ISSN: 2278-0181
NCICN-2015 Conference Proceedings

Enhancement of the Quality of Interactive Multi-View Video Streaming

Radhika K R[1*], Dr. E. Arun[2*]
[1*]PG Scholar, Dept. of computer science and engineering
[2*]Professor, Dept. of computer science and engineering
TKM Institute of Technology
Kollam, India

***Abstract*— Interactive multi-view video streaming is very popular. This trend will accelerate in the future, as personal consumer devices become equipped with multiple cameras. The real-time interactive multi-view video system allows the proactive participation of the viewers during their viewing of videos and giving them the freedom to choose a viewing angle. These circumstances lead to the study of a new system framework. The system features a proxy-server situated between the server and viewer. The proxy adapts the content data sent over the wireless channels. The best possible location for the proxy server is at the junction of the wireless and wired links. Different proxy servers together provide an environment of path diversity. Compression algorithms are used to compress the data and thereby the frame error rate is reduced to a greater extent. The proposed method adds some buffer management policies to the system. The analysis shows that the video quality and transmission parameters are improved to a greater extent.**

Index Terms*— Multi-view video multiple descriptions, multipath wireless transmission, interactive immersive video streaming, network compression.

I. INTRODUCTION

Interactive video streaming applications are more sensitive to end to end transmission delay and delay variations. Video streaming over the wireless network is a very difficult task, especially if one wants to keep video quality as it is sent by the source. Multi-view video is a sequence of spatially correlated picture frames that are captured simultaneously by multiple closely situated cameras. After storing or pre-recording one representation of a multi-view sequence at the server, clients interactively request some views they are interested in. Each client requests or plays back one single view at a time out of many pre-recorded views. The meaning is that the requested data corresponds to only a small subset out of a large set of available multi-view data at the server.

The goal is to provide a desired level of view interactivity with minimum transmission bandwidth cost. Since all cameras capture the same scene from multiple viewpoints, multi-view video consists of a large amount of inter-view spatial dependencies. A frame from one camera can be guessed not only from partially related frames from the same camera or viewpoint, but also from the frames of adjacent viewpoints. These interdependencies of frames can be used for good prediction. Streaming two viewpoints in multi-view video streaming motivates the concept of multi-path transmission. The different viewpoints are sent to the server through multiple paths in response to the client request. Path diversity decreases the load through the transmission channels. Multipath transmission provides the better utilization of the available bandwidth or the bandwidth wastage can be avoided to a greater extent.

Network compression is used to compress the video data. The probability of successful delivery of data can be increased by network compression and it also preserves the bandwidth efficiently. The rest of the paper is organized as follows. Section II explains the prior works in the field area of multi-view video streaming. The system architecture is discussed in section III. Analysis of the new framework is explained in section IV and the whole work is concluded in section V.

II. RELATED WORK

Multi-view video streaming is an emerging field. There are so many challenges happen during the streaming. Transmission delay, bandwidth usage and buffer management are the major issues of this area. The prior work [1] contains a proxy server situated next to the last hop. Multiple proxies in the network provide the path diversity concept. Path diversity and network compression is used to minimize the frame losses and error probabilities.

Transmission rate is an important factor while streaming. Frames are encoded into different formats [2] for efficient storage and to match with different transmission rates. Thus the clients get an uninterrupted service as the viewpoint of their selection changes. There are different transmission actions for multi-view video streaming. MVC and video plus depth [3] are the most popular encoding techniques. A strong comparison between them provides an optimization approach and that provides a better way to reconstruct the missing frames.

Network compression increases the probability of successful delivery of video data. Compression techniques such as MVC and Layered Coding are compared and provide a new technique MVC-CC (Multi video coding-Conditional Compression) [4] for compression. It conveys the interdependencies between the frame structures. The problem of streaming a packet through a lossy packet network introduces the concept of proxy server [5]. Proxy server is situated at the last hop and it dynamically adapts the video content sent to the client. Proxy eliminates the streaming problems in a Rate Distortion Optimization way.

The delay occurred during Network Compression (NC) introduces the Expand Window NC [6]. Frames are inserted into the window in the order of the RD-Optimized scheduler.

An international standard is described in [7] and this standard is used in most of the network environments. It is also widely used in video conferencing, digital video storage etc.. Proxy based media delivery has a greater performance compared to the server based streaming [8]. Proxy compares the backbone network conditions from the server side and also from client side. Multiple active agents strongly increase streaming performance and it depends on the network quality and packet losses.

