♦ Spanning a Multimedia Session Across Multiple Devices

Jasper Aartse Tuijn and Dennis Bijwaard

This paper describes an innovative solution that allows a multimedia session to be distributed over multiple devices that are in close proximity to a user. This would typically be useful when a user engaged in an audio/video conference enters a meeting room. Upon entering, all media session components are transferred from the personal digital assistant (PDA) to the projector, Hi-Fi set, and webcam present in the meeting room. When the user leaves the room, all media session components are transferred back to the PDA. Transferring parts of a multimedia session between different devices is defined as partial session mobility (PSM). Numerous methods exist for PSM. None of these methods is directly suitable for initiating PSM from a dedicated network node such as an IP Multimedia Subsystem (IMS) application server, but they do provide a starting point. Because of a number of advantages in the invite-based method, this method is taken as a basis for network-initiated PSM. This paper compares the existing methods, describes the proposed network-initiated method to support PSM, and investigates how to combine it with terminal-initiated PSM. It also describes how PSM could be supported in IMS. © 2008 Alcatel-Lucent.

Introduction

The use of mobile devices has grown dramatically in recent years, and they are now used daily by most people in developed countries. The size of mobile devices has shrunk, while the technical capabilities have grown rapidly. For example, today mobile devices support multiple network technologies as well as videoconferencing. The size of current powerful mobile devices also has a downside; the usability of all these technical capabilities can be limited by the small size.

Typical Use Case

A typical use case for spanning a session across multiple devices can be envisioned in a business environment where people use videoconferencing on their mobile devices to communicate with colleagues or customers. While having a video conference on a mobile device, entering a conference room could trigger a proposal for transferring the video to a big screen located in that room. This proposal could typically be shown on the mobile device of the user, and after the user agrees, the video stream could be automatically transferred to the big screen without any noticeable interruption. The mobile device could also offer the possibility to retrieve the video stream, for example, when leaving the conference room.

Motivation

Recent developments in the Session Initiation Protocol (SIP) [14] standard on session mobility [16]

Bell Labs Technical Journal 12(4), 179–194 (2008) © 2008 Alcatel-Lucent. Published by Wiley Periodicals, Inc.
Published online in Wiley InterScience (www.interscience.wiley.com) • DOI: 10.1002/bltj.20275

Panel 1. Abbreviations, Acronyms, and Terms

3GPP—3rd Generation Partnership Project
3PCC—Third party call control
AS—Application server
B2BUA—Back-to-back user agent
CN—Correspondent node
DAIDALOS—Designing advanced network interfaces for the delivery and administration of location independent, optimized personal services
Hi-Fi—High fidelity
IMS—IP Multimedia Subsystem
IP—Internet Protocol
LN—Local node
MDS—Multi-device system
MDSM—Multi-device system manager
MN—Mobile node
PDA—Personal digital assistant
PSM—Partial session mobility
SDP—Session Description Protocol
SIP—Session Initiation Protocol
SIP-UA—SIP user agent
SSC—Sub-session controller
URI—Uniform resource identifier
VCC—Voice call continuity

aim to use auxiliary devices discovered in the vicinity of the user to enhance ongoing multimedia sessions. These developments focus specifically on terminal-initiated transfer of multimedia stream endpoints to other devices. The main goal of the work described in this paper is to initiate such a transfer from a dedicated node in the network, in order to be able to support these transfers as a network service without necessarily having to change all client devices. Network-initiation can also be preferable because terminal-initiated discovery of candidate devices and transfers may prove quite complex in terms of processing and (mobile) network usage. Additionally, guest use of devices and associated privacy and charging could be easier for an operator since it already has privacy and charging in place for other purposes. Transferring media stream endpoints across devices in a multimedia session is defined here as partial session mobility. Network-initiated partial session mobility would typically be triggered by external events like network-side discovery of nearby multimedia devices, such as a big screen. To offer network-initiated transfers of session parts as a service, the dedicated network node would typically be an application server (AS) when applied in the IP Multimedia Subsystem (IMS), standardized by 3rd Generation Partnership Project (3GPP*) [1] and 3GPP2 [3]. Since IMS is aware of all available multimedia devices and could be or become aware of their capabilities and location as explained later in this paper, an application server is a good candidate to initiate partial session mobility from the network. Device discovery from the mobile device is expected to consume more bandwidth, time, processing and associated costs, and battery power consumption.

Objectives

Since users may not always want to enhance their multimedia sessions with additional devices, for example, because they will not stay around the devices for a long period of time, the user needs to be involved in the decision to change session parameters. This involvement is guaranteed by first proposing the session change to the mobile device. The following information is assumed to be sufficient in such a proposal:

- The media stream endpoint to be transferred,
- The device to transfer the stream endpoint to, and
- The media parameters to be used for the media stream.

After a media stream endpoint has been transferred to another device, it is useful to enable its transfer back to the originating mobile terminal, e.g., when the user starts moving. This transfer may be initiated by either the dedicated network node or the user. Since the user is more likely to know when to stop using auxiliary devices in a session, it should also be possible to initiate these transfers from the mobile device. Alternatively, when a device that has been added disconnects, the associated media stream endpoints are expected to be restored to the mobile device.

To ease deployment of the functionality described above, it is important that no upgrades of legacy SIP clients are required for basic functionality.

The objectives described above can be summarized as follows:

- Transfer of individual media stream endpoints to other devices.
  - Support for both network-initiated and terminal- initiated transfer.
  - User involvement in a network-initiated transfer decision.
  - Transferred media streams can be transferred back to the mobile device.
  - The initiator of the transfer should be able to influence the media parameters of the transferred media stream.
- Compatibility with current SIP user agents (SIP-UAs).
- Robustness when involved nodes suddenly disconnect.

Structure of This Paper

This paper first introduces related technologies and other background. Then the concepts behind partial session mobility, and especially partial session mobility initiated from the network, are explained. After analysis of existing technologies, this paper then introduces a new complementary method that supports network-initiated partial session mobility that can be combined with terminal-initiated partial session mobility.

Background and Related Work

A number of related methods for transferring part of a session to another device and a number of approaches for discovery of potential devices are briefly described below.

Related Methods

As described by Shacham and his co-authors [16], different techniques exist to enable session mobility in SIP. The mobile node control mode uses third party call control [13] to let the mobile node (MN)—that is, the user terminal—coordinate the transfer of media streams to another device. With this technique, the mobile node stays in the signaling path, while the media streams exist directly between both endpoints. The session handoff mode uses SIP REFER [17] to transfer the complete session to the other device. With this technique, the mobile node does not stay in the signaling path of the session and therefore cannot retrieve the session directly using re-INVITEs.

Shacham [16] also describes how both the mobile node control mode and the session handoff mode can be used to transfer part of the media streams to another device. The mobile node control mode uses Session Description Protocol (SDP) [12] capabilities to define different endpoints for each media stream. The session handoff mode uses a multi-device system (MDS), where the multi-device system manager (MDSM) acts as a gateway for both SIP signaling and media, distributing the media streams to different devices.

