3GPP TSG-CN Meeting #24 **NP-040268**
02 – 04 June 2004, Seoul, KOREA

Source: **CN5 (OSA)**

Title: **2 Rel-6 CRs 29.198-04-4 OSA API Part 4: Call control; Sub-part 4: Multi-Media Call Control SCF**

Agenda item: **9.7 (OSA Enhancements [OSA3])**

Document for: **APPROVAL**

Doc-1st-	Spec	CR	Rev	Phase	Subject	Cat	Version	Doc-2nd-	Workite
NP-040268	29.198-04-4	016	-	Rel-6	Correction of description in superviseVolumeRes - Align with Rel-5	F	6.1.0	N5-040113	OSA3
NP-040268	29.198-04-4	017	-	Rel-6	Correction of method references in MMCC - Align with Rel-5	F	6.1.0	N5-040114	OSA3

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) N5-040113
Meeting #26, Atlanta, GA, USA, 16-20 February 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-04-4** **CR** **016** ⌘ **rev** **-** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correction of description in superviseVolumeRes - Align with Rel-5
Source:	⌘	CN5 Parlay Gareth Carroll (Open API Solutions)
Work item code:	⌘	OSA3
Date:	⌘	09/02/2004
Category:	⌘	**F**
Release:	⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The description of superviseVolumeRes is incorrect. It states that the method "… is invoked as a response to the request also when a tariff switch happens in the network during an active call." There is no appropriate value in TpCallSuperviseReport to indicate any tariff change, only to indicate that the supervision timer has expired, the call has ended, a warning tone has been applied, or UI has completed. The corresponding superviseVolumeReq method also does not mention tariff change notification.
Summary of change:	⌘	We propose that the text in superviseVolumeRes that references tariff changes should be removed. This is easier than introducing a new value into TpCallSuperviseReport and makes the superviseVolumeRes correspond closer to the description given in superviseVolumeReq.
Consequences if not approved:	⌘	Failure to remove this incorrect statement could lead to implementations invoking superviseVolumeRes with an inappropriate and irrelevant value simply because a tariff has changed in the network and there is no appropriate value to use.

Clauses affected:	⌘	6.4.1			
Other specs affected:	⌘	**Y**	**N**		
			X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

6.4 Interface Class IpAppMultiMediaCall

6.4.1 Method superviseVolumeRes()

This asynchronous method reports a call supervision event to the application when it has indicated its interest in these kind of events.

It is also called when the connection is terminated before the supervision event occurs. Furthermore, this method is invoked as a response to the request also when ~~a tariff switch happens in the network during an active call or~~ the Quality of Service parameters were renegotiated during the active call.

Parameters

`callSessionID : in TpSessionID`

Specifies the call session ID of the call

`report : in TpCallSuperviseReport`

Specifies the situation which triggered the sending of the call supervision response.

`usedVolume : in TpCallSuperviseVolume`

Specifies the used time for the call supervision (in milliseconds).

`qualityOfService : in TpDataSessionQosClass`

Specifies the newly negotiated Quality of Service parameters for the multimedia call.

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) N5-040114
Meeting #26, Atlanta, GA, USA, 16-20 February 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-04-4** **CR** **017** ⌘ **rev** **-** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correction of method references in MMCC - Align with Rel-5
Source:	⌘	CN5 Parlay Gareth Carroll (Open API Solutions)
Work item code:	⌘	OSA3
Date:	⌘	09/02/2004
Category:	⌘	**F**
Release:	⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	There are a number of incorrect method references in the Multi-media call control specification. Sequence diagram 4.4 references routeReq which no longer exists on IpMultiPartyCall, and routeRes, which no longer exists on IpAppMultiPartyCall.
Summary of change:	⌘	We propose to correct the incorrect method references.
Consequences if not approved:	⌘	Failure to correct these method references can lead to confusion and, potentially, incorrect and non-interoperable implementations.

