GSTV: An Integrated, Adaptive and Scalable Digital Multimedia Content Distribution System

Zoran Djuric and Ognjen Joldzic
Faculty of Electrical Engineering, University of Banja Luka, Banja Luka, Bosnia and Herzegovina
Email: {zoran.djuric, ognjen.joldzic}@etfbl.net

***Abstract*—Multimedia content distribution systems, in the first place OTT (Over-the-Top) and IPTV (Internet Protocol Television), have rapidly gained popularity with the widespread expansion of network availability. Although both types of systems employ IP networks to offer the same services, architecturally they are importantly different. There are several reasons why these systems are merging in some kind of hybrid solution, in the last few years. This paper presents an overview of GSTV - an integrated, adaptive and scalable digital multimedia content distribution system. This system is designed to overcome the deficiencies of legacy content distribution systems and to meet all important requirements and recommendations in order to be classified as a modern multimedia content distribution system. Some of the most important features of this system are: support for all IP capable devices that can display the media, support for different video resolutions/formats suitable for all supported devices, quality of service through adaptive bit-rate streaming support, advanced identity management support, and support for customization of informational and advertising messages for end users.**

***Index Terms*—IPTV, OTT, multimedia, streaming, adaptive bit-rate, quality of service, scalability**

I. Introduction

Multimedia content distribution systems have rapidly gained popularity with the widespread expansion of network availability and the recent increase in available network speeds and accompanying technologies [1]-[4]. This expansion has, at the same time, raised the demands in terms of quality of the content distributed to the consumers [4].

One of the most important technical principals behind today's content distribution environments is the convergence of modern networks. Convergence represents a unification of technologies employed in transporting different types of content (voice, video, etc.) over the same underlying infrastructure, thus effectively eliminating the need for networks that are purposely built for a single type of content. Such substantial reduction of costs has certainly contributed to the rapid emergence of content delivery networks in recent period.

Strictly speaking, there are two main types of multimedia content distribution systems: Over-The-Top (OTT) and Internet Protocol Television (IPTV) [4]-[6]. Architecturally these systems are importantly different, although they employ IP networks to offer the same services (live television, video on demand, etc.) to the end users. The main differences between these types of systems are the following. OTT services can be distributed to any connected IP enabled device (globally), while IPTV services are usually distributed only in a closed managed proprietary network to users set-top boxes or computers. Although OTT services are typically provided over an un-managed network (by a third-party, not the network operator), the usage of different adaptive techniques at the end points is resulting in the best possible service quality.

In last few years IPTV and other IP video services (like OTT) are merging in some kind of hybrid solution, where IPTV is starting to support new technologies for delivery, while complementary IP services bring Internet “anywhere” access and Web-based interactivity to IPTV [4], [5]. Some of the most important reasons for this are the increased mobility of users and the increased demand to receive the same services on different devices. A wide variety of platforms, especially mobile platforms, with significantly different device dimensions and hardware capabilities are all factors that must not be overlooked when dealing with multimedia content distribution systems. Therefore, any system for multimedia content distribution must be designed to accommodate any of the client platform specifics, while still providing a satisfactory quality of service expected of such systems.

This paper presents an overview of an integrated, adaptive and scalable digital multimedia content distribution system named GSTV (Global Streaming TeleVision). Section II discusses some of the most important properties of a modern multimedia content distribution system. On the basis of these properties the objectives for the system reported in this paper, GSTV, are created and presented in Section III. The main components and features of the GSTV system, including transcoder, streaming server, video container formats, session controller, web application, web and mobile clients, database, billing and messaging system are described in Section IV.

II. Background

Multimedia distribution systems have evolved from legacy one-way cable television systems (CATV) [7].

Manuscript received October 19, 2013; revised December 6, 2013.

doi: 10.12720/joace.2.4.381-387

The most important difference is their adaptation to two-way communication in order to enable interactive content and session control. Furthermore, the fact that modern video transport protocols use the same communication channels used for standard communication over the Internet (e.g. general purpose application layer protocols, such as HTTP, SMTP, etc.) enables them to use many of the control techniques employed with other standard protocols, thus reducing the cost of distribution system implementation. Modern multimedia streaming systems receive the original video content from a headend device that is usually found in classic CATV systems [8]. This headend device is not part of the multimedia streaming systems itself, but provides the basic source of video content that is distributed to clients after necessary processing.