Mani and Crespi [9] introduce a network-side mobility server that acts as the MDSM in the session handoff mode. They assert that the mobile node control mode is not acceptable in IMS because a transferred session will be controlled by a terminal which is no longer involved in the session. This might be true for full session mobility; however, the mobile node could still be involved in the session (e.g., as mediator in the session to change or aggregate the session parts later, and as one of the possible media endpoints), and it could be charged for the whole or part of the session.

Within the DAIDALOS project [8, 15] another method has been proposed to support the transfer of certain media stream(s) to another device. The DAIDALOS method also uses the REFER method [17], extended with multiple refer [7]. With this method, the mobile node sends a REFER message to the device of the other user in the session, the correspondent node (CN). This message contains references to the different devices to be invited for a specific part of the original session. This technique does not yet specify how the mobile node can indicate which part of the original session must be transferred to what device.

A "mobility" SIP header is proposed by Peng in [10]. This header contains the call-ID of the session between the mobile node and correspondent node, and information about the specific part of the session that must be transferred. This method is also based on REFER messages, but in contrast to the methods

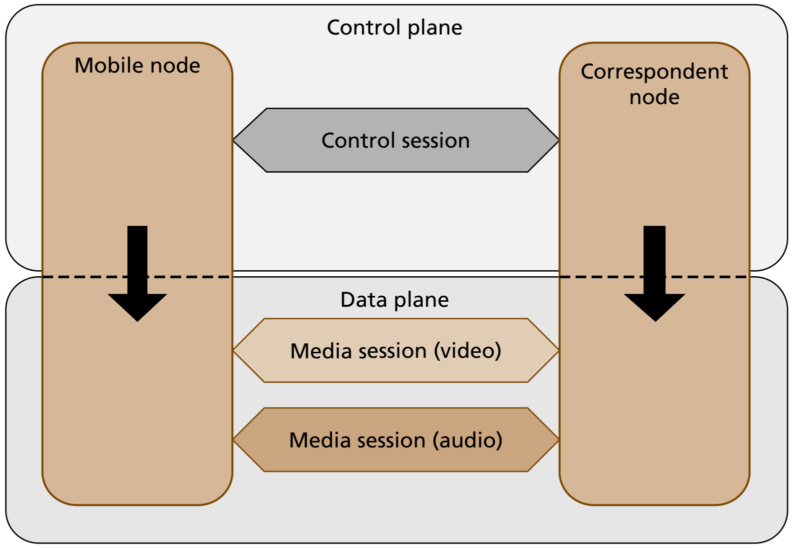

Figure 1.
Concept of session.

previously described, it uses a session-structure to correlate the different sessions that the correspondent node sets up with the additional devices to the session between the correspondent node and mobile node.

On the network side, other mechanisms exist for session transfer between a terminal's IP and circuit voice interface and are specified in the 3GPP voice call continuity (VCC) standards [2]. These mechanisms currently do not target usage of multiple devices at each side of the session.

Finding Nearby Devices

Multiple solutions can be envisaged to discover devices in the neighborhood [5] to enhance a session. This discovery can be done from the mobile device, from the candidate devices, or from within the network. From a mobile device or candidate devices, this would typically require short-range communication/detection and/or presence information via the mobile network. Within the network, presence information enriched with physical location data could be used to find capable devices in the neighborhood of the mobile device. With the location and multimedia capability of fixed multimedia devices available, the actual location of the mobile device would be sufficient to find candidate multimedia devices in the neighborhood. Context information is more general than presence and location information; Brok, Vemuri, Meeuwissen, and Batteram [6] describe a framework to utilize context information from various sources in an IMS application server.

Concepts

This section explains the basic concepts behind the ability to transfer a media stream endpoint to another device and the ability to initiate these transfers from a dedicated network node.

A basic SIP session entails a control session on the control plane that defines and controls media session endpoints on the data plane. **Figure 1** shows how the control session on the control plane and the media session endpoints at the data plane are related to each other for a typical audio/video session between a mobile node and a correspondent node.

In SIP, a media stream endpoint (data plane) can only change if the change is initiated from the control plane. **Figure 2** shows control sessions mediated by the mobile node that specify video streams between the mobile node and correspondent node, and an audio stream between the audio node and correspondent node. This situation would typically occur

 DOI: 10.1002/bltj

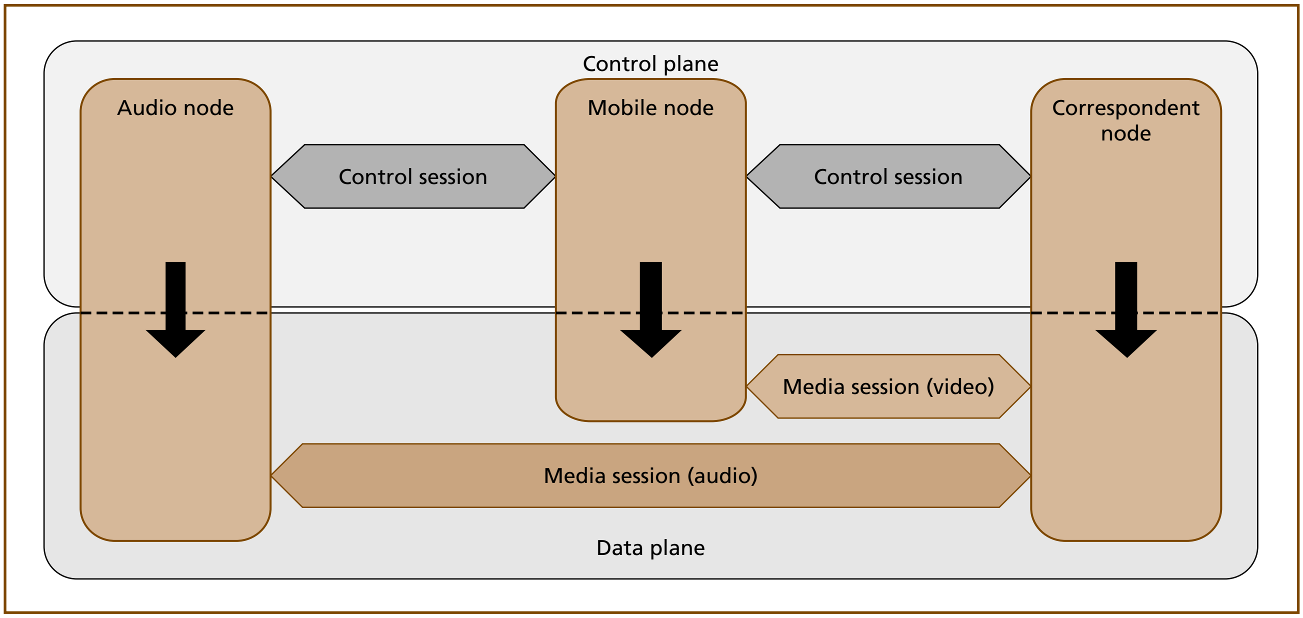

Figure 2.
Concept of partial session mobility.

in the mobile node control mode method, where the mobile node from Figure 2 sets up an additional control session with a newly involved audio node and instructs the audio node and the correspondent node to act as endpoints in the session. In other methods, the correspondent node or additional nodes could act as mediating node. The control session between the mediating node and the additional node is from now on called sub-session.

