The International Journal Of Engineering And Science (IJES)
||Volume|| 3 ||Issue|| 01 || Pages || 41-47 || 2014||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

An Effective Live Video Streaming System

Rajasekhar Muppavarapu[1,] Mastan Rao Kale[2]
[1]*Department of Computer Science, QIS College of Engineering & Technology :: Ongole*
Email: raja07.mca@gmail.com
[2]*Department of Computer Science, QIS College of Engineering & Technology :: Ongole*
Email: mastanraokale@yahoo.com

----------ABSTRACT----------

A number of commercial peer-to-peer systems for live streaming have been introduced in recent years. The behavior of these popular systems has been extensively studied in several measurement papers. Due to the proprietary nature of these commercial systems, however, these studies have to rely on a "black-box" approach, where packet traces are collected from a single or a limited number of measurement points, to infer various properties of traffic on the control and data planes. Although such studies are useful to compare different systems from end-user's perspective, it is difficult to intuitively understand the observed properties without fully reverse-engineering the underlying systems. In this paper we describe the network architecture of Zattoo, one of the largest production live streaming providers in Europe at the time of writing, and present a large-scale measurement study of Zattoo using data collected by the provider. To highlight, we found that even when the Zattoo system was heavily loaded with as high as 20,000 concurrent users on a single overlay, the median channel join delay remained less than 2 to 5 seconds, and that, for a majority of users, the streamed signal lags over-the-air broadcast signal by no more than 3 seconds.

KEYWORDS *: Peer-to-peer system, live streaming, network architecture*

Date of Submission: 21 December 2013 Date of Acceptance: 25 January 2014

I. INTRODUCTION

Current generation broadcast network for TV is DTH which will be slowly replaced by the next generation Internet Protocol Television (IPTV) network. There is an emerging market for IPTV. Numerous commercial systems now offer services over the Internet that is similar to traditional over-the-air, cable, or satellite TV. Live television, time-shifted programming, and content-on-demand are all presently available over the Internet. Increased broadband speed, growth of broadband subscription base, and improved video compression technologies have contributed to the emergence of these IPTV services [1][6]. IPTV systems deliver video and audio channels to viewing devices by switching a single channel to multiple sources. IP Television networks are primarily constructed of computer servers, gateways, access connections and end user display devices. Servers control the overall system access and processing of channel connection requests and gateways convert the IP television network data to signals that can be used by television media viewers. Content aggregation is the process of combining multiple content sources for distribution through other communication channels. A head end is part of a television system that selects and processes video signals for distribution into a television distribution network. The core network is the central network portion of a communication system. The core network primarily provides interconnection and transfer between edge networks. An access network is a portion of a communication network (such as the public switched telephone network) that allows individual subscribers or devices to connect to the core network.

A premises distribution network (PDN) consists of the equipment and software that are used to transfer data and other media in a customer's facility or home. A viewing device is a combination of hardware and software that can convert media such as video, audio or images into a form that can be experienced by humans. The network archicture of IPTV is shown in Fig1. Peer to peer multiplexing is mainly used for Live streaming of video and audio. User draw a distinction between three uses of peer-to-peer (P2P) [1] networks: delay-tolerant file download of archival material, delay-sensitive progressive download (or streaming) of archival material, and real-time live streaming. In the first case, the completion of download is elastic, depending on available bandwidth in the P2P network. The application buffer receives data as it trickles in and informs the user upon the completion of download. The user can start playing back the file for viewing in the case of a video

file. Bit torrent and variants are examples of delay-tolerant file download systems. In the second case, video playback starts as soon as the application assesses it has sufficient data buffered that, given the estimated download rate and the playback rate, it will not deplete the buffer before the end of file. If this assessment is wrong, the application would have to either pause playback or rebuffed or slow down playback. While users would like playback to start as soon as possible, the application has some degree of freedom in trading off playback start time against estimated network capacity. Most video-on-demand systems are examples of delay-sensitive progressive-download application. The third case, real-time live streaming has the most stringent delay requirement. While progressive download may tolerate initial buffering of tens of seconds or even minutes, live streaming generally cannot tolerate more than a few seconds of buffering. Taking into account the delay introduced by signal ingest and encoding, and network transmission and propagation, the live streaming system can introduce only a few seconds of buffering time end-to-end and still be considered ―live‖. The Zattoo peer-to-peer live streaming system was a free-to-use network serving over 3 million registered users in eight European countries at the time of study, with a maximum of over 60 000 concurrent users on a single channel. The system delivers live streams using a receiver-based, peer-division multiplexing scheme.

