International Journal of Software Engineering and Its Applications
Vol.8, No.2 (2014), pp.151-158
http://dx.doi.org/10.14257/ijseia.2014.8.2.15

An Efficient P2P-based Mobile Social Media Delivery for Real-time MobileCast

Changhee Cho[1], Ha Tran-Thi-Thu[2], Sanghyun Park[2] and Jinsul Kim[2*]

[1]Graduate School of Interdisciplinary Program of E-Commerce, Chonnam National University, Gwangju, 500-757, Korea
[2]School of Electronics & Computer Engineering, Chonnam National University Gwangju, 500-757, Korea
xisom@xisom.com, thuhabkhn@gmail.com, sanghyun079@gmail.com, jsworld@jnu.ac.kr

Abstract

With traditional video live streaming technologies, the quality of the video degrades as more viewers tune in. Big viewers need a lot of bandwidth to support them. In this paper, we propose a new algorithm with which mobile user is both capable of broadcasting and watching live video at the same time. It is called MobileCast. MobileCast allows users broadcast their own video in real-time to neighbor mobiles based on peer-to-peer network using BitTorrent protocol technology. With MobileCast, the quality and reliability of the broadcast actually improves every time a new viewer joins, since every new viewer becomes a miniature broadcaster and amplifies the stream. The idea is to use some of the idle sending capacity at the end mobile users. When mobile user captures and sends video in real-time, the mobile device becomes a broadcaster then it can send this video for other mobile devices which again would forward the stream to other mobile, and so on.

Keywords: *peer–to–peer, live video streaming, file sharing, broadcasting*

1. Introduction

Nowadays, peer-to-peer (P2P) live streaming is becoming an increasingly popular technology. The demand for real-time multimedia services, including VoIP (Voice over Internet Protocol), audio and video streaming, has been growing rapidly so that multimedia streaming applications have become dominant in present communications systems. Consequently, the demand for innovative smart phone applications that allow users to receive and deliver live or on-demand rich content has increased dramatically. One of the main challenges is to provide a good quality of service (QoS) in spite of the dynamic behavior of the network [1]. Live video streaming is a technology that allows an audio and video source broadcast live to be viewed in real-time by viewers who are connected. Uses generate a video stream that is transmitted live to streaming server using encoding software, broadcast software. Broadcasting video direct to any listener who is connected anywhere in the world. So you can allow listeners to have access directly to the video stream via their devices. Streaming generally is a method for intelligent broadcasting of data on the mobile phone. It differs from conventional downloading as it isn't necessary to wait for the end of the process in order to be able to start playing back the data. It is important to understand this difference as it is the whole point of broadcasting in real-time since it allows considerable bandwidth

* Corresponding Author

ISSN: 1738-9984 IJSEIA

savings resulting in less costs for the traffic generated. The audio stream is compressed using an audio codec such as Mp3, Vorbis or AAC. The video stream is compressed using a video codec such as H264 or VP8. Encoded audio and video streams are assembled in a container bitstream such as Mp4, FLV, or ISMA. The bitstream is delivered from a streaming server to a streaming client using a transport protocol, such as MMS (Microsoft Media Server) or RTP (Real-time Transport Protocol). The streaming client may interact with the streaming server using a control protocol, such as MMS or RTSP (Real Time Streaming Protocol). When using any of the technologies described up until now, at the fix time, the mobile users receive a live video streaming but do not broadcast anything but with MobileCast, user can watching live video and also broadcast this video to other mobile [2]. So, anyone can create a resilient video stream that would be instantly accessible by thousands of people around the world.

2. Related Research

2.1. Peer-to-Peer Live Video Streaming

P2PTV refers to peer- to-peer software applications designed to redistribute video streams in real-time on a P2P network, as shown in Figure 1 the distributed video streams are typically TV channels from all over the world but may also come from other sources. The draw to these applications is significant because they have the potential to make any TV channel globally available by any individual feeding the stream into the network where each peer joining to watch the video is a relay to other peer viewers, allowing a scalable distribution among a large audience with no incremental cost for the source.

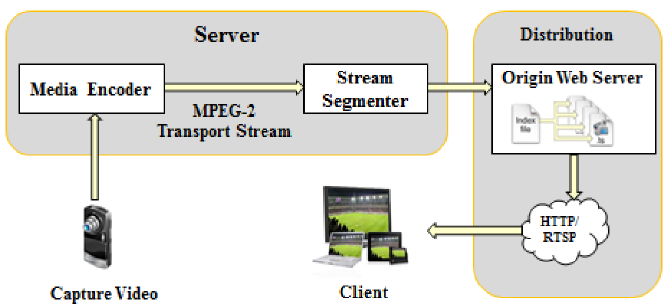

Figure 1. P2PTV Conventional Live Streaming Structure

P2P streaming systems can be broadly classified into two categories based on the overlay network structure. They are tree-based and mesh-based. Video streaming can be classified into two categories: live and on-demand. In a live streaming session, a live video content is disseminated to all users in real-time [3]. The video playbacks on all users are synchronized. To the contrary, video on- demand users enjoy the flexibility of watching whatever video clips whenever they want. The playbacks of the same video clip on different users are not synchronized. In this section, we introduce several P2P live streaming systems using different overlay structures.