Modern multimedia content distribution systems are determined by several important properties. Some of the most important are: services which they deliver to users, types of supported devices, accessibility, mobility of users and user experience.

Modern multimedia content distribution systems deliver live television, video on demand, personal video, as well as other services to different types of devices by using IP [2]. These services are not delivered only over dedicated access networks. Instead, they are delivered over any combination of cable, DSL (Digital Subscriber Line), FTTH (Fiber to the Home) or mobile access, regardless of the physical location of end users [4].

The TV set is no longer the dominant device for delivery of multimedia content. PC's, smart mobile phones, tablets and similar smart devices are all becoming increasingly important [9].

Next, the multimedia content should be accessible worldwide, from any location that has Internet connectivity. This implies that modern multimedia content distribution systems should have advanced security capabilities, including support for digital rights management (DRM) and protection of customer privacy, as well as advanced Quality of Service (QoS) capabilities [4].

User experience is also very important in modern multimedia content distribution systems. As content can be personalized, viewing experience no longer conforms to pre-defined broadcast schedules or channels. This personalization requires powerful and interactive graphical user interfaces, as well as adequate information models to capture and manage user preferences [4].

All these facts imply that encoding of multimedia, as well as transport formats must be network agnostic and adaptable to different devices and different access networks capabilities and bandwidth. Also, applications running on these devices and their graphical user interfaces should be interactive and should allow users to customize all important features, like display formats and video quality.

III. GSTV OBJECTIVES

In order to make scalable and secure hybrid multimedia content distribution system we created the list of objectives. This list is based on requirements and recommendations for similar systems [4], [10]. The most important objectives for GSTV multimedia content distribution system are as follows:

- It should support any IP capable device that can display the media, such as Set-Top Box (STB), PC, smart mobile phones and tablets;
- It should provide different video resolutions/formats suitable for all supported devices;
- In order to utilize the best effort IP network effectively, it should support adaptive bit-rate streaming;
- It should have an advanced identity management support allowing the same user to be associated with different devices;
- It should provide information about current and scheduled broadcast;
- It should support customization of informational and advertising messages for individual users, as well as for group of users, based on different criteria. The systems should support broadcast of these messages in real time, as well as scheduled;
- Graphical user interface should be interactive and capable of capturing and managing user preferences;
- Categorization of video content in the VOD service should be supported, as well as searching of content.

IV. GSTV MULTIMEDIA CONTENT DISTRIBUTION SYSTEM

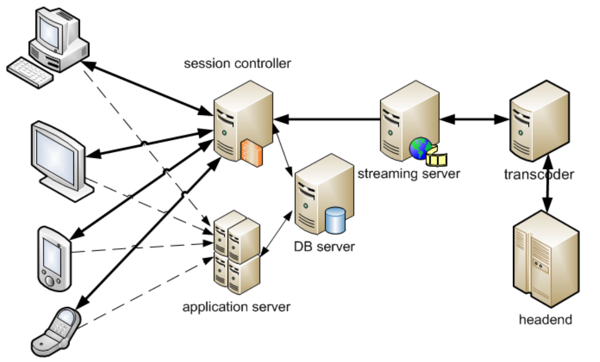

Figure 1. Architecture of GSTV multimedia content distribution system.