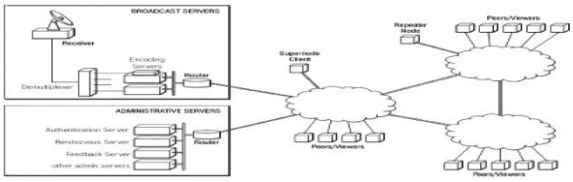

Fig. 1 Architecture of IPTV

To ensure real-time performance when peer uplink capacity is below requirement, Zattoo subsidizes the network's bandwidth requirement, as described in. After delving into Zattoo's architecture in detail, large-scale measurements collected during the live broadcast of the UEFA European Football Championship, one of the most popular one-time events in Europe, in June 2008 [5]. During the course of that month, Zattoo served more than 35 million sessions to more than 1 million distinct users.

II. SYSTEM ARCHITECTURE

The Zattoo system rebroadcasts live TV, captured from satellites, onto the Internet. The system carries each TV channel on a separate peer-to-peer delivery network and is not limited in the number of TV channels it can carry. Although a peer can freely switch from one TV channel to another, thereby departing and joining different peer-to-peer networks, it can only join one peer-to-peer network at any one time. Users are required to register themselves at the Zattoo Web site to download a free copy of the Zattoo player application. To receive the signal of a channel, the user first authenticates itself to the Zattoo *Authentication Server*. Upon authentication, the user is granted a ticket with limited lifetime. The user then presents this ticket, along with the identity of the TV channel of interest, to the Zattoo *Rendezvous Server*. If the ticket specifies that the user is authorized to receive signal of the said TV channel, the Rendezvous Server returns to the user a list of peers currently joined to the P2P network carrying the channel, together with a signed channel ticket. If the user is the first peer to join a channel, the list of peers it receives contain only the Encoding Server. The user joins the channel by contacting the peers returned by the Rendezvous Server, presenting its channel ticket, and obtaining the live stream of the channel from them. Zattoo uses the Reed–Solomon (RS) error correcting code (ECC) for forward error correction [2]. The RS code is a systematic code: of the n packets sent per segment, k<n packets carry the live stream data, while the remainder carries the redundant data [3]. Due to the variable-bit rate nature of the data stream, the time period covered by a segment is variable, and a packet may be of size less than the maximum packet size.

III. RECEIVER BASED P2P MULTIPLEXING

1. P2P MULTIPLEXING

When a new peer requests to join an existing peer, it specifies the substream(s) it would like to receive from the existing peer [4]. These substreams do not have to be consecutive. Contingent upon availability of bandwidth at existing peers, the receiving peer decides how to multiplex a stream onto its set of neighboring peers, giving rise to our description of the Zattoo live streaming protocol as a receiver-based, peer-division multiplexing protocol. To minimize per-packet processing time of a stream, the Zattoo protocol sets up a virtual circuit with multiple fan outs at each peer. When a peer joins a TV channel, it establishes a peer-division multiplexing (PDM) scheme among a set of neighboring peers by building a virtual circuit to each of the neighboring peers. Baring departure or performance degradation of a neighbor peer, the virtual circuits are maintained until the joining peer switches to another TV channel. With the virtual circuits set up, each packet is forwarded without further per-packet handshaking between peers.