2.3. BitTorrent Protocol

BitTorrent is a widely popular P2P protocol for content distribution. BitTorrent is one of the most popular ways of sharing and downloading files as shown in Figure 2. In BitTorrent, files are split into pieces, allowing peers that are still downloading content to share the pieces they already have to others [4]. Corresponding to each file available for down-load, there is a central component called tracker that keeps track of the nodes currently in the system. When a new peer joins, it contacts the tracker to obtain a list of a random subset of these nodes. Some implementations of the BitTorrent protocol also make use of a DHT (distributed hash table - which was not part of the original design), next to the tracker, to enhance peer discovery. Each node then establishes persistent connections with a large set of peers, called its neighborhood, and uploads data to a subset of this neighborhood. More specifically, each peer equally divides its upload capacity into a number of upload slots. Each peer maintains its neighborhood informed about the pieces it owns. The information received from its neighborhood is used to request pieces of the file according to the Local Rarest First policy. This policy determines that each peer requests the pieces that are the rarest among its neighbors.

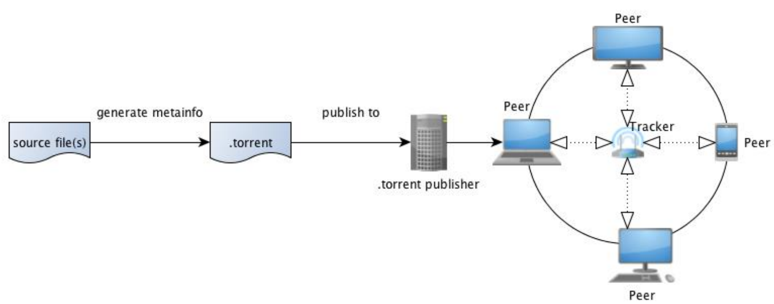

Figure 2. BitTorrent Protocol Procedure

The procedure of sharing a file is implemented as follows:

- A torrent file containing information about the file and a central server called a tracker is published, usually on a web page.
- The user downloads the torrent file, and opens it with a BitTorrent client application. The client informs the tracker of its existence, and receives a list of other peers downloading or sharing the file in question.
- The peers exchange information about who has which pieces, and download those pieces that are interesting. When a piece is successfully downloaded by a peer, it is announced as available to other peers.
- When the complete file has been downloaded, the peer may choose to remain online, to share the file.

3. System Design

3.1. System Structure

MobileCast is a live video streaming sharing system that cooperates transparently by using idle bandwidth on a user's mobile to deliver live video or on-demand broadcasts. It is used for the purpose of improving performance, scalability, and cost efficiency, of delivering live video streaming to end mobile users through the use of a media plug-in as show in Figure 3. MobileCast is peer-to-peer file sharing system such as BitTorrent [5, 6], but has the advantage of providing a robust live or on-demand streaming solution. The original high bitrate stream is split up into smaller bit-rate streams that are shared through user's mobiles, by incorporating a plug-in that reconstructs the smaller streams back to the original high bit-rate stream. MobileCast constructs several smaller data streams a lower bit-rate from an original higher bit-rate stream. In MobileCast, none of the data streams is identical. As an example, assume the live stream is a 400 Kbit/s signal and the MobileCast solution constructs multiple data streams at a size of 100 Kbit/s. Now, an end user receiving any four of the different data streams at 100 Kbit/s may use these four data streams to construct the original live stream back to 400 Kbit/s, and thus the movie can be played in real-time at the end mobile user [7], [8].

