Network Working Group
Request for Comments: 5631
Category: Informational

R. Shacham
H. Schulzrinne
Columbia University
S. Thakolsri
W. Kellerer
DoCoMo Euro-Labs

October 2009

Session Initiation Protocol (SIP) Session Mobility

Abstract

Session mobility is the transfer of media of an ongoing communication session from one device to another. This document describes the basic approaches and shows the signaling and media flow examples for providing this service using the Session Initiation Protocol (SIP). Service discovery is essential to locate targets for session transfer and is discussed using the Service Location Protocol (SLP) as an example. This document is an informational document.

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright and License Notice

Table of Contents

1. Overview

As mobile devices improve and include more enhanced capabilities for IP-based multimedia communications, they will remain limited in terms of bandwidth, display size, and computational power. Stationary IP multimedia endpoints (including hardware IP phones, videoconferencing units, embedded devices and software phones) allow more convenience of use, but are not mobile. Moving active multimedia sessions between these devices allows mobile and stationary devices to be used concurrently or interchangeably in mid-session, combining their advantages into a single "virtual device". An approach to session mobility based on the Session Initiation Protocol (SIP) [1] was described first in [20], where two different methods are proposed: third-party call control (3pcc) [2] and the REFER method [3].

This document expands on this concept, defining a framework for session mobility that allows a Mobile Node to discover available devices and to include them in an active session. In particular, the framework for session mobility presented in this document describes basic approaches for using existing protocols and shows the signaling and media flow examples for providing session mobility using SIP. It is intended as an informational document.

The devices selected as session transfer targets may be either personal or public. Personal devices are ones used by a single individual, such as one’s PC or phone. Public devices are ones available for use by a large group of people and include large conference-room displays. Two capabilities are required to transfer sessions:

Device Discovery - At all times, a user is aware of the devices that 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 sessions to one or more devices.

This document describes session mobility examples for SIP. It does not mandate any particular protocol for device discovery. Many different protocols exist and we discuss the tradeoffs involved in choosing between them. For our examples, we use the Service Location Protocol (SLP) [17], primarily because it is the only such protocol standardized by the IETF.

2. Requirements

This session mobility framework seeks to fulfill the following requirements:

- REQ 1: Backward Compatibility - We distinguish two kinds of devices. Enhanced devices support the call flows described in Section 5 and can perform discovery, while basic devices can do neither and only have basic SIP capabilities. Devices initiating session mobility must have enhanced functionality, while all others can be either basic or enhanced devices. This includes the transfer destinations, such as the local video camera, as well as the device being used by the remote participant.

- REQ 2: 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.

- REQ 3: Minimal Disruption - Session transfer should involve minimal disruption of the media flow and should not appear to the remote participant as a new call.

3. Roles of Entities

Session mobility involves five types of components: A Correspondent Node (CN), a Mobile Node (MN), one or more local devices used as targets for session transfer, an SLP [17] Directory Agent (DA), and, optionally, a transcoder. The Correspondent Node (CN) is a basic multimedia endpoint being used by a remote participant and may be located anywhere. It may be a SIP User Agent (UA), or a Public Switched Telephone Network (PSTN) phone reachable through a gateway. The Mobile Node (MN) is a mobile device, containing a SIP UA for standard SIP call setup, as well as specialized SIP-handling capabilities for session mobility, and an SLP [17] User Agent (UA) for discovering local devices. The local devices are located in the user’s local environment for discovery and use in his current session. They may be mobility-enhanced or basic. Basic devices, such as IP phones, are SIP-enabled, but have no other special capabilities. Mobility-enhanced devices 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 5.4.3. The SLP Directory Agent (DA) keeps track of devices, including their locations and capabilities. The use of SLP will be described in more detail in Section 4. SIP-based transcoding services [18] are used,

when necessary, to translate between media streams, as described in Section 6.

4. Device Discovery

A Mobile Node must be able to discover suitable devices in its vicinity. This is outside the scope of SIP, and a separate service location protocol is needed. It is outside the scope of this document to define any service location protocol. This section discusses various options, and describes one of them in more detail.

While having a global infrastructure for discovering devices or other services in any location would be desirable, nothing of this sort is currently deployed or standardized. However, this document assumes that such an infrastructure is unnecessary for discovering devices that are in close proximity, such as in the same room. It is possible for such devices to be discovered through direct communication over a short-range wireless protocol such as the Bluetooth Session Description Protocol (SDP). Two other categories of service discovery protocols may be used, assuming that devices that are physically close to each other are also within the same network and/or part of the same DNS domain. Multicast-based protocols, such as SLP [17] (in its serverless mode) or Bonjour (mDNS-SD [30]), may be used as long as the Mobile Node is within the same subnet as the local devices. When devices are part of the same DNS domain, server-mode SLP or non-multicast DNS Service Discovery (DNS-SD) [29] are possible solutions. Such protocols can discover devices within a larger geographical area, and have the advantage over the first category in that they allow for the discovery of devices at different location granularities, such as at the room or building level, and in locations other than the current one. In order to discover devices in a specific location, location attributes, such as room number, must be used in the search, e.g., as service attributes in SLP or as a domain name in DNS-SD. The mobile device must ascertain its location in order to perform this search. We note that many of these techniques could be difficult to implement in practice. For example, different wireless networks may be deployed by different organizations, which could make it unrealistic to have a common DNS setup.

We describe here how SLP is used in server mode in general, then how it may be used to discover devices based on their location. As mentioned before, this is only one of many ways to perform service discovery. 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. The attributes may be information such as vendor, supported media codecs, and display resolution. SLP defines three roles: Service Agents (SAs), which send descriptions of services;

User Agents (UAs), which query for services; and Directory Agents (DAs), which receive the registrations and queries. An SA registers a service description to a 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 a 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. It may then ascertain the specific attributes of each service using an attribute request (AttrRqst) message.

This document assumes the following use of SLP for discovering local devices. Devices have a service type of "sip-device" and a SIP URI as the service URI. Section 5.2 describes the form of this URI. Attributes specify device characteristics such as vendor, supported media codec, display resolution, as well as location coordinates, such as street address and room number. SAs are co-located with SIP UAs on session-mobility enhanced devices, while a separate SA is available to send SrvReg messages on behalf of basic devices, which do not have integrated SLP SAs.

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. 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 that are not in the current room. A SrvRply message is sent to the mobile device with a list of SIP URIs for all devices in the room. A separate attribute request (AttrRqst) is then sent for each URL to get the attributes of the service. The MN displays for the user the available devices in the room and their attributes. Figure 1 shows this protocol flow.

