

Delivering a personalised video service using IPTV

Zelalem S. Shibeshi¹, Shange Ndakunda², Alfredo Terzoli³, Karen Bradshaw³

Department of Computer Science

Rhodes University, P. O. Box 94, Grahamstown 6140

Tel: +27 46 6038247, Fax: +27 46 6361915

zelalems@rucus.ru.ac.za¹; g05n5057@campus.ru.ac.za²; {A.Terzoli, K.Bradshaw}@ru.ac.za³

Abstract— Video oriented services are taking the lead in terms of revenue in the current Internet environment. In order to tap this revenue, Telcos are resorting to serious measures such as moving to Next Generation Network (NGN) where they can provide QoS for services that demand this feature, such as, IPTV. IPTV is more than just what its name implies and includes various types of additional services to make the TV a better entertainment and communication environment. Various innovative IPTV services have been proposed and developed by researchers and practitioners, including personalized advertisement, personalized EPG (Electronic Program Guide), and personalized VoD (Video on Demand). In this paper we introduce a new IPTV service called, the personalized dynamic video delivery service, which sends new video alerts (like breaking news alert) to subscribed users and notifies them when the video is available. If desired, users can watch the video automatically by switching from what they are currently watching without requiring explicitly a new video setup. Users can configure their profile and also set a policy on the way they wish to be notified and the types of video they wish to be alerted to. The paper describes the issues surrounding the delivery of this service and presents the architecture and the various open source technologies that are used to develop the service.

Keywords— IPTV, Personalize video services, Dynamic video delivery, Mobicents

I. INTRODUCTION

The development of digital video has led to the introduction of high quality video and various innovative video services. But the mushrooming video oriented services on the Internet couldn't meet the demand of QoS that most multimedia services require. This was an opportunity for Telcos to come up with an infrastructure that provide QoS for the services that demand it. Accordingly, research on telecommunication technologies lead to the development of the Next Generation Network (NGN) which paved the way to deliver interactive multimedia services by providing the necessary QoS.

Although there are many competing visions of how service delivery can be achieved in NGN, one the framework that is currently gaining considerable momentum is the IP Multimedia System framework (IMS) [1]. Another service development environment that is the focus of much attention in the research community is Mobicents [2]. In this paper we used both platforms for the demonstration of our service. The

introduction of NGN and other telecommunications technologies created a platform to easily deliver bandwidth hungry multimedia services, like IPTV. As specified in [3], IPTV are multimedia services delivered over IP based networks managed to support the required level of QoS/QoE (Quality of Service/Experience), security, interactivity and reliability. As such IPTV has opened opportunities to service developers to come up with ideas for various innovative services unlike the traditional Linear TV delivery.

There are different proposals of implementation standards for IPTV by different standard bodies. The major IMS standard bodies are: the 3rd Generation Partnership Project (3GPP) [4], European Telecommunications Standards Institute (ETSI) Telecommunications and Internet converged Services and Protocols for Advanced Networking European Telecommunications Standards Institute (TISPAN) [5], and the Telecommunication Standardisation Sector of International Telecommunications Union (ITU-T) [6]. However, the specification that has received the greatest emphasis for the development and testing of IPTV service is the one proposed by ETSI-TISPAN.

Multimedia session delivery involves the use of a session control protocol to control the session and a media control protocol to control the media delivery. As per the TISPAN IPTV specification, the media delivery for IPTV is carried out by a component called the IPTV Media Function (MF). The control unit of the IPTV MF is the MF Controller (MFC) and the delivery unit is the MF Delivery (MFD). The media delivery unit is supposed to be implemented by a fully-fledged media sever. However, because of lack of free and open source media servers, researchers mostly use open source streaming servers to develop and test media services. Nevertheless, the major open source streaming servers don't allow the delivery of media to a different place other than the initiator of the streaming session. As a result, a work around is needed to solve this problem. We will explain later how we tackle this problem.