The PDM establishment process consists of two phases: the *search* phase and the *join* phase. *Search Phase:* To obtain a list of potential neighbors, a joining peer sends out a SEARCH message to a random subset of the existing peers returned by the Rendezvous Server. The SEARCH message contains the substream indices for which this joining peer is looking for peering relationships. The joining peer continues to wait for SEARCH replies until the set of potential neighbors contains at least a minimum number of peers, or until all SEARCH replies have been received. *Join Phase:* Once the set of potential neighbors is established, the joining peer sends JOIN requests to each potential neighbor. The JOIN request lists the substreams for which the joining peer would like to construct virtual circuit with the potential neighbor.

IV. STREAM MANAGEMENT

The IOB is referenced by an *input pointer*, a *repair pointer*, and one or more *output pointers*. The input pointer points to the slot in the IOB where the next incoming packet with sequence number higher than the highest sequence number received so far will be stored. The repair pointer always points one slot beyond the last packet received in order and is used to regulate packet retransmission and adaptive PDM [3] as described later. Different peers may request for different numbers of, possibly nonconsecutive, sub streams. To accommodate the different forwarding rates and regimes required by the destinations, we associate a packet map and forwarding discipline with each output pointer. Fig. 3 shows the packet map associated with an output peer pointer where the peer has requested sub streams 1, 4, 9, and 14. Every time a peer pointer is repositioned to the beginning of a sub-buffer of the IOB, all the packet slots of the requested sub streams are marked Needed and all the slots of the sub streams not requested by the peer are marked SKIP. When a Needed packet arrives and is stored in the IOB, its state in the packet map is changed to READY. As the peer pointer moves along its associated packet map, READY packets are forwarded to the peer and their states changed to SENT. A slot marked needed but not READY, such as slot n+4

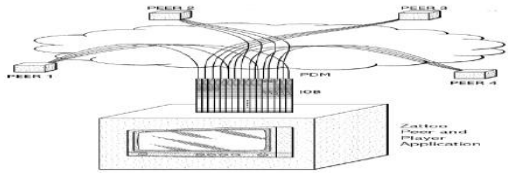

Fig 2. Peer system with an IOB

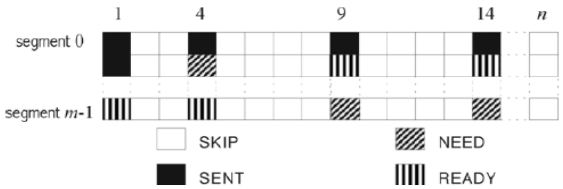

Fig. 3 Packet Map Associated with peer map

In addition to achieving lossless recording, we use retransmission to let a peer recover from transient network congestion. A peer sends out a retransmission request when the distance between the repair pointer and the input pointer has reached a threshold of R packet slots, usually spanning multiple segments. A retransmission request consists of an R bit packet mask, with each bit representing a packet, and the sequence number of the packet corresponding to the first bit. Marked bits in the packet mask indicate that the corresponding packets need to be retransmitted. When a packet loss is detected, it could be caused by congestion on the virtual circuits forming the current PDM or congestion on the path beyond the neighboring peers. In either case, current neighbor peers will not be good sources of retransmitted packets.

V. ADAPTIVE PDM

Peers on the Zattoo network can redistribute a highly variable number of substreams, reflecting the high variability in uplink bandwidth of different access network technologies [7]. For a full-stream consisting of 16 *constant*-bit rate substreams, our prior study showed that based on realistic peer characteristics measured from the Zattoo network, half of the peers can support less than half of a stream, 82% of peers can support less than a full-stream, and the remainder can support up to 10 full streams (peers that can redistribute more than a full stream is conventionally known as super nodes in the literature) [5]. With variable- bit rate streams, the bandwidth carried by each sub stream is also variable. To increase peer bandwidth usage, without undue degradation of service, we instituted measurement-based admission control at each peer. In addition to controlling resource commitment, another goal of the measurement-based admission control module is to continually estimate the amount of available uplink bandwidth at a peer.