Mobile users tend to abort more videos while streaming. This is due to the inefficient buffer management techniques. A new system CBM (Cost aware Buffer Management) in [9] introduces a good buffering technique according to the channel conditions and load. A number of encoding techniques are compared in [10] and concluded that UDMVT outperforms other techniques. The two step encoding technique reduces the latency and packet losses. A scheduling algorithm in [11] orders the frames according to the spatial correlation between them and thereby reducing the redundant data

III. SYSTEM MODEL

A. Server-Client Interaction

The video either captured or pre-recorded one. Client requests a particular video content and server streams the data in response to the client's feedbacks. The proxy server is located at the last hop. The server sends the encoded video content by comparing the adjacent viewpoints. Proxy is located between the wired (wd) and wireless (ws) channels. The framework is shown in fig 1.

Data storage and transmission delay are the major problems in video streaming. In the proposed system server contains different buffers for each proxy. The data is stored as different frames in the buffers and each of them is taken in a FIFO manner. Each proxy contains separate buffers and the frames are stored in the order send by the server. Client selects the proxies according to a buffer management algorithm.

B. Network Compression

Server compresses the video content into group of frames before transmission and for that the LZ4 compression algorithm is used. Transmission delay can be reduced by the data compression strategy. Compression process uses the fast instance of the LZ4 algorithm. Outcome of the algorithm is a flow of sequences and each sequence is less than 64kb in size. During decompression, the size can be equal to 64kb. But the last sequence is comparatively so small. The video data transmission uses the sequence identifiers and each sequence contains a pointer to the location of the next sequence. While transmission server sends two viewpoints V_j and V_{j+1} that constitute the requested multi view video V.

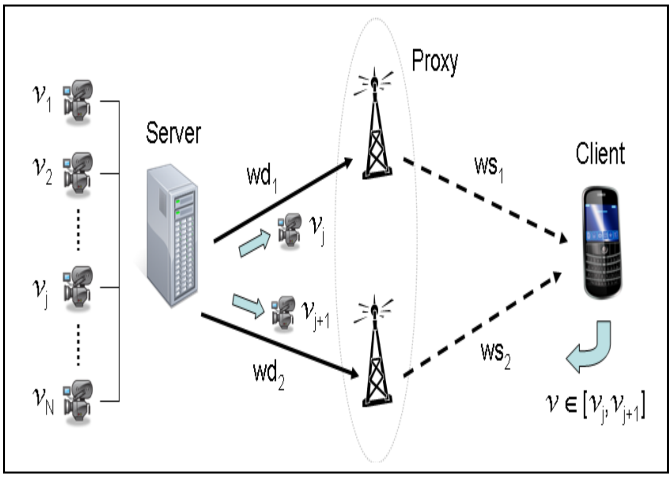

Fig. 1. Interactive multi-view video streaming

Algorithm 1. Data Compression Algorithm

Definition:
X: Offset in the compressed data stream
Y: Offset in the uncompressed data stream

1: Initialize X=0 and Y=0
2: **For** All Compressed Data Block do 3-14
3: Decode Token Byte and increment X
4: **If** Literal part is there, do 5-8
5: Compute Literal Length and increment X
6: Set Source = Compressed Data[X], Destination = Uncompressed Data [Y] and Length = Literal Length
7: Increment X as Literal Length
8: Increment Y as Literal Length
9. **End if**
10: **If** Match part is there, do 10-15
11: Get Match Offset
12: Increment X by 2 and compute Match Length
13: Increment X as the number of bytes used in Match Length
14: Set Source = Uncompressed Data[Y-Match Offset], Destination = Uncompressed Data [Y] and Length = Match Length
15: Increment Y as the Match Length
16: **End if**
17: **End for**

After compression data is stored as group of frames in the buffer. Here another problem arises that how to select the order of the proxies and a buffer management algorithm solves this problem generally. Proxy buffers are holds the major role in this buffer management.

C. Buffer Management

As the connection progress, the load on the proxy servers increases and it is difficult to select efficient proxies for each client. The procedure is based on the load at the proxy section relative to the client section. At each iteration client evaluates the capacity of each proxy and depending on the decreasing order of capacity of each proxies, connection is established between the proxies and clients. By this method clients can select efficient proxies and consequently the transmission delay also decreases and data retrieval speed increases. Algorithm 1describes the proxy selection algorithm.

