A Scalable Video Retrieval In Peer To Peer Network

Linta John[1], Smita C Thomas[2], Rincy Roy Oommen[3]
[1, 2, 3] Mount Zion College of Engineering Kadamanitta, Pathanamthitta

Abstract- *Peer-to-peer networking is the scalable way for sharing the multimedia video. For a large amount of data which is distributed among different nodes, to perform retrieval function, peer-to-peer networks is an important but challenging problem. But many of the existing methods is focus on the indexing high dimensional visual features. This type of methods have scalability limitations. PEER-TO-PEER (P2P) and peer-assisted streaming systems have emerged as promising approaches for delivering multimedia content to large-scale user communities. These servers are referred to as seed servers. In current P2P streaming systems, a video is encoded at a certain bitrate, typically ranging from 300 kbps to 1 Mbps .In order to support a wider range of receivers, it is preferred to encode and serve a lower-bitrate video, but this will provide a low quality for everyone. First focus is given on the problem of efficiently allocating the resources of seed servers to requesting peers according to their demands and contributions. This allocation plays a critical role for providing a high-quality streaming service. In the proposed work recent scalable video coding technique H.264/SVC, is considered to improve this coding efficiency Congested networks and overloaded servers resulting from the ever growing number of Internet users contribute to the lack of good quality video streaming over the Internet. Caching system for streaming media utilizes its local memory and disk resources to reduce network and server load, while also improving the video and audio quality perceived by end users. In this system peers are cached based on the popularity of the segment request and a novel approach for cache replacement is considered in case of zero free space. The selfish nodes are eliminated based on the unfaithful behavior of the nodes in the network.*

Keywords- Peer to Peer network, Seed server.

I. INTRODUCTION

Peer to peer (P2P) networks is a network which consists of nodes, which are self-organized and have equal privilege. This is the most favorable architectures for data sharing.. Web pages which mostly consist of textual documents like news, articles or forums. The multimedia files have a vital role in most P2P networks which contain multimedia data. For large scale multimedia retrieval applications such as content-based image sharing, and copyright infringement detection, there is a need of P2P because of the computational power of P2P networks. In file sharing P2P networks are well known for their efficiency, scalability and robustness. During the past two decades, video coding technology has matured and state-of-the-art coding standards have become very important part of the video industry. Standards such as MPEG-2 and H.264/AVC provide strong support for digital video transmission, storage and streaming applications. PEER-TO-PEER (P2P) and peer-assisted streaming systems have emerged as promising approaches for delivering multimedia content to large-scale user communities. When concentrating on file sharing the constraint of the server's upload bandwidth and the fairness between upload and download amounts at each peer is a factor. The seed servers can be boosted to complement the limited upload capacity offered by peer and the peers are cached to obtain fairness in uploads and download. Data's are usually replicated at nodes, other than the original owners , to increase data accessibility .Some selfish peers may not follow to increase their payoff. Such nodes need to be blocked to improve the capacity in adaptive streaming. In order to solve these issues and provide data services to heterogenous users, the project is developed.

II. LITERATURE SURVEY

In this paper[1] pointed out that to improve rendered video quality and serve more receivers, peer-to-peer(P2P)video-on-demand streaming systems usually deploy seed servers. These servers complement the limited upload capacity offered by peers. In this paper, how capacity of seed servers can be optimally managed is considered, especially when scalable videostreams are served to peers. Scalable video streams are encoded in multiple layers to support heterogeneous receivers. It is showed that the problem of optimally allocating the seeding capacity to serve scalable streams to peers is NP-complete. An approximation algorithm is considered to solve it. Using the proposed allocation algorithm an analytical model is developed to study the performance of P2P video-on-demand streaming systems and to manage their resources. The analysis also provides an upper bound on the maximum number of peers that can be admitted to the system in flash crowd scenarios.