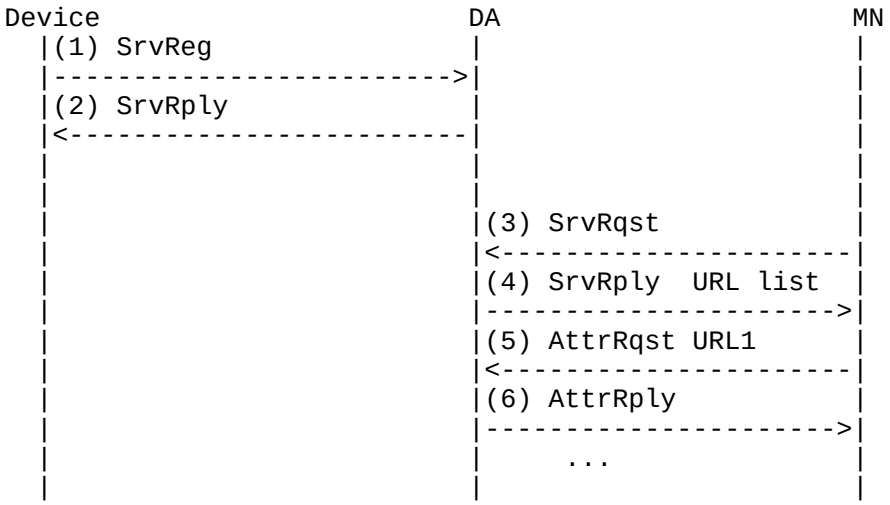

Figure 1. SLP message flow for the device to register its service and the MN to discover it, along with its attributes.

5. Session Mobility

5.1. Options for Session Mobility

5.1.1. Transfer and Retrieval

Session mobility involves both transfer and retrieval of an active session. A transfer means to move the session on the current device to one or more other devices. A retrieval causes a session currently on another device to be transferred to the local device. This may mean returning 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 to that device; when he walks away, he returns the stream to his mobile device for continued communication. One may also move a session to a device that had not previously carried it. For example, a participant in an audio call on his stationary 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.

5.1.2. Whole and Split Transfer

The set of session media may either be transferred completely to a single device or split across multiple devices. For instance, a user may only wish to transfer the video component of his session while maintaining the audio component on his PDA. Alternatively, he may find separate video and audio devices and wish to transfer one media

type 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.

5.1.3. Transfer Modes

Two different modes are possible for session transfer, Mobile Node Control (MNC) mode and Session Handoff (SH) mode. We describe them below in turn.

5.1.3.1. Mobile Node Control (MNC) Mode

In Mobile Node Control mode, the Mobile Node uses third-party call control [2]. 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 sessions with the CN. The shortcoming of this approach is that it requires the MN to remain active to maintain the sessions.

5.1.3.2. Session Handoff (SH) Mode

A user may need to transfer a session completely because, for example, the battery on his mobile device is running out or he is losing radio connectivity. Alternatively, the user of a stationary device who leaves the area and wishes to transfer the session to his mobile device 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. In such a case, Session Handoff mode, which completely transfers the session signaling and media to another device, is useful.

Based on our experiments, we have found MNC 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.

5.1.4. Types of Transferred Media

A communication session may consist of a number of media types, and a user should be able to transfer any of them to his device of choice. This document considers audio, video, and messaging. Audio and video are carried by RTP and negotiated in the SDP body of the SIP requests and responses. Three different methods exist for carrying text messages, and possibly other MIME types, that are suitable for SIP endpoints. RTP may be used to transport text payloads in real time,

based on [9]. Any examples given for audio and video will work identically for text, as only the payloads differ. For the transfer of entire messages (as opposed to a small number of characters in RTP), either the SIP MESSAGE method [21] or the Message Session Relay Protocol (MSRP) [7] may be used. MESSAGE is used to send individual page-mode messages. The messages are not associated with a session, and no negotiation is done to establish a session. Typically, a SIP UA will allow the user to send MESSAGE requests during an established dialog, and they are sent to the same contact address as all signaling messages are sent in mid-session. We discuss later how these messages are affected by session mobility. MSRP, on the other hand, is based on sessions that are established like the real-time media sessions previously described. As such, transferring them is similar to transferring other media sessions. However, this document will point out where special handling is necessary for these types of sessions.

5.2. Addressing of Local Devices

As stated before, this document assumes both personal and public devices. We assume that public devices use a dedicated Address of Record (AOR), such as sip:device11@example.com. A personal device already uses the owner's AOR, so that he should be reachable there; that AOR could also be used for transferring sessions. However, it is preferable to distinguish the role of a device as a transfer target from its existing role. Therefore, all devices are assumed to have dedicated AORs.

Since every transfer device has its own AOR, there is a one-to-one mapping between AOR and device. Therefore, a transfer request could be addressed to the AOR, which would resolve to the device. However, in Section 5.4.3, we present a model where devices create multi-device systems to pool their capabilities. Therefore, a single device must be reachable by multiple URIs representing different combinations of devices. The appropriate solution is to define each combination of devices with a Globally Routable UA URI (GRUU) [12].

Therefore, we assume the following addressing for the remainder of the document. As mentioned earlier, a device has a unique AOR. It registers a separate contact URI for itself and for each system of devices that it controls. Each contact has an associated GRUU, which is registered with SLP as the Service URI, and may be directly addressed by another device in a request sent through the proxy. When the proxy forwards the request to the device, it will replace the GRUU with the contact URI, as described in [12]. Therefore, the contact URI, not the associated GRUU, will be used by devices to determine whether the request is for the device itself or for a multi-device system. We assume that the public GRUU is used.

5.3. Mobile Node Control Mode

5.3.1. Transferring a Session to a Single Local Device

5.3.1.1. RTP Media

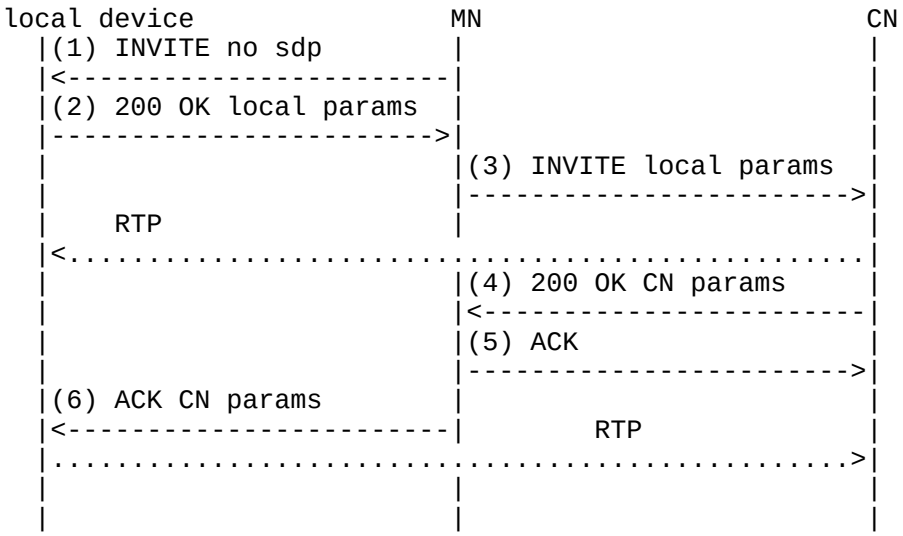

Figure 2. Mobile Node Control mode flow for transfer to a single device.

