices by discovering other local devices, and including them in their active sessions, creating a "virtual device." We have implemented our concept based on Columbia University's multimedia environment and we show its feasibility by a performance analysis.**

***Index Terms*—Internet multimedia, mobile communications, ubiquitous computing, location-based services**

I. Introduction

WHILE mobile devices are improving, with more enhanced capabilities for IP-based multimedia communications, they remain limited in terms of bandwidth, display size and computational power. The appearance of stationary IP multimedia endpoints, including hardware IP phones, videoconferencing units, embedded devices and software phones in more and more offices, meeting rooms and homes sets the stage for these two types of devices to be logically merged. The seamless transition between these devices will allow them to be used concurrently or interchangeably in mid-session, combining the advantages of both into a single "virtual device." Since the Session Initiation Protocol (SIP) [1] has been chosen by the Third Generation Partnership Project (3GPP) as its standard for session establishment in the Internet Multimedia Subsystem (IMS) [2] and it is being deployed in hardware and software IP multimedia clients, we are building an architecture based completely on SIP and its extensions, in addition to a service discovery protocol, to provide this seamless, ubiquitous service. We describe in this paper the architecture and report on our current implementation. In Section II, we discuss related work in the area of session mobility. In Section III, we specify the system components and requirements. In Sections IV and V, we discuss in detail the system components and associated protocol flows. In Section VI, we discuss how to reconcile differences in device capabilities. In Section VII, we analyze the performance of our system. In Section VIII, we discuss how security and privacy concerns in the system may be handled. We conclude in Section IX by discussing our current implementation and future plans.

This work was supported by DoCoMo Eurolabs, Munich, Germany

Ron Shacham is with the Department of Computer Science, Columbia University, New York, NY 10027 USA (phone: 212-939-7040;e-mail: rs2194@cs.columbia.edu).

Henning Schulzrinne is with the Department of Computer Science, Columbia University, New York, NY 10027 USA (e-mail: hgs@cs.columbia.edu).

Srisakul Thakolsri is with the Future Networking Laboratory at DoCoMo Eurolabs, Munich, Germany (e-mail: thakolsri@docomolab-euro.com).

Wolfgang Kellerer is with the Future Networking Laboratory at DoCoMo Eurolabs, Munich, Germany (e-mail: kellerer@docomolab-euro.com).

II. Related Work

Several existing approaches address the seamless transfer of IP-based multimedia sessions between devices to enable a user to switch terminals in the middle of a session. For example, the session-layer API method described in [3] provides an end-to-end session mobility by introducing session layer middleware which has to be installed on every terminal. A drawback of this approach is that the remote participant must have special capabilities for handling signaling during session migration from one terminal to another terminal. The service mobility proxy approach in [4] uses a special-purpose proxy to handle all session migration and media adaptation to different terminals. The disadvantage of this approach is that the user data flow between two calling parties must always traverse the proxy, regardless of whether session migration is desired, introducing triangular routing.

A standard SIP-based approach for terminal mobility was first given in [5], where two different methods are proposed, third-party call control (3PCC) [6] and the REFER method [7]. In this paper, we expand on this concept, defining a complete system for session mobility. Using a standard SIP solution to provide session mobility is desirable because it can fit well into emerging Third Generation (3G) wireless systems and other SIP-based systems. It does not require any special servers, and only requires the devices in the mobile user's environment to support additional capabilities.

Focusing on real-time conversational services, our work complements other session transfer concepts for streaming [8] or data sessions [9].

III. System Components and Requirements

In order for the Mobile Node to know about available devices and to include them in an active session, the architecture consists of two main components:

- *Service Location:* At all times, a user is aware of the devices which are available in his local area, along with their capabilities.
- *Session Mobility:* While in a session with a remote participant, the user may transfer any subset of the active media services to one or more devices.

The system has the following requirements:

- *Interoperability:* No special capabilities should be required of the remote participant in the call, as long as he is using a SIP-compliant device or there is a PSTN gateway between them. The device should be capable of handling a transfer by the mobile user utilizing only features specified in Internet Engineering Task Force (IETF) Requests For Comments (RFCs) and mature Internet Drafts.
- *Backward Compatibility:* Both mobility-enhanced and basic devices should be integrated into the system. Mobility-enhanced devices are those that are controlled by software enhanced to support special call handling and service discovery. At the same time, commercial IP phones and embedded devices are basic since they cannot be enhanced by the end user, but should nevertheless be available as destinations for transfer.
- *Flexibility:* Differences in device capabilities should be reconciled. Transfer should be possible to devices that do not support the codec being used in the session, and even to devices that do not have a codec in common with the remote participant. A transfer should also take into account device differences in display resolution and bandwidth.
- *Seamlessness:* Session transfer should be as seamless as possible. It should involve minimal disruption of the media flow and the should not appear to the remote participant as a new call.

Figure 1 shows the architecture of our system. The Correspondent Node (CN) is a basic SIP multimedia device being used by the remote participant, and may be anywhere. The Mobile Node (MN) is a SIP-enabled mobile device, containing a SIP User Agent (UA) for standard SIP call setup, as well as specialized SIP-handling capabilities for session mobility (SIP SM) and an SLP [10] User Agent (UA) for service querying. The local devices may be mobility-enhanced or basic. Basic devices, such as the IP phone, are SIP-enabled, but have no other special capabilities. Mobility-enhanced devices, such as the video display in the figure, have SLP Service Agent capabilities for advertising their services and session mobility handling. They also contain an SLP UA, whose purpose will be explained in the discussion of multi-device systems in Section 5. The SLP Directory Agent (DA) keeps track of devices based on their location and capabilities. SLP will be described in more detail in Section 4. SIP is used for the signaling behind various session mobility scenarios described in Section 5. The Real-Time Transport Protocol (RTP) [11] is used for all media transport, and SIP-based transcoding services [12] are used, when necessary, to translate between streams, as described in Section 6.

