Collaboration Projects

Session Mobility: Service Continuity across Terminals

We had been working on location-based session mobility, which allows a mobile user to seamlessly transfer his/her ongoing multimedia sessions across different terminals in close proximity including his/her handheld devices. This research was conducted jointly with the Department of Computer Science (Professor Henning Schulzrinne and Ron Shacham), Columbia University, New York, USA.

Srisakul Thakolsri and Wolfgang Kellerer

1. Introduction

Internet multimedia communication and services will become available in many communication environments. Making use of end devices that are available in such ubiquitous environments allows a user to enjoy his personalized services from any place, anytime and on any device regardless of the type of access network. In addition, a user could also benefit from the advantages of mobile and stationary devices. For instance, stationary devices have better capabilities in terms of data rate, display size and computational power, while mobile devices are easily carried around.

Location-based session mobility is one type of a ubiquitous service. It allows a user to discover and acquire available devices in close proximity and to seamlessly continue his/her ongoing multimedia session on any available and appropriate device without terminating and re-establishing the call to the corresponding party. A number of approaches supporting session mobility have been proposed, however, their approaches have several drawbacks in terms of interoperability, backward compatibility, flexibility, and system deployment.

In this article, we describe the session mobility framework, which avoids the above drawbacks, by expanding the SIP[*1] based approach proposed in [1]. In particular, we focus on conversational multimedia sessions with different options of session transfer. Our research result is in the process of standardization in IETF[*2] [2]. It is also currently discussed in ETSI TISPAN[*3] and 3GPP.

2. System Architecture

The session mobility system architecture shown in **Figure 1** is based on three standardized protocols: Service Location Protocol (SLP) for device discovery, Session Initiation Protocol (SIP) and its extensions for signaling in order to support session transfer, Real-Time Transport Protocol (RTP) for all media transport such as audio and video. In Fig. 1, the Mobile Node (MN), which is invoking a session transfer, is a SIP-enabled mobile device. It implements standard SIP and the described SIP Session Mobility (SM) extension, which handles SIP signaling for session transfer. Local Devices, which are discovered by the MN, may be standard SIP-enabled devices or they may include a session mobility extension as well. At least one local device must have this extension, which is, for example, the video display in Fig. 1. The Correspondent Node (CN) is a basic SIP device implementing a SIP User Agent (UA), capable of setting up SIP calls. A transcoder is only used when the reconciliation of different capabilities during the session transfer is needed. For example, transcoding is needed when there is no codec of the CN available on a transfer target local device. We have chosen to implement device discovery using SLP, as it is a standardized protocol and allows discovery in different granu-

*1 SIP: A standard protocol for initiating, modifying and terminating a multimedia session among several clients.

*2 IETF: A standardization organization that develops and promotes standards for Internet technology.

*3 ETSI TISPAN: A non-profit, independent standardization organization for Internet-related technologies.

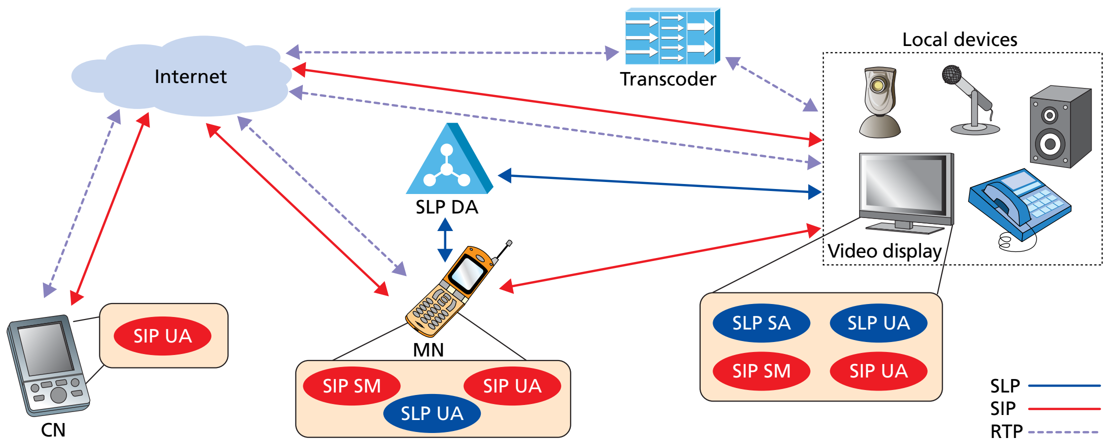

Figure 1 Session mobility system architecture

larities, such as in a specific room or on a floor. In SLP, the SLP Service Agent (SA) is responsible for advertising services, while the SLP User Agent (UA) queries for services. The SLP Directory Agent (DA) is the centralized directory keeping track of devices based on location and capabilities. Nevertheless, our architecture is not dependent on any specific discovery protocol.