Clauses affected:	⌘	4.3, 4.4, 6.1.3

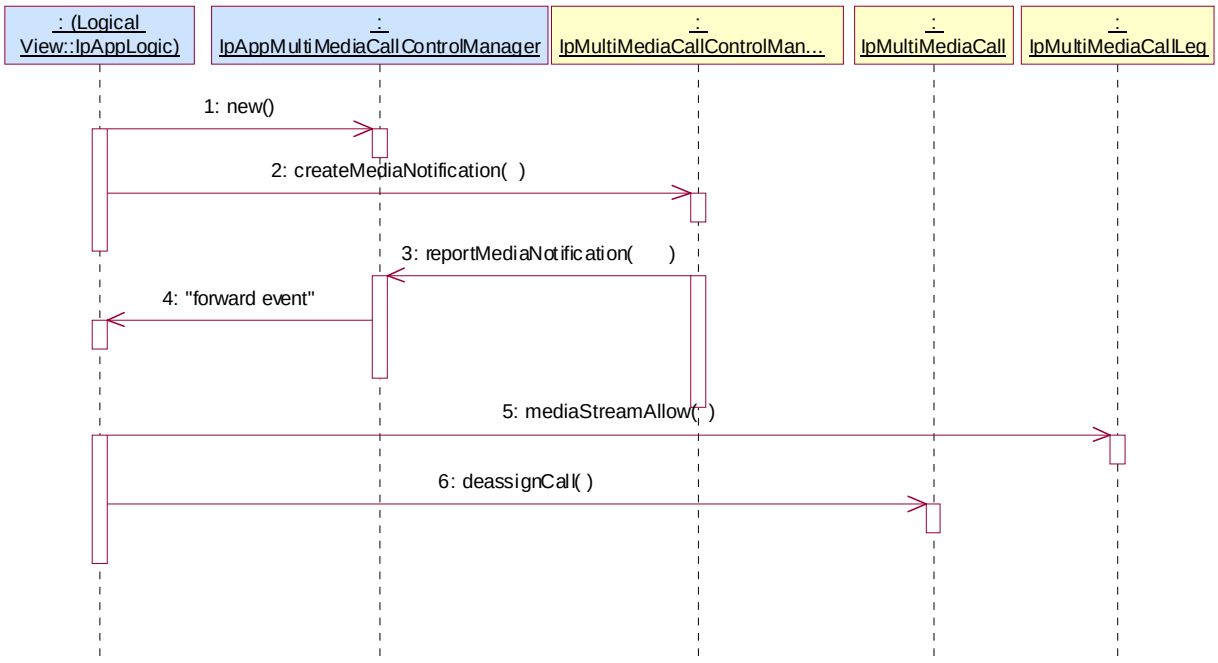
		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

4 MultiMedia Call Control Service Sequence Diagrams

4.1 Barring for media, simple

This sequence illustrates how an application can block the establishment of video streams for a certain user.



1: The application starts a new AppMultiMediaCallControlManager interface for reception of callbacks.

2: The application expresses interest in all calls from or to subscriber A that use video. The just created App interface is given as the callback interface.

3: Subscriber A makes a call with the SIP INVITE with SDP media stream indicating video.

4: The message is forwarded to the application.