Algorithm 2. Proxy Selection Algorithm:

Definition:
f_r: Frame rate of each proxy
L: Load on each proxy
E_i: Expected load on each proxy

1. Initialize x
2. Compute $E_i = f_r / L$
3. Sort E_i in decreasing order
4. **For** each proxy i in P
5. $x = x + E_i$
6. Add P_i to S
7. **if** (x>=A)
8. break;
9. **End for**
10. Connect the client to all S proxies

Buffer allocation at the proxy side is carried out using a buffer management algorithm. This method improves the storage capacity and data retrieval speed of the system. Thereby the transmission speed also improved in the proposed system. This algorithm relates the transmission speed between the client and proxy servers. Rate of loss of client requests can be avoided by this method. Transmission delay can be avoided by minimizing the delay for data retrieval. Algorithm 3 explains the buffer management algorithm.

Algorithm 3. Buffer Management Algorithm

Definition:
B= $\{b_1, b_2, \ldots\ldots\ldots b_3\}$ Set of all buffers
C_i : Initial storage of each proxy
b: maximum buffer storage required among i proxies

1. Set B as total buffer size
2. **For** each proxy i
3. $b = \max((B/C_i)*E_i)$
4. Distribute buffer size b_i among I proxies.
5. **End for**
6. Reserve buffer size R=B-b

/* As connection progress*/

7. **if** each buffer is full for each proxy i
8. **if** R is less than each buffer size
9. **for** each proxy i
10. Distribute R
11. **End for**
12. **End if**
13. **End if**
14. **Else** wait for a connection termination

IV. PERFORMANCE EVALUATION

The proposed framework provides a better buffer management technique for wireless multi-view video streaming. This algorithm improves the data streaming to a greater extend. This means that efficient data storage makes the data retrieval easier and it also reduces the transmission delay. Compression reduces the delay up to 40 percentages and the application of buffering technique improves the output more than the later. Data compression includes the system security also because the data is enveloped in a sequence or frame format.

The variation of transmission delay is shown in fig 2. Buffer management algorithm reduces the loss of packets from the clients. As video is send to the client, requests are collected in a FIFO manner. Variation in data loss is shown in fig 3. Prosy server acts as an intermediate between the server and clients and the proxy dynamically adapts the video data while transmission. Different buffers for each proxy in the server achieve the path diversity. Addition of buffer management algorithm in proxy increases the outcome of path diversity. Proxy reduces the amount of video data that needs to be send to the client.

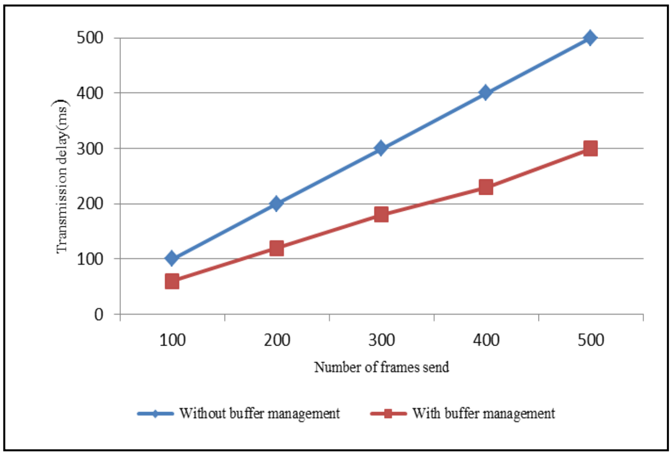

Fig. 2. Transmission delay versus number of frames

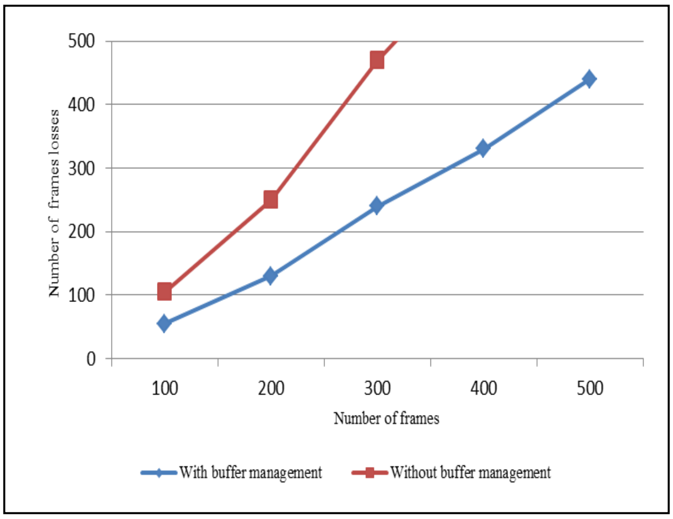

Fig. 3. Packet loss with increase in number of packets

TABLE I. KEY NOTATIONS

Key Notation	Explanation
D	Transmission delay
F	Number of frames lost
B	Set of all buffers
f	Frame rate of each proxy