The amount of available uplink bandwidth at a peer is initially estimated by the peer sending a pair of probe packets to Zattoo's Bandwidth Estimation Server. Once a peer starts forwarding sub streams to other peers, it will receive from those peers quality-of-service feedbacks that inform its update of available uplink bandwidth estimate. A peer sends quality-of-service feedback only if the quality of a substream drops below a certain threshold [9]. Upon receiving quality feedback from multiple peers, a peer first determines if the identified substreams are arriving in low quality. If so, the low quality of service may not be caused by limit on its own available uplink bandwidth.

VI. SERVER-SIDE MEASUREMENTS

In the Zattoo system, two separate centralized collector servers collect usage statistics and error reports, which we call the "stats" server and the "user-feedback" server respectively. The "stats" server periodically collects aggregated player statistics from individual peers, from which full session logs are constructed and entered into a session database. The session database gives a complete picture of all past and present sessions served by the Zattoo system. A given database entry contains statistics about a particular session, which includes join time, leave time, uplink bytes, download bytes, and channel name associated with the session. We study the sessions generated on three major TV channels from three different countries (Germany, Spain, and Switzerland), from June 1st to June 30th, 2008. Throughout the paper, we label those channels from Germany, Spain, and Switzerland as *ARD, Cuatro*, and *SF2*, respectively. Euro 2008 games were held during this period, and those three channels broadcast a majority of the Euro 2008 games including the final match. See Table I for information about the collected session data sets.

The "user-feedback" server, on the other hand, collects users' error logs submitted asynchronously by users. The "user feedback" data here is different from peer's quality feedback used in PDM reconfiguration described in Section II-C. Zattoo player maintains an encrypted log file which contains, for debugging purposes, detailed behavior of client-side P2P engine, as well as history of all the streaming sessions initiated by a user since the player startup. When users encounter any error while using the player, such as log-in error, join failure, bad quality streaming etc., they can choose to report the error by clicking a "Submit Feedback" button on the player, which causes the Zattoo player to send the generated log file to the user-feedback server. Since a given feedback log not only reports on a particular error, but also describes "normal" sessions generated prior to the occurrence of the error, we can study user's viewing experience (e.g., channel join delay) from the feedback logs. Table II describes the feedback logs collected from June 20th to June 29th. A given feedback log can contain multiple sessions (for the same or different channels), depending on user's viewing behavior. The second column in the table represents the number of feedback logs which contain at least one session generated on the channel listed in the corresponding entry in the first column. The numbers in the third column indicate how many distinct sessions generated on said channel are present in the feedback logs.

Overlay Size and Sharing Ratio

We first study how many concurrent users are supported by the Zattoo system, and how much bandwidth is contributed by them. For this purpose, we use the session database presented in Table I. By using the join/leave timestamps of the collected sessions, we calculate the number of concurrent users on a given channel at time i. Then we calculate the average sharing ratio of the given channel at the same time. The average sharing ratio is defined as total users' uplink rate divided by their download rate on the channel. A sharing ratio of one means users contribute to other peers in the network as much traffic as they download at the time. We calculate the average sharing ratio from the total download/uplink bytes of the collected sessions. We first obtain all the sessions which are active across time i. We call a set of such sessions Si. Then assuming uplink/download bytes of each session are spread uniformly throughout the entire session duration, we approximate the average sharing ratio at time *i* as $\frac{\sum_{i \in S_i} uplink\_bytes(i)/duration(i)}{\sum_{i \in S_i} download\_bytes(i)/duration(i)}$

Channel Switching Delay

When user clicks on a new channel button, it takes some time (a.k.a. channel switching delay) for the user to be able to start watching streamed video on Zattoo player. The channel switching delay has two components. First, Zattoo player needs to contact other available peers and retrieve all required sub-streams from them. We call the delay incurred during this stage "join delay." Once all necessary sub-streams have been negotiated successfully, the player then needs to wait and buffer the minimum amount of streams (e.g., 3 seconds) before starting to show the video to the user. We call the resulting wait time "buffering delay." The total channel switching delay experienced by users is thus the sum of join delay and buffering delay.