Besides mediation from the media endpoints, it is also possible to start and manage the control sessions via a dedicated network node that operates on the control plane. The network node here is called the sub-session controller (SSC), and it would typically act as a SIP back-to-back user agent (B2BUA). **Figure 3** shows how the sub-session controller would be positioned within the control sessions. Before setting up the sub-session between mobile node and audio node, the sub-session controller would first need to propose the transfer of the audio session endpoint from the mobile node to the audio node.

Evaluating Current Methods

As described in the background section, different methods currently exist to initiate transferring session endpoints from a mobile device to another device, called a local node (LN). This section describes the evaluation of these methods to find the most suitable method that could form the basis of a method that supports partial session transfers initiated from the user-terminal combined with network-initiated partial session mobility.

The methods are evaluated against the objectives stated in the Introduction. Additionally, separation of concerns identifies a general guideline to keep solutions simple by making sure certain logic is located on the most appropriate place in the architecture.

Mobile Node Control Mode

Using this method, the mobile node stays in the signaling path after the media stream(s) have been transferred to another device. This has a number of consequences:

- The MN can easily retrieve a media stream by sending a re-INVITE message to the correspondent node.
- The MN has direct influence on the body of the media parameters of the transferred media stream(s) because the MN actually sends the INVITE message(s) to the local node(s).
- If a local node used in the session suddenly disconnects, the MN can change the session

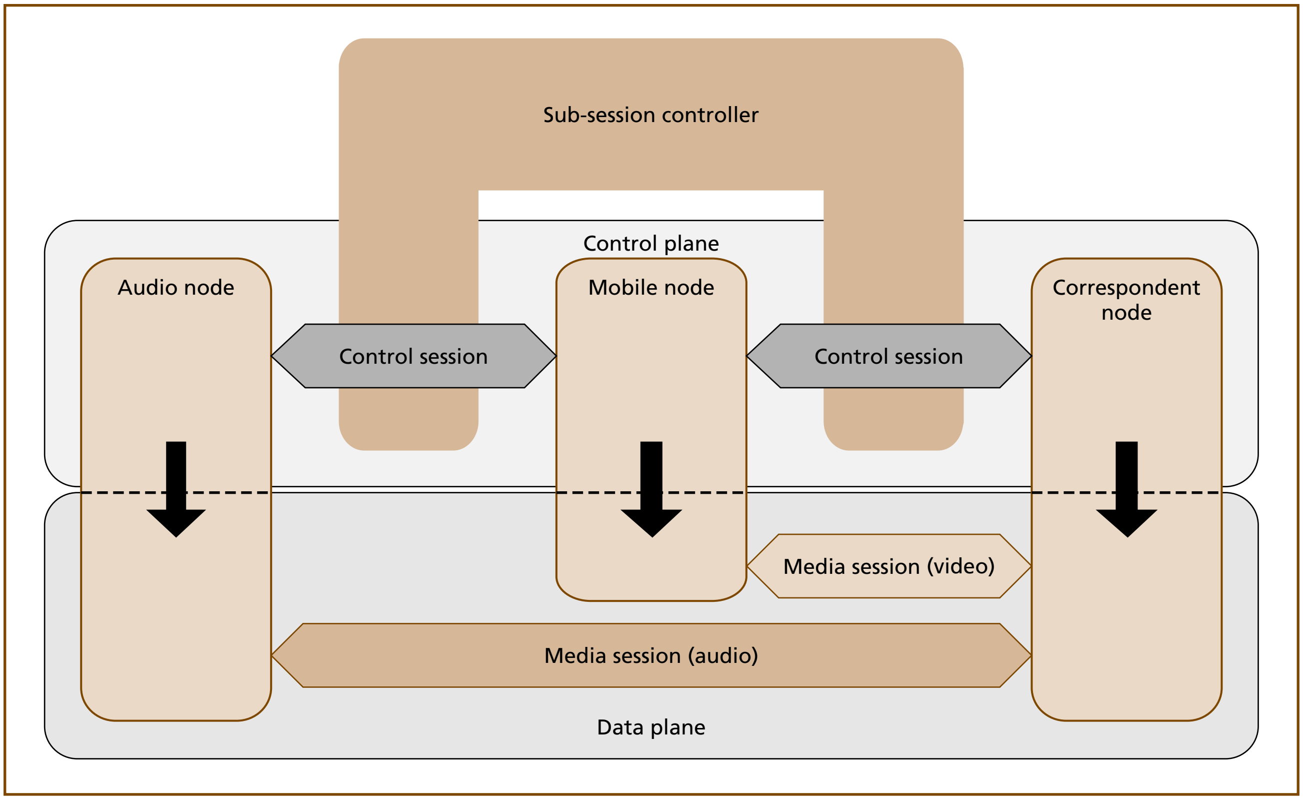

Figure 3.
Concept of network-initiated partial session mobility.

accordingly by issuing a re-INVITE message to the CN.

- If the MN suddenly disconnects, the media streams between CN and LNs remain in place while the signaling path is broken. This means these sessions cannot be closed properly.

Because this method uses third party call control [13], it is also possible to let the sub-session controller manage the sessions and sub-sessions of the MN. In this case, the SSC must be located in the signaling path of these sessions. Also, because the mobile node control mode is based on existing functionality of SIP, no additional extensions are necessary.

Session Handoff Mode

Using this method, the MN does not stay in the signaling path, but the whole session is transferred to the multi-device system manager. This has the following consequences:

- The MN cannot simply send a re-INVITE message to the CN to retrieve the session; it can only "ask" the MDSM, using a nested REFER message [18], if it wants to transfer the streams back.
- The MN does not control the media parameters of the media stream that is transferred to the LN because it cannot indicate this in the REFER message that is being sent to the MDSM.
- In case the MN suddenly disconnects after session transfer to the MDSM, the media streams that are not located on the MN will continue to exist; however, the user cannot stop the session. If the MDSM suddenly disconnects, the user is also not able to retrieve the streams, because a nested REFER message has to be sent to the MDSM to do so.
- When the mobile node suddenly disconnects after a media stream has been transferred, the signaling path would not be broken, meaning the CN or MDSM could close the session properly. On the other hand, if the MDSM suddenly disconnects, even if the MN is still connected, the signaling path would be broken.

A dedicated network node could send a REFER message to the MSDM, to let it replace the session between the CN and MN. With this mechanism, the dedicated node that initiates the transfer should be authorized to let the MSDM replace the session.