A. Transcoder

The first major component of the GSTV system is the part of the system responsible for the conversion of MPEG-2 [11] streams received from the headend into a format more suitable for network transport, called MPEG-4. MPEG-4 [12] is a predictive lossy encoding standard that is capable of providing high-resolution video output, while at the same time preserving the low bandwidth consumption. The process of encoding (or *transcoding*) is performed in real time by a modular component called transcoder (Fig. 1), which can be implemented in one of two ways:

- Software transcoder operates as a software process that produces MPEG-4 output from a MPEG-2 input feed. The main advantage of using a software transcoder is its inherent flexibility and expandability. On the other hand, a software transcoder puts a significant performance strain on the server running the process, which can limit the number of streams that can be concurrently processed. The system presented in this paper uses a software-based transcoder, developed on top of FFMpeg library, which is used in a variety of signal processing solutions [13], [14],
- Hardware transcoder operates as a dedicated hardware device that accepts MPEG-2 streams and converts them to MPEG-4. Hardware implementation is more robust and brings a performance improvement and added stability, but at a significantly higher cost and complexity. The system is designed to enable a seamless transition between different types of transcoders, regardless of their types and features.

MPEG-4 video content is enclosed in a container format that is suitable for transport and delivery to the end users. Most legacy IPTV systems use UDP transport layer protocol coupled with multicast streaming in order to deliver the content to the customers [15]-[18]. These protocols provide a one way distribution path, with only one sender and multiple receivers. As a positive aspect, the utilization of network resources is optimal, the system scales effortlessly to a great number of simultaneous users and sessions, and the overhead of different PDUs is reduced to a minimum, thus minimizing delay. However, this approach requires multicast routing enabled throughout the Internet in order to be able to reach any user connected to their respective networks. Although the IP multicast maximizes utilization of the network resources, deployment by network operators across the Internet has been slow [19]-[21]. It further makes any form of session control virtually impossible, as the source of the transmission is never aware of the number or the identity of its clients. Another important difference is witnessed in the inability of playback control in cases where content is not streamed in real-time, but instead users are allowed to choose the segment of the content they wish to reproduce (Video-on-Demand service). Multicast sources provide the same content to all of their users, who can only opt whether they want to leave the playback session or continue receiving the content at the same rate as all the other users.

Most OTT streaming platforms choose TCP (or a TCP-based protocol) as the transport layer protocol in order to provide a foundation for session control procedures and an added layer of interactivity between the sender and each of the receivers [22]. The slightly bigger TCP segment header results in a bigger total overhead for the entire stream, and a possible increase in delay between the video provided by the headend and the video received by the clients. The security benefits of such an approach are reflected in the ability of the system to identify each of the users connected to the system, regardless of their client network environment and configuration.

B. Streaming Server and Video Container Formats

The video provided by the transcoder is encapsulated into an appropriate container format by a component called a streamer, or streaming server (Fig. 1). The streamer imposes an insignificant processing strain on the server, as no CPU-heavy processing is performed at this point. Usually, the streamer is the first component that interacts with the end-user. As it can be seen in Fig. 1, GSTV system streamer is not directly accessible by end users. Still, this is the segment of the system where the increase in the number of users affects the performance of the streamer. The transcoding process is performed only once for each piece of video content, but the streaming process encapsulates the transcoded content for each of the connected users separately.

Commonly used streaming servers support several video container formats:

- FLV is a popular protocol for video streaming that gained widespread popularity through wide availability of browser plugins (primarily the Adobe Flash plugin) that enable a seamless reproduction of content. This format transports video content encoded using H.263 or H.264 codec, while the audio content is encoded as mp3,
- WebM is an open-source container format used generally for web applications. Its specification is integrated into the HTML5 draft. It primarily uses VP8 for video and Vorbis codec for encoding audio content,
- ASF (Advanced Systems Format) is a proprietary container format developed by Microsoft. Its commercial nature somewhat limits its scope, since it uses WMV and WMA for video and audio encoding, respectively, so the reproduction of such content is limited to Microsoft client technologies,
- HTTP Live streaming (HLS) is another proprietary container format developed by Apple Corporation. It operates by dividing the original stream into smaller segments (called chunks), usually not longer than 10 seconds. At the beginning of the session, the clients request an M3U8 playlist containing the list of chunks that make up the original stream. This playlist can be statically generated (if the streamed content has a finite duration), or dynamically updated by the streaming server in case of real-time video. Each chunk within the playlist is identified by an EXTINF entry containing the address of the chunk (absolute or relative) and its duration. HLS protocol is suitable for mobile clients, since its official implementation employs several adaptive techniques that enable the client to select the appropriate stream quality for the current connection.