Figure 2 shows the message flow for transferring a session to a single local device. It follows Third Party Call Control Flow I (specified in [2]), 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, but does not include an SDP body. The local device's response contains an SDP body that includes the address and port it will use for any media, as well as a list of codecs it supports for each. The MN updates the session with the CN by sending an INVITE request (re-INVITE) 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 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 5400 RTP/AVP 31 34
a=rtpmap:31 H261/90000
a=rtpmap:34 H263/90000
```

Sending both audio and video media lines will transfer both media sessions of an existing audio/video call to the local device. Alternatively, the MN may select a subset of the media available on the local device, and use the local device's parameters for those media in the request sent to the CN, while continuing to use its own parameters for the rest of the media. For example, if it only wishes to transfer an audio session to a local device that supports audio and video, it will isolate the appropriate media line for audio from the response received from the local device and put it in the request sent to the CN, along with its own video parameters. The CN will send 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 will send an ACK message to the local device, which includes these parameters in the body. The MN will establish a session with the local device and maintain its session with the CN, while the media flow will be established directly between the CN and the local device. Only the MN, who will be in an ongoing session with the CN, will later be allowed to retrieve the media session. Parsing of unknown SDP attributes by the controller is discussed in [2].

5.3.1.2. MSRP Sessions

In figure 2, the message sequence for transferring an MSRP message session using MNC mode is identical to that used for audio or video, although the contents of the messages differ. To simplify the example, we assume that an MSRP session, with no other media, is being transferred to a local messaging node, MSGN. In the following flow, we refer to the corresponding messages in Figure 2. An empty INVITE request (1) is sent to the local messaging node, MSGN, as follows:

```
INVITE sip:msgn@example.com;gr=urn:uuid:jtr5623n SIP/2.0
To: <sip:msgn@example.com;gr=urn:uuid:jtr5623n>
From: <sip:bob@example.com>;tag=786
Call-ID: 893rty@mn.example.com
Content-Type: application/sdp
```

The messaging node responds with all of its media capabilities, including MSRP, as follows (2):

```
SIP/2.0 200 OK
To: <sip:msgn@example.com;gr=urn:uuid:jtr5623n;tag=087js>;tag=087js
From: <sip:bob@example.com>;tag=786
Call-ID: 893rty@mn.example.com
Content-Type: application/sdp
```

```
v=0
c=IN IP4 msgn.example.com
m=message 52000 msrp/tcp *
a=accept-types:text/plain
a=path:msrp://msgn.example.com:12000/kjhd37s2s2;tcp
m=audio 4400 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 5400 RTP/AVP 31 34
a=rtpmap:31 H261/90000
a=rtpmap:34 H263/90000
```

The same request is then sent by the MN to the CN (3), but containing the MSRP media and attribute lines with the path given in the MSGN response above. The CN responds (4) with its own path. The MN includes this in the ACK that it sends to the MSGN (6).

MSRP sessions are carried over a reliable connection, using TCP or TLS (Transport Layer Security). Therefore, unlike in the case of real-time media, this connection must be established. According to the MSRP specifications, the initiator of a message session, known as the "offerer", must be the active endpoint, and open the TCP connection between them. In this transfer scenario, the offerer of both sessions is the MN, who is on neither end of the desired TCP connection. As such, neither endpoint will establish the connection. A negotiation mechanism could be used to assign the role of active endpoint during session setup. However, while MSRP leaves open this possibility, it is not currently included in this document due to complexity. The only other way that such session transfer would be possible is if both the CN and the local device ordinarily use an MSRP relay [8], since no direct connection must be established between them. When each new endpoint receives the INVITE request from the MN, it will create a TLS connection with one of its preconfigured relays if such a connection does not yet exist (the CN will already have one because of its session with the MN) and receive the path of the relay. In its response to the MN, it will include the entire path that must be traversed, including its relay, in the path attribute. For instance, the response from the MSGN will look as follows:

```
SIP/2.0 200 OK
To: <sip:msgn@example.com;gr=urn:uuid:jtr5623n;tag=087js>;tag=087js
From: <sip:bob@example.com>;tag=786
Call-ID: 893rty@mn.example.com
Content-Type: application/sdp
```

```
v=0
c=IN IP4 msgn.example.com
m=message 52000 msrp/tcp *
a=accept-types:text/plain
a=path:msrp://relayA.example.com:12000/kjhd37s2s2;tcp \
     path:msrp://msgn.example.com:12000/kjhd37s2s2;tcp
```

Since the CN and the local device each establish a TLS connection with their relay, as they would for any session, and the relays will establish a connection between them when a subsequent MSRP message is sent, neither party needs to establish any special connection. The existing protocol may therefore be used for session transfer.

5.3.2. Transfer to Multiple Devices

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 appropriate media parameters it receives in their responses. For instance, in order to transfer an audio and video call to two devices, the MN initiates separate sessions with each of them, combines the audio media line from one response and the video media line from the other, and sends them together as the request to the CN, as follows:

```
v=0
m=audio 48400 RTP/AVP 0
c= IN IP4 audio_dev.example.com
a=rtpmap:0 PCMU/8000
m=video 58400 RTP/AVP 34
c= IN IP4 video_dev.example.com
a=rtpmap:34 H263/90000
```

The CN responds with its own parameters for audio and video. The MN splits them and sends one to each local device in the ACK that completes each session setup.

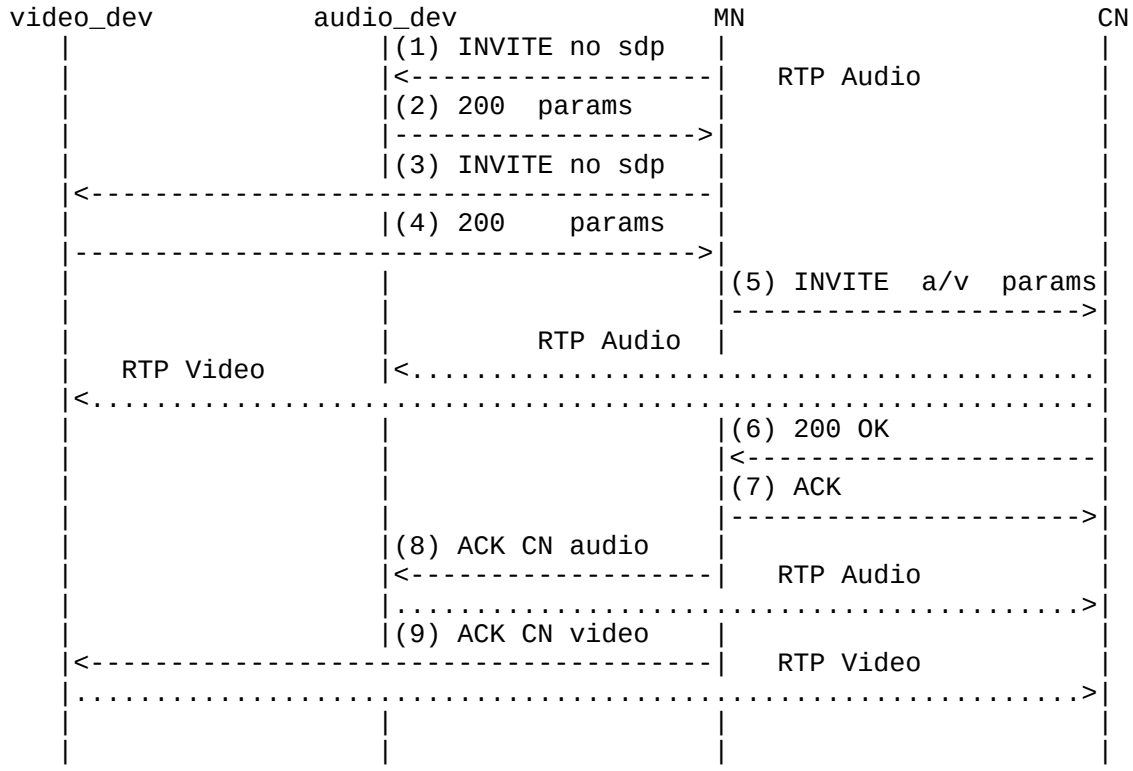

Figure 3. Mobile Node Control mode flow for transfer to multiple devices.

Splitting a full-duplex media service, such as video, across an input and an output device, such as a camera and a video display, 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 the local devices into two different sessions, but does not include any SDP body. They each respond with all of their available media. If they only support unidirectional media, as is the case for a camera or display-only device, they will include the "sendonly" or "recvonly" attributes. Otherwise, the MN will have to append the appropriate attribute to each one’s media line before sending the combined SDP body to the CN. That body will look as follows:

