

DISPATCH Working Group
Internet-Draft
Intended status: Informational
Expires: August 24, 2010

G. Camarillo
S. Loreto
Ericsson
February 20, 2010

Disaggregated Media in the Session Initiation Protocol (SIP)
draft-loreto-dispatch-disaggregated-media-01.txt

Abstract

Disaggregated media refers to the ability for a user to create a multimedia session combining different media streams, coming from different devices under his or her control, so that they are treated by the far end of the session as a single media session. This document lists several use cases that involve disaggregated media in SIP. Additionally, this document analyzes what types of disaggregated media can be implemented using existing protocol mechanisms, and the pros and cons of using each of those mechanisms. Finally, this document describes scenarios that are not covered by current mechanisms and proposes new IETF work to cover them.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 24, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Table of Contents

1.	Introduction	3
2.	Disaggregated media: Use Cases	3
2.1.	Using Two Separate devices to Start a Conversation	3
2.2.	Showing a Pre-recorded Video During a Conversation	4
2.3.	Sending a File from a PC During a Conversation	4
2.4.	Including Live Video in a Conversation	4
2.5.	Including Remote Live Video in a Conversation	5
2.6.	Answering a call using Two Separate Devices	5
2.7.	Other possible use cases	6
3.	Existing Mechanisms to Implement Disaggregated Media	6
3.1.	Message Bus (Mbus)	7
3.1.1.	Mbus issues	8
3.2.	Megaco (H.248)	8
3.2.1.	Megaco issues	9
3.3.	Third Part Call Control (3pcc)	9
3.3.1.	3pcc issues	10
4.	Scenarios Not Covered by Existing Mechanisms	11
5.	Security Considerations	12
6.	IANA Considerations	12
7.	Informational References	12
	Authors' Addresses	13

1. Introduction

Disaggregated media refers to the ability for a user to create a multimedia session combining different media streams, coming from different devices under his or her control, so that they are treated by the far end of the session as a single media session.

The SIP specification [[RFC3261](#)] defines a multimedia session as "an exchange of data between an association of participants". SDP (Session Description Protocol) is the default session description format in SIP. The SDP (Session Description Protocol) specification [[RFC4566](#)] defines a multimedia session as "a set of multimedia senders and receivers and the data streams flowing from senders to receivers".

Generally, a given participant uses a single device to establish (or participate in) a given multimedia session. Consequently, the SIP signaling to manage the multimedia session and the actual media streams are typically co-located in the same device. In scenarios involving disaggregated media, a user wants to establish a single multimedia session combining different media streams coming from different devices under his or her control. This creates a need to coordinate the exchange of the those media streams within the media session.

The remainder of this document is organized as follows. [Section 2](#) contains use cases where different media streams, coming from different devices, are combined to establish a multimedia session. [Section 3](#) describes what types of disaggregated media can be implemented using existing protocol mechanisms, and the pros and cons of using each of those mechanisms. [Section 4](#) describes scenarios that are not covered by current mechanisms and proposes new IETF work to cover them.

2. Disaggregated media: Use Cases

This section lists several use cases where users participate in a multimedia session using multiple devices. That is, either the user initiating the session uses several devices in parallel during the session, or the user receiving the session invitation uses several devices in parallel during the session, or both.

2.1. Using Two Separate devices to Start a Conversation

Laura is at her office. On her desk, she has a PC with a soft client and a desk phone. The PC has a low-quality built-in microphone and is connected to high-quality speakers.

Laura wants to establish a voice session with Bob using the desk phone as the microphone and the soft client as the speaker.

Laura wants Bob to treat the send-only audio stream from her deskphone and the receive-only audio stream from her softclient as part of the same communication space. That is, Laura wants Bob to treat both streams as belonging to the same multimedia session.

2.2. Showing a Pre-recorded Video During a Conversation

Bob has a voice-only phone and an IP-TV device. Laura has an integrated advanced multimedia phone with camera.

Bob is talking on his voice-only phone with Laura, who is on her multimedia phone.

Bob wants to show Laura part of the TV show he recorded last night. Bob interacts, using his voice-only phone, with his IP-TV device and sends a video stream to Laura's device.

Bob talks about the show with Laura on his voice-only phone while Laura watches the show.

Bob wants Laura to treat the video stream from his IP-TV device and the voice stream from his voice-only phone as part of the same communication space. That is, Bob wants Laura to treat both streams as belonging to the same multimedia session.

2.3. Sending a File from a PC During a Conversation

Bob has a voice-only phone and a PC with a soft client. Laura has an integrated advanced multimedia phone with support for file transfers.