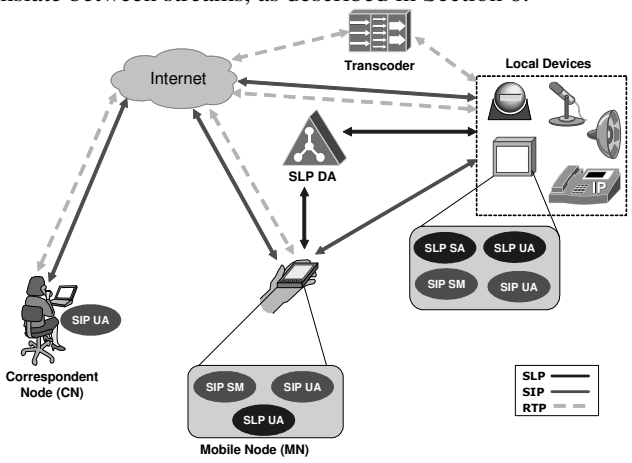

Figure 1 The location-based session mobility architecture.

IV. Service Location Architecture

A mobile node must be able to discover suitable devices in its vicinity. It is possible for devices in close proximity to be discovered by direct methods such as Bluetooth without the use of centralized servers. On the other hand, many centralized directory-based service discovery protocols exist, such as the Service Location Protocol (SLP). These protocols are general and are not based on physical locations of services, but they may be easily adapted by adding location attributes to the service description [14]. They have the advantage of allowing discovery of devices at different location granularities, such as at the room or building level, and in a location other than that of the device. They have the disadvantage of requiring mobile devices to discover their location in order to perform such queries. We have chosen to implement device discovery in our system using a general protocol, namely SLP. However, our architecture is not dependent on this protocol, and may use any mechanism that provides location-based information related to devices and their capabilities.

Many standard technologies may be used to update the mobile node of its location for use in an SLP query. Indoors, the node can receive its civic coordinates (ie., street address, room number, etc.) either directly or indirectly. With direct methods, the location is sent to the node itself by means such as a Bluetooth beacon. Indirect methods use external location sources such as swipe cards to update the user's location. The mobile node subscribes to the user's location through the mechanism in [17] and receives the location updates in the format standardized in [18]. Outdoors, a mobile device may use direct methods such as the Global Positioning System (GPS) [19] to update it on its geospatial coordinates. Online databases can translate these to civic coordinates.

Once the node has obtained its location, it uses SLP to query for available devices in the vicinity that may be included in the "virtual device." SLP identifies services by a "service type," a "service URL," which can be any URL, and a set of attributes, defined as name-value pairs. A SIP device in our architecture is of a service type that we have created called "sip-device," and its SIP URL, such as sip:audio_rm123@example.com or sip:audio@device1.example.com serves as its service URL. The first URL mentioned is an address of record that is used to connect to the device through the local proxy server, while the second URL includes the name of the host on which the service is provided, "device1.example.com," allowing a point-to-point session to be established with the device. Either of these models may be used, although the proxy model should make the device available after an address change more quickly than in the point-to-point model, where the DNS entry would take longer to be updated unless dynamic DNS were used. The service description also includes attributes specifying device characteristics (e.g., vendor, supported media codec, display resolution) and location parameters.

SLP defines Service Agents (SAs) which send descriptions of services using the Service Registration (SrvReg) message, and User Agents (UA) which query for services using the Service Request (SrvRqst) message. In our architecture, SAs are co-located with SIP UAs on mobility-enhanced devices, while a separate SA is available to provide descriptions of services offered by basic devices, which do not have integrated SLP SAs. SLP provides two models for a UA to query for services. In the distributed model, the UA sends requests through multicast and SAs reply directly with the details of the service they provide. In the centralized model, a Directory Agent (DA) is used, to which the SAs register and the UAs request services. For our examples, we use the centralized model, though either could work in our system. The SA registers its service description to the DA with a service registration (SrvReg) message that includes its service type, service URL and attribute-value set. A UA queries for services by sending the service request (SrvRqst) message, narrowing the query based on service type and attribute values. It receives a reply (SrvRply) that contains a list of URLs of services that match the query.

The Mobile Node includes an SLP UA that discovers available local devices and displays them to the user, showing, for example, a map of all devices in a building or a list of devices in a current room. Figure 2 shows the protocol flow by which a display registers its service and the MN discovers it, as well as its attributes. Once the MN receives its current location in some manner, its SLP UA issues a SrvRqst message to the DA requesting all SIP devices using the location attributes to filter out those which are not in the current room. A SrvRply message is sent to the mobile device with a list of SIP URIs for all devices on the floor. A separate Attribute Request (AttrRqst) is then sent for each URL to get the attributes of the service, which include the room where the

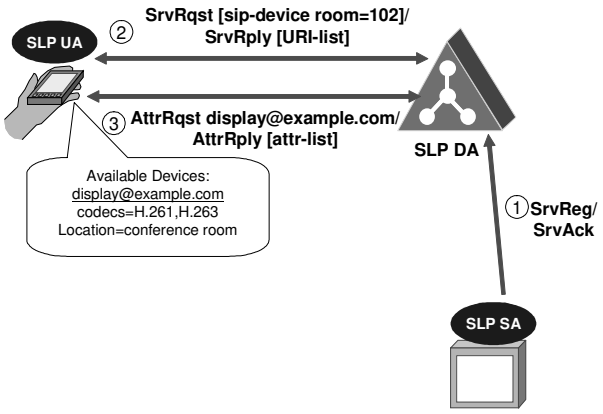

Figure 2 SLP protocol flow for registering a video display, and the subsequent discovery by the mobile node.

device is located. The MN displays for the user the available devices in the room, and their attributes.