```
m=video 50900 RTP/AVP 34
a=rtpmap:34 H263/90000
a=sendonly
c=IN IP4 camera.example.com
m=video 50800 RTP/AVP 34
a=rtpmap:34 H263/90000
a=recvonly
c=IN IP4 display.example.com
```

In updating an SDP session, according to Section 8 of [4], the i-th media line in the new SDP corresponds to the i-th media line in the previous SDP. In the above cases, if a media type is added during the transfer, the media line(s) should follow the existing ones. When an existing media is transferred to a different device, the media line should appear in the same place that it did in the previous SDP, as should the lines for all media that have not been altered. When a duplex media stream is being split across an input and output device, the stream corresponding to the input device should appear in place of the duplex media stream. Since this new stream is the one that will be received by the CN, including it in place of the old one ensures that the CN views the new stream as a replacement of the old one. The media line corresponding to the output device must appear after all existing media lines. In the last example, if the SDP had initially contained a video line followed by an audio line, the updated SDP sent to the CN would look as follows:

```
m=video 50900 RTP/AVP 34
a=rtpmap:34 H263/90000
a=sendonly
c=IN IP4 camera.example.com
m=audio 45000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 50800 RTP/AVP 34
a=rtpmap:34 H263/90000
a=recvonly
c=IN IP4 display.example.com
```

During the course of the session, the CN may send a MESSAGE request to the MN containing text conversation from the remote user. If the mobile user wishes to have such messages displayed on a device other than the MN, the request is simply forwarded to that device. The forwarded message should be composed as though it were any other message from the MN to the local device, and include the body of the received message. The local device will send any MESSAGE request to the MN, who will forward it to the CN.

5.3.3. Retrieval of a Session

The MN may later retrieve the session by sending an INVITE request to the CN with its own media parameters, causing the media streams to return. It then sends a BYE message to each local device to terminate the session.

5.4. Session Handoff (SH) mode

5.4.1. Transferring a Session to a Single Local Device

Session Handoff mode uses the SIP REFER method [3]. This message is a request sent by a "referrer" 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 [5] header is used to give the referrer’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 Secure/Multipurpose Internet Mail Extensions (S/MIME) to authenticate the REFER request. Figure 4 shows the flow for transferring a session.

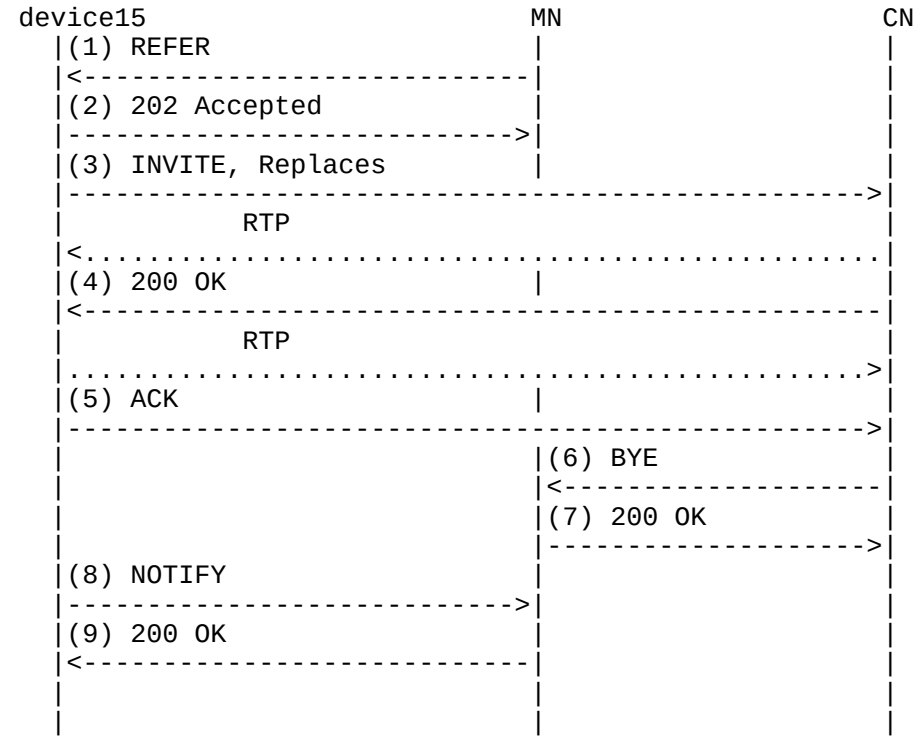

Figure 4. Session Handoff mode flow for transfer to a single device.

The MN sends the following REFER request (1) to a local device:

```
REFER sip:device15@example.com;gr=urn:uuid:qfnb443ccui SIP/2.0
To: <sip:device15@example.com;gr=urn:uuid:qfnb443ccui>
From: <sip:bob@example.com>
Refer-To:<sip:corresp@example.com;gr=urn:uuid:bbb6981;audio;video?
      Replaces="1@mn.example.com;
          to-tag=bbb;from-tag=aaa">
Referred-By: <sip:bob@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 in the Refer-To URI are callee capabilities [10]. Here they are used to inform the referee that it should initiate an audio and video session with the CN. Also included in the URI is the Replaces header field, specifying that a Replaces header field should be included with the specified value in the subsequent INVITE request. The Replaces

header identifies an existing session that should be replaced by the new session. Here, the local device will request that the CN replaces its current session with the MN with the new session. According to [6], the CN should only accept a request to replace a session from certain authorized categories 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 provides 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.

Once the local device has established a session with the CN, it sends a NOTIFY request to the MN, as specified in [3]. This NOTIFY contains the To (including tag), From (including tag), and Call-ID header fields from the established session to allow the MN to subsequently retrieve the session, as described in Section 5.4.2.

Once a session is transferred, the destination for MESSAGE requests moves automatically. Since a new session is established between the CN and the local device, any subsequent MESSAGE requests will be sent to that device.

The transfer flow described above for media sessions may also be used to transfer an MSRP session. The local device will initiate an MSRP session in message (4), along with the other sessions. The REFER request (1) indicates that an MSRP session should be established using callee capabilities in the Refer-To header field, as it does for audio and video. Such a media feature tag, "message" has already been defined [11]. Once the local device receives the REFER request, it initiates an MSRP session with the CN. As the initiator, it will establish a TCP connection in order to carry the session (as specified in [7]), or will set up the session through its relay if configured to do so.