In this paper[2] focus on peer cache adaptation where each peer adjusts its cache capacity adaptively tomeet the server's upload bandwidth constraint and achieve the fairness is described. *Multiple video approaches* have been proposed recently. In these approaches, a peer stores videos that have been watched before. When a peer wants to watch a video, it first contacts its neighbors to find it from their caches. Hence, the upload burden of the server can be shared by many other peers. But the previous works based on the multiple video approach have not considered the constraint of the server upload bandwidth which is crucial to solve the scalability problem. If a peer wants segment *j* that is not cached atany peer, it should request it to the server. Each peer caches the segments it watched at its local storage and sends these when requested by some other peers. When a new segment needs to be cached, an existing cache should be deleted to create a memory space ,for that a cache replacement algorithm is used. This paper manages cache information by using the search DHT. For a cache replacement, a peer finds the number of caches for each cached segment through the search DHT and chooses a segment to be deleted in a probabilistic way. Simulation results showed this meets the server upload bandwidth constraint and achieves the fairness well at each peer.

In this paper[3]Provides a description on high-quality streaming over peer-to-peer (P2P) systems faces multiple challenges such as limited upload capacity of peers and high heterogeneity of receivers in terms of download bandwidth and screen resolutions. In this thesis, the design of a P2Plive streaming system that uses scalable video coding (SVC) as well as network coding is considered. The proposed design is simple and modular. Therefore, other P2P streaming systems could also benefit from various components of the proposed design to improve their performance. An extensive quantitative analysis to demonstrate the expected performance gain from the proposed design is conducted. The combination of video coding and network coding in live peer to peer is beneficial as it achieves: (i) significant improvement in the visual quality perceived by peers (ii) smoother and more sustained streaming rates, (iii) higher streaming capacity by serving more requests from peers.

In this paper [4] describes about the simple concept of topology in p2p live video streaming and presents a review for p2p live video streaming topology, in this paper three main types of topology that most used in p2p live video streaming systems has been discussed and shows their strength and weakness for each one of them, beside briefing for 18 studies tried to hybrid between two of the main types or using some artificial intelligence tools to improve topology performance.

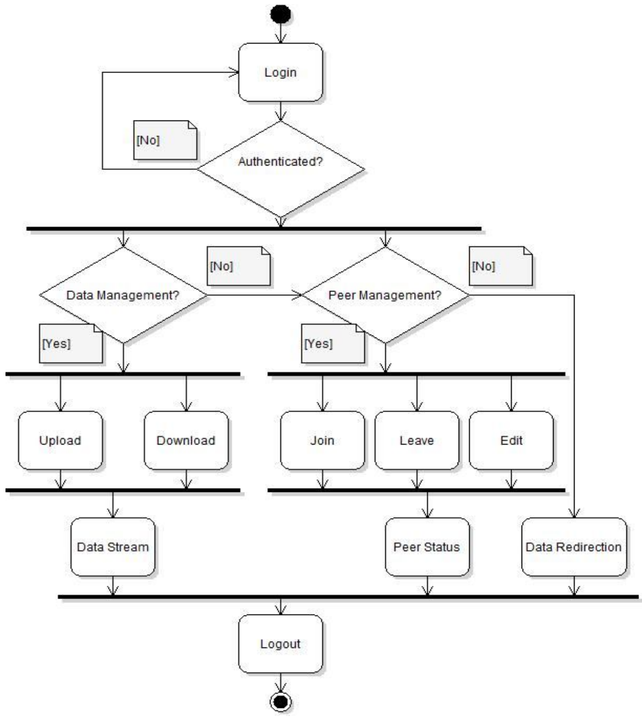
In this paper [5]implements an authentication scheme for H.264/SVC streams in a prototype called *svcAuth*, which is available as an open source library and can be employed by any multimedia streaming application as a transparent software add-on, without requiring changes to the encoders/decoders. For authentication of non scalable videos, Hash chaining is one of the simplest techniques employed. In this scheme packets of the stream are divided into blocks, each of size *n* packets. Then, the hash of each packet is attached to its previous packet, and the first packet of each block is digitally signed. Due to the one-way property of the hash function, the signature authenticates the whole block. Analysis of hash chaining is straightforward. For a block of *n* packets, hash chaining computes *n* hash values and verifies one digital signature. Hash chaining does not tolerate any packet losses. There is no receiver buffer requirement for this scheme as packets can be verified as they arrive after receiving the first packet with the signature. The sender, however, needs to wait for *n* packets to be generated, because hash chaining starts at the last packet in the block. To enable the hash chaining scheme to tolerate packet losses, the hash value of a packet is replicated and attached to multiple packets. Other authentication schemes include Augmented Hash Chaining, Butterfly Hash Chaining, SAIDA and eSAIDA, cSAIDA. The best features of cSAIDA (communication efficiency) and TFDP (computation efficiency) are combined in designing an efficient authentication scheme for streaming of pre encoded streams. The contributions of this thesis include a comprehensive analysis and *quantitative* comparison of the main schemes proposed in the literature to authenticate multimedia streams. This paper proposes a new authentication scheme for non scalable streams, which combines the advantages for on-demand streaming applications. This work but fails to propose authenticating scalable video streams to support the full flexibility of recent scalable streams.