The selection of the streaming encapsulation protocol for any distribution system depends on several factors - most importantly, the number of client platforms and the security features built into the protocol design. The GSTV divides the client platforms into two groups, so the selection of the streaming protocol is made according to the characteristics of the client platforms. The first group is comprised of web-based clients used on any of the desktop-oriented systems today supporting Adobe Flash. Support for Adobe Flash is widely available on any of these platforms, and implements all the security features necessary to implement server-based session control. Therefore, all web clients use FLV as their container format, with specially developed client software that is integrated into a Java-based web application (Fig. 2).

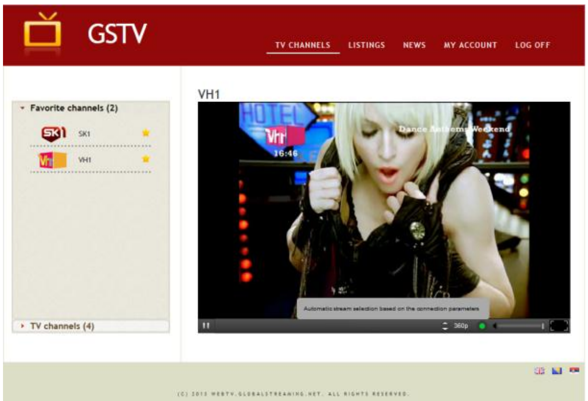

Figure 2. GSTV Web client.

A dedicated web client enables an implementation of additional security mechanisms, as well as an adaptive bitrate switching technique that is compatible with any streaming server. It is important to note that while WebM format with HTML5 would certainly provide a lightweight platform, there is still no single video format that is supported across all major browsers used today [23], and is therefore not supported by any of the major streaming servers.

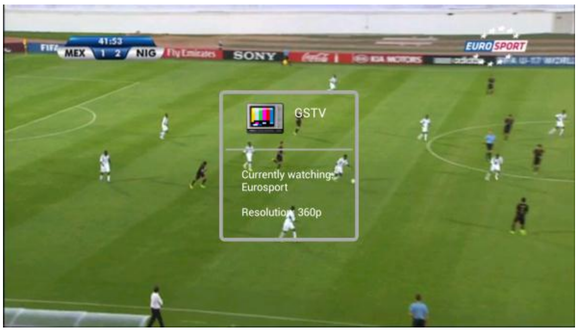

Figure 3. GSTV mobile client - screenshot.

Mobile client applications include the Android and the iOS application (Fig. 3). Both applications are designed without restrictions regarding the dimensions of the device or other hardware characteristics. The adaptive nature of the application itself enables it to even run on older devices with less processing power and memory than the newest generation devices. Applications have been thoroughly tested on both smartphones and tablet devices, with various versions of the Android and iOS operating systems. Several functions have been developed exclusively for the mobile platforms, to make the most out the mobility provided by these types of devices.

Client applications can easily be configured with alerts attached to certain entries in the TV schedule listing. It is important to note that these mobile platforms lack Flash support, since the plugin is deemed a security risk by the majority of mobile operating system manufacturers [24]. One of the container formats supported by all major platforms and operating systems is Http Live Streaming.

The system presented in this paper contains a custom adaptive bitrate selection algorithm implemented on top of the original protocol in order to retain compatibility with different streaming servers.

C. Session Controller

One of the most important components of the entire system is the session controller (Fig 1). The session controller is a software component placed between the streaming server and its clients, with the primary purpose of controlling requests made by clients and forwarding the content from the streaming server to the clients. This component is implemented as Java-based application that works in close conjunction with the web application that processes the login and billing requests from the clients.

Each request for content is checked against a database of valid sessions populated by the web interface at login time. If a session is valid, the client is allowed access to the requested content, while the illegal sessions are automatically disconnected from the server. Sessions are maintained using a special application-layer messaging protocol, through a series of messages sent in regular intervals from the clients to the server.