While streaming there may be some delay occur between the audio and video section or otherwise the audio and video do not occur simultaneously. Loss of frame sequences causes the video become discontinuous and it makes the viewer uncomfortable. Thereby we can conclude that the video is distorted. The transmission parameters that are used for the analysis are transmission delay and number of frames lost. By data compression the frame data can be made more secured and so the data loss decreases. By buffer management data storage mechanism improves and the time taken for data retrieval decreases by the efficient buffer management algorithm. Thereby the transmission delay D also decreases.

V. CONCLUSION

The system provides an intermediate proxy server that dynamically adapts the content sends over the wireless network path. The use of multiple paths through the transport network for streaming overcomes the problem of loss and delay that affects the communication. Buffer management algorithm improves the data transmission by minimizing the transmission delay.

REFERENCES

[1] J. Chakareski.”Wireless streaming of interactive multi view video via Network compression and path diversity”. IEEE Transaction on communications.Vo.62.No.4.April 2014.

[2] Gene Cheung, Senior Member, IEEE, Antonio Ortega, Fellow, IEEE, and Ngai-Man Cheung, Member, IEEE. “Interactive Streaming of Stored Multi view Video Using Redundant Frame Structures”. IEEE transactions on image processing, vol. 20, no. 3, March 2011.

[3] J. Chakareski.” Transmission Policy Selection for Multi-View Content Delivery over Bandwidth Constrained Channels”. IEEE transactions on image processing, vol. 23, no. 2, February 2014.

[4] J. Heide, J. Sørensen, R. Krigslund, P. Popovski, J. Chakareski, and T. Larsen, “Cooperative media streaming using adaptive network compression,” in Proc. 2008 IEEE Workshop on Mobile Video Delivery at the Int’l Symp.World of Wireless, Mobile and Multimedia Networks, pp.1–7.

[5] J. Chakareski and P. Chou, “RaDiO Edge: rate-distortion optimized proxy-driven streaming from the network edge,” IEEE/ACM Trans. Networking, vol. 14, no. 6, pp. 1302–1312, Dec. 2006.

[6] Irina Delia Nemoianu, Claudio Greco, Marco Cagnazzo, B´eatrice Pesquet-Popescu. “Multi-view Video Streaming Over Wireless Networks with RD-optimized Scheduling of Network Coded Packets”.TELECOM-ParisTech, TSI department. Mar 19.2013.

[7] Telecom. Standardization Sector of ITU, “Advanced video coding for generic audiovisual services,” Recommendation ITU-T H.264—International Standard ISO/IEC 14496-10, Jan. 2012.

[8] Jacob Chakareski.” In-Network Packet Scheduling and Rate Allocation: A Content Delivery Perspective”. IEEE Transactions on Multimedia, vol. 13, no. 5, October 2011.

[9] Jian He, Zheng Xue, Di Wu, Member, IEEE, Dapeng Oliver Wu, Fellow, IEEE, and Yonggang Wen, Member, IEEE.” CBM: Online Strategies on Cost-Aware Buffer Management for Mobile Video Streaming”. IEEE transactions on multimedia, vol. 16, no. 1, January 2014.

[10] Takuya Fujihashi, Member, IEEE, Ziyuan Pan, Member, IEEE, and Takashi Watanabe, Member, IEEE.” UMSM: A Traffic Reduction Method on Multi-View Video Streaming for Multiple Users”. IEEE transactions on multimedia, vol. 16, no. 1, January 2014.

[11] Laura Toni, Member, IEEE, Thomas Maugey, Member, IEEE and Pascal Frossard, Senior Member, IEEE.” Correlation-Aware Packet Scheduling in Multi-Camera Networks”. IEEE transactions on multimedia, vol. 16, no. 2, February 2014.