In addition to the media delivery and control unit, IMS services, like IPTV, are controlled by a service controller unit, which in the case of IPTV is the IPTV Service Control Function (SCF), also called the Application Server (AS). So, if a user knows about the service description of an IPTV service, s/he contacts the SCF to obtain the desired service. The SCF, in turn, will contact the MFC to initiate the delivery

of media. The MCF then initiates the media delivery by instructing the MDF to send the stream directly to the user. In general, the MCF initiates the media request on behalf of the user and the media server (MDF) delivers the stream directly to the user.

From the above brief description of the architecture IPTV, the User Equipment (UE) initiates the IPTV service and should in advance know about the service description for the specific service that the user is interested in. However, this is not always the case. There are cases where the service controller, based on certain condition, wants to change the media during the media delivery process and it also needs to be done dynamically. The architecture described above does not have a way to deliver such a service. As a result, an investigation is done to see how such a service can be implemented and presented in this paper.

The paper is organised as follows: Section 2 gives background information, Section 3 describes related works, Section 4 explains the proposed architecture suitable for the delivery of such an IPTV service, Section 5 discusses implementation, Section 6 presents discussion points and finally Section 7 presents conclusions and future work.

II. BACKGROUND

We use the IPTV architecture proposed by ETSI-TISPAN to present and discuss the service. The major components of this architecture specific to IPTV are: Service Discovery and Selection, Service Controller, and Media Control and Delivery. Figure 1 shows these functional units of the IMS-based IPTV architecture as proposed by the ETSI-TISPAN standards body. The next sections discuss the different functions of the service.

A. The IPTV Media Setup

For the purpose of describing an IPTV service, we will refer the IPTV specification specified in [9]. As per the specification there are two types of media setup for IPTV services, referred to as Method 1 and Method 2. The distinction relates to where the streaming session setup commands (RTSP) are initiated from. In the Method 1 session setup technique, the media session is initiated and the setup originates from the MFC (the media controller). Whereas, in Method 2, it is initiated from the UE, but again the UE sends all RTSP commands to the MFC. Basically the UE, after identifying and selecting the service, sends the request to the IPTV service control function (SCF).

The SCF then performs service authorisation and credit control, selects the relevant IPTV Media Function, and forwards the request to the MCF which is responsible for controlling the media.

The MCF is responsible for initiating the media session by contacting the MDF that is supposed to serve the user. Once the media is setup correctly, the MCF notifies the UE of the status of the media session, which will then send the RTSP "PLAY" command to the MCF, to start the media delivery.

The MCF accordingly will forward the request to the MDF and the delivery of media then starts from the MDF to the UE.

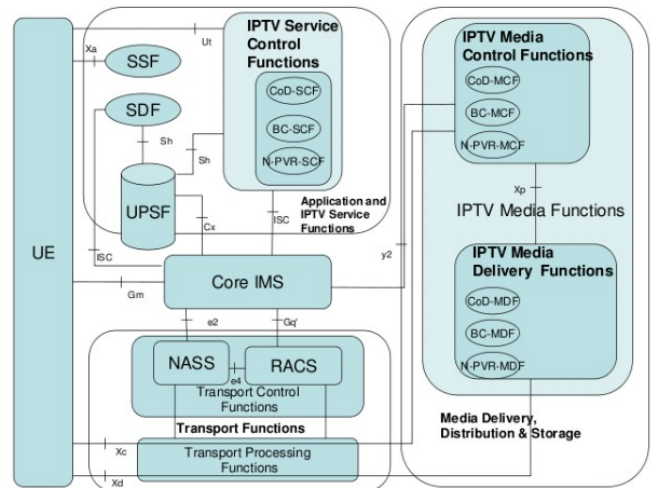


Figure 1: Functional architecture of IPTV

As mentioned previously the delivery of media is done using streaming servers. However, this makes it difficult to have a separate media control unit for IPTV service in IMS, if one wants to use a streaming server as an MDF unit. For this reason, researchers tend to combine the MFC and MDF units as one unit and initiate the media delivery and control from the UE. Another approach to use streaming servers as MDF is by using them together with an RTSP proxy. The advantage with this particular approach is the media can also be changed easily if needed. The Convergence Research Group of Rhodes University is using this approach to develop and test IPTV services. As it will be explained later, this approach can also help implement the media switching functionality easily.