The design of this messaging protocol enables the server to control all clients, regardless of their current location or network characteristics, since the control mechanism is invulnerable to any NAT (Network Address Translation) or DHCP (Dynamic Host Configuration Protocol) related issues that might present a problem in this situation.

D. Web Application

Finally, the main web application implements the majority of the business logic of the entire system. It consists of three segments:

- Client web application - which implements an interface for communication with the end users, regardless of their respective platforms. Web users access the application interface directly, through the browser. This interface is responsible for client authentication and authorization. Electronic programming guide, user account settings, channels usage overview, etc., are also important functionalities of this application. Specially developed client software (described in previous section) is also integrated into this application (Fig. 2).
- Administrative web interface (Fig. 4) is used to configure and control the operation of the entire system. This part of the application is intended for use by the administrative staff, and should not be publicly available to all users of the system. Some

of the options that are configurable through this interface include: channel information, content type, stream and resolution configuration, account management, pricing and billing configurations, load balancing, etc. The management portion of the administrative application provides a statistical overview of the system, giving the manager the ability to perform analysis of the operation of the system. All user sessions are aggregated to show the most frequently watched, or the number of client accounts sorted by type or client platform. All web application interfaces support internationalization, and can be easily extended to use any language for the interface,

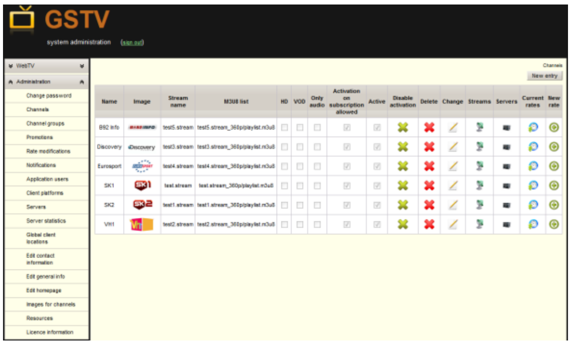

Figure 4. GSTV administrative web interface.

- Integration interface is a SOAP-based web service which enables the users to integrate the platform into their own CRM (Customer Relationship Management) system, without having to keep double user records or use two or more applications simultaneously for the same purpose of configuring user accounts and available channels. The use of SOAP-based communication enables the administrator to seamlessly integrate the streaming platform with the existing system, regardless of its underlying infrastructure and programming specifics. SOAP communication is not language dependent and provides a unified interface for integration,
- Accompanying applications and service provide the system with added functionalities which affect the user experience. These applications include a TV schedule parser, which is able to provide accurate TV schedule listings on a daily basis for all of the included channels, and a geolocation module which provides detailed information about the geographic locations of the users accessing the system.

E. GSTV Database

In order to synchronize the data between the central interfaces and the proxy server, both parts of the system communicate through a centralized database in which all information concerning the sessions are stored. The database is the only part of the system that cannot be distributed, due to the importance of data synchronization. The use of open-source database management system allows the use of load balancing clusters in case of high server loads, so the central database should not be viewed as a bottleneck or a single point of failure. Other segments of the system are highly scalable and distributable, enabling the system to handle high processing loads generated by a great number of users. This distribution is implemented as part of the messaging protocol, which enables the administrator to configure simple adaptive load balancing. The system transparently assigns the clients to the streaming server with the least processor load among the configured stream sources at the time of the request.

F. User Management and Billing

User account information is organized in a hierarchical manner. Top layer (also called the *customer layer*) contains detailed personal information about the entities. The middle (or the *account*) level contains the billing information and the account balance. Finally, the bottom level contains subscription information, such as login credentials, available channels, special offers etc. This hierarchy is implemented with a one-to-many cardinality, enabling a single account to contain an unlimited number of subscriptions, while, at the same time, each subscription is attached to exactly one account. During the authorization stage, a user is allowed to establish a session only if his account information is valid and the requested channel is available at the time of the request. Messages that are exchanged periodically as part of the session maintenance process also allow the system to dynamically calculate the cost for the elapsed time since the beginning of the session and, if necessary, terminate playback for all subscriptions whose account balance is insufficient for the price of the current channel. The billing subsystem supports the use of custom modifications that affect the price for a single subscription or channel, according to the specific needs of the company that deployed the platform.