Multiple Refer

In contrast with the session handoff mode, in the multiple refer method, the CN is the node combining related sessions instead of the MDSM. The MN uses a REFER message to let the CN transfer a media stream to another node. Using REFER messages, the MN cannot prescribe the content of the INVITE message, meaning the MN has no control over the media parameters.

Because the MN is not in the signaling path of the created sub-sessions, it does not have all the information necessary to decide when it must transfer which stream to what device. However, it could initiate a partial session transfer to retrieve a media stream back to the MN, for which it would use a REFER message with a replace header. This way the MN is also able to retrieve a media stream.

As with the session handoff mode, another node besides the MN could send the REFER message to the CN; however, this node would need the call-ID of the session between the CN and MN. If this node does have access to this information, network-initiated partial session transfer is possible.

With this method, the correspondent node contains the intelligence to relate a session with the sub-sessions needed for a service provided by the media node. However, ideally, the CN should not be controlling a partial session transfer requested by the MN, due to separation of concerns. Because the CN sets up the sub-session with the LNs, it would be hard to relate both volume- and signaling-based [11] sub-session charging to the MN.

In addition, when the MN suddenly disconnects after a media stream has been transferred, the part that has been transferred can be closed properly because the MN is not located in the signaling path of the session between the CN and LNs.

Mobility Header

As with the multiple refer method, in the mobility header method, the CN is the node that combines the different sessions together. However, with this method the MN sends a REFER message to the LN if it wants the LN to take over a media stream.

Peng [10] does not explicitly define how retrieval works; however, a media stream that has already been transferred can be retrieved because it has an ongoing session with the CN. This session can be "re-activated" with a re-INVITE message.

Because all involved nodes should support the mobility header, support for partial session mobility for the MN depends on the capabilities of the CN and involved LNs. Just as with the multiple refer method, the MN does not have control of the actual media parameters of the transferred media stream.

With respect to separation of concerns, for this method, as well as the multiple refer method, ideally the CN should not be saddled with controlling the transfer of a media stream to another device. The REFER message sent to the LN by the MN to transfer a media stream could also be sent by a SSC. When it is necessary to close the session between the CN and MN, the SSC should be able to act as a B2BUA in the session.

Because the CN is the central node between the sessions, it is the weakest link with respect to robustness. At the moment the CN suddenly disconnects, the MN has no direct influence on the session between the CN and the local devices, although the MN might have to pay for those sessions. As with the multiple refer method, it also would be hard to relate charging of the sub-sessions to the MN.

Recommendation

The previous four sections described the advantages and disadvantages of the methods currently available to execute partial session mobility using SIP, based on the objectives mentioned in the Introduction. The following short list shows the evaluation criteria:

1. Potential to support network-initiated partial session transfers,
2. Transfer of a media stream back to the user device,
3. Control of the media parameters by the initiator,
4. Robustness,

Table I. Evaluation results.

	1	2	3	4	5	6
Mobile node control	+	+	+	−	+	+
Session handoff	±	±	−	±	+	+
Multiple refer	±	+	−	±	±	−
Mobility header	+	+	−	±	−	−

\+ Yes, fully fulfilled
± Partially fulfilled/fulfilled with issues
− No, not fulfilled

5. Compatibility with SIP-UAs that do not explicitly support the method, and
6. Separation of concerns.

Table I contains the evaluation results of the different methods compared with each other. When the evaluation criteria all have equal priority, this leads to the conclusion that the "mobile node control" mode is the most promising method to use.

The session handoff mode, multiple refer method, and mobility header method all use REFER messages to execute a partial session transfer, while the mobile node control mode uses re-INVITE messages. The REFER based messages have the disadvantage of not having the ability for the initiator of the partial session transfer to prescribe the exact SDP body that should be offered to the LNs. Another problem that occurs in almost all methods that use REFER messages is that a partial session transfer cannot be undone after a node transfers a media stream, because the MN is no longer directly involved in the session that contains the transferred media stream. However, with the mobility header method, the original media stream can be set up again, though full retrieval of the stream is not yet supported.

Both the multiple refer and mobility header methods involve the CN in a partial session transfer that is being executed for the benefit of the MN. With respect to separation of concerns, this is far from ideal. In the mobile node control mode, the CN is not involved in the partial session transfer on the control layer. Besides this, the mobile node control mode and the session handoff mode both do not need an extension of SIP.

Proposed Method for Network-Initiated Partial Session Mobility

The sub-session controller in the network must be able to start sessions on behalf of nodes and interpret sessions between nodes. Therefore, the SSC must implement B2BUA functionality [14]. A B2BUA can participate in multiple sessions and connect them by acting on behalf of the end nodes.

Integration with IMS

To ensure that the SSC can offer its services to a specific MN, it must be located in the signaling path of all sessions the MN has with other nodes. In IMS, an application server can be positioned in the signaling path of sessions and would therefore be the logical place to position the SSC.

The SSC itself is only able to propose and execute partial session transfers; reasoning logic is proposed in the network to determine when to transfer which stream to what device. For this reasoning logic, which is not the focus of this paper, geographical location and presence information (context information) of the mobile node and candidate local nodes are expected to be necessary. **Figure 4** shows how the SSC and context-reasoning node could be integrated in IMS, re-using a model described in [6]. Basically, the context reasoning node in the SSC can use a subscribe mechanism to be notified about the location changes of the client device (from its presence user agent, or a network-side user-location mechanism through a presence external agent) and match it with the known capabilities and location of the fixed devices, and optionally, also other mobile