The media control protocol that the MCF uses to control the MDF is not specified in the specification and it is up to the implementers to choose an appropriate protocol. In our case we used the RTSP protocol as a media control protocol. The protocol has different parameters that are helpful to extend the server's functionalities.

B. Channel Switching

Channel Switching is the ability for the client to choose a different channel to watch or listen to and have the player on the client's machine receive new channel without losing time on starting a new connection and buffering of the new stream [10]. In a similar way to the media setup mentioned above, channel switching can also be initiated either from the UE or the AS. Based on a certain criteria the AS can change the media without the involvement of the client. If we look at the IPTV specification, the only way that one can use to implement channel switching in IPTV may be through the IPTV Service Discovery and Selection function. However, this can not be possible, because this function is used at boot time and it is inapplicable for the dynamic media switching service.

There are many use cases for dynamic media switching. One interesting use case is advertising. Instead of stopping the channel and setting up the advertisement, the AS, while the user is watching the channel, can setup the advertisement and without any interruption switch to it when the advertisement is ready. Another use case that requires media switching is the dynamic insertion of new media (video feed) within an existing session. If, for example, the user wants to be notified when a breaking news is available or a goal is scored at a certain match, the AS can notify the user of these events in a friendly manner (e.g. with a banner displayed on top of the video that he/she is watching) and if the user wants to get the media, he can notify the AS by using the remote control and the AS can then setup (queue) the new media in a similar manner described above. When the new media is ready, the AS stops the existing media and switches to the new media and the user starts to watch the new media immediately.

From the media switching point of view, as far as we know there is no protocol that supports media switching directly. As we are already using RTSP protocol to control the media, we have also used this protocol to initiate the media switch request. This aspect will be discussed in detail in the Implementation section.

C. Event Notification

For event state signalling, the IETF has defined an event package framework. This is essentially an extensible framework by which SIP nodes can request notification from servers when specific events occur [11]. Event packages are based upon the SIP PUBLISH, SUBSCRIBE, NOTIFY architecture to manage events for application specific data contained in the bodies of the SIP messages. For example, the Presence, Location, and Conference event packages extend this package to share presence, location and media conferencing information respectively. The nodes on the network that publishes requests are called the Event State Publisher (ESP); the Event State Subscribers (ESSs), on the other hand, are the ones that send subscription requests to get notification; and the State Agent, which is an intermediary Back to Back User Agent (B2BUA) between the ESP and the ESS, is used for mediating the ESPs and ESSs. In this paper we propose a new SIP event package that extends the SIP event package framework for signalling dynamic video information and is explained in the Implementation section.

D. User Document Management

Presence privacy policies and buddy list documents belong to ESPs. These documents are kept at the XML Document Server (XDMS). The policies are later acquired from the XDMS by the State Agent to deliver appropriate content to the ESSs. Resource Lists are used by the State Agent to aggregate Subscription and Notifications information from different publishers for one ESS. The communications protocol used by the entities responsible for rule making is called XML Configuration Access Protocol (XCAP) [12]. This protocol

provides the means for XCAP clients to perform Creation, Reading, Updating and Deletion (CRUD) operations on their documents located at the server. The protocol is based on HTTP and XPATH and uses Uniform Resource Identifiers (URIs) to identify XML documents, elements, attributes and values in the document for the purpose of CRUD. The XCAP server sits between the XDM server and the XCAP clients and allows the two to communicate the CRUD operations while still ensuring integrity. As specified in the specification different application domains require their own application usages to manage their XML documents. In this paper we propose a new application usage for the management of user preference documents for dynamic video delivery.

III. RELATED WORK

The RTSP proxy that we used together with a streaming server for the purpose of MDF is described in [13]. The paper showed that the introduction of the Proxy doesn't bring much delay in the overall service delivery process and as such showed that it is an acceptable solution to use streaming servers as MDF. The paper also explained how additional functionalities can be implemented by using the Proxy. In fact, the work presented in this paper is also motivated by the recommendations of this paper.