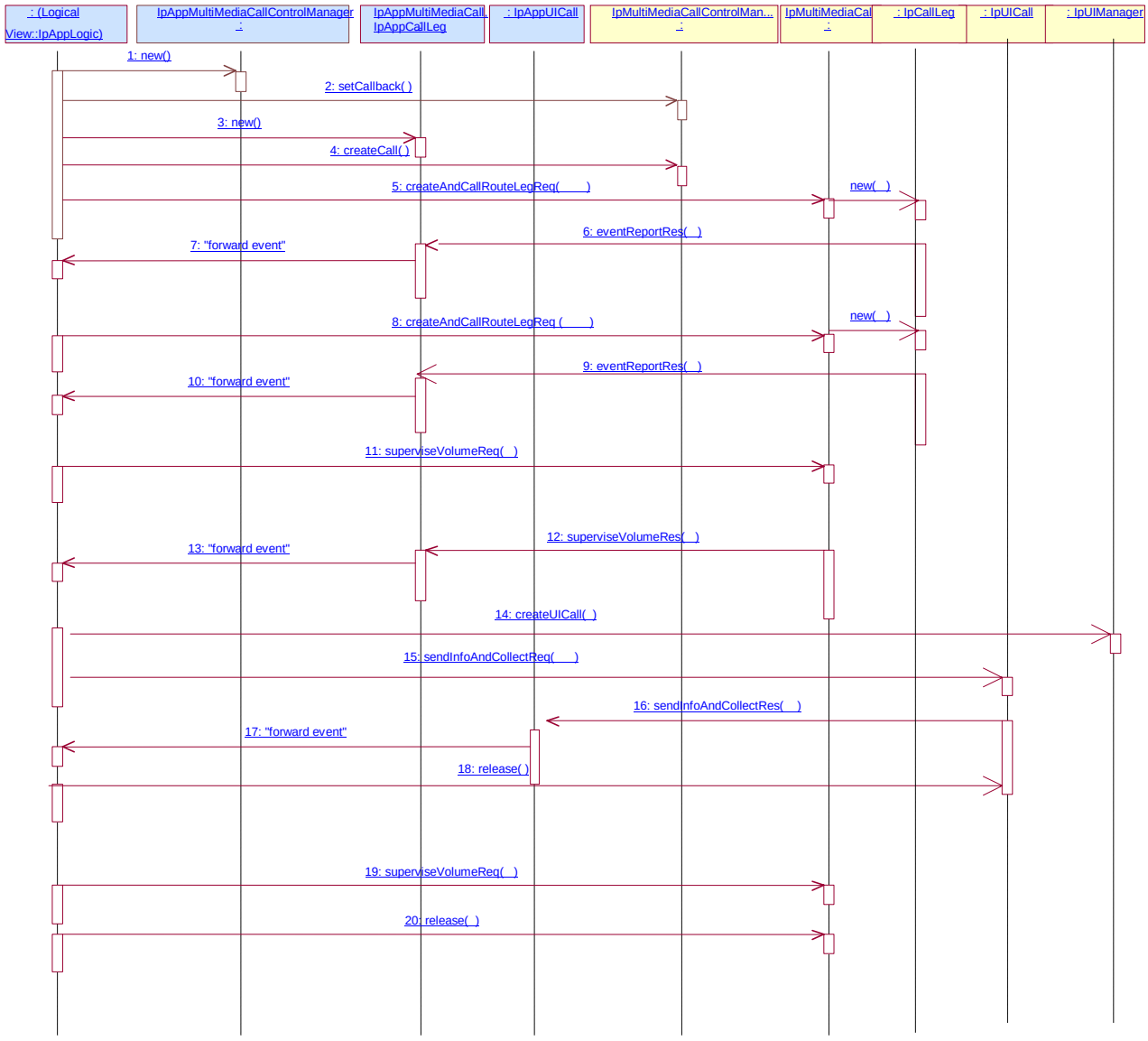
5: The application indicates that the setup of the media stream is not allowed by not including the media stream in the allowed list. This has the effect of suppressing the video capabilities in the setup.

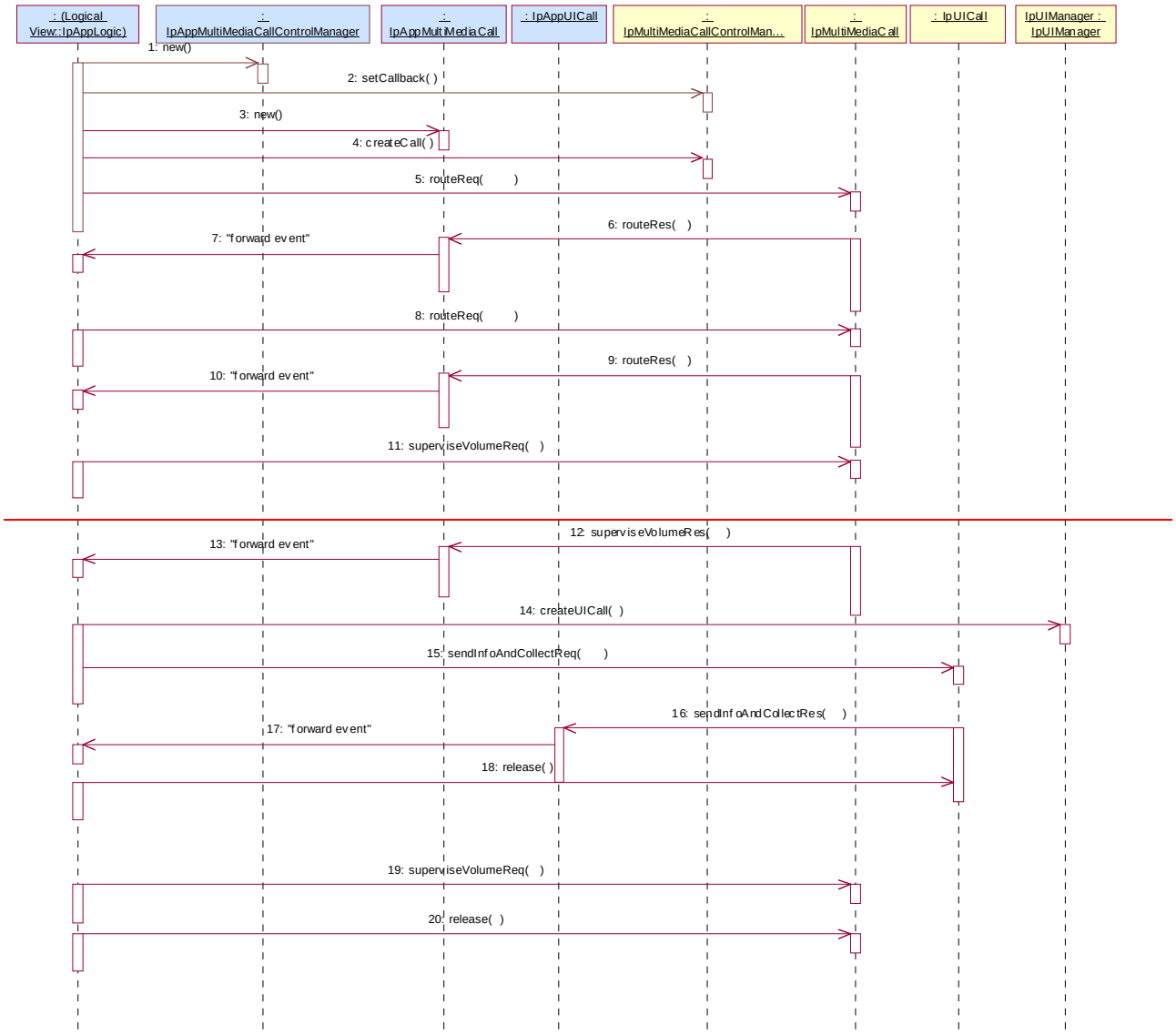
6: The application is no longer interested in the call.