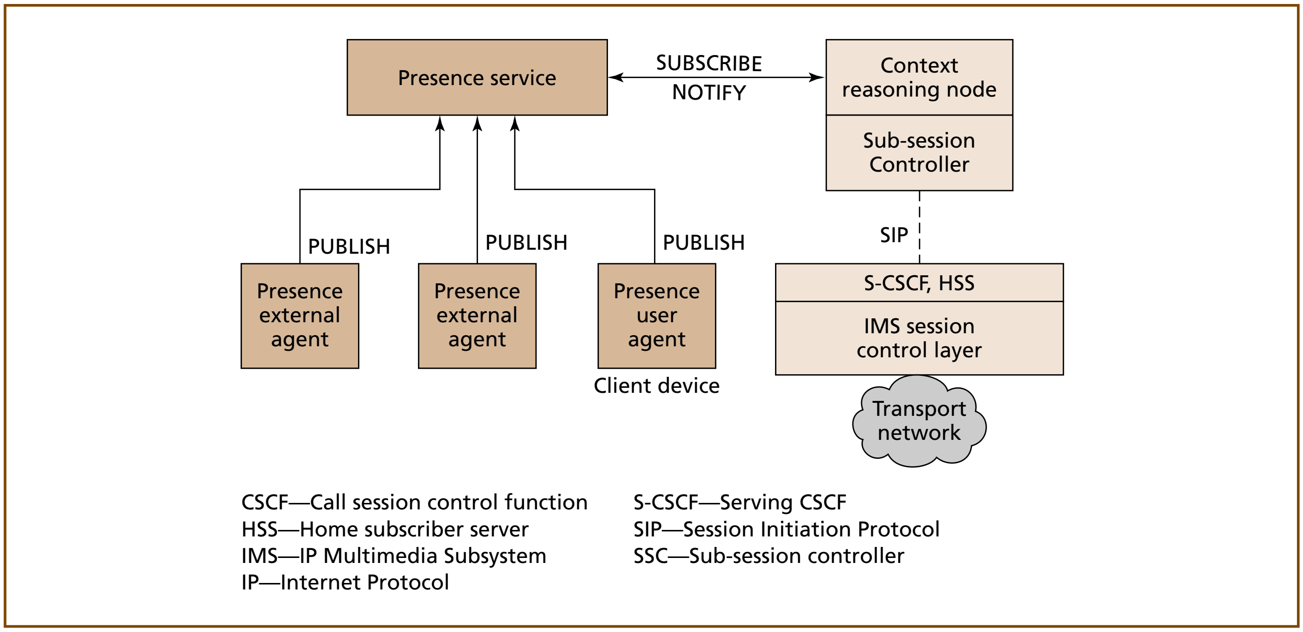

Figure 4.
SSC integration in context framework for IMS.

devices for which location and capabilities can be tracked.

Method Overview

Proposing and initiating a partial session transfer involves a number of steps. **Figure 5** shows the message sequence diagram for this method. The method assumes an ongoing session between the CN and MN (message 1). In message 2 through message 4, the SSC proposes a partial session transfer to the MN, which is accepted by the MN. In message 5 through message 8, the SSC uses third party call control to invite the LN to send audio to the CN. In message 9 through message 12, the MN is invited to stop sending audio to the CN. In message 13 through message 17, the SSC invites the CN to start sending audio to the LN instead of to the MN, without changing the video stream. The sections following provide further details.

Proposing a Partial Session Transfer

As described previously, the SSC should propose a partial session transfer to the MN. The solution illustrated in Figure 5 uses an INVITE message (message 2) sent by the SSC to the MN on behalf of the CN. This INVITE message contains the method "pst-control" in the required header.

This required header implies that the receiver of the message is only allowed to process the message if it supports that extension; if not, it should respond with a 420 (bad extension) response. When the SSC receives a 420 response, it knows the MN does not support the extension, and the SSC could apply user preferences in order to decide what to do next. **Figure 6** illustrates the sequence of events when user preferences prescribe to continue with a partial session transfer after a 420 response, as indicated in message 3. As can be seen, the SSC continues the partial session transfer (message 5 through message 17) as in Figure 5 even after the MN replies with a 420 response.

As illustrated in Figure 5, at message 3, the MN responds with a 200 (OK) response to indicate accepted partial session transfer by the MN. When the MN does not accept the transfer, it delivers a 603 (decline) response, as illustrated in **Figure 7**.

Information About the Transfer in the Proposal

The proposal issued in INVITE (message 2) contains information in an SDP body, to support the user decision whether or not to proceed with the proposed transfer (as described in the Introduction). This SDP body, however, should not be interpreted as usual in

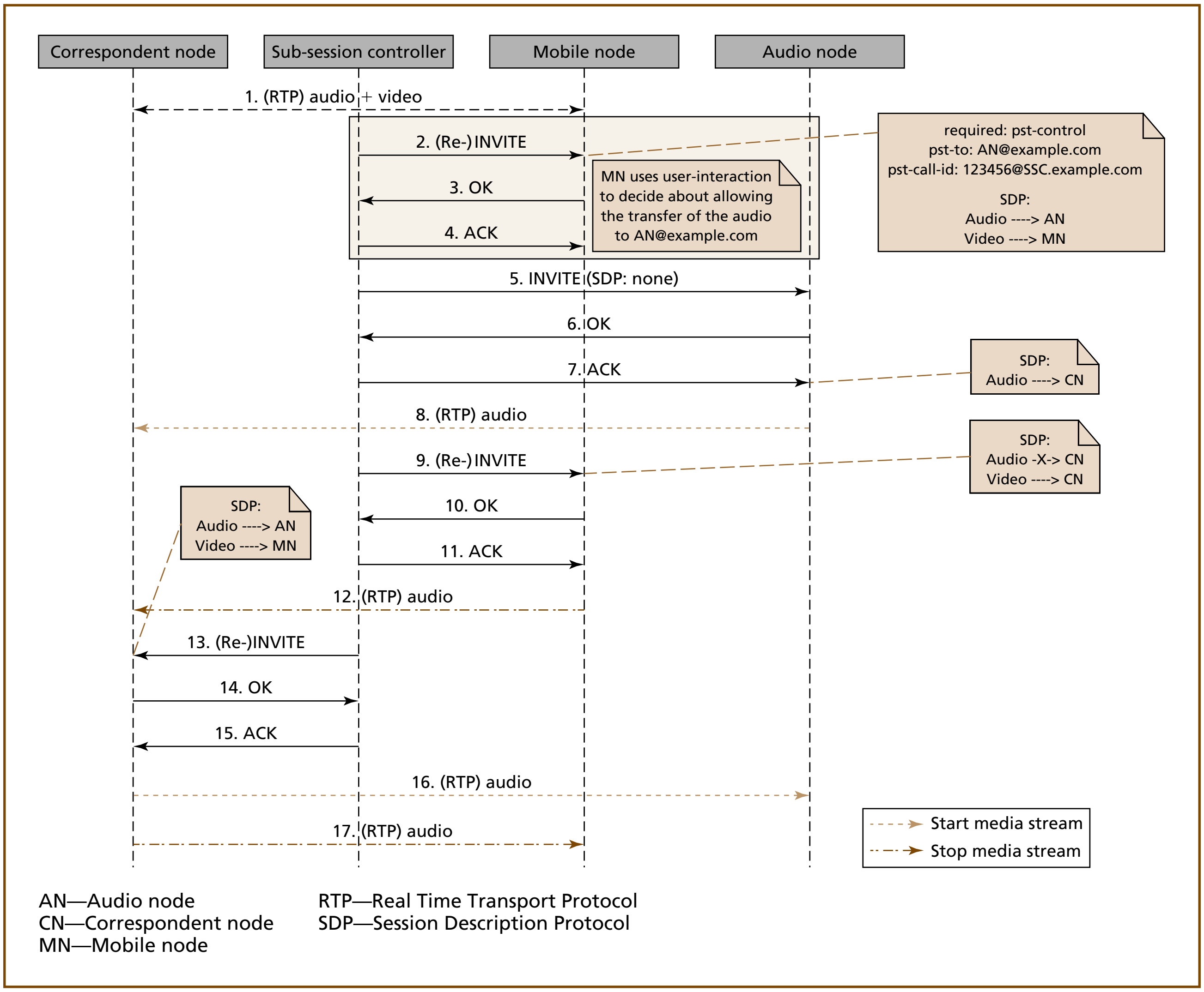

Figure 5.
Overview of the proposed method.

a normal negotiation [12]; it has another meaning as will be described below.