Most of the work related to channel switching relates to the issue of channel changing initiated from the UE (also for broadcasting services). They all try to find the best way to reduce the delay related to joining/leaving a certain channel. In this regard, [14] defined two techniques called tune-ins stream and tune-in servers to reduce the delay. Reference [10] on the other hand, showed how one can use HTTP and RTSP protocols to switch a channel using the same session with the client. The channel switch however, is again initiated from the client and the paper doesn't describe how it is done.

With regard to event subscription and notification, numerous work has been done to extend the SIP event package framework for different event types using presence and location objects which are described in [15] and [16]. In this regard, [17] implements a SIP Location Enabler in the Service Layer of the IMS framework. The implementation has two prototype services namely the location-based push-to-talk service and location based messaging service. Reference [18], on the other hand, uses the Mobicents framework for location information signalling via the SIP Event Package.

IV. ARCHITECTURE OF THE SYSTEM

The architecture that we propose is basically the same architecture specified by ETSI-TISPAN IPTV architecture with the addition of the Proxy in the MF and the two components responsible for new video alert and profile document management. Figure 2 show the block diagram of our architecture. The block diagram shows how our architecture fits into the different standard components required to provide personalised dynamic video feed in an NGN environment.

As can be seen from Fig. 2, the major components required for this service are Dynamic Video Notification Server (DVNS), the XCAP/XDMS server, the IPTV AS and the MF. For simplicity purpose we present the Streaming Server and the RTSP Proxy, detailed in [13], as MFD in the figure. The following paragraphs describe the functionalities of these components.

As discussed in the background section, the content providers in this service are the ESPs that publish information about the random video events using SIP PUBLISH request. The DVNS is the State Agent that receives the published information and composes it with older publications in a consistent manner and send it to the AS. The content of the publication request, an XML document, is carried as a SIP payload. A sample of the custom XML document designed for this purpose is shown in the next section. When new video is published on DVNS, the server also notifies the IPTV AS using the SIP NOTIFY request, which again contacts the MFC to initiate the media switching. ASs can also specify their subscriptions to different content providers as resource lists. In such a case the notification information is composed from all the content providers' (publishers from the resource list) publications.

The UE use the XCAP protocol to make its event notification interests known to the XDMS via the XCAP server. The CRUD operations can be invoked here by the UE to keep the preferences up to date. On the other hand, when the AS receives an INVITE request for IPTV session setup, it also fetches the user's preferences from the XDMS server again using XCAP protocol and later when it get the new video alert decides whether to forward the request to the MFC to initiate the media switch or not.

V. IMPLEMENTATION

We used various open source tools to develop and test the work presented in this paper. In this section we will describe the different components and specify how the new video alerting and media switching functionalities are implemented. The following sections briefly describe these aspects:

A. The Open IMS Core testbed

We used the Open IMS Core testbed from the Fraunhofer Institute FOKUS, as an IMS core. It is used for registering the client and initiating the IPTV service.

B. IMS Client

With regard to IMS User Agent, we used the UCT IMS client, which is the most popular open source IMS client in the research community. The client is designed to work with Method 2 of the IPTV media access method and also has an XCAP client in it.

C. The Mobicents Development Environment

As discussed previously, we have used Mobicents as an NGN service deployment environment to implement part of

the Application and IPTV Service Function component of Fig 1. Specifically the AS and the user profile management components are developed and deployed using Mobicents. It is a Java-based service delivery platform for the quick and easy development, deployment and management of next-generation network applications [7]. It is JAIN-SLEE 1.0 compliant, and adopts data from various IP protocols for use as Java objects. This functionality is achieved through JAIN-SLEE components called Resource Adaptors (RA) [8]. Some of the resources adaptors in Mobicents include SIP, HTTP and Diameter RAs. For our purpose the SIP and HTTP RA are particularly important. Mobicents also uses the JAIN-SLEE units known as the Service Building Blocks (SBBs) for the composition and building of services. SBBs provide interfaces that facilitate their extension and reuse by other services. The ASs are developed as SBBs.

