

LIVE MULTIMEDIA STREAMING AND VIDEO ON DEMAND ISSUES AND CHALLENGES

Amandeep Singh Walia

Student, Department of Computer Science, Rayat Educational & Research Trust, Punjab, India

Abstract

Live Streaming and Video on Demand are the trending technologies nowadays over the internet. It provides the mechanism to deliver multimedia content such as audio or video to the large number of audience. However internet based services face the problem of QOS (Quality of Service) due to the instability faced in networks. Performance gets degraded when serving content to large number of consumers. Despite following the modern architectural design, precise estimate of resources such as bandwidth and server load is a challenging task. In this paper we delve into the architectural and performance issues of running these kinds of services. Our study demonstrates that the streaming architecture and Security issues are the challenges faced by these technologies. Moreover resources such as bandwidth and design of networks degrade the quality of multimedia data delivered to users. Thus in order to have best experience of streaming and Video on demand services, these issues must be addressed.

Keywords— Live Streaming, VOD (Video On Demand), P2P Streaming, Client-Server Model.

1. INTRODUCTION

With the advancement in technology and global connectivity throughout the world, it had made possible to deliver live content to the large number of end users and provide them the capability to play, seek forward and reverse the media provided by VOD. Users don't have to wait for the entire content to be downloaded; they can play as it is coming from the provider. One of the traditional approaches used to deliver the content is Client-Server model. In this model, client makes the request to the video server, server then process the request and if that content is available on the server, stream is provided to the client. CDN (Content Delivery Network) is another method used which is based on the Client-Server Model. In this model, rather than single video server, multiple servers are deployed, users than makes the request from the nearest server available. It reduces the time used to start the video and traffic on the network but major problem faced by this model is scalability. Quality of the stream is the utmost feature required by the end users. Major challenge faced by the CDN is, as the user base grows, bandwidth of the network also surges. Load balancing and replication of data helps in providing the reliable services but at the cost of increased expenditure.

Another model which can overcome these obstacles is P2P (Peer to Peer) Live Streaming model. In P2P model client can act both as server or client and fulfil the request of other clients also called as peers. Users can also share the data i.e. upload the stream slices which they have downloaded from main servers or other peers. This enables the communication between the different peers with the help of p2p network. It reduces the bandwidth load on main servers as resources of peers i.e. uploading bandwidth is used to fulfil the request of the users.

P2P technology is quite famous in file sharing applications

such as Bittorent [1] and Vuze [2]. In these services user has to wait until the video download is complete then only user can view the video, but in the case of live streaming user need to receive the video in particular order in real time. Some services like Peercast [3] have made it success to deliver media by this model to large users. As a result of these P2P VOD technology also emerged and it was tough challenge to design and deploy this service due to the random requests made by users and provide functioning like backward, forward and pause as in the case of CD player. These types of system are very complex to design and operate but if designed logically these can solve the issues of server load and bandwidth shortage.

The rest of the paper is organized as the following: Section II describes the traditional and most used Client-Server model of both streaming and VOD. Section III describes the existing P2P model of both technologies. Section IV & Section V describes issues in both P2P and VOD systems. Section VI focuses on other issues faced by these systems. Finally, paper is concluded in Section VII with summary of our work.

2. CLIENT SERVER MODEL

Video Streaming services are based on the two categories: Live Streaming and VOD (Video on Demand). Live streaming provides the capability to provide the real time viewing of the multimedia content. User is synchronized with the server and live stream is provided to the user. In the case of Video on Demand user can view video anytime regardless of timing. Users have the capability to seek forward, reverse and pause the video as in the case of the CD player. The following sections describe the different architectures used to construct these services.

2.1 CLIENT SERVER ARCHITECTURE

These types of network architecture consist of powerful computers known as servers. These special purpose computers are equipped with dedicated networks, storage devices and other resources. Client send request to the video server, it than fulfil the request of the client by providing them with the stream or video requested. Video Servers main goal is to record, capture, and store and provide the playing capability. It also performs additional tasks such as encoding the video which is input the server, then later on decoding it on the users end. Compression techniques are also used to decrease the size of the video or the streams.