PPLive reports channel switching delay around 20 to 30 seconds, but can be as high as 2 minutes, of which join delay typically accounts for 10 to 15 seconds [7]. We measure the join delay experienced by Zattoo users from the feedback logs described in Table II. Debugging information contained in the feedback logs tells us when user clicked on a particular channel, and when the player has successfully joined the P2P overlay and starting to buffer content. One concern in relying on user-submitted feedback logs to infer join delay is the potential sampling bias associated with them. Users typically submit feedback logs when they encounter some kind of errors, and that brings up the question of whether the captured sessions are representative samples to study. We attempt to address this concern by comparing the data from feedback logs against those from the session database. The latter captures the complete picture of user's channel watching behavior, and therefore can serve as a reference. In our analysis, we compare the user arrival time distribution obtained from the two data sets. For fair comparison, we used a subset of the session database which was generated during the same period when the feedback logs were collected.

Repeater Node Assignment

As illustrated in Fig. 5, the live coverage of Euro Cup games brought huge flash crowds to the Zattoo system. With typical users contributing only about 25–30%of the average streaming bit rate, Zattoo must subsidize the balance. As described in Section III, the Zattoo's Subsidy System assigns Repeater nodes to channels that require more bandwidth than its own globally available aggregate upload bandwidth.

VII. CLIENT-SIDE MEASUREMENTS

To further study the P2P overlay beyond details obtainable from aggregated session-level statistics, we run several modified Zattoo clients which periodically retrieve the internal states of other participating peers in the network by exchanging SEARCH/JOIN messages with them. After a given probe session is over, the monitoring client archives a log file where we can analyze control/data traffic exchanged and detailed protocol behavior. We run the experiment during Zattoo's live coverage of Euro 2008 (June 7th to 29th). The monitoring clients tuned to game channels from one of Zattoo's data centers located in Switzerland while the games were broadcast live. The data sets presented in this paper were collected during the coverage of the championship final on two separate channels: *ARD* in Germany and *Cuatro* in Spain. Soccer teams from Germany and Spain participated in the championship final. As described in Section II-A, Zattoo's peer discovery is guided by peer's topology information. To minimize potential sampling bias caused by our use of single vantage point for monitoring, we assigned "empty" AS number and country code to our monitoring clients, so that their probing is not geared towards those peers located in the same AS and country.

Sub-Stream Synchrony

To ensure good viewing quality, peer should not only obtain all necessary sub-streams (discounting redundant sub streams), but also have those sub-streams delivered temporally synchronized with each other for proper online decoding. Receiving out-of-sync sub-streams typically results in pixilated screen on the player. As described in Sections 1 and II-C, Zattoo's protocol favors sub-streams that are relatively insync when constructing the PDM, and continually monitors the sub-streams' progression over time, replacing those sub streams that have fallen behind and reconfiguring the PDM when necessary. In this section we measure the effectiveness of Zattoo's Adaptive PDM in selecting sub-streams that are largely in-sync.To quantify such inter-sub stream synchrony, we measure the difference in the latest (i.e., maximum) packet sequence numbers belonging to different incoming sub-streams. When a remote peer responds to a SEARCH query message, it includes in its SEARCH reply the latest sequence numbers that it has received for all sub-streams. If some sub-streams happen to be lossy or stalled at that time, the peer marks such sub-streams in its SEARCH replies. Thus, we can inspect SEARCH replies from existing peers to study their inter-sub stream synchrony.

Peer Synchrony

While sub-stream synchrony tells us stream quality different peers may experience, "peer synchrony" tells us how varied in time peers' viewing points are.With small scale P2P networks, all participating peers are likely to watch live streaming roughly synchronized in time. However, as the size of the P2P overlay grows, the viewing point of edge nodes may be delayed significantly compared to those closer to the Encoding Server. In the experiment, we define the *viewing point* of a peer as the median of the latest sequence numbers across its sub-streams. Then we choose one peer (e.g., a Repeater node directly connected to the Encoding Server) as a reference point, and compare other peers' viewing point against the reference viewing point.