Bob wants to send a file to Laura from his PC during his conversation with Laura on his voice-only phone.

Bob interacts, using his voice-only phone, with his PC and starts a file transfer session to Laura's multimedia phone.

Bob wants Laura to treat the file transfer from his PC and the voice stream from his voice-only phone as part of the same communication space. That is, Bob wants Laura to treat both streams as belonging to the same multimedia session.

2.4. Including Live Video in a Conversation

Bob has a voice-only phone and a PC which has a soft client, a Webcam, and a low-quality built-in microphone. Laura has an

integrated advanced multimedia phone with camera.

Bob wants to send a live video to Laura from his PC during his conversation with Laura.

Bob interacts, using his voice-only phone, with his PC and starts live video stream to Laura's multimedia phone.

Bob wants Laura to treat the live video stream from his PC and the voice stream from his voice-only phone as part of the same communication space. That is, Bob wants Laura to treat both streams as belonging to the same multimedia session.

2.5. Including Remote Live Video in a Conversation

Bob, who is at his office, has a multimedia phone. At his summer cottage, Bob has a webcam-phone (e.g. a video-surveillance system). Laura has an integrated advanced multimedia phone with a camera.

During his conversation with Laura, Bob wants to show her the new summer cottage he just bought. Bob interacts, using his multimedia phone, with his webcam phone at this summer cottage and start live video stream to Laura's multimedia phone.

Bob wants Laura to treat the live video stream from his webcam-phone and the voice stream from his voice-only phone as part of the same communication space. That is, Bob wants Laura to treat both streams as belonging to the same multimedia session.

2.6. Answering a call using Two Separate Devices

Laura has a PC with a softclient and a desk phone. Bob has an integrated advanced multimedia phone with camera.

Laura receives on her desk phone an incoming voice-video call from Bob.

Laura decides to answer Bob's session invitation by establishing a voice session with Bob using the desk phone and the video session using her multimedia phone. Two SIP dialogs need to be established: one between Bob's device and Laura's desk phone and one between Bob's device and Laura's multimedia phone.

Laura wants Bob to treat the audio stream from her deskphone and the video stream from her softclient as part of the same communication space. That is, Laura wants Bob to treat both streams as belonging to the same multimedia session.

2.7. Other possible use cases

This section just enumerates, for sake of completeness, other possible use cases, similar to the one elaborated in the previous sections.

A user wants to start or answer a call combining:

- o Voice and video streams from a deskphone and application sharing from a computer.
- o Voice stream from a deskphone and video stream to/from a TV attached to a set top box with a camera built in.
- o The User Interface (UI) for the call on a mobile phone and the audio streaming coming in/out of a speakerphone that is in the same room where he is.

3. Existing Mechanisms to Implement Disaggregated Media

Figure 1 shows the media flow in the most generic scenario where both the Caller and the Callee use disaggregated media, involving in the multimedia session different devices under their control.

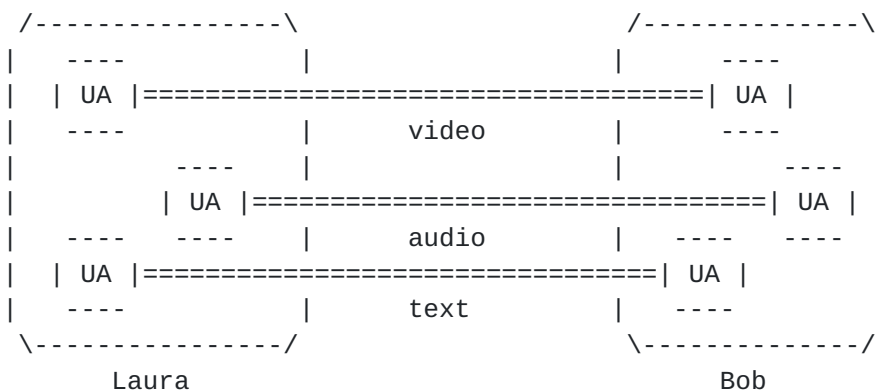


Figure 1: Media Flows in Disaggregated Media

All existing mechanisms to implement disaggregated media in SIP use a centralized approach whereby the far end of the session receives the same SIP signaling flow that it would receive if all the media streams came from a single device. This makes it transparent to the far end of the session the fact that the caller is using separate devices for different media.