G. GSTV Messaging System

The basic functionality behind the messaging protocol described previously has also been extended to include several non-critical functionalities. Among these functionalities is the ability to send various status or advertisement messages to selected clients. The receivers can be selected based on current viewing channel, type of the client, an administratively created customer group, or by any other arbitrary criteria. This feature is very useful for notifying users about upcoming promotions, system maintenance outages or targeted marketing. The notifications can be scheduled for repeated sending, starting with any time period, with no administrator intervention required.

Messages exchanged within the messaging protocols use an authentication mechanism that enables the communicating parties to assert the identity of the other participant. This is done using a hashing mechanism, while at the same avoiding the use of pre-shared keys on client side of the system. The constraints on the strength of encryption algorithms that can be used on the mobile platforms are imposed by the export regulations for countries in which the application store servers are hosted

[25]. These regulations were the primary reason the GSTV system uses a combination of account data and current session information to identify the requests sent towards the streaming server, instead of using the information embedded into the client device itself.

H. Implementation

All web interfaces, including the integration web service are implemented as Java web applications using JavaServer Faces and AXIS libraries, respectively. The system itself is based on open-source software, namely Apache Tomcat Application Server, MySQL DBMS (Database Management System) and Linux operating system, CentOS distribution. Mobile applications are developed as native applications, in order to maintain a high level of security and to enable efficient execution. Android and iOS mobile client applications have the same functionalities in order to create a unified user experience, regardless of the platform.

V. Conclusion

Although legacy multimedia distribution systems, like IPTV, are still largely used, they have several important deficiencies. They have very expensive closed architecture and their coverage is geographically limited. On the other hand, converged networks have led to a rapid emergence of a new type of multimedia distribution systems, like OTT and hybrid systems. These systems do not need any special provisioning and they exist on very simple principles. The lack of control over the network has been overcome by using adaptive bit-rate streaming techniques. The use of these techniques has created a distribution method which is highly effective. The most important market advantage of these systems over the legacy IPTV systems is that anyone who holds the rights to content can distribute it globally, not only in a closed network.

This paper presented an integrated, adaptive and scalable digital multimedia content distribution system, named GSTV. This system has all necessary properties to be classified as a modern multimedia content distribution system. Additionally, this system satisfies all important requirements and recommendations for similar systems [4].

References

[1] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672-1687, 2008.

[2] S. Meng, L. Liu, and J. Yin, "Scalable and reliable IPTV service through collaborative request dispatching," in *Proc. IEEE International Conference on Web Services*, 2010, pp. 179-186.

[3] Z. Xiao and F. Ye, "New insights on internet streaming and IPTV," in *Proc. International Conference on Content-based Image and Video Retrieval*, 2008, pp. 645-645.

[4] M. J. Montpetit, T. Mirlacher, and M. Ketcham, "IPTV: An end to end perspective," *Journal of Communications*, vol. 5, no. 5, pp. 358-373, 2010.

[5] A. Blair, G. Parr, S. Bryan, S. Appleby, and M. Nilsson, "A unified architecture for video delivery over the internet," in *Proc. 12th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool, United Kingdom, 2011.

[6] P. Tan and J. Slevinsky, "Multi-screen IPTV: Enabling technologies and challenges," *IEEE International Conference on Consumer Electronics*, 2011, pp. 1-2.

[7] K. Liu and M. Little, "The many faces of IPTV and internet TV services: A taxonomy," in *Proc. IEEE International Conference on Consumer Electronics*, 2010, pp. 1-2.

[8] U. Jennehag, "Bandwidth efficient IPTV distribution," Ph.D. dissertation, Department of Information Technology and Media, Mid Sweden University, Sundsvall, Sweden, 2007.

[9] M. Mu, J. Ishmael, W. Knowles, M. Rouncefield, *et al.*, "P2P-based IPTV services: Design, deployment and QoE measurement," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1515 - 1527, 2012.