Effectiveness of ECC in Isolating Loss

Now we investigate the effects of overlay sizes on the performance scalability of the Zattoo system. Here we focus on client-side quality (e.g., loss rate). As described in Section 1, Zattoo-broadcast media streams are RS encoded, which allows peers to reconstruct a full stream once they obtain at least k of n sub-streams. Since the ECC-encoded stream reconstruction occurs hop by hop in the overlay, it can mask sporadic packet losses, and thus prevent packet losses from being propagated throughout the overlay at large.To see if such packet loss containment actually occurs in the production system, we run the following experiments. We let our monitoring client join a game channel, stay tuned for 15 seconds, and then leave. We wait for 10 seconds after the 15-second session is over. We repeat this process throughout the 2-hour game coverage and collect the logs. A given log file from each session contains a complete list of packet sequence numbers received from the connected peers during the session, from which we can detect upstream packet losses. We then associate individual packet loss with the peer-path from the Encoding Server. This gives us a rough sense of whether packets traversing longer hops would experience higher loss rate. In our analysis, we discount packets delivered for the first 3 seconds of a session to allow the PDM to stabilize.

VIII. PEER-TO-PEER SHARING RATIO

The premise of a given P2P system's success in providing scalable stream distribution is sufficient bandwidth sharing from participating users. Section IV-A shows that the average sharing ratio of Zattoo users ranges from 0.2 to 0.35, which translates into bandwidth uplinks ranging from 100 kbps to 175 kbps. This is far lower than the numbers reported as typical uplink bandwidth in countries where Zattoo is available. Aside from the possibility that user's bandwidth may be shared with other applications, we find factors such as users' behavior, support for variable-bit rate encoding, and heterogeneous NAT environments contributing to suboptimal sharing performance. Designers of P2P streaming systems must pay attention to these factors to achieve good bandwidth sharing. However, one must also keep in mind that improving bandwidth sharing should not be at the expense of compromised user viewing experience, e.g., due to more frequent uplink bandwidth saturation.

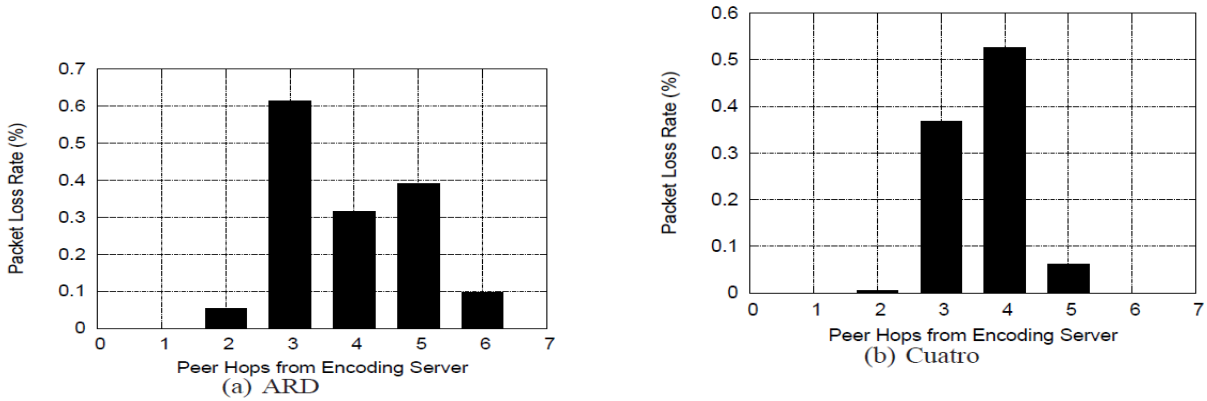

Fig. 4. Packet loss rate vs. peer hops from Encoding Server.