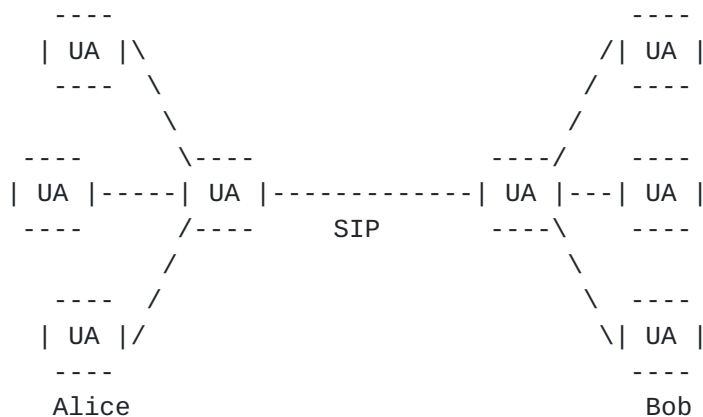


Figure 2: Centralized scenario

Figure 2 shows the generic signaling flow common to all centralized solutions, where a Central Node is able to manage all signaling messages needed to coordinate the different user's devices and hide from the far end of the session all the mechanisms used to distribute the media among different devices.

[Section 3.1](#), [Section 3.2](#) and [Section 3.3](#) analyze how different existing mechanisms can be used to implement disaggregated media in a centralized way.

3.1. Message Bus (Mbus)

The Message Bus (Mbus) [[RFC3259](#)] is a light-weight local coordination protocol for developing component-based distributed applications. Mbus provides a simple and flexible message oriented communication channel for a group of components that may be distributed on multiple hosts in a local network. The transport services include useful features such as peer location, point-to-point and group communication and security.

In a disaggregated media scenario a user can use Mbus to coordinate the different devices under his control in a loosely coupled conference control model and so involve them in the call. The different devices can communicate with one another using Mbus messages, and then let only one device, a call control engine, to initiate, manage and terminate call control relations to other SIP endpoints. In this case the fact that the caller is using separate devices for different media becomes transparent to the callee.

Figure 3 shows an example of the relation between a call control engine in an Mbus enabled conferencing system.

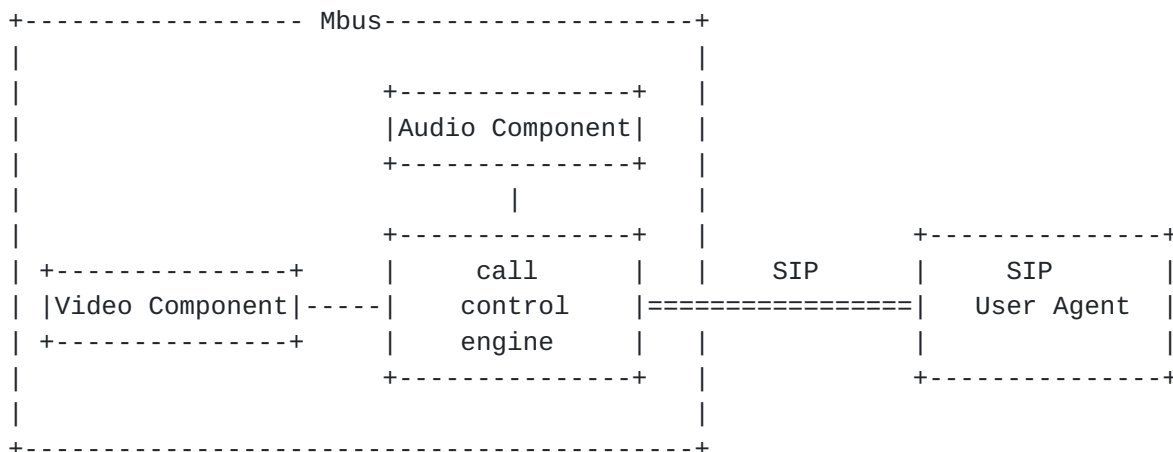


Figure 3: Mbus Architecture

3.1.1. Mbus issues

The Mbus protocol introduces the following issues.

Mbus support: All the different user's devices need to support the Mbus protocol.

Central point: Because the call control engine has a complete control over the call, it needs to be involved during the whole duration of the session. It cannot leave the session before the whole session ends (unless it transfers the controller role to one of the other user's devices).

Local network: Mbus uses multicast with a limited scope for message transport. The multicast limits the coordination mechanism only to groups of devices that are connected to a local network. So Mbus can be used in a disaggregated media scenario only if all the different devices under the user control, or at least the one the user wants to involve in the communication, are attached to the same local network.