III. PROPOSED METHOD

In the proposed method, only the server can upload videos and the client can download the videos. As this is a peer to peer network, there are a number of peers. The server may be the authenticated one. Through the network the client can download the video. Here the videos are downloaded as the packets. At the same time more than one client can download the same video but not the same packet. After the completion of the packet, the other needed peer will get that packet.

Here a video player is needed to create for playing the video which is downloaded. Once the video is downloaded it will get stored in the client system. There are three modules, The first one is the seed based capacity allocation: here the seed is the one who downloading the video. The video is downloaded as packets. The seed does not give the packet which is currently downloading by him and later it will allow to download. The second module is the broadcasting video stream: the clients must receives the list of videos which is available. The third module is the video segmentation: Asd said the video is downloaded as packets or segments, this module is for the segmentation.

And also the three algorithms which is used is the seed allocation algorithm, Cache management module give importance to cache capacity required for peers and the replacement of cache policy .This is to be followed in case of full cache. Each peer request segments it want to watch. The segment search is implemented. If a peer wants segment 'j' that is not cached at any peer it should be requested from the server. Each peer caches the segment it watches at its local cache and sends when requested by some other peers. When a new segment needs to be cached if there is no storage space available then an existing cache should be deleted to create a memory space. When peer 'i' request segment 'j' ,it can find peer 'k' which manages the list of peers that store segment 'j' peer 'k' picks up a peer1 from the list. When a peer stores a segment it may cache some other segments of the same video with high probability. If a peer 'i' played segment 'j' received from peer1 it is reasonable to search peer1 first for some other segments of the same video. Two case needs to be considered in deciding segment length. First is the case that the segment length exceeds the cache capacity of a peer. A possible problem of using a longer segment occurs when a peer watches a part of it and stops watching it or jumps to some other segments. In this case the peer is not able to cache the corresponding segment completely. Second is the case that the segment length is too short incur much overheadbecause of too many search requests. So the segment length should be decided considering these 0together. Tracker maintains per each [peer, video] pair a list of keeping track of the layers that the peer holds from each segment of the video. The number of entries in this list is the number of video segments and the peers watching and seeding others. The other two algorithma are Packet segmentation algorithm and the broadcasting algorithm.



IV. CONCLUSION

In this paper, a method is introduced for the retrieval of the video from the server as packets. This is works in a peer to peer network. More than one user can download the video. This method is a good scalable approach for the video retrieval in peer to peer network.For improving the retrieval performance, the proposed method is very effective.

V. ACKNOWLEDGMENT

We would like to thank, first and foremost, Almighty God, without his support this work would not have been possible. We would also like to thank all the faculty members of Mount Zion College of Engineering, for their immense support.

REFERENCES

[1] K. Mokhtarian and M. Hefeeda,"A Capacity Management of Seed Servers in Peer-to- Peer Streaming Systems With Scalable Video Streams" IEEE Transactions on Multimedia, vol. 15, no.1, January 2013,pp 181-194.

[2] Jongtack Kim and Saewoong,"Peer Cache Adaptation for Peer-to-Peer Video-on-Demand Streaming,"JOURNAL OF COMMUNICATIONS AND NETWORKS, VOL. 14, NO. 3, JUNE 2012.

[3] ShabnamMirshokraie, "Live peer-to-peer streaming with scalable video coding and network coding," Master's thesis, Simon Fraser Univ., Surrey, BC, Canada,2010.

[4] AmmarWaysiAlTuhafi,“A Review on Peer-to-Peer Live Video Streaming Topology” in International Journal of Computer Applications (0975 – 8887) Volume 68– No.5, April 2013.

[5] K. Mokhtarian, “Efficient and secure delivery of scalable video streams,” Master’s thesis, Simon Fraser Univ., Surrey, BC, Canada,2009.