5.4.2. Retrieval of a Session

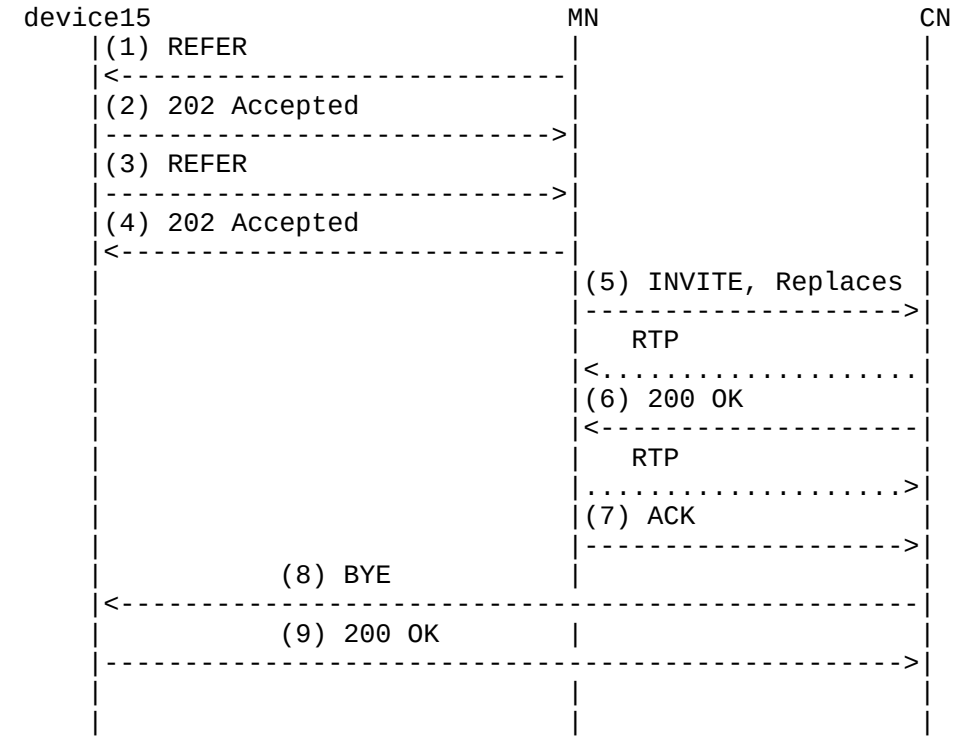

Figure 5. Session Handoff mode flow for session retrieval.

Figure 5 shows the flow for retrieval by the MN of a session currently on a local device. In order to better motivate the message flow, we start by describing the final INVITE (5) and work backwards. 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 (5):

```
INVITE sip:corresp@example.com;gr=urn:uuid:bbb6981 SIP/2.0
To: <sip:corresp@example.com;gr=urn:uuid:bbb6981>
From: <sip:bob@example.com>
Replaces: 1@device15.example.com;to-tag=aaa;from-tag=bbb
Referred-By: <sip:device15@example.com>

   [S/MIME authentication body]
```

Since the users on the MN and the local device are different identities, 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. Also, the user may wish to execute the transfer while running out of the office with mobile device in hand. In order for the MN to prompt the REFER from the local device, it sends a "nested REFER" [5], a REFER request for another REFER. 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 subsequent INVITE request. The REFER request from the local device to the MN (3) is composed as follows:

```
REFER sip:bob@example.com;gr=urn:uuid:ytav223h67gb3 SIP/2.0
To: <sip:bob@example.com;gr=urn:uuid:ytav223h67gb3>
From: <sip:device15@example.com>
Refer-To: <sip:correspondent@example.com;gr=urn:uuid:bbb6981;audio;
                    video?Replaces="1@device15.example.com;to-tag=aaa;
                    from-tag=bbb">
Referred-By: <sip:device15@example.com>

    [S/MIME authentication body]
```

A header field 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 (1):

```
REFER sip:device15@example.com;gr=urn:uuid:qfnb443ccui SIP/2.0
To: <sip:device15@example.com;gr=urn:uuid:qfnb443ccui>
From: <sip:bob@example.com>
Refer-To:<sip:bob@example.com;gr=urn:uuid:ytav223h67gb3;method=REFER
       ?Refer-To="<sip:correspondent@example.com;gr=urn:uuid:bbb6981;
               audio;video?Replaces=%221@device15.example.com;
               to-tag=aaa;from-tag=bbb%221>">
```

The Refer-To header specifies that the MN is the refer target and that the referral be in the form of a REFER request. The header field specifies that the REFER request contains a Refer-To header containing the URI of the CN. That URI, itself, contains the "audio" and "video" callee capabilities that will tell the MN to initiate an audio and video call, and a header field specifying that the ultimate INVITE request contains a Replaces header. If the MN had previously transferred the session to the local device, it would have received

these in the NOTIFY sent by the local device following the establishment of the session. If, on the other hand, the MN is retrieving a session it had not previously held, as mentioned above in Section 5.1.1, it gets these parameters by subscribing to the Dialog Event Package [13] of the local device. Such a subscription would only be granted, for instance, to the owner of the original device that carried the session. Even when these parameters are provided in the Replaces header, the local device does not accept the REFER request from anybody except the original participant in the session or the owner of the device. The MN receives the REFER request from the local device, sends the INVITE request to the CN, which accepts it, and, once the session is established, terminates its session with the local device.

5.4.3. Transfer to Multiple Devices

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 the requirement of seamlessness.

This document describes the use of multi-device systems to overcome this problem. A local device's SLP UA queries for other devices and joins with them to create a "virtual device", or a Multi-Device System (MDS). We refer to the controlling device as the Multi-Device System Manager (MDSM). In a system that includes at least one mobility-enhanced device, one of them may act as the MDSM. In a system consisting entirely of basic devices, either a dedicated host or another local device from outside of the system acts as MDSM. When the MDSM subsequently receives a REFER request, it uses third-party call control to set up media sessions between the CN and each device in the system. Specifically, it invites each local device into a separate session, and uses their media parameters (and possibly its own) in the INVITE request it sends to the CN.

A single device may act as an MDSM for several different groups of devices, and also act as an ordinary device with only its native capabilities. There must be a way to address a request to a device and specify whether it is to the device itself or one of the multi-device systems it controls. As mentioned above in Section 5.2, a device registers a separate contact for itself and for each of its

multi-device systems. For example, the device with AOR "sip:device11@example.com" and hostname "device11.example.com" will register a contact "sip:device11@device11.example.com" that represents its own capabilities. Once it discovers other devices and creates an MDS, it will register a new contact, "sip:av1@device11.example.com". It associates a GRUU with each of these contacts. The device itself and any new system is registered in SLP using the GRUU. When the proxy receives a request addressed to a GRUU, it will rewrite it as the contact URI before forwarding the request to the device. The device will use this unique contact to determine whether to handle the request natively or with one of its systems.

Figure 6 shows the transfer of a session to a multi-device system. The audio device has previously discovered the video device and created a multi-device system. The REFER request sent to "sip:device11@example.com;gr=urn:uuid:893eeeyuinm981" prompts the audio device to invite the video device into a session to ascertain its SDP, and then to invite the CN into a session using its own SDP and that of the video device.