3.2. Megaco (H.248)

The Megaco [[RFC3525](#)] protocol is used to establish, and terminate calls across terminations (end points) connected to Media Gateways (MGs). Megaco instructs Media Gateways (MG) to connect streams coming from outside a packet network on to a packet stream such as RTP. Master-MGC issues commands to send and receive media from addresses, generate tones, and to modify configuration. The architecture requires a Media Gateway Controller (MGC) controlling the Media Gateway(s). However it does not constitute a complete system. To establish communication between MGC(s) is necessary a session initiation protocol. SIP is the protocol normally used to

establish calls across domains or across MGCs.

Megaco can be used in a disaggregated media scenario to let one of the user's devices act as a Media Gateway Controller, coordinating all the other devices under the user's control, which in this case will act as Media Gateways. In this case the fact that the caller is using separate devices for different media becomes transparent to the callee.

3.2.1. Megaco issues

The Megaco protocol introduces the following issues.

Megaco support: All the different user's devices need to support the Megaco protocol.

Central point: Because the Media Gateway Controller has a complete control over the call, it needs to be involved for all the session time. It can not leave the session before the whole session ends.

3.3. Third Part Call Control (3pcc)

3pcc [[RFC3725](#)] allows one entity (called the Controller) to setup and manage a communications relationship between two or more other parties using SIP.

In a disaggregated media scenario, a user can use 3pcc mechanism only if at least one among the different devices under his control supports this mechanism and is able to become a Controller for the other in the call. In this case become transparent for the callee the fact that the caller is using separate devices for different media. In fact the Controller is a central point for the signaling on the caller side, and has complete control over the call, making everything in one dialog for the callee.

The call flow for disaggregated media using 3pcc is shown in Figure 4. It is based on Flow IV documented in [Section 4.4 of \[RFC3725\]](#) and recommended for calls to unknown entities, or to entities known to represent people. The flow requires some SDP manipulation by the Controller to convert offer2 to offer2' and offer2'', and then to convert answer2' and answer2'' to answer2.