3. Session Mobility Options

Our architecture supports various options of session transfer as described in the following:

- Transfer and Retrieval: After a session is transferred to local devices, a user may want to have his session back on the original device before the transfer happens.
- Complete Session Transfer (CST) or Splitting Session Transfer (SST): A user has a choice to either transfer his/her ongoing session consisting of several media types (e.g., audio and video media in a video call) completely to a single local device (CST) or split it across local devices (SST).
- Session Control Retention or Relinquishment: A user may or may not want to keep the control on the originating device initiating the session transfer. Our architecture provides two different modes: Mobile Node Control (MNC) mode and Session Handoff (SH) mode.

3.1 MNC Mode

MNC mode uses the SIP Third Party Call Control (3PCC) mechanism, in which the MN establishes a SIP session with each local device selected as transfer target and updates its existing session with the CN in order to setup a media flow between the CN and local devices. **Figure 2** shows the protocol flow for transferring a session from the MN to a local device with CST option. At first, the MN has established a SIP session and media flow with the CN, and they are communicating using audio and video media. In order to move the session, the MN initiates a new SIP session with the local device by sending an INVITE request (Fig. 2(1)). This request message has to contain the CN's media parameters (i.e., its IP address and media ports for transmitting and receiving) in order to set up a media flow between the local device and the CN. After establishing the session, the MN updates its existing SIP session with the CN by sending an INVITE including the local device's media parameters in its message body (Fig. 2(2)). Now, the media flow is established between the local device and the CN (Fig. 2(3)) whereas the SIP session is still kept between the MN and CN (Fig. 2(4)).

The flow for SST in MNC is similar to Fig. 2. The only difference is that the MN will have several sessions established, one for each local device. Also, the step of session update with the CN (Fig. 2(2)) has to include all media parameters from each local device. A retrieval of the session back to the MN can be done by updating the existing SIP session with the CN again. This update contains the media parameters of the MN in its

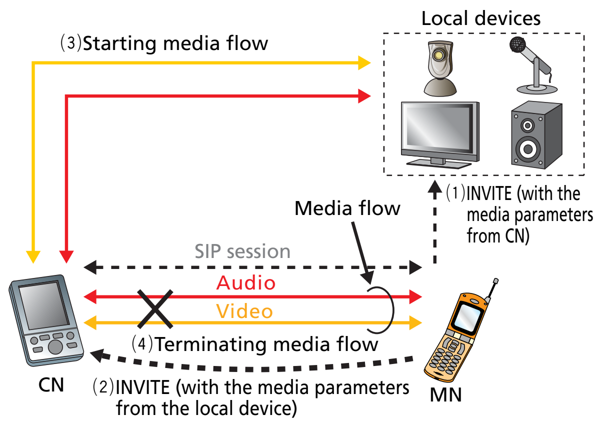

Figure 2 Protocol flow for MNC mode with CST

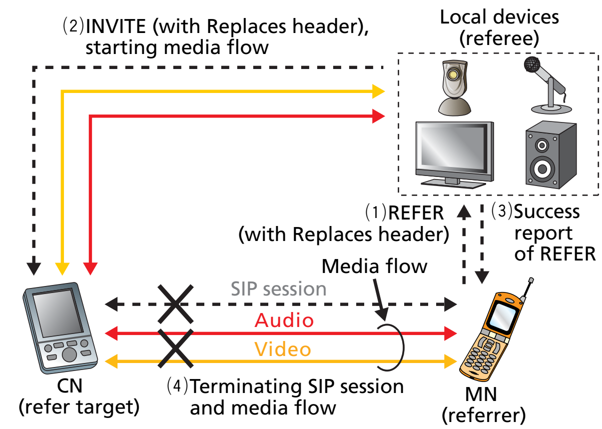

Figure 3 Protocol flow for SH mode with CST

body message. After session retrieval, the SIP sessions previously set up with local devices during the session transfer and the media flows between the CN and local devices will be terminated.