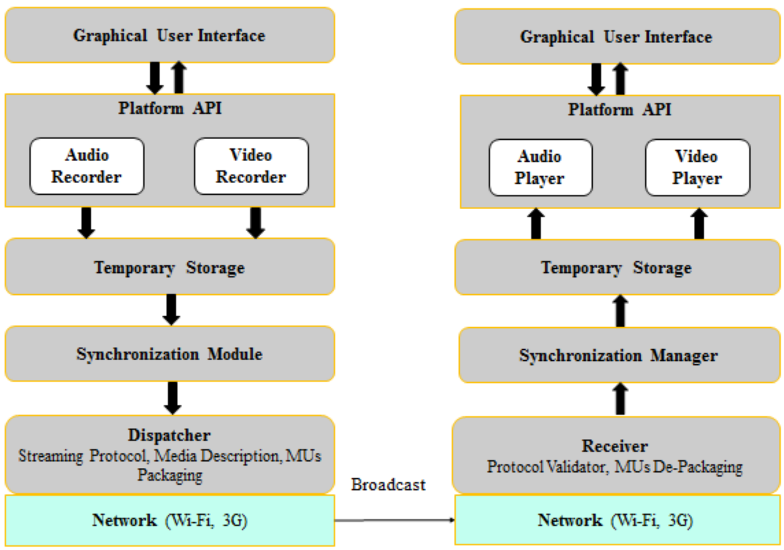

Figure 3. MobileCast System using BitTorrent Protocol

The video quality of the channels usually depends on how many users are watching, the video quality is better if there are more users. For example, if a mobile user wants to see a live video, he contacts a tracker server for that channel in order to obtain addresses of peers who distribute that channel; it then contacts these peers to receive the feed [9-11]. The tracker records the user's address, so that it can be given to other users who want to view the same channel. In effect, this creates an overlay mobile network on top of the regular internet for the distribution of real-time video content.

3.2. MobileCast Algorithm:

In this paper, we propose a scenario where smartphone users are interested in watching the same live video at the same time. However, each phone's individual cellular connection may not be sufficient for providing high video quality. Consider, for example, a user wants to show to his friends a live video which capture by his mobile camera while being in a stadium, his friends who want to watch a live soccer match on their phones at the same time. In this case, some or all of the users may have poor or intermittent cellular connectivity, depending on the coverage of their providers, or may face congestion in the local network (*e.g.*, when they use Wi-Fi or 3G to download). Furthermore, when every phone has multiple parallel connections (*e.g.*, 3G, Wi-Fi, and Bluetooth), there are even more available resources that, if properly used, can further improve the user experience.

In MobileCast, we consider a local network consisting of four phones: A, B, C, and D. After finishing downloading segment *s* using 3G or Wi-Fi, mobile phone A periodically broadcast the segments that it currently has to its neighbors - B, C, and D. B, C, and D then send requests (want to see segment of video which A is watching) for segment *s* to A as shown in Table 1.

Table 1. MobileCast algorithm

```
WHEN a packet p is received from mobile phone A
  IF p is an advertisement containing s THEN
// subsequent pulls
    Request A for the dimensions of s
  ELSE IF p is a request for d dimensions of s THEN
    IF there are other similar requests THEN
    Let d be largest requested dimension
    Remove these requests from the request queue
    END IF
    Send d dimensions of s to A
  ELSE IF p is a dimension of s THEN
    Decode s using p
    IF s is received fully THEN
      // initial push
      Send all dimensions of s to a neighbor
      Add s to the list of segments to be advertised
    END IF
  END IF
END WHEN
```

With *s*: segment, *p*: packet, *d*: dimension.

When mobile phone B requests a segment *s* from mobile phone A, it takes into account previously overheard dimensions of the subspace representing segment *s*. In particular, it explicitly indicates in the request how many additional dimensions it needs to receive to decode *s*. This reduces the number of dimensions to be sent. Upon receiving the request, mobile phone A sends the requested segments to mobile phone B.

When mobile phone A sends a segment *s* requested by mobile phone B, it first checks if there are pending requests for the same segments from other neighbors. If there are, it finds the maximum number of dimensions requested among these requests. Denote this maximum dimension by *d*. After sending a segment *s* to B, A notifies all phones that requested some dimensions of segment *s*. Upon receiving the notification, these phones check if they received all the necessary dimensions to decode *s*. If not, they send requests to A and B for additional dimension. At this time, mobile phone B is watching a segment which receives from mobile A and also broadcast this segment for other mobile phone such as C, D and so on.

4. Conclusion

In this paper, we proposed MobileCast algorithm for MobileCast application which is implemented as an Android application with P2P system using BitTorrent protocol technology. Video is captured from the mobile's camera, is broadcasted and transmitted to other mobiles using the P2P approach manner of service and depending on the support can significantly reduce the cost of the server using the rental cost and speed of the service. Each mobile user, while downloading a live video stream, is simultaneously also uploading that stream to other users. By using N to N configuration, the quality of the video increases as more viewers tune in P2P network.

Acknowledgements

This research was supported by Chonnam National University and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2013R1A1A2013740).