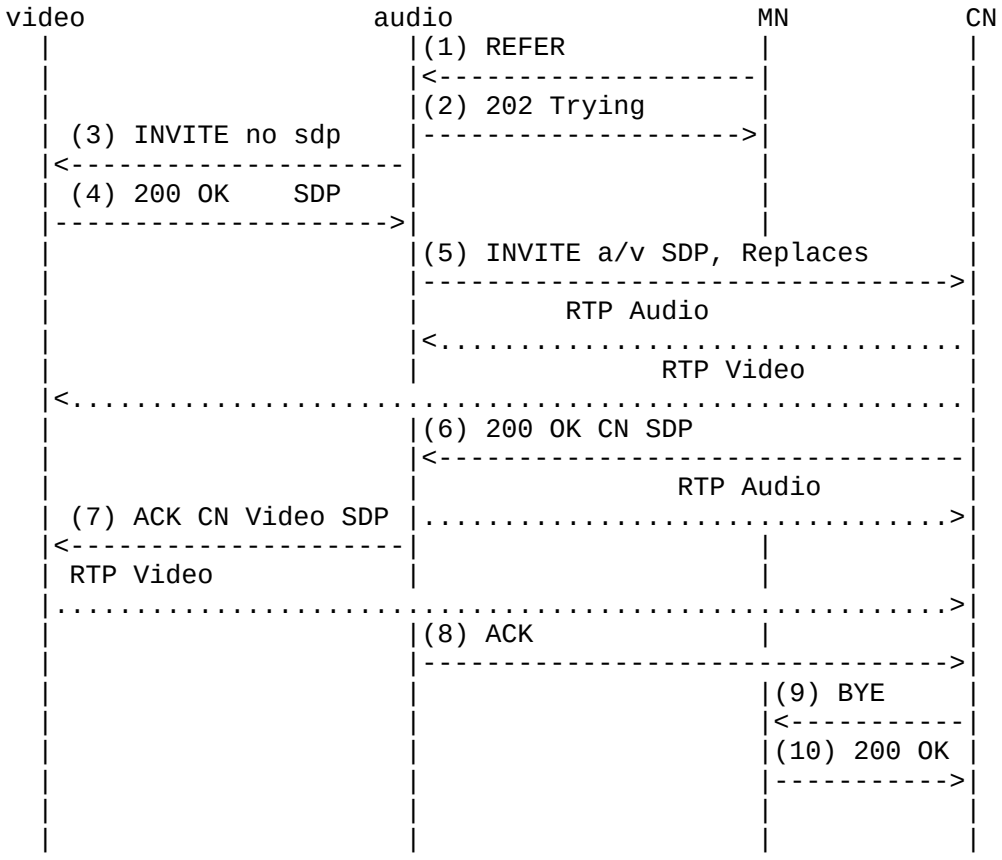

Figure 6. Session Handoff to a multi-device system.

5.5. Distributing Sessions for Incoming Call

The examples presented above have involved an established session that a user transfers to one or more devices. Another scenario would be for an incoming call to be immediately distributed between multiple devices when the user accepts the call. In such a case, the initial session would not yet be established when the transfer takes place.

The transfer could be carried out in either of the transfer modes. However, complete handoff to a separate device, which is done in Session Handoff mode, could be achieved through existing means, such as proxying or redirection. MNC mode would be useful in a case where the user wishes to automatically include an additional device in a call. For instance, a user with a desk IP phone and a PC with a video camera could join the two into a single logical device. The

SIP UA on the PC would, for any incoming call, send an INVITE request to the desk phone, setting the display name in the From header field to "Bob Jones (audio portion)", for instance, so that the user can identify the caller on the phone. The user could then either accept or reject, as he would with a call coming directly to the phone. If he accepts, the PC UA, acting as the controller, would respond to the caller with its video parameters and the phone's audio parameters in the SDP body. The final ACK from the Correspondent Node would then complete the session establishment.

If the desk phone is registered as a contact for the user, it would also ring in response to the direct call being proxied there, in addition to the INVITE request sent by the controller, causing confusion to the user. The use of caller preferences can solve this problem, as the caller would indicate that the call should preferentially be proxied to devices with audio and video capabilities. It is likely that the caller would use caller preferences in any case, if they were available to him, to avoid the callee unknowingly picking up the IP phone when he has a video-capable device available. However, since caller preferences are not yet widely supported on commercial devices, the callee must ensure the proper routing of the call. One solution would be for the PC to register its contact with a higher priority than the one given to the phone. The Call Processing Language (CPL) [22] (the "proxy" node) could then be used to specify that forking should be done to the set of user devices in sequence, rather than in parallel. Since all calls would first be sent to the PC as long as it were online, it would redirect any request that included only audio in its SDP.

5.6. Use of ICE in Session Mobility

Interactive Connectivity Establishment (ICE) [27] is a protocol for Network Address Translator (NAT) traversal that may be used with SIP. Rather than negotiating addresses and ports used for media sessions directly in SDP, a list of possible address/ports (candidates) is exchanged, and the Session Traversal Utilities for NAT (STUN) [28] protocol is used to check which pairs of candidates may be used. ICE could be used in the call flows described in this section. In MNC mode, the candidates would be sent by each local device to the MN, who would exchange them with the CN. Afterward, each device would perform checks with the CN to determine an appropriate candidate. In SH mode, where the local device establishes a session with the CN, ICE would work no differently than in the standard case.

6. Reconciling Device Capability Differences

Session mobility sometimes involves the transfer of a session between devices with different 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 limitations that must be taken into account when carrying out a session transfer.

6.1. Codec Differences

Before executing a session transfer, the device checks 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 is suggested to be used to query the CN, while SLP is suggested to 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 flow will negotiate that this should become the codec used in the media session. In MNC mode, the MN forwards the response from the local device to the CN, who will choose a codec it supports from those available. 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 during their session establishment. No special behavior of the MN is required.

If the MN sees that a common codec does not exist, it executes 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 Mobile Node discovers available transcoders through some means, including SLP.

Using transcoding services in SIP is defined in [18] using third-party call control. In MNC mode, the Mobile Node establishes one media session between the transcoder and the CN, and one between the transcoder and the local device. This differs from the normal transcoding case, where one party establishes a media session between itself and the transcoder and one between the transcoder and the other party. The MN starts by sending an INVITE request to the local device with no body; it receives in the response the list of codecs that the device can use. It then repeats this for the CN, and receives its available codecs. It chooses one codec from each side,

along with the address and port of each device, and combines them in an INVITE request sent to the transcoder. The transcoder responds with the ports on which it will accept each stream. The appropriate port information is sent individually to the CN and the local device. Once the three sessions have been established, two media sessions exist, and the transcoder translates between them. This flow is shown in Figure 7.