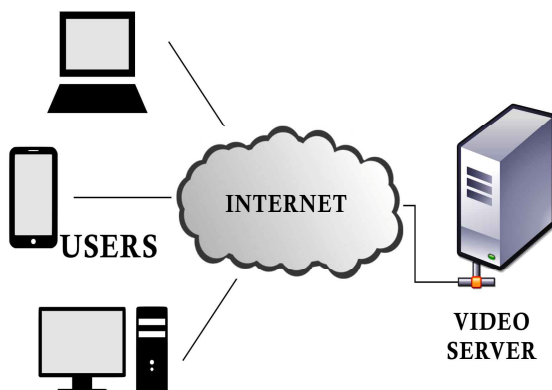


Fig. 1 Client-Server Architecture & different typed of clients attached.

Video Servers play vital role in providing streaming as well as VOD services. They handle streaming data in real time and then process the data. They make use of various protocols in Live Streaming such as RTMP (Real Time Messaging Protocol), HLS (HTTP Live Streaming) and many others [4]. They also provide synchronization with the clients and QOS (Quality of Service) mechanism so that content is delivered properly to the user. In case of VOD, when the user request the desired multimedia files server then looks for the presence of the data available on its storage. If file is found then stream is generated for the video file else error is generated. These Video Servers make the smart use of cache. Whenever there is huge demand of the similar file. Server than stores that file in the cache and provide that file from cache to the users. It saves the processing time of the servers and help in saving precious resources. These types of model generally suffer from server load problems, when there are large numbers of users requesting the stream or video files, it creates huge load on server side and leads to the bottleneck condition. This problem is generally handled by creating multiple servers.

2.2 CDN (CONTENT DELIVERY NETWORK)

As the number of users grows there is the need to increase the capacity on server side in order to provide the better services. This approach is used to manage the server load efficiently. Content Caching and Replication are the methods to efficiently handle the load on the server.[5] Web caching is used to decrease the latency and access time,

similar requests. Replication technique creates multiple copies of the data on the different servers. Thus Load Balancing can be achieved and load can be distributed among different servers thus preventing the bottleneck condition and overloading of servers.



Fig. 2 Content Delivery Network (CDN).

Fault tolerance can be achieved by the replication strategy. In case one server fails, all requests can be diverted to other servers and high up time is guaranteed, moreover latency and bandwidth can be managed in order to provide the quality service.

3. P2P ARCHITECTURE

Distributed networks are mostly used in the Internet. There are number of clients which make requests to the servers. Server capacity become inadequate as content is to be served 24x7 and QOS is to be provided equally all the time. Despite having specially designed architecture of the Server, resources are still not enough to provide continuous access as they are spread in limited circle. P2P (Peer to Peer) architecture got quite famous due to its capability of solving the problem of limited resources and providing the continuous access to the data. P2P technology makes the efficient use of resources available such as networks, storage and computing power. P2P enables the use of distributed approach and avoid the centralized approach. P2P networks consist of peers (nodes), these nodes communicate with other nodes and exchange data among themselves thus reducing the role of the central server. Nodes can themselves become server or client and serve content to other peer or receive from them at the same time. Peers can connect and leave freely and all this is handles by P2P approach automatically, without creating much impact on neighbouring peers. This architecture provides the relief to the server and bandwidth of the networks is utilised to its full extent. Peers can interactively exchange data between other peers and as the number of peers grows, data availability also increases and more number of requests can be easily fulfilled. It is opposite to the central architecture, in which the performance degrades as the number of users grows. Another feature provided by P2P technology is fault tolerance, as data is present on multiple nodes and even if the one node fails to provide data, other nodes are present to share the data. Problem of latency can also be solved as

peers can connect to their nearby location peers and content can be exchanged without any excessive delay. This leads to the better use of the bandwidth and network at the peer's side. Following diagram represents the basic design of the P2P (Peer to Peer Network) and shows the communication between the different peers.

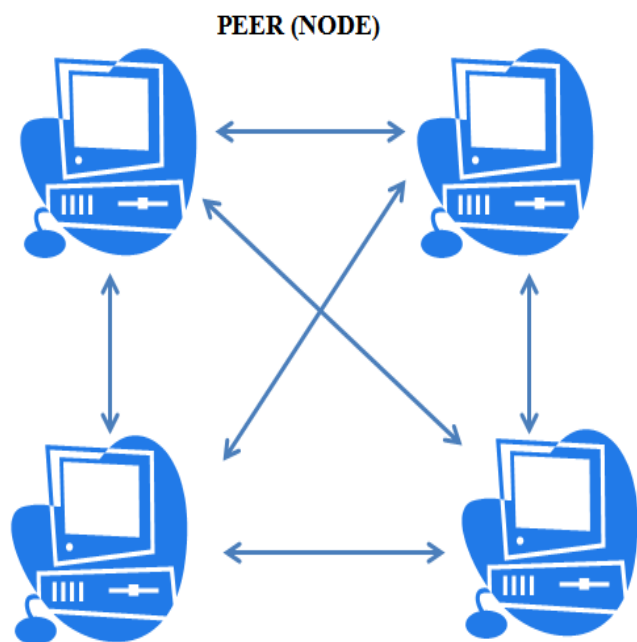


Fig. 2 Basic design of P2P Network.