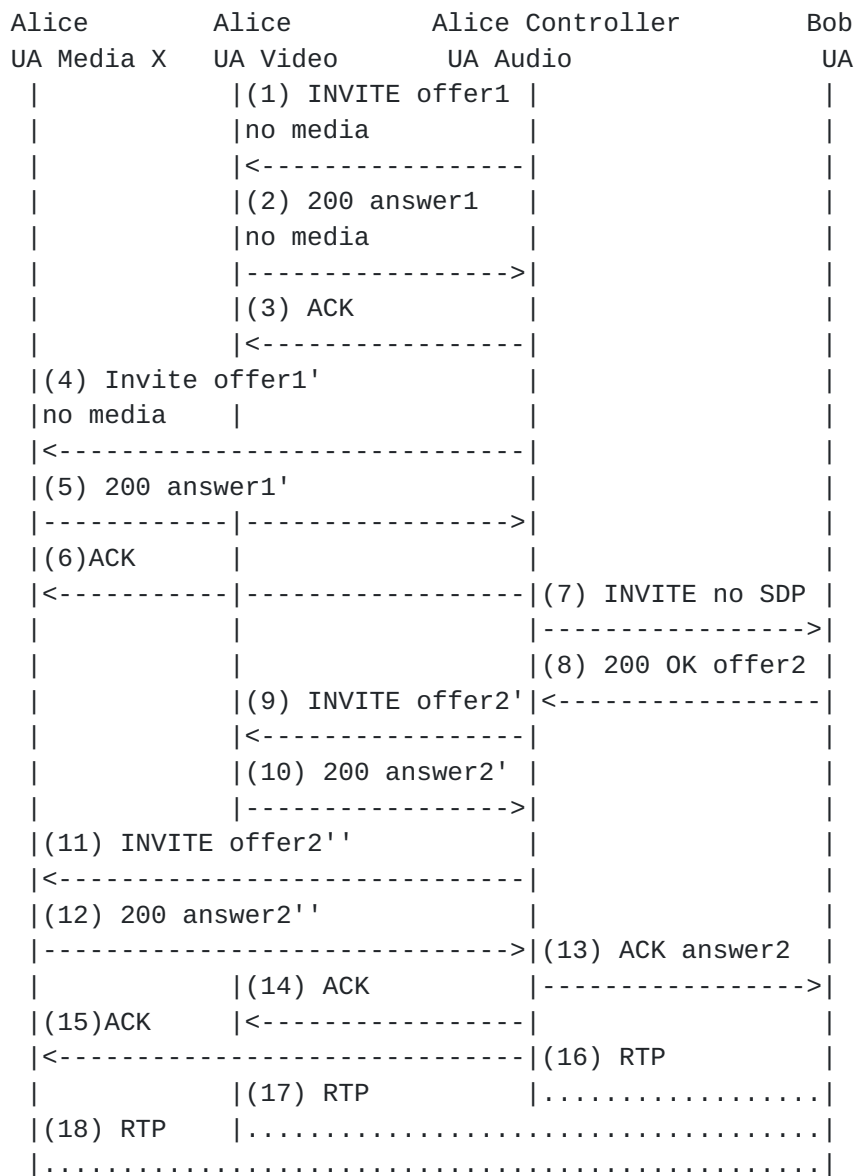


Figure 4: 3pcc call flow

3.3.1. 3pcc issues

The 3pcc mechanism introduces the following issues.

Complexity: Third party call control only uses protocol mechanism specified in [[RFC3261](#)]. However the usage of third party call control becomes more complex when aspects of the call utilize SIP extensions or optional features of SIP like resource reservation.

Central point: Because the Controller has a complete control over the call, it needs to be involved during the whole duration of the session. It cannot leave the session before the whole session ends (unless it transfers the controller role to one of the other user's devices).

User experience: 3ppc results in a suboptimal user experience because the slave phones are not aware that they are involved in a disaggregated media call scenario. Indeed, the slave phones behave as they were just involved in a normal call with the Controller. Moreover the slave phones will be alerted without any media having been established yet.

SDP manipulation: the Controller cannot "proxy" the SIP messages received from one of the parties. In many cases, it is required to modify the SDP exchanged between the participants in order to affect the changes.

4. Scenarios Not Covered by Existing Mechanisms

As discussed previously, all existing mechanisms implement disaggregated media using a centralized approach. Scenarios not covered by existing mechanisms include those where none of the nodes can act as a controller because it does not support the necessary functionality or because it will not participate in the whole session (transferring the SIP dialog from a controller to a new one using REFER and Replaces is complex and requires support from the far end). These scenarios are better implemented using a more distributed approach.

In a distributed scenario, the far end of the session receives directly signaling messages from each of the devices involved in the multimedia session. That means that the receiver needs to treat all the signaling and media coming from different devices of the same user as part of the same media session.

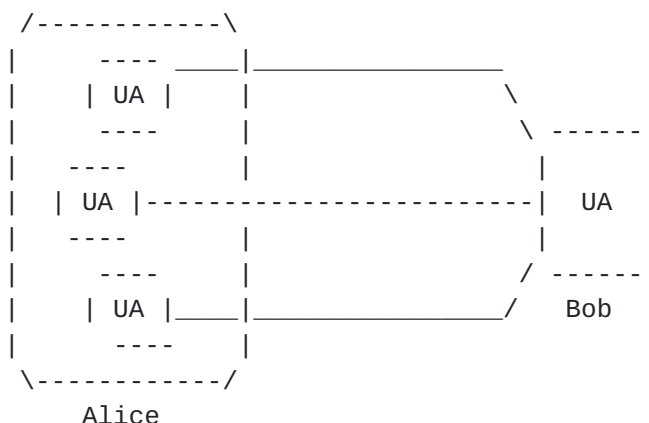


Figure 5: Distributed scenario

Figure 5 shows the generic signaling flow in a Distributed Scenario, where all the signaling messages go from the different user's devices to the far end of the session.

Since this type of scenario is not covered by existing mechanisms, we propose to initiate work on SIP extensions to support it. These extensions may require support from the far end of the session. While this may limit the usability of these extensions in some scenarios, scenarios where an administrator can deploy devices with support for a given extension (e.g., in an enterprise) could still benefit from it.

5. Security Considerations

To be done.

6. IANA Considerations

This document does not require any actions by the IANA.

7. Informational References

- [RFC3087] Campbell, B. and R. Sparks, "Control of Service Context using SIP Request-URI", [RFC 3087](#), April 2001.
- [RFC3259] Ott, J., Perkins, C., and D. Kutscher, "A Message Bus for Local Coordination", [RFC 3259](#), April 2002.
- [RFC3261] 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.
- [RFC3525] Groves, C., Pantaleo, M., Anderson, T., and T. Taylor, "Gateway Control Protocol Version 1", [RFC 3525](#), June 2003.
- [RFC3725] 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.
- [RFC3911] Mahy, R. and D. Petrie, "The Session Initiation Protocol (SIP) "Join" Header", [RFC 3911](#), October 2004.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.

Authors' Addresses

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: salvatore.loreto@ericsson.com