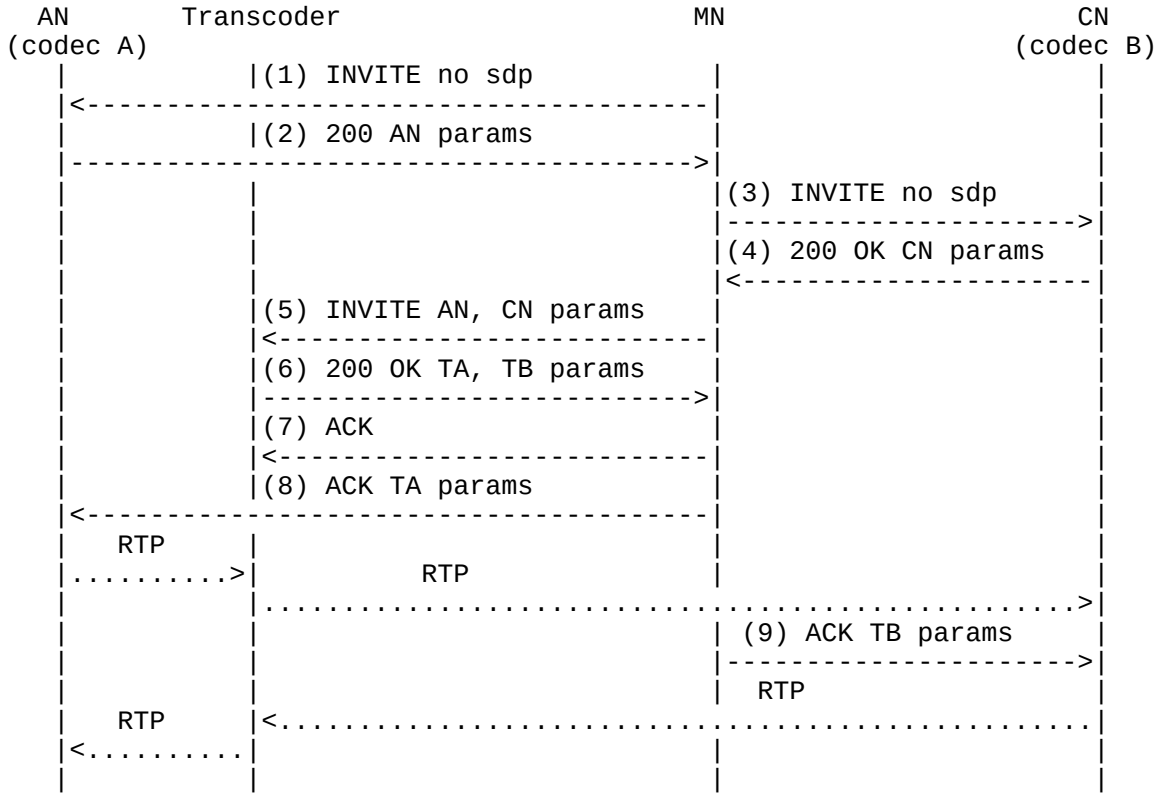

Figure 7. Transfer of a session in Mobile Node Control mode through a transcoder to translate between native codecs of CN and an audio node AN, where they share no common codec.

In Session Handoff mode, the local device itself establishes 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 [18].

6.2. Display Resolution and Bandwidth Differences

Other differences in device capabilities, such as display resolution and bandwidth limitations, are also suggested to be reconciled during transfer. 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 frame rate. 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 frame rate.

Changing the image resolution and frame rate requires no special handling by the MN. An SDP format is defined [19] for specifying these and other parameters for the H.263+ codec, for example. The suitable image formats and corresponding MPIs (Minimum Picture Interval, related to the frame rate) supported for the given codec are listed following the media line, in order of preference. For example, the following lines in SDP would 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 MNC mode, the response by the local device (Figure 2, message 2) to the initial INVITE request sent by the MN includes this line in the SDP body, and the MN then includes it in the INVITE request sent to the CN (3). In Session Handoff mode, the local device includes this parameter in the INVITE request sent to the CN (Figure 4, message 3) after receiving the REFER request. If the local device is not configured to include the supported image sizes during session establishment, the information could be made available through SLP. The MN then includes it in the INVITE request sent to the CN in Mobile Node Control mode. However, this information is not sent in Session Handoff mode unless the local device was configured to send it. In both modes, the MN sends its own resolution and frame rate preferences in the body of the INVITE request sent to retrieve the session.

7. Simultaneous Session Transfer

A session transfer may be carried out by one call participant after the other participant has transferred the session on his side. If the first transfer was done in MNC mode, a subset of the original session media is now on local devices. The MN receives either a re-INVITE from the other participant or an INVITE request from a local device on the other side. This message carries the new media parameters of the session. The MN, therefore, must send a re-INVITE

to any local devices with these parameters. It then includes the parameters returned from these devices in the 200 OK response. If the first transfer was done in SH mode, the local device will directly receive the session transfer message from the other party and will follow the normal procedure for responding to an INVITE request. If it is controlling other local devices for this session as part of an MDS, it follows the procedure above, where the first transfer was done in MNC mode.

It may occur that both participants attempt a transfer at the same time. In MNC mode, each node initiates a session with a local device, then sends a re-INVITE to the other node. Section 14.2 in [1] mandates a 491 response when a re-INVITE is received for a dialog once another re-INVITE has already been sent. Once both parties receive this response, they each generate a random timer with staggered intervals. Once its timer fires, each participant attempts the re-INVITE again. The first to receive it from the other participant responds to it with the SDP parameters of its local device. Both participants then send an ACK request to their local device containing the new parameters obtained from the other one during the re-INVITE process.

In SH mode, if both participants attempt a transfer at the same time, after one node sends a REFER request to the local device, it receives the INVITE request from the local device on the other end. The appropriate protocol definition could mandate that a 491 response be sent in this case, as well. This response would be returned to the referrer in a NOTIFY indicating the status of the referred session establishment. The staggered timer solution described above could work. The MN would cancel the REFER request sent to the local device, then wait a random amount of time before sending it again.

8. Session Termination

Once a session has been transferred, the user may terminate it by hanging up the current device, as he would do in a call originating on that device. This should be true even when the session is using several local devices. In MNC mode, when the user hangs up the current device, a BYE request is sent to the controller. The controller must then send a BYE request to each device used in the transfer and a BYE request to the CN. An MDSM used for SH mode must follow the same procedure. In SH mode, the current device has previously initiated an ordinary session with the CN in response to the REFER request, and the BYE it sends to the CN on hang-up requires no special handling.

9. Security Considerations

As this work is based heavily on the work in [2], [3], and [5], the security considerations described in those documents apply. We discuss here the particular issues of authorizing use of local devices, providing media-level security following transfer, and the issue of flooding attacks in MNC mode.

9.1. Authorization for Using Local Devices

It is necessary that the use of a local device be limited to authorized parties. As stated earlier, this document assumes both personal and public devices, and these have different authorization policies. A personal device only accepts transfer requests from a single identity, the device owner. Therefore, the most appropriate means of access control is to maintain a list of identities representing the device owner authorized to transfer sessions to the device. As mentioned before, the device is configured with an AOR representing its status as a transfer device, in addition to the user's AOR. Only requests made to the device AOR follow the access list, while incoming requests to the user's AOR are accepted from anyone (provided that a white or blacklist or other policy does not preclude their request from being accepted). The SIP-Identity header [25] is used to securely identify the initiator of a SIP request. That specification can be used in our use-cases when the local device must ensure that the INVITE or REFER request in MNC or SH mode, respectively, is indeed from the owner of the device.

Public devices accept transfer requests from a large number of identities. Access lists may be used for this purpose. Alternatively, since devices are often available to categories of users, such as "manager" or "faculty member", an appropriate solution may be to use trait-based authorization [23]. Using this mechanism, a user may acquire, from a trusted authorization service, an "assertion" of his user status and permissions. The assertion, or a reference to it, is included in the request to use the device.

9.2. Maintaining Media Security During Session Mobility

9.2.1. Establishing Secure RTP Using SDP

Confidentiality, message authentication, and replay protection are necessary in internet protocols, including those used for real-time multimedia communications. The Secure Real-time Transfer Protocol (SRTP) [14] provides these for RTP streams. Since SRTP may be used to carry the media sessions of SIP devices, such as the MN and CN, we

discuss how to ensure that the session continues to use SRTP following the transfer to another device. This is also discussed in less detail in [2].