[10] D. Cymbalák, F. Jakab, and M. Michalko, "Next generation IPTV solution for educational purposes," in *Proc. 9th International Conference on Emerging eLearning Technologies and Applications*, 2011, pp. 41-46.

[11] H.262 - Generic Coding of Moving Pictures and Associated Audio Information: Video, ITU-T Recommendation - 2012.

[12] H.264: Advanced Video Coding for Generic Audiovisual Services, ITU-T Recommendation - 2013.

[13] H. Zeng and Y. Fang, "Implementation of video transcoding client based on FFMPEG," *Advanced Materials Research*, vol. 756-759, pp. 1748-1752, 2013.

[14] Y. Cheng, Q. Liu, C. Zhao, X. Zhu, and G. Zhang, "Design and implementation of multimedia format converter based on FFmpeg," *Software Engineering and Knowledge Engineering: Theory and Practice, Advances in Intelligent and Soft Computing*, vol. 115, pp. 857-865, 2012.

[15] F. M. V. Ramos, "Channel smurfing: Minimising channel switching delay in IPTV distribution networks," in *Proc. IEEE International Conference on Multimedia and Expo*, 2010, pp. 1327-1332.

[16] P. H. Wu and Y. H. Hu, "Optimal layered video IPTV multicast streaming over mobile WiMAX systems," *IEEE Transactions on Multimedia*, vol. 13, no. 6, pp. 1395 - 1403, 2011.

[17] H. Joo, C. Yoon, T.W. Um, and H. Song, "A novel fountain code-based mobile IPTV multicast system architecture over WiMAX network," *Journal of Visual Communication and Image Representation*, vol. 23, no. 1, pp. 161–172, 2012.

[18] Z. Li, X. Zhu, A.C. Begen, and B. Girod, "IPTV multicast with peer-assisted lossy error control," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 3, pp. 434-449, 2012.

[19] J. F. Buford and M. Kolberg, "Hybrid overlay multicast simulation and evaluation," in *Proc. 6th IEEE Consumer Communications and Networking Conference*, 2009, pp. 1-2.

[20] X. Jin; W. Tu, and S.-H.G. Chan, "Challenges and advances in using IP multicast for overlay data delivery," *IEEE Communications Magazine*, vol. 47, no. 6, pp. 157-163, 2009.

[21] M. Brogle, D. Milic, L. Bettosini, and T. Braun, "A performance comparison of native IP Multicast and IP Multicast tunneled through a Peer-to-Peer overlay network," *International Conference on Ultra Modern Telecommunications*, 2009, pp. 1-6.

[22] Z. Fu, X. Meng, and S. Lu, "A transport protocol for supporting multimedia streaming in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, pp. 1615-1626, 2003.

[23] F. Daoust, P. Hoschka, C. Z. Patrikakis, R. S. Cruz, M. S. Nunes, and D. S. Osborne, "Towards video on the web with HTML5," *NEM Summit*, Barcelona, 2010.

[24] D. Gavalas and D. Economou, "Development platforms for mobile applications: status and trends," *IEEE Software*, vol. 28, no. 1, pp. 77-86, 2011.

[25] *Revised U.S. Encryption Export Control Regulations*, Bureau of Export Administration, Department of Commerce, United States, 2000.

Zoran Djuric (born April 8th, 1977) is a professor at the Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina. He received his Master of Science degree from the Faculty of Electrical Engineering, University of Belgrade, Serbia, in 2004. He received his Ph.D. in Electrical Engineering from the Faculty of Electrical Engineering, Banja Luka University, Bosnia and Herzegovina, in 2008.
His main areas of interest lie in computer security and privacy, particularly in software security, network security and data protection,

as well as programming languages, web technologies and network protocols.
Prof. Djuric is a member of IEEE.

Ognjen Joldzic (born August 14^{th}, 1986) works as a research and teaching assistant at the Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina. He received his Master's degree in Computing and Informatics from the Faculty of Electrical Engineering, Banja Luka University, Bosnia and Herzegovina, in 2010.
He is currently a PhD student, and his major interests include computer architecture, web development and computer networks.
Mr. Joldzic is a member of IEEE.