3.2 SH Mode

The SIP REFER method is used for the SH mode. It is a request message sent by a "referrer" to the "referee," who is asked to establish a new session with the "refer target." **Figure 3** shows the protocol flow for transferring a session in SH mode with CST option. The MN (referrer) sends a REFER to the selected local device (referee) (Fig. 3(1)) using a "Replaces" header and a "Referred-By" header that provide particular information about the session to be replaced and about the referrer (MN) respectively. This information is necessary for authentication and authorization as part of the session replacement. Consequently, the local device asks the CN (refer target) to replace its existing session (SIP session and media flow) with the MN by sending an INVITE, containing the Replaces header as specified in the REFER (Fig. 3(2)). Once the session is replaced, the local device notifies the MN about the success of session transfer (Fig. 3(3)). Finally, the MN terminates the SIP session and media flow with the CN (Fig. 3(4)). Retrieval of the session can be done either from the user interface on the local device, which is similar to the steps of performing a new, independent transfer, or using a nested REFER (For more details, see [2]).

SST in SH mode requires a Multi-Devices System (MDS), which combines multiple local devices and presents them virtually as one single system. This function can either be included with a local device that implements the session mobility extension or it can be implemented on a dedicated separate host. The protocol flow will be slightly different from Fig. 3, as the MDS will first establish a session to each local device and use their media parameters from their responses to invite the CN into a new session. The rest of signaling steps are the same.

4. Implementation and Performance Measurement

We have implemented a prototype of our architecture using Columbia University's SIP UA, and have enhanced it to support session mobility and device discovery based on location including the capability of creating a MDS. The location update is done through tags, so called iButtons®*4, or via Bluetooth®*5. To evaluate the performance of our system, we measured three parameters as described below, while performing the session transfer:

- Total Transfer Time (TTT): Time period starting from sending the transfer request made by the MN until receiving the media on the local device.
- MN Lapse Time (MNLT): Time period of media disruption caused by the update of the media parameters when the CN

*4 iButton®: A mechanical packaging standard containing a computer chip, in which we can store some information. A registered trademark of Dallas Semiconductor Corp. in the United States.

*5 Bluetooth®: A short-range wireless communication standard for interconnecting mobile terminals such as cell phones, notebook computers, and PDAs. A registered trademarks of Bluetooth SIG Inc. in the United States.

stops sending media to MN until it starts sending media to the local device.

- CN Lapse Time (CNLT): Time period of media disruption that occurs if the MN stops sending media to the CN before the local device starts sending media to the CN.

We set up a trans-Atlantic call between two SIP UAs (Munich, Germany and New York, USA) over the Internet, and measured the above performance criteria for different transfer options. For CST, we transfer an audio-only call to a local device, while in SST, the audio and video call is transferred to two different local devices.

Table 1 shows the average values of 10 measurements performed. Our results shown in the CNLT row verify that the CN will experience only a small audio disruption in SH mode, and no disruption at all in MNC mode. This is due to the fact that the local device already starts sending media to the CN, while the MN updates the existing session with the CN. However, due to the startup of the implemented video application (not due to the signaling), the media disruption for transferring video is larger than expected (see numbers in parentheses). This disruption will disappear, if the video application could start sending media without delay. This also applies to the measurement for TTT and MNLT. Here, the delay is about 2 seconds due to the restart of the video application.

MNLT is minimal in MNC mode. A few hundreds milliseconds are added in the SH mode due to the authorization check before session replacement. The TTT in both modes is less than one second, which is fast enough so that a user will consider that an invoked session transfer is working properly. The additional lapse time for video of the CN in SST is caused by the MDS system forwarding the signaling to the video node. In our experiment, we implement the MDS function on the audio node, so the lapse for audio is lower than for video.

Table 1 Performance measurement results

	CST (only audio)		SST (video call)			
	MNC	SH	MNC Audio	MNC Video	SH Audio	SH Video
TTT (ms)	815	420	397	400 (2,361)	906	900 (2,967)
MNLT (ms)	81	238	99	100 (2,098)	461	500 (2,523)
CNLT (ms)	0	154	0	0 (1,155)	193	600 (1,756)

5. Conclusion

We described a session mobility system architecture enabling a user to discover local devices available in close proximity and to make use of them by transferring an ongoing conversational session across those local devices. A prototype has been implemented and shows the feasibility of our approach. The performance measurement proves the effectiveness and seamlessness of the session transfer. Using this result, we will go ahead with the standardization.

References

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

[2] R. Shacham, H. Schulzrinne, S. Thakolsri and W. Kellerer. "Session Initiation Protocol (SIP) Session Mobility," IETF Internet Draft, draft-shacham-sipping-session-mobility-03, Nov. 2006.