V. Session Mobility Architecture

Session mobility in our architecture is based entirely on SIP and its extensions, documented in RFCs and mature Internet Drafts. We therefore preface our discussion of session mobility with an introduction to SIP.

A. *Introduction to SIP*

SIP is a protocol for the establishment of IP Multimedia sessions. It consists of several methods, such as INVITE, BYE and REFER, which specify actions to be taken by the participants involved. Each method consists of a request message sent by one participant to another, and a number of response messages that may be sent back. For instance, an INVITE request asks the recipient to establish a multimedia session, to which it may respond with a "200 OK" response to accept. In the INVITE method, an ACK is then sent by the original sender to complete the transaction. The messages contain a number of headers.

While SIP is not used to transport the session media itself, it uses its message bodies to describe the media so that it can be transported using RTP or another protocol. SIP uses the Session Description Protocol (SDP) [15], which was standardized for use in SIP and other protocols by [16]. The initiator of a session sends an SDP body describing its media capabilities in the initial INVITE request or in the subsequent ACK. The receiver of the request must respond with its media capabilities that are relevant to the session. The following shows the essential lines in an incomplete SDP body sent in an INVITE request:

```
v=0
c= IN IP4 host1.macrosoft.com
m=audio 4400 RTP/AVP 0
m=video 5400 RTP/AVP 34
```

Here, the session initiator, on the host called "host1.macrosoft.com" requests the other participant to initiate two media sessions, each represented by a media line. The

first session is audio, received on port 4400, using RTP and G.711 (format 0). The second session is video, received on port 5400, using RTP and H.263 (format 34). The original "c=" line gives address information that applies to all sessions, but each media line may optionally be followed by another "c=" line to indicate that the particular media session should be set up with a host other than that listed in the original line.

B. Session Mobility Options

We provide for both the transfer and retrieval of an active session. Retrieval means to remotely transfer a session currently on another device to the local device. This may mean to return a session to the device on which it had originally been before it was transferred to another device. For example, after discovering a large video monitor, a user transfers the video output stream from the other participant to that device. When he walks away, he returns the stream to his mobile device for continued communication. One may also retrieve a session to a device that had not previously carried it. For example, a participant in an audio call on his IP phone may leave his office in the middle of the call and transfer the call to the mobile device as he is running out the door.

Our architecture further allows for session media to either be transferred completely to a single device or to be split across multiple devices. For instance, a user may only wish to transfer the video of his session while maintaining the audio on his PDA. Alternatively, he may find separate video and audio devices and wish to transfer one media service to each. Furthermore, even the two directions of a full-duplex session may be split across devices. For example, a PDA's display may be too small for a good view of the other call participant, so the user may transfer video output to a projector and continue to use the PDA camera.

In SIP, the session signaling need not follow the same path as the session media. This allows our architecture to provide two different modes of session transfer, Mobile Node Control mode and Session Handoff mode. In Mobile Node Control mode, the Mobile Node uses third-party call control [6]. It establishes a SIP session with each device used in the transfer and updates its session with the CN, using the SDP parameters to establish media sessions between the CN and each device, which take the place of the current media session with the CN. The shortcoming of this approach is that it requires the MN to remain active to maintain the sessions. A user may need to transfer a session completely because the battery on his mobile device is running out. Also, if the user of a stationary device leaves the area and wishes to transfer the session to his mobile device, he will not want the session to remain on the stationary device when he is away, since it will allow others to easily tamper with his call. Therefore, we provide Session Handoff mode, which completely transfers the session signaling and media to another device. Still, we have found Mobile Node Control mode to be more interoperable with existing devices used on the CN's side. The remainder of this section describes the transfer, retrieval and splitting of sessions in each of the two session transfer modes.

C. Mobile Node Control mode

We follow Third Party Call Control Flow I specified in [6] which is recommended as long as the endpoints will immediately answer. The MN sends a SIP INVITE request to the local device used for the transfer, requesting that a new session be established. The local device's response contains