The establishment of secure RTP communications through SDP is defined by two documents. The "crypto" attribute [15] is a media-level attribute whose value includes the desired cryptographic suite and key parameters used to perform symmetric encryption on the RTP packets. Since the key information is sent in the SDP body with no dedicated encryption or integrity protection, a separate protocol such as S/MIME must be used to protect the signaling messages. Another document [16] specifies the "key-mgmt" attribute used to provide parameters for a key management protocol, such as MIKEY. Using this attribute, the two participants exchange keys encrypted by a public or shared key, or negotiate a key using the Diffie-Hellman method.

The use of cryptographic parameters in SDP does not change the message flows described earlier in this document. For instance, in MNC mode shown in Figure 2, the response from the local device (2) will include, in addition to any supported media type, cryptographic information for each type. This cryptographic information will be a list of attribute lines describing the crypto suite and key parameters using either of the two attributes mentioned. These lines will be sent by the MN to the CN in the subsequent request (3). The CN will choose a cryptographic method and return its own key information in the response (4). Maintaining a secure media session in SH mode requires the local device to negotiate a cryptographic relationship in the session that it establishes following its receipt of the REFER request.

It is noted in [2] that establishing media security in third party call control depends on the cooperation of the controller. In this document, the Mobile Node (MN) in Mobile Node Control mode (MNC) has the role of controller in 3pcc, while in the Session Handoff (SH) mode, MN uses the REFER method instead. The following is an excerpt from that document:

> End-to-end media security is based on the exchange of keying material within SDP. The proper operation of these mechanisms with third party call control depends on the controller behaving properly. So long as it is not attempting to explicitly disable these mechanisms, the protocols will properly operate between the participants, resulting in a secure media session that even the controller cannot eavesdrop or modify. Since third party call control is based on a model of trust between the users and the controller, it is reasonable to assume it is operating in a well-behaved manner. However, there is no cryptographic means that can

> prevent the controller from interfering with the initial exchanges of keying materials. As a result, it is trivially possibl[e] for the controller to insert itself as an intermediary on the media exchange, if it should so desire.

We note here that given the model presented in this document, where the controller is operated by the same person that uses the local device, i.e., the MN user, there is even more reason to believe that the controller will be well-behaved and will not interfere with the initial transfer of key exchanges.

9.2.2. Securing Media Using the Transport Layer

The exchange of media could alternatively be secured at the transport layer, using either TLS or Datagram Transport Layer Security (DTLS) [24]. The one consideration for use of these protocols in session mobility would be assigning the client and server roles. In SH mode, it may be assumed that the local device, the referee, would act as the client, since it is initiating the signaling session with the CN. However, in MNC mode, these roles would be unclear. The same problem was mentioned above in establishing a secure connection for an MSRP session transferred in MNC mode. This problem could be solved through the use of Connection-Oriented Media (COMEDIA) [26], which specifies the "setup" SDP attribute to negotiate these roles.

We describe here briefly how this is done. In the MNC exchange shown in Figure 2, the local device chooses whether to specify a media session over a secured transport in its response to the MN. If so, it includes under the media line a "setup" attribute set to either "active", "passive", or "actpass". This is sent on to the CN. Assuming it agreed to such a session, it responds with a "setup" attribute, as per the COMEDIA specifications. This is then sent by the MN to the local device. If the local device and CN agreed on their roles, the appropriate session could be established, through which the media would be transmitted. Before they transmit media between them, the CN and local device exchange messages to establish the TLS or DTLS session. This same approach could be used to establish an SRTP security context over DTLS, as per [31].

9.3. Flooding Attacks in MNC Mode

The MNC call flows in this document, where one device instructs another device to send an RTP flow to a third one, present the possibility of a flooding attack. This is a general problem that relates to any use of 3pcc. In this document, it is only a concern where the device is public, as described at the beginning of this section, and a large group of people can transfer media to it, since there may not be a very strong trust relationship between the device

owner (e.g., an institution) and the users. Obviously, where a device is private and only its owner can transfer to it, the concern does not exist, given the use of the Identity header mentioned earlier. A possible solution may be the use of ICE [27], since both sides confirm that they want to receive each other's media.

10. Acknowledgments

We would like to acknowledge the helpful comments made about this document by the SIP community, in particular Jon Peterson, Joerg Ott, and Cullen Jennings.

11. References

11.1. Normative References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[2] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, April 2004.

[3] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[4] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[5] Sparks, R., "The Session Initiation Protocol (SIP) Referred-By Mechanism", RFC 3892, September 2004.

[6] Mahy, R., Biggs, B., and R. Dean, "The Session Initiation Protocol (SIP) "Replaces" Header", RFC 3891, September 2004.

[7] Campbell, B., Ed., Mahy, R., Ed., and C. Jennings, Ed., "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

[8] Jennings, C., Mahy, R., and A. Roach, "Relay Extensions for the Message Sessions Relay Protocol (MSRP)", RFC 4976, September 2007.

[9] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, June 2005.

[10] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

[11] Camarillo, G., "Internet Assigned Number Authority (IANA) Registration of the Message Media Feature Tag", RFC 4569, July 2006.

[12] Rosenberg, J., "Obtaining and Using Globally Routable User Agent URIs (GRUU) in the Session Initiation Protocol (SIP)", RFC 5627, October 2009.

11.2. Informative References

[13] Rosenberg, J., Schulzrinne, H., and R. Mahy, Ed., "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", RFC 4235, November 2005.

[14] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[15] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006.

[16] Arkko, J., Lindholm, F., Naslund, M., Norrman, K., and E. Carrara, "Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)", RFC 4567, July 2006.

[17] Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999.

[18] Camarillo, G., Burger, E., Schulzrinne, H., and A. van Wijk, "Transcoding Services Invocation in the Session Initiation Protocol (SIP) Using Third Party Call Control (3pcc)", RFC 4117, June 2005.

[19] Ott, J., Bormann, C., Sullivan, G., Wenger, S., and R. Even, Ed., "RTP Payload Format for ITU-T Rec. H.263 Video", RFC 4629, January 2007.

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

[21] Campbell, B., Ed., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[22] Lennox, J., Wu, X., and H. Schulzrinne, "Call Processing Language (CPL): A Language for User Control of Internet Telephony Services", RFC 3880, October 2004.

[23] Peterson, J., Polk, J., Sicker, D., and H. Tschofenig, "Trait-Based Authorization Requirements for the Session Initiation Protocol (SIP)", RFC 4484, August 2006.

[24] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[25] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[26] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005.

[27] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", Work in Progress, October 2007.

[28] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[29] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", Work in Progress, September 2008.

[30] Cheshire, S. and M. Krochmal, "Multicast DNS", Work in Progress, September 2008.

[31] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing an SRTP Security Context using DTLS", Work in Progress, March 2009.

Authors' Addresses

Ron Shacham
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA

EMail: shacham@cs.columbia.edu

Henning Schulzrinne
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA

EMail: hgs@cs.columbia.edu

Srisakul Thakolsri
DoCoMo Communications Laboratories Europe
Landsberger Str. 312
Munich 80687
Germany

EMail: thakolsri@docomolab-euro.com

Wolfgang Kellerer
DoCoMo Communications Laboratories Europe
Landsberger Str. 312
Munich 80687
Germany

EMail: kellerer@docomolab-euro.com