In SIP [14], a media stream is defined by the combination of the SDPs of each endpoint of the session, and each endpoint has its own "view" of the media session defined in the corresponding SDP. The SDP body in INVITE (message 2) contains the proposed session description to be sent to the CN. The MN can compare each media description in this SDP body with the corresponding media description in the SDP body that was last sent to the CN for setting up or changing the session.

Since the IP-address and/or port number changes when doing partial session transfer, the media description in the proposal changes accordingly. This enables the MN to recognize the stream to be transferred, and possibly the changed media parameters, based on the changed media description. Because the SDP does not include a SIP uniform resource identifier (URI) to identify the device the stream is being transferred to, the new header "pst-to" is added to the proposing INVITE message (message 2) which contains it. Figure 5 also shows this header in the comment at message 2.

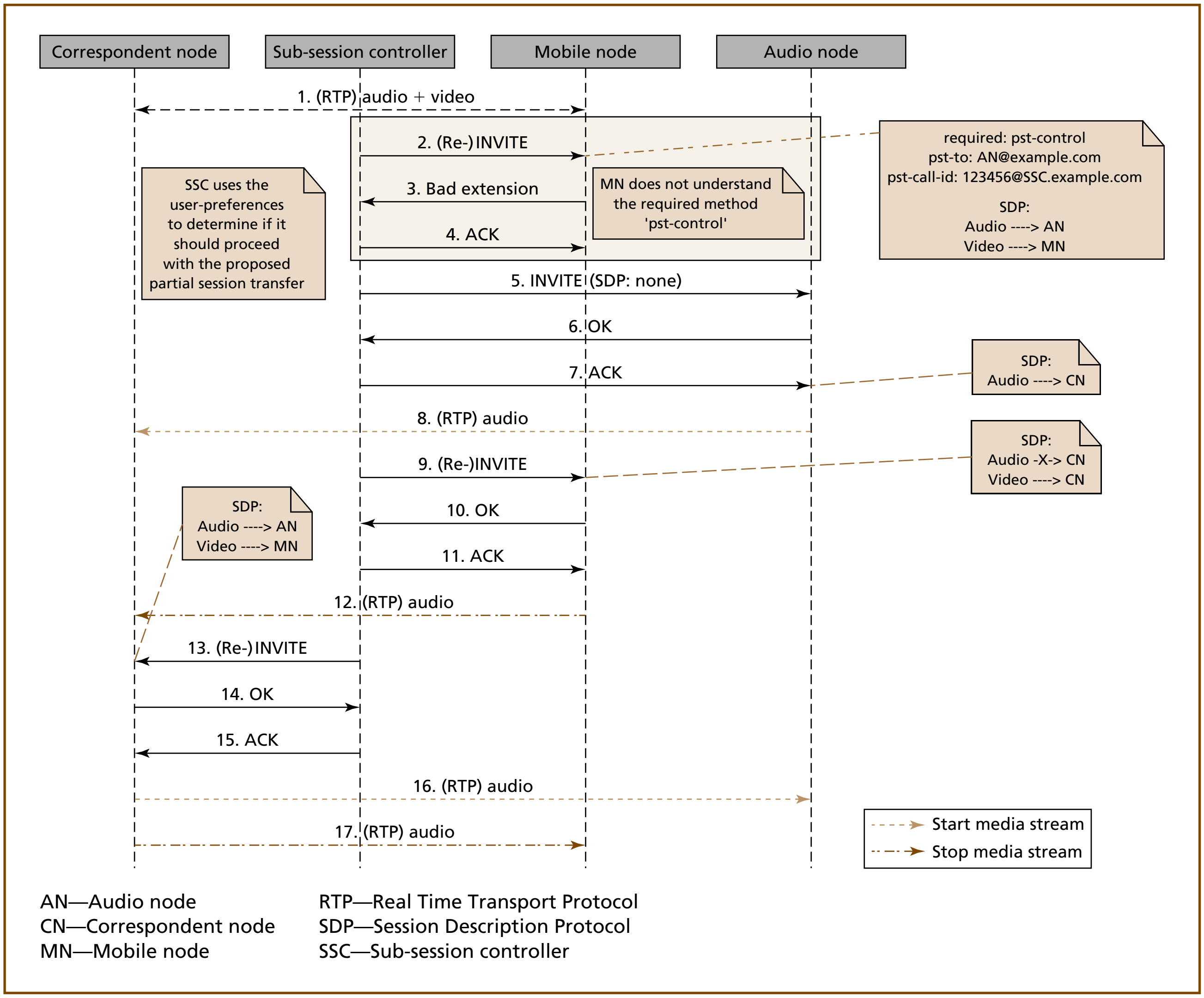

Figure 6.
The MN does not support the extension.

Retrieve or Move an Already Transferred Media Stream

After the SSC or MN has executed a partial session transfer, it must be possible to transfer it again, back to the MN or to another LN. A network-initiated partial session transfer is only possible after a terminal-initiated transfer when the SSC correctly interpreted the signaling between the MN and other nodes. A terminal-initiated partial session transfer after a network-initiated partial session transfer is more complicated, because the MN does not yet have an explicit session with the specified LN. During the network-initiated partial session transfer, the SSC did set up a session with the LN on behalf of the MN, but the MN was not involved in the session setup and does therefore not know the call-ID of this session.

Because of this, the new pst-call-ID header is contained in the proposal INVITE message, with the call-ID. This new header is also illustrated in Figure 5 in the comment at message 2. With this call-ID, the MN can close or change the session with the LN when it wants to. The SSC contains the logic to make sure SIP messages sent with this call-ID are delivered at the LN. **Figure 8** illustrates the situation where the MN retrieves the media stream (message 3 through message 7) and closes the sub-session with the LN (message 8 through message 10).

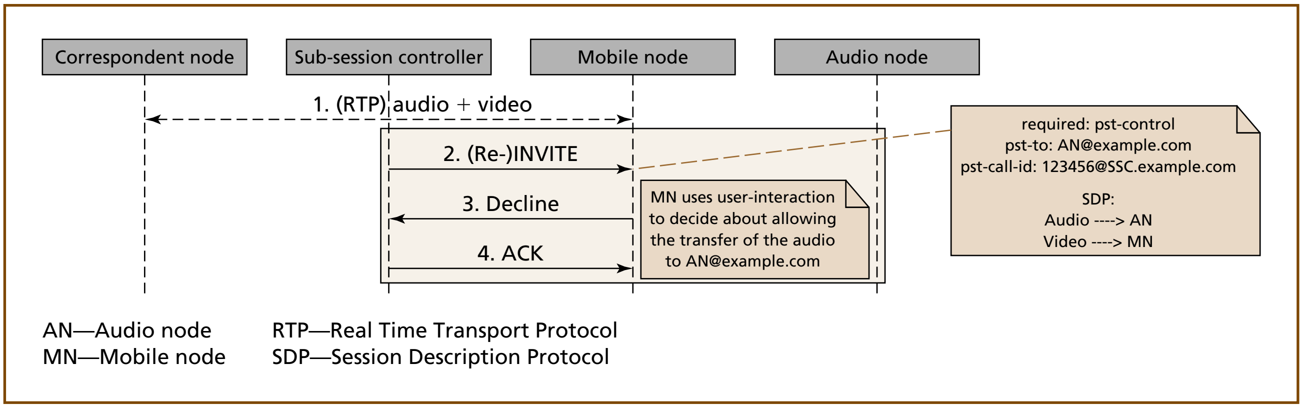

Figure 7.
User declines the proposed partial session transfer.