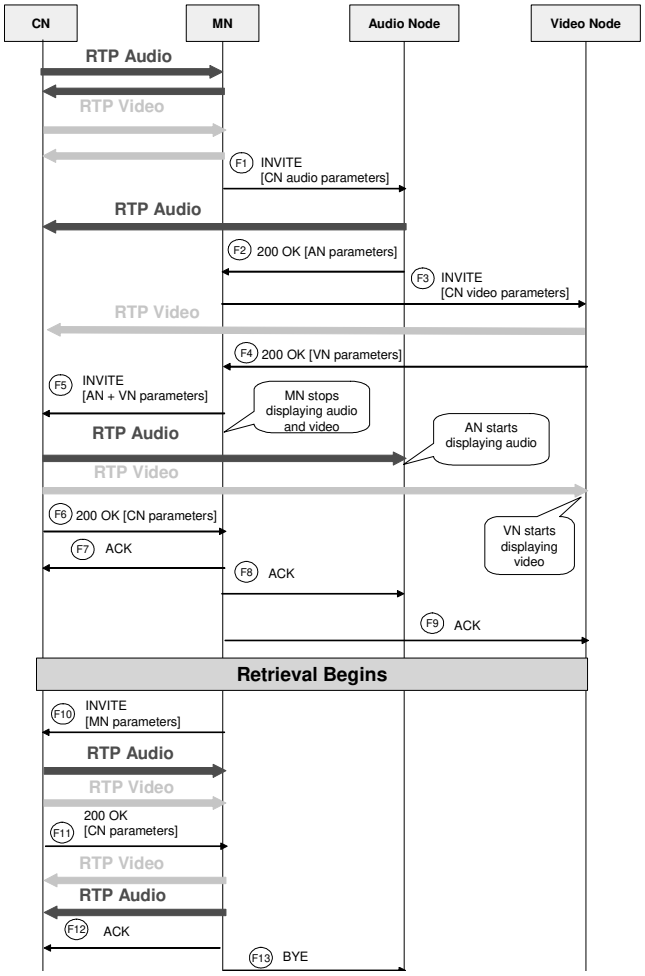

Figure 3 SIP protocol flow for transferring a session to two different devices in Mobile Node Control Mode and subsequently returning the session to the mobile node.

an SDP body that includes the address and ports it will use for any media. The MN updates the session with the CN by sending an INVITE message (which is referred to as a re-INVITE, since it is in mid-session) containing the local device's media parameters in the SDP body, as follows:

```
v=0
c= IN IP4 av_device.example.com
m=audio 4400 RTP/AVP 0
m=video 5400 RTP/AVP 34
```

The CN sends a response, and includes, in its body, the media parameters that it will use, which may or may not be the same as the ones used in the existing session. The MN sends an ACK message to the local device, which includes these

parameters in the body if they have changed. The MN has established separate SIP session with the CN and the local device, but a media flow has been established between the CN and the local device. In order to split the session across multiple devices, the MN establishes a new session with each local device through a separate INVITE request and updates the existing session with the CN with an SDP body that combines the media parameters it receives in their responses. For instance, in order to transfer an audio and video call to two devices, it creates an audio session with one device and a video session with another, and combines the SDP bodies from both to reINVITE the CN, as follows:

```
v=0
m=audio 4400 RTP/AVP 0
c= IN IP4 audio_dev.example.com
m=video 5400 RTP/AVP 34
c= IN IP4 video_dev.example.com
```

Splitting a full-duplex media service such as video across two devices is a simple extension of this approach. The signaling is identical to that of Figure 3, with the audio and video devices replaced by a video output and a video input device. The SDP, however, is slightly different. The MN invites each local device into a unidirectional media session, using the "recvonly" and "sendonly" parameters. Using their responses, the MN constructs the following SDP body to re-INVITE the CN:

```
m=video 8900 RTP/AVP 34
a=recvonly
c=IN IP4 display.example.com
m=video 8800 RTP/AVP 34
a=sendonly
c=IN IP4 camera.example.com
```

The MN may later retrieve the session by sending a re-INVITE to the CN with its own media parameters, causing the media streams to return, then sends a BYE message to each local device to terminate their session. Figure 3 shows the flow used to transfer an audio and video session to two separate devices and to subsequently retrieve the session.

D. Session Handoff (SH) mode

Session Handoff mode uses the SIP REFER method [7]. This message is a request sent by a "referer" to a "referee," which "refers" it to another URI, the "refer target," which may be a SIP URI to be contacted with an INVITE or other request, or a non-SIP URI, such as a web page. This URI is specified in the "Refer-To" header. The "Referred-By" [20] header is used to give the referer's identity which is sent to the refer target for authorization. Essential headers from this message may also be encrypted and sent in the message body as S/MIME to authenticate the REFER request.

Figure 4(a) shows the flow for transferring a session. The MN sends the following REFER request (F1) to a local device:

```
REFER sip:av@local_device.example.com SIP/2.0
To: <sip:av@local_device.example.com>
From: <sip:mn@example.com>
Refer-To:<sip:cn@host1.macrosoft.com;audio;video?
    Replaces="1@mob.example.com;
    to-tag=bbb;from-tag=aaa>
Referred-By: <sip:mn@example.com>

[S/MIME authentication body]
```

This message refers the local device to invite the refer target, the CN, into a session. The "audio" and "video" tokens

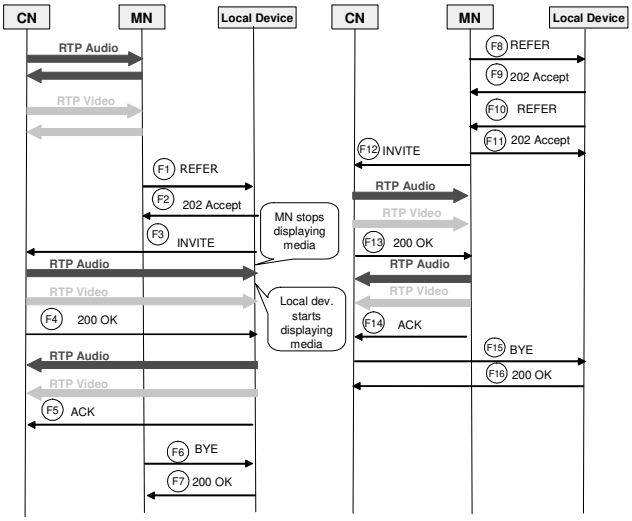

Figure 4 SIP protocol flow for the (a) transfer and (b) retrieval of an audio and video session in Session Handoff mode.