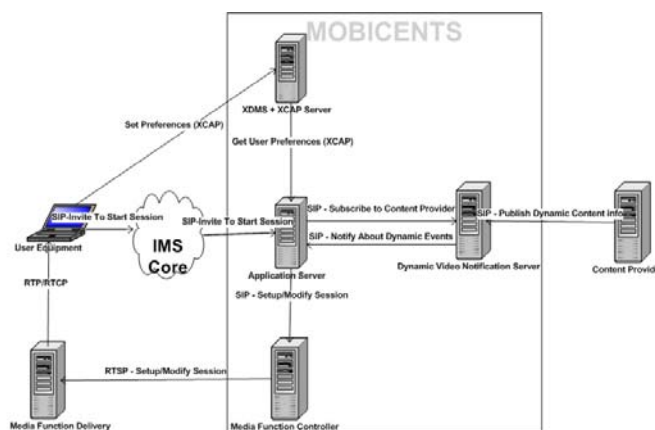


Figure 2: Architecture of the system

D. New media alert

The new media alert is implemented using the SIP event package framework mentioned in the previous section. For our purpose a custom event with different headers was defined. An application usage and preference documents were also defined for the purpose of specifying users' preferences. The following sub sections briefly describes how these functionalities are implemented.

1) *The Event Package Headers and DVN Document.* As it is specified in the event package framework specification [19], custom event packages that extend the package must specify their event type in the event-type header and content type in the content-type header. The event type we designed for our service is called dvn (dynamic video notification). The content-type header is set to application/dvn+xml. The events on which notifications are provided are random, but there is still a need for an expires header to indicate that whether or not the ESP at the Content Provider is alive. All the other SIP headers and behaviours specified at [11] still remain the same in our event package. The rest of the information is specified in an XML document, which we called it *DVN Document*, and is carried as part of the SIP PUBLISH, SUBSCRIBE and

NOTIFY payloads. The following is a sample DVN document.

```
<dv n id="sip:convergence@ru.ac.za">
  <media type="sports">
    <event type="1" time="2010-07-21T09:44:00" ttl="3600">
      <description> Goal: Inter Millan have scored a goal in the UEFA
        Quarter Final game the score is now Inter millan 2 - 1
        Real Madrid
      </description>
      <link>rtsp://convergence.ru.ac.za:554/uefa</link>
    </event>
  </media>
  <media type="news">
    <event type="1" time="2010-07-21T09:46:00" ttl="7200">
      <description> Breaking news: 39 Chille miners rescued
        successfully after 69 days trapped underground</description>
      <link>rtsp://convergence.ru.ac.za:554/news1</link>
    </event>
    <event type="2" time="2010-07-21T09:46:00" ttl="6500">
      <description> 39 Chille miners rescued
        successfully after 69 days trapped underground</description>
      <link>rtsp://convergence.ru.ac.za:554/news2</link>
    </event>
  </media>
</dv n>
```

Extract 1: Sample DVN Document

The root *dv n* element contains an *id* attribute which takes the name of the content provider, in our case, it is sip:convergence@ru.ac.za. This element can have one or more *media* elements. The media elements describe events related to a type of video media such as *sports* or *news*.

Multiple *event* elements can be contained within the *media* elements describing the type of event that has occurred. The *type* attribute of the *event* element takes an integer value regarding the type of event in the context of the video media type. For instance, in the context of sports, a *type* value of 1 signals a goal in soccer or hockey; a try in rugby and so on. A 2 may indicate a secondary event such as a penalty, a red card and suchlike. The *time* attribute takes the time at which the event was published, or when it started happening. The *ttl* is the time-to-live field and it indicates in seconds for how long the random event is relevant. If this time passes, the event is discarded.

Finally, the *description* element contains human-readable information about the event which eventually be displayed to the user by the MDF.