References

[1] C. B. Hays, "Bypassing internet service provider traffic shaping with peer-to-peer file sharing through deliberate false positives", IEEE Transactions on Multimedia, vol. 14, no. 6, **(2012)**, pp. 1515-1527.
[2] C. Cho, H. Tran-Thi-Thu and J. Kim, "P2P-based Mobile Social Media Delivery Methodology for Beaming Broadcast Service", Workshop on Mobile and Wireless 2013 Second at Jeju National University International Center, Jeju Island, Korea, **(2013)**.
[3] M. Sanna and E. Izquierdo, "Live Scalable Video Streaming on Peer-to-Peer Overlays with Network Coding", IEEE Latin America Transactions, vol. 11, no. 3, **(2013)**, pp. 962-968.
[4] M. Mu, J. Ishmael, W. Knowles, M. Rouncefield, N. Race, M. Stuart and G. Wright, "P2P-Based IPTV Services: Design, Deployment, and QoE Measurement", IEEE Transactions on Multimedia, vol. 14, no. 6, **(2012)**, pp. 1515-1527.
[5] H.-Han Shuai, D.-N. Yang, W.-H. Cheng and M.-S. Chen, "MobiUP: An Upsampling-Based System Architecture for High-Quality Video Streaming on Mobile Devices", IEEE Transactions on Multimedia, vol. 13, no. 5, **(2011)**, pp. 1077-1091.
[6] A. Paula Couto da Silva, E. Leonardi, M. Mellia and M. Meo, "Exploiting Heterogeneity in P2P Video Streaming", IEEE Transactions on Computers, vol. 60, no. 5, **(2011)**, pp. 667-679.
[7] A. Nafaa and M. Gaucher, "Implementation and Analysis of a Peer-to-Peer Retransmissions System for Live Video Services", IEEE Transactions on Multimedia, vol. 18, no. 2, **(2011)**, pp. 60-71.
[8] H. Chang, S. Jamin and Wenjie Wang, "Live Streaming With Receiver-Based Peer-Division Multiplexing", IEEE/ACM Transactions on Networking, vol. 19, no.1, **(2011)**, pp. 55-68.
[9] Y. Chen, B. Zhang, C. Chen, W. Meng and C. Li, "Performance Investigation on Request Collisions in Peer-To-Peer Live Streaming Systems: Problems and Algorithms", IEEE Journal on Selected Areas in Communications, vol. 31, no. 9, **(2013)**, pp. 189-199.
[10] D. Kim, E. Kim and C. Lee, "Efficient peer-to-peer overlay networks for mobile IPTV services", IEEE Transactions on Consumer Electronics, vol. 56, no. 4, **(2010)**, pp. 2303-2309.
[11] A. Ksentini, M. Naimi and A. Guéroui, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture", IEEE Communications Magazine, vol. 44, no.1, **(2006)**, pp. 107-114.

Authors

Changhee Cho received the Master Degree from the School of Interdisciplinary Program of E-Commerce, Chonnam National University in 2010. He has been studying PhD Degree from 2010 to now in School of Interdisciplinary Program of E-Commerce, Chonnam National University. Currently, he works as a CEO in XiSom Company. His research interests are E-commerce, Automation Systems, Mobile Interface Devices, and Cloud Computing.

Ha Tran-Thi-Thu is currently a M.S candidate at Smart Mobile and Media Computing Laboratory, School of Electronics and Computer Engineering, Chonnam National University, South Korea. She received B.S degree from the School of Electronics and Telecommunications, HaNoi University of Science and Technology, VietNam in 2011. She was a solution engineer at Asian Communication Solution, JSC in 2012. Her major interests are in the research areas of Mobile Cloud Computing, Next Generation of Mobile Platform, Mobile Operating System, Peer-to-Peer Network.

Sanghyun Park received his B.S. Degree in Computer and Information from the University of Korea Nazarene. He worked as an engineer in System Development Team of Media Flow Company from 2010 to 2012. He is now studying Master Degree in Electronics and Computer Engineering at Chonnam National University. His research interests are Interactive Media, Systems Development, Embedded of systems, Digital Media and Cloud computing.

JinSul Kim received the B.S. Degree in computer science from University of Utah, Salt Lake City, Utah, USA, in 2001, and the M.S. and Ph.D degrees in digital media engineering, department of information and communications from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2005 and 2008. He worked as a researcher in IPTV Infrastructure Technology Research Laboratory, Broadcasting/Telecommunications Convergence Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea from 2005 to 2008. He worked as a professor in Korea Nazarene University, Chon-an, Korea from 2009 to 2011. Currently, he is a professor in Chonnam National University, Gwangju, Korea. He has been invited reviewer for IEEE Trans. Multimedia since 2008. His research interests include QoS/QoE, Measurement/ Management, IPTV, Mobile IPTV, Smart TV, Multimedia Communication and Digital Media Arts.