following the URI are callee capabilities. Here they are used to inform the referee that it should initiate an audio and video session with the CN. Also included is a URI parameter [25] that specifies the content of the "Replaces" header to be sent in the INVITE request. The "Replaces" header identifies an existing session that should be replaced by the new session. Here, the local device requests that the CN replace its current session with the MN with the new session. According to standards [21], the CN should only accept a request to replace a session by certain authorized groups of users. One such type of user is the current participant in the session. The MN may, therefore, refer the local device to replace its current session with the CN. However, it must provide authentication by encrypting several headers from the original REFER request in an S/MIME body that it sends in the REFER. The local device sends this body to the CN. This keeps a malicious user from indiscriminately replacing another user's session. Once the local device receives the REFER request, it sends an INVITE request to the CN, and a normal session setup ensues. The CN then tears down its session with the MN.

Figure 4(b) shows the flow for retrieval by the MN of a session currently on a local device. In order for a device to retrieve a session in Session Handoff mode, it must initiate a session with the CN that replaces the CN's existing session. The following message is sent by the MN to the CN (F12):

```
INVITE sip:cn@host1.macrosoft.com SIP/2.0
To: <sip:cn@host1.macrosoft.com>
From: <sip:mn@example.com>
Replaces: 1@local_device.example.com;to-tag=aaa;from-
    tag=bbb
Referred-By: <sip:av@local_device.example.com>

[S/MIME authentication body]
```

The MN needs to be referred by the local device and include its URI in the "Referred-By" header, in addition to including an S/MIME authentication body from the local device, in order to be permitted to replace the session. Therefore, the MN must receive a REFER request from the local device referring it to send this INVITE request. The user could use the user interface of the local device to send this REFER message. However, such an interface may not be available, and the user may also wish to execute the transfer while running out of the office with mobile device in hand. In order to prompt the REFER from the Mobile Node, a "nested REFER," [20] a REFER request for another REFER, is sent. In this case, the second REFER is sent back to the Mobile Node. That REFER must specify that the "Replaces" header be included in the target INVITE request. The REFER request from the local device to the MN (F10) is composed as follows:

```
REFER sip:mn@example.com SIP/2.0
To: <sip:mn@ example.com>
From: <sip:av@local_device.example.com>
Refer-To:<sip:cn@host1.macrosoft.com;audio;video?
    Replaces="1@local_device.example.com;to-tag=aaa;
    from-tag=bbb">
Referred-By: <sip:av@local_device.example.com>

[S/MIME authentication body]
```

A URI parameter is included in the "Refer-To" URI to specify the value of the "Replaces" header in the target INVITE request. In order to have this message sent to it, the MN must send the following REFER request (F8):

```
REFER sip:av@local_device.example.com SIP/2.0
To: <sip:av@local_device.example.com>
From: <sip:mn@example.com>
Refer-To:<sip:mn@host1.macrosoft.com;method=REFER
    ?Refer-To="<sip:cn@host1.macrosoft.com;audio;
    video?Replaces="1@local_device.example.com;to-
    tag=aaa;from-tag=bbb">">
```

The "Refer-To" header specifies the MN as the refer target and that the referral be in the form of a REFER request. The URI parameter specifies that the REFER request should contain a "Refer-To" header containing the URI of the CN.

That URI, itself, should contain the "audio" and "video" callee capabilities that will tell the MN to initiate an audio and video call, and a URI parameter specifying that the ultimate INVITE request should contain a "Replaces" header. Obviously, the local device must not grant the request to anybody and allow the session to be taken over. Only the original participant in the session or the owner of the device should be authorized to request such a REFER.

The CN accepts the INVITE request and, once the session is established, will terminate its session with the local device.

Splitting a session in SH mode requires multiple media sessions to be established between the CN and local devices, without the MN controlling the signaling. This could be done by sending multiple REFER requests to the local devices, referring each to the CN. The disadvantage of this method is that there is currently no standard way to associate multiple sessions as part of a single call in SIP. Therefore, each session between the CN and a local device will be treated as a separate call. They may occupy different parts of the user interface, their media may not be available simultaneously, and they may have to be terminated separately. This certainly does not fulfill our requirement of seamlessness.

A better solution, which we propose, is to use multi-device systems. A local device's SLP UA queries for other devices and joins with them to create a "virtual device." It then chooses a single SIP URI to address it, such as sip:a_v@dev.example.com, and registers the service in SLP. We refer to the controlling device as the Multi-Device System Manager (MDSM). In a system that includes at least one mobility-enhanced device, a device may act as the MDSM. However, in order to support systems consisting entirely of basic devices, a dedicated host is rquired. Once the MN discovers this system, it may hand off a session by sending a REFER request to the MDSM URI. When the device receives the request sent to this URI, it uses third-party call control to set up media sessions between the CN and each device in the system (including itself). Specifically, it invites each local device into a separate session, and uses their media parameters to invite the CN into a session. Figure 5 shows the transfer of a session to a multi-device system consisting of four devices, with the video camera acting as the MDSM.