2) *The DVN Application Usage and User Preference Document.* As specified before, an application usage (appusage) is required to specify the user preferences regarding the dynamic video event notification. We name our appusage as dv n-services and this is what we use in our XCAP URIs to perform CRUD operations on DVN preference documents (which we call devprefs) from the XCAP server. The content type for the devprefs requests is application/dv n+xml. Every user will have his/her own preference document and so their SIP identity needs to be included in the URI as per XCAP specifications. For instance we invoke the URI *http://convergence.ru.ac.za:8080/mobicents/dv n-services/users/sip:shange@convergence.ru.ac.za/index* to add, modify and delete the entire dv n

preferences document for user *shange@convergence.ru.ac.za*. We use the digest authentication method to ensure the CRUD operations are secure. Extract 2 shows a sample user preference document.

```
<dv nprefs>
  <media type="sports">
    <event type="1" expires="2010-07-21T10:00:00" />
    <event type="2" expires="2010-07-21T11:00:00" />
    <publishers>
      <publisher id="sip:convergence@ru.ac.za">
      <publisher id="sip:sports.convergence@ru.ac.za">
    </publishers>
  </media>
  <media type="news">
    <event type="1" expires="2010-07-21T11:00:00" />
    <publishers>
      <publisher id="sip:news.convergence.ru.ac.za">
    </publishers>
  </media>
</dv nprefs>
```

Extract 2: Sample User Preference Document

In the preferences document, shown above, the user specifies the video media type that he/she is interested in using the *type* attribute in the *media* element. Multiple *event* elements can also be specified within the *media* elements. Each one specifies the type of event which is identified by a number. A date and time of expiry information is also included to indicate the time frame that a particular type of event is relevant. Optionally, the user can specify from which publisher he/she wants to get the video event notifications.

The AS can also retrieve this document using the XCAP interface from the XDM Server.

The Mobicents Service Delivery Platform (SDP) includes the presence service and a pres-rules *appusage* (for presence policy). The presence service runs on top of a sip-event package (an implementation of the SIP Event Package Framework which is especially crafted to allow for extensions). We also extended the sip-event package to develop a publication service called the Dv nPublicationSBB, which is a Mobicents SBB. This SBB handles only the headers and the content that are specific to the DVN package. The sip-event package handles the rest for us. We use JAXB (Java Architecture for XML Binding) together with our DVN document schema for marshalling and unmarshalling of *dv n* documents. We also put together a Dv nSubscriptionSBB service that serves the incoming SIP SUBSCRIBE requests and sends SIP NOTIFICATION requests using the JAXB context. The DVN *appusage* is added to the Mobicents XDM server following the process discussed in [7]

We used the libvlc media player as MDF and used the *mediacontrol_display_text* function to display the new video alert as a banner on the client side.

E. Media switching

Media switching involves the communication between the MFC and the Proxy (which passes the request to MDF). When the MFC receives the notifications of the new media, it requests the MDF to send the notification as a banner to notify

the user about the occurrence of the event and request if he/she want to switch to the new media. If the user shows interest, then the media switching follows. The media switching is implemented as follows.

The Proxy has different handlers on the client and server side for both the RTSP and RTP/RTCP sessions. On the client side, the RTSP session is created with the MFC and the RTP/RTCP session is created with the client (UE). So when a media switch request arrives from the MFC, the Proxy will create a different RTSP and RTP/RTCP session for the new media on the server side and sends/receives the RTSP requests/responses by itself. When the new media setup is finished it uses the same RTP/RTCP session on the client side to deliver the media. With regard to sending the media switch command to the Proxy, in order to minimise communication between the MFC and the Proxy, we have extended the RTSP OPTION command by defining fields named “switchto”, “session”, and “tracks” to send the “media switch” request. The RTSP Proxy is modified to support the OPTION command with these parameters.

The MFC unit is implemented as an SBB and uses an open source SDP API to extract media related information from the INVITE message for use of Method 1 media session setup. It has also RTSP capability.

VI. DISCUSSION

Dynamic video delivery, as specified in this paper, is one of the innovative IPTV services that service providers can provide to their clients. The event notification and subscription aspects of the service are well handled using the SIP event package and XDMS protocols, and easily be implemented in Mobicents. However, there are not enough standards for media modification and we modified the RTSP protocol to demonstrate the service. For this purpose, we propose that the IMS specification needs to include standards to be used for media switching and modification.

As presented in the paper the dynamic video delivery service involves the use of different technologies and protocols. Because the Mobicents/JAIN SLEE environment is designed and optimized for event driven applications, it is an ideal choice to develop and implement such type of service. Service developers only need to concentrate on the service logic and not on implementation of different standards.