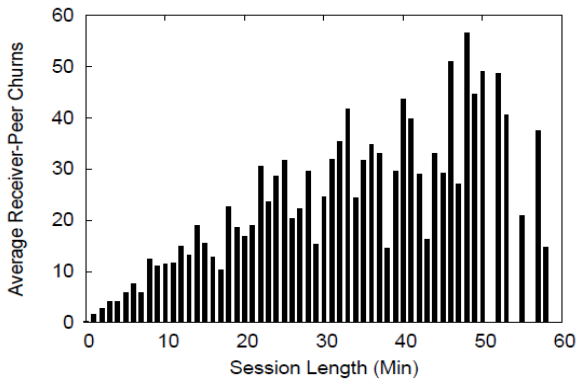

Fig. 5. Receiver-peer churn frequency.

IX. RELATED WORKS

Aside from Zattoo, several commercial peer-to-peer systems intended for live streaming have been introduced since 2005, notably PPLive, PPStream, SopCast, TV Ants, and UUSee from China, and Joost, Livestation, Octoshape, and RawFlow from the EU. A large number of measurement studies have been done on one or the other of these systems Many research prototypes and improvements to existing P2P systems have also been proposed and evaluated. Our study is unique in that we are able to collect network core data from a large production system with over 3 million registered users with intimate knowledge of the underlying network architecture and protocol.

X. CONCLUSION

A receiver-based, peer-division multiplexing engine to deliver live streaming content on a peer-to-peer network. The same engine can be used to transparently build a hybrid P2P/CDN delivery network by adding Repeater nodes to the network. By analyzing a large amount of usage data collected on the network during one of the largest viewing events in Europe, we have shown that the resulting network can scale to a large number of users and can take good advantage of available uplink bandwidth at peers. We have also shown that error-correcting code and packet retransmission can help improve network stability by isolating packet losses and preventing transient congestion from resulting in PDM reconfigurations.

REFERENCES

[1] R. auf der Maur, ―Die Weiterverbreitung von TV- und Radioprogrammenüber IP-basierte Netze, in *Entertainment Law*, F. d. Schweiz, Ed., 1st ed. Bern, Switzerland: Stämpfli Verlag, 2006.

[2] Euro2008, UEFA [Online]. Available: http://www1.uefa.com/ [3] S. Lin and D. J. Costello Jr., *Error Control Coding*, 2nd ed. Englewood Cliffs, NJ: Pearson Prentice-Hall, 2004.

[3] S. Xie, B. Li, G. Y. Keung, and X. Zhang, ―CoolStreaming: Design, theory, and practice, *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1661–1671, Dec. 2007.

[4] K. Shami *et al.*, ―Impacts of peer characteristics on P2PTV networks scalability,‖ in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2736–2740.

[5] Bandwidth test statistics across different countries, Bandwidth- Test.net [Online]. Available: http://www.bandwidth-test. net/stats/country/

[6] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, ―Insights into PPLive: A measurement study of a large-scale P2P IPTV system, in *Proc. IPTV Workshop, Int. World Wide Web Conf.*, May 2006.

[7] B. Li *et al.*, ―An empirical study of flash crowd dynamics in a P2Pbased live video streaming system, in *Proc. IEEE GLOBECOM*, 2008, pp. 1–5.

[8] J. Rosenberg *et al.*, ―STUN—Simple traversal of User Datagram Protocol (UDP) through network address translators (NATs), RFC 3489, 1993.

[9] A. Ali, A. Mathur, and H. Zhang, ―Measurement of commercial peer-to-peer live video streaming, in *Proc. Workshop Recent Adv. Peer-to-Peer Streaming*, Aug. 2006.

AUTHORS PROFILE

Mr.RAJASEKHAR MUPPAVARAPU (M.Tech) from QIS College of Engineering and Technology, Ongole Affiliated to JNTU, Kakinada. His Interested areas include Secure Computing and Computer networks.

Mr Mastan Rao Kale, M.Tech(CSE) is currently working as an Asst. Professor in Department of CSE, QIS College of Engineering & Technology :: Ongole. He has published several papers at various national and International Journals and Conferences. His Research areas include Image Processing, Data mining and Network Security.