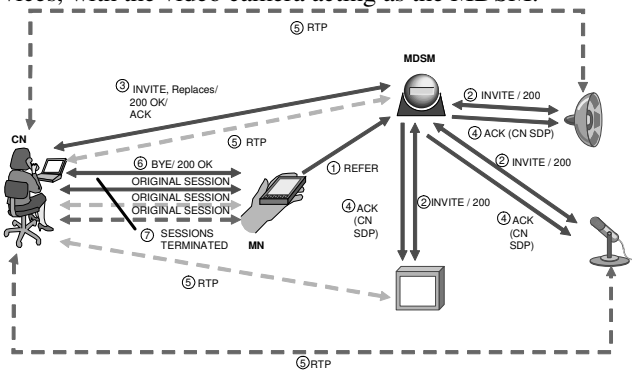

Figure 5 Session handoff to a multi-device system.

VI. Reconciling Device Capability Differences

Session mobility sometimes involves the transfer of a session between devices with differing capabilities. For example, the codec being used in the current session may not be available on the new device. Furthermore, that device may not support any codec that is supported by the CN. In addition to codecs, devices may have different resolutions or bandwidth

requirements which should be taken into account when carrying out session transfer.

Before executing a session transfer, the device must check the capabilities of the CN and the new device. These may be found through either the SIP OPTIONS method, used in SIP to query a device's media capabilities, or may be included as SLP service attributes. Since the OPTIONS method is standard, it should be used to query the CN, while SLP should be used to get the media capabilities of local devices, since it is already being used for them.

If the CN and the local device are found to have a common codec, the transfer should be carried out so that it becomes the codec used in the new media session. In Mobile Node Control Mode, the flow is identical, but the SDP bodies include the desired codec. In Session Handoff Mode, the MN sends a REFER request to the local device and allows it to negotiate a common codec with the CN.

If a common codec does not exist, the MN must execute the transfer through an intermediate transcoding service. Rather than establishing a direct media session between the CN and the local device, separate sessions are established between the transcoder and each of them, with the transcoder translating between the streams. The transcoding service need not be geographically local, and the Mobile Node may discover available transcoders through SLP.

Current standardization [12] uses third-party call control for transcoding. The standard case involves a controller, party A, that initiates a media session with party B through a transcoder. The controller invites the transcoder into a session and provides the media parameters of itself and B. The transcoder responds with a "200 OK" that includes its own media parameters, namely the ports on which it will receive each stream. The controller then establishes a separate session with B, in which it gives it the address and port of the transcoder as the destination of its media. It also establishes its own media session with the transcoder. When both sessions are established, two media streams have been established through the transcoder.

In Mobile Node Control mode, the Mobile Node establishes a media session between the transcoder and the CN, and one between the transcoder and the local device. The initial INVITE sent by the MN to the transcoder includes a session description referring to the CN and the local device, rather than including its own parameters as in the standard case. It then establishes a separate session with each of those nodes. Once the three sessions have been established, two media sessions exist, and the transcoder translates between them.

In Session Handoff mode, the local device itself must establish a session with the CN through the transcoder. After receiving the REFER request, it uses the OPTIONS method to find the capabilities of the CN. It will then use a common codec, if available, in the session setup, or set up the transcoded session using third-party call control as in [12].

Other differences in device capabilities, such as display resolution and bandwidth limitations, should also be reconciled during tranfer. For example, a mobile device, limited both in its display size and bandwidth, will likely be receiving the video stream from the other call participant at a low resolution and framerate. When the user transfers his video output to a large-screen display, he may start viewing much higher quality video at the higher native resolution of the display and at a higher framerate. These two parameters may be handled using existing protocol standards.

Changing the image resolution and framerate requires no special handling by the MN. An SDP format is defined [13] for specifying these and other parameters for the H.263+ codec. The suitable image formats and corresponding MPIs (Minimum Picture Interval, related to the framerate) supported for the given codec are listed following the media line, in order of preference. For example, the following lines in SDP would

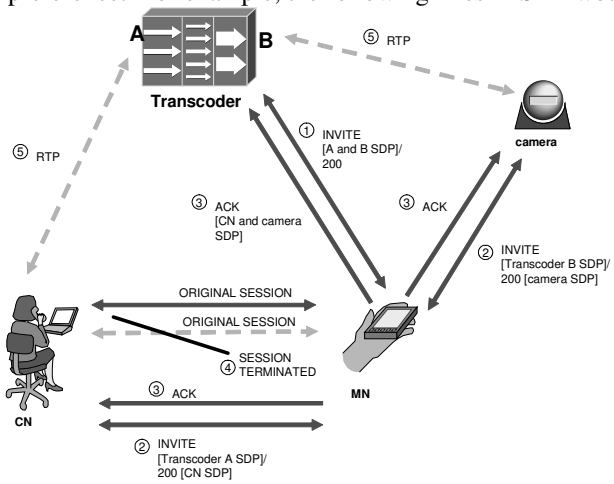

Figure 6 Transfer of a session in Mobile Node Control mode through a transcoder to translate between native codecs of CN and a camera, where they share no common codec.

indicate that a device supports the H.263 codec (value 34) with the image sizes of 16CIF, 4CIF, CIF and QCIF (with the MPI for each format following the "="):

```
m=video 60300 RTP/AVP 34
a=fmtp:34 16CIF=8;4CIF=6;CIF=4;QCIF=3
```

In Mobile Node Control mode, the response by the local device (Figure 3, F2 and F4) to the initial INVITE request sent by the MN would include this line in the SDP body, and the MN would then include it in the INVITE request sent to the CN (F5). In Session Handoff mode, the local device would include this parameter in the INVITE request sent to the CN (Fig. 4(a), F3) after receiving the REFER request. If the local device is not mobility-enhanced, and is, therefore, not configured to include the supported image sizes during session establishment, the information could be made available through SLP. The MN would then include it in the INVITE request sent to the CN in mobile node control mode. However, this information would not be sent in Session Handoff mode unless the local device were configured to send

it. In both modes, the MN would send its own resolution and framerate preferences in the body of the INVITE request sent to retrieve the session.