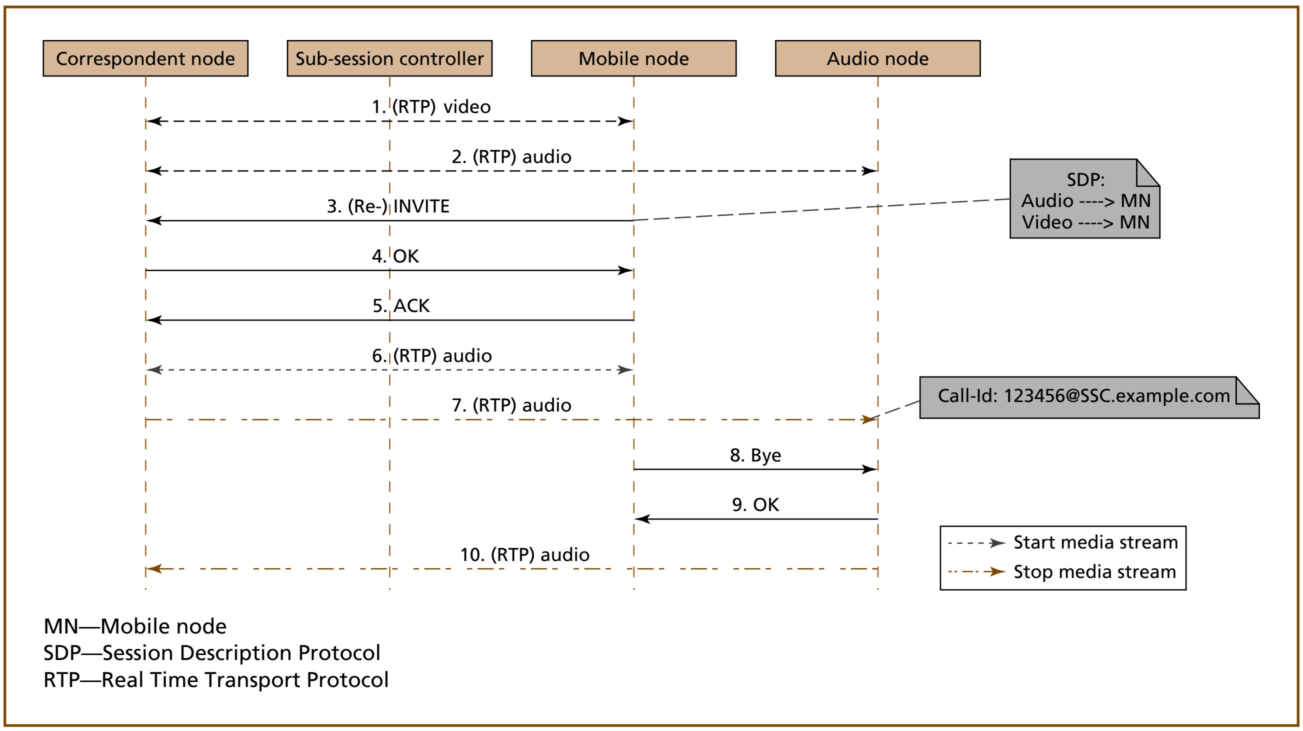

Figure 8.
Transfer back a media-stream.

Managing Sub-Sessions

In both network-initiated and terminal-initiated partial session mobility, sub-sessions are set up between the mobile node and the local nodes. This section describes which component is responsible for dealing with these sub-sessions and in what situation. The basic principle for managing sub-sessions is that the node (MN or SSC) that last changed a sub-session is responsible for the sub-session. For example, when the MN sends a re-INVITE to the CN after a network-initiated partial session transfer, the SSC would be responsible for making corresponding changes to the sub-sessions with the LNs when the MN did not change the sub-sessions. Since the MN can only change sub-sessions when it initiates them or when it supports the pst-control extension, the SSC can easily deduce when it is responsible for sub-sessions with the following rules:

- When MN supports pst-control
  - The node that made the last change to an individual sub-session is responsible for that sub-session. So, after a network-initiated partial session transfer, the SSC is responsible for the involved sub-session until the MN changes that sub-session.
  - This also means that if the MN is responsible for an individual sub-session, and if it does change the session with the CN, it should also make sure the sub-session with the LN is still consistent with the session between the MN and CN.
  - When the MN closes or changes a sub-session, it is responsible for changing the session with the CN accordingly.
- When the MN does not support pst-control:
  - When an MN supports terminal-initiated partial session mobility, the node that last changed a sub-session is responsible for that sub-session. This means that the MN is no longer responsible for an initiated sub-session, once the SSC modifies it.
  - When an MN does not support terminal-initiated partial session mobility, the SSC needs to make sure that the sub-session(s) remain consistent when the MN changes something in the session with the CN.

Please note that the combination of partial session mobility with full session mobility is not yet considered. Thus, one known issue with the rules above is a session transfer from an MN that does support the pst-extension to one that does not.

As described earlier, the "mobile node control" mode does have a problem related to robustness at the moment the MN suddenly disconnects while one of the media streams has already been transferred to another device. In this situation, the media stream involved continues to exist, and the user is not able to use his mobile terminal to stop the media stream. In this case, the user should use the LN interface to stop the stream. With the solution described here, the SSC is always involved in all sub-sessions related to the MN. Therefore, the SSC could easily stop all related sub-session activity when it is notified of a disconnection of the MN.

Validation

The sub-session controller was prototyped in order to validate the following aspects of network-initiated partial session mobility [4]:

- Proposing a partial session transfer to the user terminal.
- Executing the network initiated partial session transfer.
- Re-transfer of an already transferred media stream (also back to the mobile node).
- Controlling the media parameters of the transferred media stream.
- Compatibility with current SIP-UAs that do not support the extensions introduced in this paper.
- Minimizing the disruption of the media stream (theoretical approach).