VII. CONCLUSION

In this paper we have showed how the IPTV framework can be used to deliver a personalised dynamic video delivery service to IPTV users. Particularly we have showed how one can use open source tools to develop such a service. The different components required to deliver the service are discussed and also the schemas and events used to transfer information between the different components are explained. We hope this will serve as a base for the development of other innovative services.

ACKNOWLEDGEMENT

This work is being carried out in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom, Comverse, Stortech, Tellabs, Amatole Telecom Services, Bright Ideas 39, and THRIP.

REFERENCES

- [1] E. Burger. “Media Services in the IMS: Evolution for Innovation”, Cantata Technology, 2006.
- [2] (2010) Mobicents-Public. Available: URL: <http://groups.google.com/group/mobicents-public/web>
- [3] E. Mikoczy and P. Podhradsky. Evolution of IPTV Architecture and Services towards NGN. Recent Advances in Multimedia Signal Processing and Communications, Studies in Computational Intelligence, 2009, Volume 231/2009, 315-339, DOI: 10.1007/978-3-642-02900-4_13
- [4] (2010) The 3rd Generation Partnership Project. 3GPP. [Online]. Available: <http://www.3gpp.org/>.
- [5] (2010) ETSI Telecommunications, Internet converged Services, and Protocols for Advanced Networking (ETSI-TISPAN). TISPAN. [Online]. Available: <http://www.etsi.org/tispan/>.
- [6] (2010) The Telecommunication Standardization Sector of ITU (ITU-T). Standardisation (ITU-T). [Online]. Available: <http://www.itu.int/ITU-T/>
- [7] Redhat, “Mobicents SIP Presence Service User Guide,” 2009 <http://hudson.jboss.org/hudson/job/MobicentsBooks/lastSuccessfulBuild/artifact/sip-presence/index.html>.
- [8] SUN Microsystems “JAIN-SLEE 1.0 Specification, Final Release”, January 2004.
- [9] ETSI Telecommunications, Internet converged Services, and Protocols for Advanced Networking (ETSI-TISPAN). IMS-based IPTV stage 3 specification. Protocol Specification. TS 183 063, European Telecommunications Standards Institute, December 2009.
- [10] RealNetworks. Fast Channel Switching. RealNetworks White Paper Series. 2008
- [11] (2010) Session Initiation Protocol (SIP)-Specific Event Notification. Available : <http://www.ietf.org/rfc/rfc3265.txt>.
- [12] The Extensible Markup Language (XML) Configuration Access Protocol (XCAP). Available : <http://www.ietf.org/rfc/rfc4825.txt>.
- [13] Z. S. Shibeshi, A. Terzoli, and K. Bradshaw. *Using an RTSP Proxy to implement the IPTV Media Function* via a streaming server. In ICUMT'10: Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems. Moscow, Russia, 18 to 20 October 2010
- [14] H. Fuchs and N. Färber. Optimizing channel change time in IPTV applications. Broadband Multimedia Systems and Broadcasting, 2008 IEEE International Symposium on. March 31 2008-April 2 2008.
- [15] (2010) A Presence-based GEOPRIV Location Object Format. <http://www.ietf.org/rfc/rfc4119.txt>.
- [16] (2010) Presence Information Data Format (PIDF). <http://www.ietf.org/rfc/rfc3863.txt>.
- [17] M. Mosmondor, L. Skorin-Kapov, R. Filjar, and M. Matijasevic” Conveying and handling location information in the IP Multimedia Subsystem”, 2005.
- [18] S. Ndakunda, M. Wright and A. Terzoli. “Adding Location-Based Services Support to the Mobicents Service Delivery Platform”, September 2010
- [19] V. K. Singh, H. Schulzrinne, P. Boni, B. Elman, and D. Kenneson. “Presence Aware Location-Based Service For Managing Mobile Communications”, 2007. Available: <http://sites.google.com/site/geopriv/>.

Mr. Zelalem S. Shibeshi holds an MSc in Information Science, Diploma in Computer Science, and BSc in Physics, all from Addis Ababa University, Ethiopia, and is currently working towards his PhD in the Computer Science Department at Rhodes University.