VII. Performance Analysis

In order to enjoy the advantages of session mobility, the transfer should minimize the disruption of service, and should be quick. Below, we analyze the expected performance of our system, based on these criteria.

A. Disruption of Media During Transfer

The most critical disruption is the "dead time" between tear-down of the existing media stream and the establishment of the new stream. Transfer could cause a segment of media to be unplayed during this time, which we refer to as "lapse." This occurs when one side begins sending media to a device not ready to display it, because the necessary session establishment has not been done. It may also occur if the device involved in the current session stops sending media before the new device begins sending media. Even if there is no lapse, there may be a delay during which one or both sides do not receive media on any device. The flows already shown in Figures 3 and 4 ensure that this delay is no longer than the time it takes a single packet to travel between the remote participant and the local network, and that the lapse is either non-existent or negligibly small. Therefore, the mobile user need not warn the participant at the Correspondent Node: "Wait for me until I complete this transfer."

In the Mobile Node Control mode flow of Figure 3, even though the CN receives a new destination address for its media in message F5, it will not stop receiving and displaying the incoming flows from the MN until it starts receiving media streams from the local devices, initiated after messages F1 and F3. While the MN has no way of knowing when this happens, it can wait a safe amount of time, such as a second, before it stops sending. Therefore, the CN will not experience any disruption of media flow. The mobile user may experience a negligibly short delay in incoming media service. The CN stops sending media to the MN and starts sending to the local devices after receiving the INVITE request with new media parameters, as shown in the figure. Since the local devices are already listening for the media following the INVITE requests received by the MN, they will begin displaying the media as soon as they receive it. Therefore, no media lapse will occur. The only delay will be the time it takes for the RTP packets to travel to the devices. During some of that time, as well, old media packets will still be received by the MN. The delay will, therefore, be extremely short.

In Session Handoff mode (Fig. 4), also, there will be no media lapse. After the CN receives the INVITE request (F3) from the local device, it immediately redirects the media from the MN to the local device. Once the media stream reaches the local device, it will immediately begin displaying it. Therefore, all media recorded by the CN will be displayed on some device for the mobile user. There may be a negligibly short delay between the time the MN stops receiving the media and the local device begins displaying it. The CN will not experience any lapse or delay, since it will display the media from the MN until it starts receiving the streams from the local device, following message F4.

B. Total Transfer Latency

A less critical, but important, concern is the total delay in transferring a session—from the time that the mobile user makes the request until he begins to receive the CN's media streams on the new device. Firstly, the user may be away from the current device when transfering to a second one. For example, the mobile user may be retrieving a session onto his mobile device while walking out of the office. Even if the user has both devices in front of him, the latency should be small enough to provide seamless use of the devices. Furthermore, a long delay may lead the mobile user to believe that the transfer is not working, as mentioned regarding ordinary telephone call setup in [22]. Therefore, we give an estimate of the transfer latency in a typical network.

Both models presented in the paper use flows that consist of signaling between the local environment (MN and local devices) and the CN, which may traverse a long distance, and signaling within the local environment. Previous work [22] has measured transcontinental call setup delay in SIP to be below one second. We assume that the network-layer packet loss and latency over the wireless links connecting the MN, and possibly the CN and local devices, will not significantly affect this figure. Therefore, we use this figure for call setup between the local environment and the CN. Delay in session creation with the local devices depends on the route taken. If the sessions are established through the proxy servers and the mobile user's proxy server is far away, the setup may require a triangular route to be traversed. If, on the other hand, either the mobile user's proxy is local or peer-to-peer setup is done, without the proxy, setup time should be negligible. We assume that this setup time is very short.

In Mobile Node Control mode, the call flow consists of session creation with each local device, followed by a session update with the CN, which has the same signaling as a normal call setup. We therefore estimate that the transfer delay should not be much longer than a second. In Session Handoff mode, as well, the call flow consists of a single call setup between the local network and the CN, and signaling between the MN and the local devices, such as the REFER request. Here, too, the transfer should not take much longer than a second.

VIII. Security and Privacy Considerations

Many security concerns must be addressed in the local device environment. Here we give ways to handle two such concerns, the unauthorized use of devices and the use of input devices for surveillance [14].

A. Authorization for using local devices

Public devices generally have a group of users who are authorized to use and to transfer their sessions to them. It is essential that any other users are not allowed access, in order

not to limit the usage of authorized users. Two solutions may be used to allow the device to authenticate the user without keeping a list of users. The devices are likely to have a category of users who are authorized to transfer their sessions to them, rather than a list of individual users. For instance, a service provider who installs such devices in a public area will have a group of customers who pay for the service. Since a user should only be required to show evidence that he falls into the given category and his personal identity is not important, trait-based authorization [26] may be used. The user authenticates himself to a server called an authorization service and sends it the SIP message it intends to send to the device. The server returns a digital signature asserting that the sender of that SIP message is in the category of users authorized to use the device. The mobile node then sends the signature to the local device in the body of the same INVITE or REFER request. The device checks the signature and trusts its authenticity, so it accepts the request. Another solution is the use of Authentication, Authorization and Accounting (AAA) in SIP [24].