After partial session transfers were proposed and accepted by the mobile node, the prototyped sub-session controller could successfully transfer media stream endpoints multiple times between a number of local nodes including the mobile device. Measurements and analysis indicated that a full-fledged SSC would not influence the continuity of the multimedia streaming because it would apply "make before break." This streaming continuity would be mainly affected by a) the time it takes a particular SIP UA to process SIP messages and initiate streaming and b) network-level delays of both signaling and streams.

Conclusions and Future Research

This paper describes a network node that can initiate partial session mobility in order to span a multimedia session across multiple devices. Major challenges involved finding a method that keeps the user informed and in control, that works together with terminal-initiated partial session mobility, and that has (limited) support for legacy SIP user agents. To overcome these challenges, the most appropriate terminal-initiated method was modified to be applied from within the signaling session between session peers while still allowing terminal initiation and extended to allow user awareness and control. SIP UAs that do not support the extension could use a user profile setting to indicate when to pursue a transfer.

The validation of the prototyped sub-session controller shows that the network-initiated partial session mobility method described in this paper works in practice and indicated continuity of multimedia streaming in real deployment of the method.

For future research, validation of the combination of terminal- and network-initiated partial session mobility is needed, for which a SIP UA needs to be extended with terminal-initiated partial session mobility and the proposed SIP extension. Other interesting research topics include scalability, combination with user mobility, merging voice call continuity with the proposed method, adding broadcast and multicast streams to an existing SIP session, and supporting group sessions for multicast group setup and switching between unicast, multicast, and/or a conference server.

Acknowledgments

The work described in this paper is based on results of IST FP6 Integrated Project DAIDALOS II, which receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

*Trademark

3GPP is a trademark of the European Telecommunications Standards Institute.

References

[1] 3rd Generation Partnership Project, "IP Multimedia (IM) Session Handling, IM Call Model, Stage 2 (Release 6)," 3GPP TS 23.218, v6.3.0, Mar. 2005, <http://www.3GPP.org/ftp/Specs/html-info/23218.htm>.

[2] 3rd Generation Partnership Project, "Combining Circuit Switched (CS) and IP Multimedia Subsystem (IMS) Services, Stage 2 (Release7)," 3GPP TS 23.279, v7.5.0, Dec. 2006, <http://www.3gpp.org/ftp/Specs/html-info/23279.htm>.

[3] 3rd Generation Partnership Project 2, "IP Network Architecture Model for cdma2000 Spread Spectrum Systems," 3GPP2 S.R0037-0, v 3.0, Aug. 21, 2003, <http://www.3gpp2.org/Public_html/specs/S.R0037-0_v3.0_111303.pdf>.

[4] J. Aartse Tuijn, Partial Session Mobility in Context Aware IP-Based Multimedia Subsystems, M.Sc. Thesis, University of Twente, Mar. 2007.

[5] S. I. Ahamed, M. Zulkernine, and S. Anamanamuri, "A Dependable Device Discovery Approach for Pervasive Computing Middleware," Proc. 1st Internat. Conf. on Availability, Reliability and Security (ARES '06) (Vienna, Aus., 2006), pp. 66–73.

[6] J. Brok, B. Kumar, E. Meeuwissen, and H. J. Batteram, "Enabling New Services by Exploiting Presence and Context Information in IMS," Bell Labs Tech. J., 10:4 (2006), 83–100.

[7] G. Camarillo, A. Niemi, M. Isomaki, M. Garcia-Martin, and H. Khartabil, "Referring to Multiple Resources in the Session Initiation Protocol (SIP)," IETF Internet Draft, June 22, 2006, <http://tools.ietf.org/wg/sipping/draft-ietf-sipping-multiple-refer/draft-ietf-sipping-multiple-refer-06.txt>.

[8] DAIDALOS, Designing Advanced Network Interfaces for the Delivery and Administration of Location Independent, Optimized Personal Services, <http://www.ist-daidalos.org>.

[9] M. Mani and N. Crespi, "Session Mobility Between Heterogeneous Accesses with the Existence of IMS as the Service Control Overlay," Proc. 10th IEEE Singapore Internat. Conf. on Commun. Syst. (ICCS '06) (Singapore, 2006), pp. 1–6.

[10] C.-J. Peng, SSIP: Split a SIP Session Over Multiple Devices, M.Sc. Thesis, National Chung Cheng University, June 2005.

[11] D. J. Plas and W. Romijn, "Online Charging in a Converged Network Environment: An Overview and Results From an Initial Implementation," Proc. 10th Internat. Conf. on Convergence in Services, Media and Networks (ICIN '06) (Bordeaux, Fr., 2006).

[12] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)," IETF RFC 3725, Apr. 2004, <http://www.apps.ietf.org/rfc/rfc3725.txt>.

[13] J. Rosenberg and H. Schulzrinne, "An Offer/Answer Model With the Session Description

Protocol (SDP)," IETF RFC 3264, June 2002, <http://tools.ietf.org/html/ rfc3264>.
[14] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002, <http://www.apps.ietf.org/rfc/rfc3261.txt>.
[15] A. Sarma, R. Aguiar, D. Bijwaard, and B. Farschian, "Pervasive Services for Next Generation Heterogeneous Networks," Proc. World Telecommun. Congress (WTC '06) (Budapest, Hungary, 2006).
[16] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer, "Session Initiation Protocol (SIP) Session Mobility," IETF Internet Draft, Feb. 2006.
[17] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method," IETF RFC 3515, Apr. 2003, <http://www.apps.ietf.org/rfc/rfc3515.txt>.
[18] R. Sparks, "The Session Initiation Protocol (SIP) Referred-by Mechanism," IETF RFC 3892, Sept. 2004, <http://tools.ietf.org/rfc/rfc3892.txt>.

(Manuscript approved September 2007)

JASPER AARTSE TUIJN worked at Alcatel-Lucent while preparing his master's thesis on the subject of partial session mobility in context-aware IP-based multimedia subsystems. He studied telematics at the University of Twente in the Netherlands, graduating with an M.Sc. He is the co-founder of CaRe-Mail, a company which provides a customer relationship management (CRM) solution for car dealers in the Netherlands.

DENNIS BIJWAARD is a senior member of technical staff at Bell Labs Europe and is based in Enschede, the Netherlands. He studied computer science at the University of Twente and graduated M.Sc. at Océ. Previously, he worked at the National Aerospace Laboratory in the Netherlands on distributed software, engineering workflows, and software engineering. At Alcatel-Lucent, he has been involved as researcher, architect, and developer on quality of service (QoS) for wireless local area networks (LANs), the Alcatel-Lucent MiLifeTM Intelligent Services Gateway (ISG) Software Developers Kit (SDK), and visualization of software behavior, service platforms, and mobility for heterogeneous third generation (3G) networks in the Dutch Freeband 4th Generation Platform Launching Ubiquitous Services (4GPLUS) project. In the Information Society Technologies' IST-DAIDALOS project he was activity leader for and deliverable editor of the service provisioning architecture, and in IST-DAIDALOS-II he is currently work package leader for context-aware service provisioning and network management. ♦