New attempts to open video streams will again be indicated with a ~~createMediaNotification~~reportMediaNotification.

4.2 Call Volume charging supervision

This sequence illustrates how an application may supervise a call based on the number of bytes that are exchanged.


: (Logical View::IpAppLogic)
IpAppMultiMediaCallControlManager :
IpAppMultiMediaCall, IpAppCallLeg
: IpAppUICall
IpMultiMediaCallControlMan... :
IpMultiMediaCal :
: IpCallLeg
: IpUICall
: IpUIManager
1: new()
2: setCallback( )
3: new()
4: createCall( )
5: createAndCallRouteLegReq( )
new( )
6: eventReportRes( )
7: "forward event"
8: createAndCallRouteLegReq ( )
new( )
9: eventReportRes( )
10: "forward event"
11: superviseVolumeReq( )
12: superviseVolumeRes( )
13: "forward event"
14: createUICall( )
15: sendInfoAndCollectReq( )
16: sendInfoAndCollectRes( )
17: "forward event"
18: release( )
19: superviseVolumeReq( )
20: release( )



Note: In the sequence diagram above, a single box represents both an IpAppCall and an IpAppCallLeg for space reasons.

1: The application creates a new interface to receive callbacks on the call control manager.

2: The created interface is set as the callback interface for the call control manager.

3: The application creates a new interface to receive callback on the call.

4: The application requests the creation of a call.

5: The application initiates the call by routing to the origination. This will implicitly create a call leg. The application requests a notification when the party answers.

6: When the A party answers the application is notified.

7: The message is forwarded to the logic.

8: The application also routes the call to the destination. This implicitly creates a call leg. The application requests to be notified on answer of the B-party.

9: When the B-party answers the application is notified.

10: The message is forwarded to the logic.

11: The application requests to supervise the call. In the request the application specifies a limit on the amount of bytes that may be transferred. The application specifies that if the limit is reached the application should be notified.

12: When the limit is reached a notification is send to the application.

13: The message is forwarded to the logic.

15: The application plays an announcement to the user, asking whether the user wants to end the call or continue the call.

16: When the user answers whether the call should continue.

17: The message is forwarded to the logic.

18: The UIcall is released, since no further announcements are needed.

19: In case the user answers that the call should continue, the supervision is reset with a new maximum number of allowed bytes. (note that this might have charging consequences, not shown)

20: If the user answered that the call should not continue, the call is released.

6.1 Interface Class IpMultiMediaCallControlManager

6.1.3 Method changeMediaNotification()

This method is used by the application to change the event criteria introduced with createMediaNotification. Any stored criteria associated with the specified assignmentID will be replaced with the specified criteria.

Parameters

assignmentID : in TpAssignmentID

Specifies the ID assigned by the multi-media call control manager interface for the media stream notification. If two callbacks have been registered under this assignment ID both of them will be ~~disabled~~changed.

notificationMediaRequest : in TpNotificationMediaRequest

Specifies the new set of event specific criteria used by the application to define the event required. Only events that meet these criteria are reported.

Raises

TpCommonExceptions, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE