Session and Media Signaling for IPTV via IMS

Christian Riede
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
riede@fokus.fraunhofer.de

Adel Al-Hezmi
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
al-hezmi@fokus.fraunhofer.de

Thomas Magedanz
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
magedanz@fokus.fraunhofer.de

ABSTRACT

As the IMS provides session establishment, QoS parameter negotiation, authentication and accounting it is an overlay architecture well qualified for session signaling concerning IPTV scenarios. The management of the signaling can be integrated as an application into an IPTV Application Server. The IPTV AS is further responsible for all necessary media signaling towards a proper IPTV media server. Such a media server is similar to the ordinary media server known from the voice centric IMS but additionally provides unicast, multicast and broadcast delivery of TV content which is placed by content providers. The media signaling requires a dedicated session description and can be provided by SIP or RTSP. The signaling between user and media server is orchestrated by a service-oriented-archietctue with a session manager acting as middleware. This middleware also provides interfaces for web services triggering IPTV session establishment via Parlay X gateway. This paper proposes a general IMS-based framework for delivering Live TV services over several access networks and presents the corresponding signaling flows for content enquiring and content switching as well as related delays. It furter presents an approach for using SIP and RTSP in cooperation for realizing VOD scenarios.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Protocols—*Applications*; C.2 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communications*

General Terms

Management

Keywords

IMS, IPTV, SIP, RTSP, session signaling, media signaling

Mobilware'08 February 12-15, 2008, Innsbruck, Austria
Copyright 2008 ACM 978-1-59593-984-5/08/02 ...$5.00.

1. INTRODUCTION

As IP-based networks are more and more improving in terms of bandwidth and QoS, one of the contemporary trends concerning multimedia streaming is Internet Protocol Television (IPTV). IPTV is accepted by the consumer only if its quality of transmission concerning jitter and signaling delay, the available content to be consumed by the end customer and the price are more than comparable with traditional television distribution channels. The quality of Internet TV, as provided by YouTube and other video-podcast providers via Internet are far from acceptable quality for consumers. However, the IMS [1] guarantees Quality of Service (QoS) and provides triggering of additional value-added services like "see-what-I-see" services, presence services or call forwarding in case of incoming calls during an IPTV session and delivers a charging mechanism which allows easy pay-per-view business models. The IMS as an overlay architecture is not restricted to a certain access network but supports nearly all imaginable IP based bearer technologies like UMTS, DVB-H or WiFi. But such an IPTV provisioning requires a dedicated session management which acts as switching instance between consumer and content provider. Such a switching instance can be hosted at an IMS Application Server (AS). So the IPTV session management (SME) is more or less an application which provides control on the IPTV services including service authorization, bearer selection and media signaling towards a media server (MS) which insures media processing and media delivery via unicast, multicast and broadcast. So the SME addresses two crucial tasks:

- Multimedia Streaming Control
  - enabling the content provider to:
    - indicate about starting transmission,
    - polling for a list of pending streaming requests,
    - controlling an ongoing multimedia streaming session,
    - getting the state of media streaming session and
    - requesting that the operator applies a refund of charges to an end user account
- Multimedia Multicast Control
  - enabling an application to control the 3GPP Broadcast Multicast Service Center (BM-SC) for multicasting sessions

2. RELATED WORK

The authors of [2] have introduced an architecture and a session manager which is for realizing multimedia streaming scenarios in terms of quadruple play by means of IMS. Similar work is done by [3] and [4]. Both paper concentrate even in case of VOD on using SIP and IMS. However, the media signaling between IPTV session management and media server can be realized by SIP [5] or RTSP [6]. The latter refers to a more video-on-demand (VOD) server, which allows direct media control from the user. This paper introduces a dedicated IPTV framework via IMS and the general media signaling for an IPTV scenario based on SIP and RTSP. We further present detailed call flows for a Live TV scenario concerning SIP session and media signaling and a VOD scenario concerning SIP and RTSP in cooperation. At the end we provide measurement datas regarding content triggering as well as content switching delay for IPTV Live TV based on SIP. This paper further reflects the IPTV session manager as middleware for docking Web 2.0 web services triggering session establishment via Parlay X gateway.

3. IMS-BASED IPTV FRAMEWORK

The IMS as an overlay architecture can be supposed as an ideal signaling infrastructure on top of different IP access networks. However, the IMS does not provide the switching logic between different access networks on its own. Thus, a comprehensive IPTV framework [see figure 1] additionally requires a certain logic layer which provides services essential for enhanced IPTV scenarios. So a session management enabler (SME) in this paper just called session manager is the heart of this logic layer. The session manager is logically the switching instance between user and content provider as it keeps track of all user-to-content relationships. It furthermore enables switching between unicast, multicast and broadcast transmission and facilitates awareness as user availabiliy and current activities. The service logic layer within this IPTV framework additionally provides further services as group management (so several user can be organized in groups of same behaviour), bearer manager which determines the transmission scheme (unicast, multicast, broadcast) and the access network, furthermore a presence server which is necessary to offer "see-what-I-see" services and charging to realize easy per-per-view applications. The session manager provides interfaces to the application layer which hosts different advanced streaming services. A further key part of the IPTV framework is the IPTV media server. This server is logically the interface between session management and access network (signalized via IMS). Such a media server can be separated in several instances with a load balancer above. In this case the session manager communicates with the control instance for the IPTV media server (not included in figure 1)

The architecture is divided into four building blocks which cope with all aspects of consumer's, network provider's, service provider's and content provider's point of view.

- *Application Layer:* IPTV applications hosted on a dedicated Application Server (e.g. by a 3rd party provider).
- *Service Enabler Layer:* this layer provides the service logic to organize user in groups, to switch between different access technologies, and to keep track of user-to-content relationships

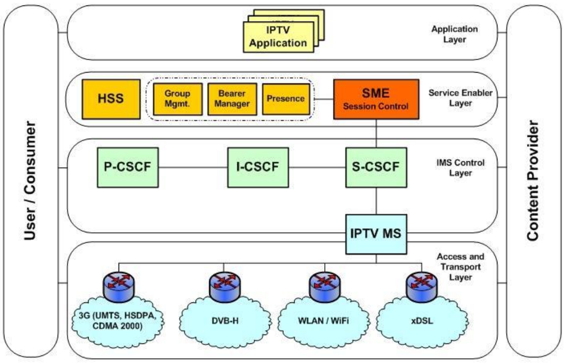

Figure 1: IPTV Framework via IMS

- *IMS Control Layer:* is represented by the IMS core for controlling and managing multimedia sessions. This layer contains logically the IPTV media server for controlling the access and transport layer.
- *Access and Transport Layer:* includes wired and wireless IP-based access technologies

4. IPTV SESSION AND MEDIA SIGNALING

Within the IMS as a secure and reliable signaling architecture, SIP can convince by its ability to establish, modify and terminate interactive user sessions. SIP has already proofed its efficiency in the scope of voice centric communication services (e.g. in the field of audio codec negotiation). However, concerning multimedia streaming, a different protocol, RTSP, is well known for session and media control. As the IMS does not speak RTSP, its interesting to face up with SIP as media signaling protocol.

4.1 Scenario Description

Subsequently we describe four general scenarios. The first scenario addresses the content provisioning. This scenario describes how to announce a content stream and suggests a session and media signaling call flow. The second scenario refers to the content stream request in case of Live TV provisioning. It details how the consumer and the session management controls the media server in respect of unicast and multicast content transmission. The third scenario describes a practical way to handle content stream switching requests. As the session management can decide to change the transmission scheme (e.g. in case of network resource economy) it is necessary to split up this scenario into two modes:

- alter transmission scheme
  - unicast to multicast
  - multicast to unicast
- do not alter tranmission scheme
  - unicast to unicast
  - multicast to multicast

The last of the four proposed scenarios is a VoD scenario [see section 4.3] which points out how SIP and RTSP may work in cooperation. It proposes a step-by-step concept of performing RTSP issues by SIP.

4.2 IPTV signaling with SIP only

4.2.1 Content stream publishing

Due to the fact that multimedia content associated to a particular service may include multiple tracks (audio, video, data) that might change over the transmission time, a dedicated protocol and a SDP offer/answer model which supports codec negotiation and re-negotiation within the same session are to be found for session signaling. As SIP and the IMS provide user authorization, and SIP additionally allows both players (service provider and content provider) to start the session, content publishing with SIP on top of IMS is a researchable issue. Announcing an available content stream from the Content Provider's perspective is quite easy to handle. The CP sends a SIP INVITE message to the session manager. The session manager checks for an adequate media server and relays the invitation to the media server. The media server responds with a Session Description Protocol (SDP) containing a media server. This SDP is transferred to the CP which knows then where to push the RTP stream [see figure 2].

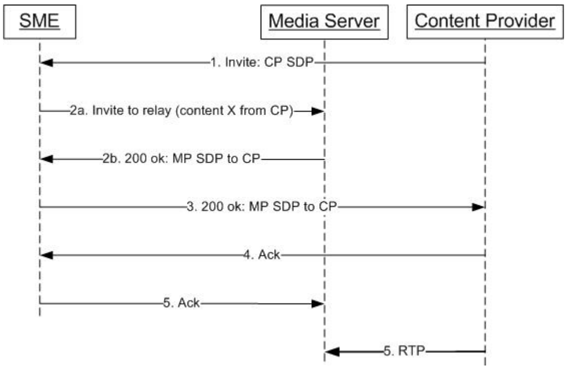

Figure 2: Content Provider: publishing live content stream

4.2.2 Live TV content stream request (unicast)

The consumer's communication end point is always the session manager. The session manager resolves a qualified media server and decides to deliver the stream per unicast or per multicast. So the session manager is a B2BUA with two legs: one leg describing the SIP session between user and session manager and the second leg the SIP session between session manager and media server. In case of unicast delivery [see figure 3] the session manager activates a dedicated unicast delivery from the media server to the user (here user 1 and user 2). This additionally requires QoS activation and reservation between user and media server. Finishing the streaming session requires a SIP BYE message from the user and finishing the SIP session between session manager and media server representing unicast delivery to the aborting user. Within a linear live TV session no consideration of handling trick functions (FF, RW, pause) is necessary as linear live TV does not need support for trick functions per definition.

4.2.3 Content stream request (multicast)

The content stream request for multicast delivery [see figure 4] is quite similar to the unicast delivery approach above.

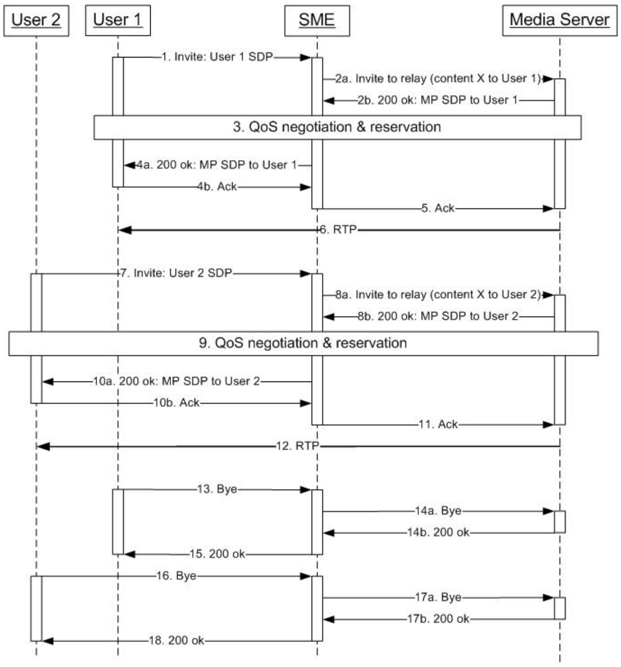
User 2
User 1
SME
Media Server
1. Invite: User 1 SDP
2a. Invite to relay (content X to User 1)
2b. 200 ok: MP SDP to User 1
3. QoS negotiation & reservation
4a. 200 ok: MP SDP to User 1
4b. Ack
5. Ack
6. RTP
7. Invite: User 2 SDP
8a. Invite to relay (content X to User 2)
8b. 200 ok: MP SDP to User 2
9. QoS negotiation & reservation
10a. 200 ok: MP SDP to User 2
10b. Ack
11. Ack
12. RTP
13. Bye
14a. Bye
14b. 200 ok
15. 200 ok
16. Bye
17a. Bye
17b. 200 ok
18. 200 ok

Figure 3: Content stream request (unicast delivery)

But the session manager has to invoke the media server one time only to deliver the content stream via multicast. Subsequent request for this content stream are to be responded by the session manager immediately with the correct multicast address for the requested content stream. The consumer has to care for the IGMP join process. The session manager has to assure that the SIP session between session manager and media server representing the multicast delivery is to be terminated in case no user is consuming this multicast content. In fact the content stream could be transmitted via multicast even if no user is demanding this content stream but this would cause unnecessary network traffic. QoS negotiation and reservation for multicast is commited only between the network nodes at the beginning of the first user request. It is to be regarded that SIP adds a session setup delay in addition to the join request.

4.2.4 Content stream switch (unicast to unicast)

The necessary information regarding the requested content is specified in the SDP. The RFC 3261 proposes SIP re-INVITE within the same SIP session dialog for updating the SDP without creating a new SIP session. This mechanism is suitable for the invocation of a content stream switch. The session manager further uses the same mechanism for switching the content stream towards the media server. As a result the media server changes the content stream delivered to the user. Figure 5 displays a possible call flow on condition the CP already provides the requested content. Here the user switches the content stream without changing the transmission scheme (unicast). It furthermore does not require QoS re-negotiation as long as session parameters (e.g. codec) keep unchanged.

4.2.5 Content stream switch (multicast to multicast)

The same mechanism as described above is applicable for the content stream switching within multicast delivery. Here

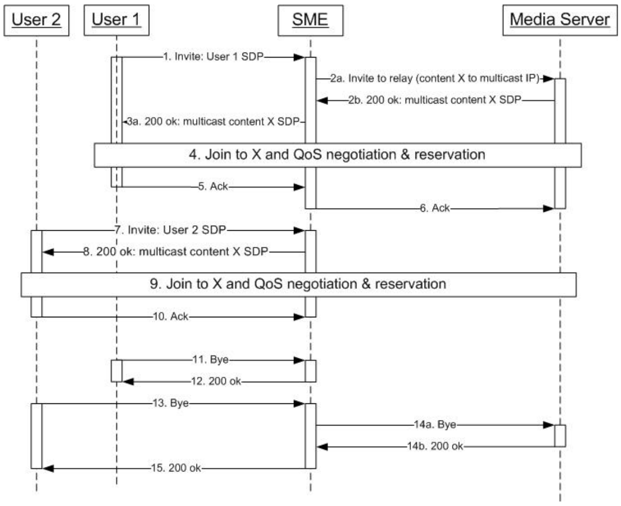
User 2
User 1
SME
Media Server
1. Invite: User 1 SDP
2a. Invite to relay (content X to multicast IP)
2b. 200 ok: multicast content X SDP
3a. 200 ok: multicast content X SDP
4. Join to X and QoS negotiation & reservation
5. Ack
6. Ack
7. Invite: User 2 SDP
8. 200 ok: multicast content X SDP
9. Join to X and QoS negotiation & reservation
10. Ack
11. Bye
12. 200 ok
13. Bye
14a. Bye
14b. 200 ok
15. 200 ok

Figure 4: Content stream request (multicast delivery)

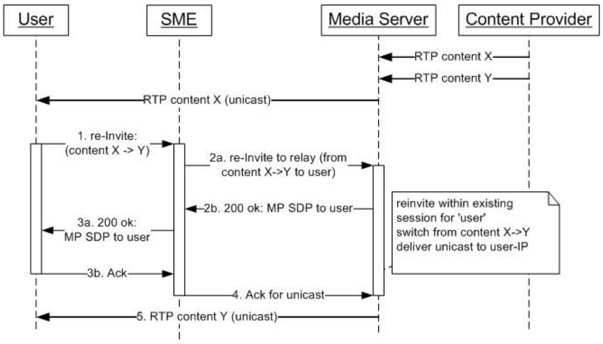
User
SME
Media Server
Content Provider
RTP content X
RTP content Y
RTP content X (unicast)
1. re-Invite: (content X -> Y)
2a. re-Invite to relay (from content X->Y to user)
2b. 200 ok: MP SDP to user
reinvite within existing session for 'user' switch from content X->Y deliver unicast to user-IP
3a. 200 ok: MP SDP to user
3b. Ack
4. Ack for unicast
5. RTP content Y (unicast)

Figure 5: Content stream switch (unicast delivery)

the session manager provides the user with a qualified SDP representing the multicast delivery of the requested content. The precondition is that the media server already delivers the requested content via multicast. So the user has to IGMP leave the old content stream and IGMP join the new requested one [see figure 6].

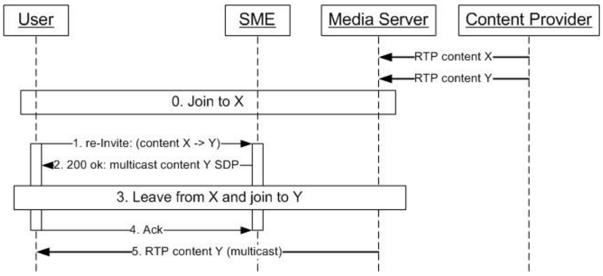
User
SME
Media Server
Content Provider
RTP content X
RTP content Y
0. Join to X
1. re-Invite: (content X -> Y)
2. 200 ok: multicast content Y SDP
3. Leave from X and join to Y
4. Ack
5. RTP content Y (multicast)

Figure 6: Content stream switch (multicast delivery)

4.2.6 *Content stream switch (unicast to multicast)*

Assuming the session manager decides to switch transmission scheme while switching the content stream, the session manager has to provide the user with a multicast SDP for the requested content. So the user may IGMP join the multicast delivery. Furthermore the session manager needs to finish the unicast delivery between media server and user by sending SIP BYE towards the media server.

Figure 7 gives an exemplary call flow. It is obvious that the synchronization of the temporary parallel multicast and unicast delivery must be handled, probably by the user's client.

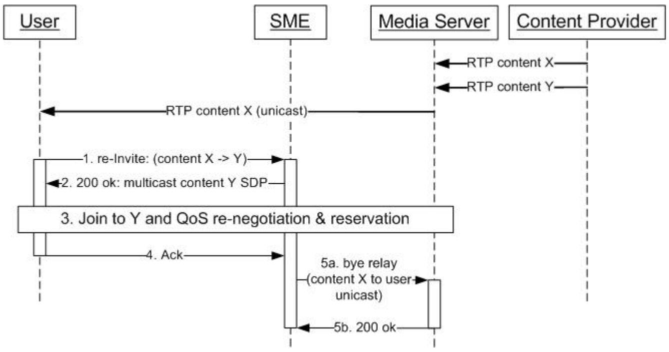
User
SME
Media Server
Content Provider
RTP content X
RTP content Y
RTP content X (unicast)
1. re-Invite: (content X -> Y)
2. 200 ok: multicast content Y SDP
3. Join to Y and QoS re-negotiation & reservation
4. Ack
5a. bye relay (content X to user unicast)
5b. 200 ok

Figure 7: Content stream switch (unicast to multicast delivery)

4.2.7 *Content stream switch (multicast to unicast)*

The content stream switching with simultaneous changing of the transmission scheme from multicast to unicast invoked by the session manager needs the session manager to create a new SIP session between session manager and media server. This SIP session represents the unicast delivery of the requested content stream to the user. After receiving a response with the necessary information from the media server (within the media server's SDP), the user invokes IGMP leave of multicast delivery and prepares for receiving a unicast stream [see figure 8].

The temporary gap between leaving multicast reception and receiving the unicast stream must probably be synchronized by the user's client to ensure seamless content stream reception. The QoS negotiation is started by the MS for this unicast transmission.

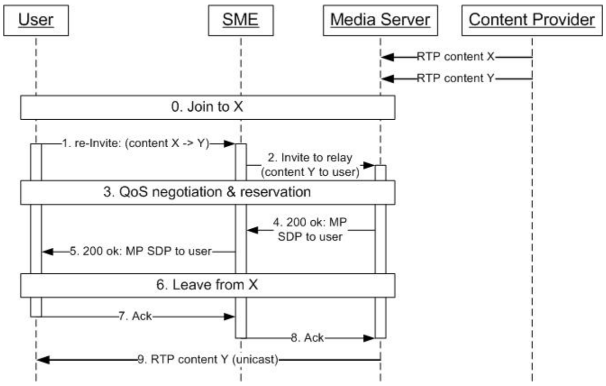
User
SME
Media Server
Content Provider
RTP content X
RTP content Y
0. Join to X
1. re-Invite: (content X -> Y)
2. Invite to relay (content Y to user)
3. QoS negotiation & reservation
4. 200 ok: MP SDP to user
5. 200 ok: MP SDP to user
6. Leave from X
7. Ack
8. Ack
9. RTP content Y (unicast)

Figure 8: Content stream switch (multicast to unicast delivery)

4.3 IPTV signaling with SIP and RTSP in cooperation

4.3.1 SIP and RTSP (primitive)

The primitive approach of using SIP and RTSP regarding activation of media streaming is using SIP for session signaling and RTSP for media signaling. The consumer passes a content resource id - in the following called crid - to the session manager instance which further resolves a dedicated media server providing the requested media stream and establishes a logical session between user and media server in which the session manager acts as B2BUA. The media server responds with a media resource link (mrl). This link is necessary for the consumer to establish the media signaling based on RTSP. The session is finished by RTSP and SIP separated. Here the strict separation of session signaling and media signaling causes unnecessary signaling messages as the media setup could be processed by SIP as well as the next section depicts.

4.3.2 SIP for session signaling, media setup; RTSP for media signaling

Activation of media streaming with SIP for session signaling **and** media setup and RTSP for media signaling is a step further. For media control on behalf of the user, a connection between user and RTSP media server has to be established. This connection may be either connection-oriented (TCP) or connectionless (UDP). Furthermore this TCP connection can be persistent (one communication channel for all subsequent RTSP messages) or non-persistent (one communication channel per request / response). In both cases the RTSP session is decoupled from the underlayed transport session and must be closed at RTSP session level. The persistent TCP connection must never be closed unless the TCP session times out (in cases the client does not reply anymore). The advantage of TCP connection-based connections is the reliability. The advantage of persistent versus non-persistent TCP connection approach is the smaller response time and smaller signaling delay as the persistent connection has not to be established each time. The disadvantage of using a persistent TCP connection is a possible run out of available connections in case of many parallel users triggering the MS.

The approach introduced here replaces the RTSP session establishment (RTSP SETUP and RTSP DESCRIBE) by using SIP. Within RTSP SETUP the client informs the MS about the transport mechanism. In our approach, the user passes its own IP and its port and a content resource id to the session manager. The session manager triggers a qualified media server with this information. The media server allocates the requested resources and allocates connection resources between user and media server based on the user IP and user port and returns a session id and the media server's IP and port. Based on this information, the user establishes a TCP connection with the MS and invokes the allocated content stream using RTSP. The session ends by sending SIP BYE message which frees resources on the media server and removes the TCP connection between user and media server.

This approach requires further research activities in the field of session hijacking as an eavesdropper can hijack the RTSP session when picking the RTSP session id. Spoofing techniques impede effective solutions and complicate quick relief.

4.3.3 RTSP Gateway

A different approach is the logical separation of the SIP scope and the RTSP scope by using SIP-to-RTSP gateway in between. Here, the activation of media streaming results from SIP for session signaling and media signaling only but a conversion of SIP media signaling into RTSP messages by a dedicated SIP-to-RTSP gateway is necessary. In this case the MS behaves like an ordinary VoD server and does not need SIP skills.

5. END-TO-END MEASUREMENT AND EVALUATION

Within this paper, the end-to-end measurement has been arranged for the SIP approach only. Thus, all necessary signaling from the user to the media server via the session manager operates within IMS over SIP. The test environment is embedded in the FOKUS Open IMS Playground [8], whereas several applications are deployed and used by multiple users. The testbed provides various access networks (UMTS, DVB-H, WiFi) with different transmission mode (uniast, multicast, broadcast). The SME was hosted at SIP Servlet Execution Environment (SIPSEE) AS [7] and is integrated in the FOKUS Media Interoperability Lab [9]. SIPSEE is the SIP Servlet container implement as core of FOKUS SIP Application Server. SIPSEE is a full implementation of SIP Servlet API 1.0 (JSR116) and provides support for HTTP/SIP converged application. As a SIP Application Server of FOKUS Open IMS project, SIPSEE is designed and implemented as an IMS compatible Application Server from very begin. The UE and the media server are simulated by SIPp [10] as the media server processing and decoding time depends on the implementation and would distort the performance measurement regarding the session management signaling delay. The signaling does neither include QoS negotiation or reservation nor IGMP join and buffering. The results reflect end-to-end measurement only.

Concerning the Live TV request scenario proposed in this paper the switching delay might be interesting as well. However, the switching delay is logically comparable to the total signaling delay; therefore it is not mentioned explicitly. The collection of data for signaling delay bases on the approach of section 4.2.2 in case of unicast transmission and on the approach of section 4.2.3 in case of multicast transmission. In turn the multicast transmission can be divided into two steps:

1. the session management already knows the multicast IP for the requested content ⇔ the media server delivers the content already via multicast

2. the session manager has to invoke multicast delivery of the requested content towards the media server

The second approach does (concerning the call flow) not differ from the invocation of a unicast delivery. So it is not explicitly mentioned here.

5.1 Values

All results base on the statistical average (μ) of 20 record sets. The standard deviation (σ) is declared additionally. To exclude statistical outliers, the median ($\bar{x}$) of the values

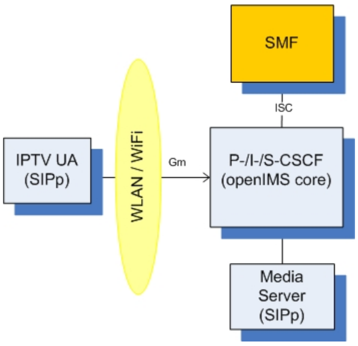

Figure 9: End-to-end measurement architecture

Access	μ in ms	σ in ms	$\bar{x}$ in ms
LAN	76	7	77
WiFi	84	30	35

Table 1: User request - total signaling delay (content stream invocation included

is given likewise. The measurement was executed via two different access networks (LAN and WiFi).

5.2 Evaluation

Considering table 1 first, evidently the difference in signaling delay concerning the access network (LAN, WiFi) is not significant here. Probably as the traffic within the LAN and on the WiFi access point were low. Average, standard deviation as well as median identify the values are quite invariant and plausible. However, the values refer only to the pure session and media signaling concerning the content stream request but not the content stream delivery in particular. Thus, they do not state the propagation delay of the RTP stream and the processing time within the media server (as already mentioned, here the media server is simulated by SIPp). It cannot be researched here, if the signaling delay is in the range of a convenient delay for the consumers. We lean on ITU-T Recommendation G.114 [11] and adopt its recommendations to the issue of signaling delay researched in this paper. [11] defines the level of acceptance for mouth-to-ear delay as not to exceed 200-280 ms to keep all users very satisfied or satisfied. Within the Live TV scenario described above [see section 4.1], most of the content stream delivery is carried on multicast transmission. The values of table 2 reflect such a multicast transmission signaling. Average, standard deviation and median are identified as quite invariant values out of this record set. The signaling delay is significant smaller than in the first case, where the content stream had to be activated toward the media server. However the delay time values does not include either multicast join (via IGMP) nor the RTP content stream delivery. The values does reflect the signaling delay concerning session and media signaling in general. Future research should delivery concrete values for enhanced measurements concerning the propagation delay for RTP content stream delivery.

Access	μ in ms	σ in ms	$\bar{x}$ in ms
LAN	95	12	101
WiFi	100	15	105

Table 2: User request - total signaling delay (multicast only)

6. CONCLUSIONS

The IMS and a dedicated IPTV signaling framework are a possible approach for the delivery of enhanced multimedia streaming scenarios like the Live TV scenario proposed in this paper. The session and media signaling delay are, regarding the end-to-end measurement values, in a range which makes future research basing on the approach IPTV via IMS applicable. It will be essentially how to integrate RTSP abilities (mainly in case of video-on-demand scenarios) in the SIP and IMS world. It furthermore depends on a efficient media server and a slick multicast delivery and switching mechanism as this will be the chief ingredient of a IPTV Live TV scenario. Beyond the basic delivery framework, more complex questions arise regarding a decision mechanism when to change the transmission scheme between unicast, multicast and broadcast.

7. REFERENCES

[1] IP Multimedia Subsystem (IMS); stage 2. *3GPP TS 23.228*, Release 7 2006.

[2] C. Riede, A. Al-Hezmi, T. Magedanz. Quadruple Play - Session Management Enabler for Multimedia Streaming. *16th IST Mobile & Wireless Communications Summit, Budapest, Hungary, July 1-5, 2007.*

[3] T. Cagenius, A. Fasbender, J. Hjelm, U. Horn, I.M. Ivars, N. Selberg. Evolving the TV experience: Anytime, anywhere, any device. *Ericsson Review, no. 03, 2006.*

[4] B. Chatras, M. Said. Delivering Quadruple Play with IPTV over IMS. *International Conference on Intelligence in Networks (ICIN) 2007, Bordeaux, France.*

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

[6] H. Schulzrinne, A. Rao, R. Lanphier. Real Time Streaming Protocol (RTSP). *IETF RFC 2326*, April 1998.

[7] K. Knuettel, T. Magedanz, L. Xie. SIP Servlet Execution Environment (SIPSee) - An approved IMS SIP Application Server for Converged Applications. *International Conference on Intelligence in Networks (ICIN) 2006, Bordeaux, France*, pages 147–152, May 2006.

[8] FOKUS Open IMS Playground. http://www.fokus.fraunhofer.de/ims

[9] FOKUS Media Interoperability Lab. http://www.mediainteroperabilitylab.org

[10] *SIPp*. http://sipp.sourceforge.net.

[11] ITU-T. One-way transmission time. *ITU-T Recommendation G.114*, May 2003.