B. *Privacy concerns for input devices*

Input devices such as cameras and microphones could be used for surveillance. This concern can be mitigated in two ways. First of all, the remote control of devices could be disallowed by requiring proof that the user is actually in the location of the device. This could be done by requiring an authentication token that is only available locally [14]. Such a token would regularly change and would be passed to the mobile device by a low-power Bluetooth beacon, with its ray restricted to a single room. This token would then be used in Digest Authentication. In order to keep a local user from transferring an ongoing session, leaving the room and eavesdropping, the device should also contain an LED to warn other users that a session is currently active.

IX. Conclusion and Future Plans

We have presented an architecture based on SIP and SLP to provide users with a seamless environment for discovering devices, as well as including multiple devices and transferring between them in mid-session. This allows mobile users to benefit from the advantages of both mobile and stationary devices. We have shown that the system should perform well during session transfer, causing minimal disruption to the parties involved in the session.

We have implemented a prototype of our architecture using sipc [23], Columbia University's SIP UA. We have enhanced sipc to query for local services when its location changes and to transfer an ongoing session to one or more devices using Mobile Node Control mode.

Our future plans include creating a prototype of the system on the Microsoft Windows Mobile™ Platform and completing the implemention of the system described here. We also plan to specificy how the services described could be integrated into emerging 3G wireless environments and to more fully handle security and privacy concerns.

References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Sparks, R.,Handley, M., Schooler, E.: SIP: Session Initiation Protocol, IETF RFC3261 (2002)

[2] 3GPP: TS 23.228: IP Multimedia Subsystem (IMS) (Stage 2) – Release 5, September 2002, ftp://ftp.3gpp.org/Specs/archive/23_series/23.228/

[3] Kaneko K., et al.: Session Layer Mobility Support for 3C Everywhere Environments, WPMC03, October 2003.

[4] Hasekawa M.: Cross-Device Handover Using the Service Mobility Proxy, WPMC03, October 2003.

[5] Schulzrinne H., Wedlund E.: Application-Layer Mobility Using SIP, ACM Mobile Computing and Communications Review, Vol. 4, No. 3, July 2000.

[6] Rosenberg, J., Peterson, J., Schulzrinne, H., Camarillo, G.: Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP), IETF RFC 3725 (2004)

[7] Sparks, R.: The Session Initiation Protocol (SIP) Refer Method, IETF RFC3515 (2003)

[8] Y. Kikuta, et al. "Design of Seamless Service Environment for Adaptive Service Transfer Among Terminals," 8th International Conference on Mobile Multimedia Communications (MoMuC 2003), Munich, October, 2003.

[9] H. Song, et al. "Browser Session Preservation and Migration," 11th International World Wide Web Conference (WWW2002), Hawaii, May 2002.

[10] Gutman, E., Perkins, C., Veizades, J., Day, M.: Service Location Protocol, Version 2, IETF RFC 2608 (1999)

[11] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications, IETF RFC 3550 (2003)

[12] Camarillo, G., Burger, E., Schulzrinne, H., Van Wijk, A.: Transcoding Services Invocation in the Session Initiation Protocol Using Third Party Call Control, draft-ietf-sipping-transc-3pcc-02, IETF Internet Draft (2004) (Work in progress)

[13] Ott, J., Sullivan, G., Wenger, S., Even, R.: RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+), draft-ietf-rfc2429-bis-03, IETF Internet Draft (2004) (Work in progress)

[14] Berger, S., Schulzrinne, H., Sidiroglou, S., Wu, X.: Ubiquitous Computing Using SIP, ACM NOSSDAV 2003 (2003)

[15] Handley, M., Jacobson, V.: SDP: Session Description Protocol, IETF RFC 2327 (1998)

[16] Rosenberg, J., Schulzrinne, H.: An Offer/Answer Model with the Session Description Protocol (SDP), IETF RFC 3264 (2002)

[17] Roach, A.: Session Initiation Protocol (SIP)-Specific Event Notification, IETF RFC 3265

[18] Peterson, J.: A Presence-based GEOPRIV Location Object Format, IETF Internet Draft (2004) (Work in progress)

[19] Global Positioning System Standard Positioning Service Specification, 2nd Edition, June 2, 1995

[20] Sparks, R.: The SIP Referred-By Mechanism, draft-ietf-sip-referredby-05, IETF Internet Draft (2004) (Work in progress)

[21] Mahy, R., Biggs, B., Dean, R.: The Session Initiation Protocol (SIP) "Replaces" Header, draft-ietf-sip-replaces-05, IETF Internet Draft (2004) (Work in progress)

[22] Eyers, T., Schulzrinne, H.: Predicting Internet Telephony Call Setup Delay, in Proceedings of First IP Telephony Workshop, Berlin, Germany, Apr. 2000

[23] Wu, X., Schulzrinne, H.: SIPc, a multi-function SIP user agent, Proceedings of 7th IFIP/IEEE Internation Conference, Management of Multimedia Networks and Services (MMNS '04), pp. 269-281, Oct. 2004, San Diego, CA

[24] Loughney, J., Camarillo, G.:Authentication, Authorization and Accounting Requirements for the Session Initiation Protocol (SIP), IETF RFC 3702 (2004)

[25] Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifiers (URI): Generic Syntax, IETF RFC 2396 (1998)

[26] Peterson, J., Polk, J., Sicker, D., Tschofenig, H.: Trait-based Authorization Requirements for the Session Initiation Protocol (SIP), draft-ietf-sipping-trait-authz-00 IETF Internet Draft (2004) (Work in progress)