The following section describes the structure of the P2P Networks and the categories into which it is divided:

3.1 TREE BASED STRUCTURE

The Tree based structure is derived from IP MULTICAST [6]. Tree like structure is formed in which there is root which is the real source and data is received by the nodes from their parent nodes. Flow of the data is generally from parent to the child node. Root node contains all the information about the source data and how it is to be delivered and shared among other nodes. Application layer is used in this structure so as to proper delivery of stream or VOD services

1) *Single-Tree Structure*: In this structure, content is served by the root node and it is delivered to the peer nodes. Hierarchy of the peers is made and they are organized in a well-defined. Root node is generally the source node and all content is first served by this root node. Flow of the stream and the multimedia file is from top to bottom i.e. Root to the Peers down in the tree. Root helps in managing the structure of the tree, when the peer join or exits the network, position of the peer and its related information is processed by the root node.

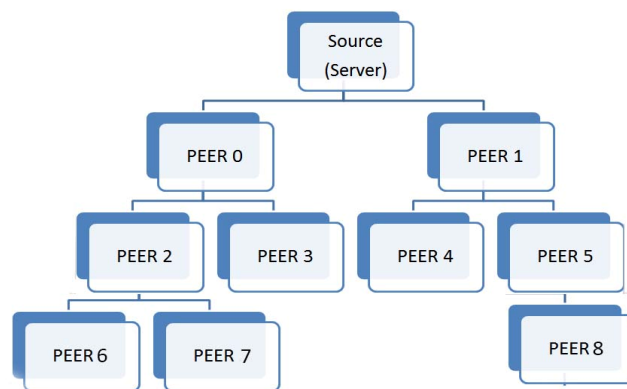


Fig. 2 Single-Tree Structure (Application Layer Multicast)

These types of trees are very difficult to manage, as peers are very dynamic in nature. They continuously enter or leave. Since data is flowing from top to down and for leaf nodes there is single flow of data, when the peer leaves, it leads to disruptions in delivery of data. So to make the recovery of the structure, Zigzag algorithm has also been developed. Despite following various measures, time taken to recover is a major concern

3.2 MULTI-TREE STRUCTURE

This structure leads to the distribution of the streams among various sub streams. Stream is provided to each of the trees coming in this structure. This helps in solving the problem of the child nodes, as now there are multiple streams of data generated from source, it prevent the structure from crashing due to the non-availability of data when the peer exits.

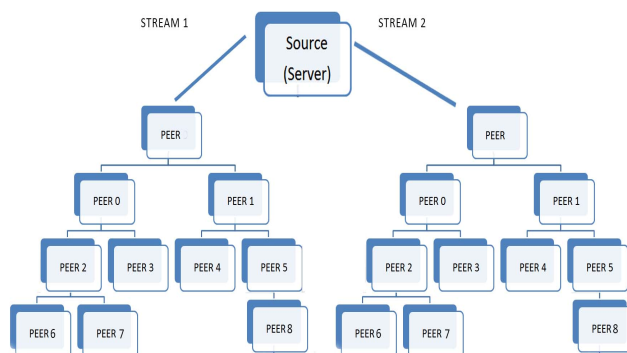


Fig. 3 Multi-Tree Structure in P2P.

4. ISSUES IN P2P LIVE STREAMING

Despite using various algorithms, there are some limitations which occur in these types of systems: These issues are as follows:

1) **UPLOAD BANDWIDTH OF PEER**: P2P models generally helps in saving the resources and provide sharing on large scale, but the whole communication depends upon the upload bandwidth available for the stream. In order to achieve the best experience of the live stream, upload speed should be greater than the rate of the stream. For achieving better upload bandwidth we need high speed connections

such as DSL or VDSL, which may lead to extra cost. If the peer is using mobile data or low speed connection, quality of the stream may be dramatically reduced.

2) **END TO END CONNECTIVITY:** Content Delivery Networks are quite famous for providing the stable and reliable stream connection from server to the client, but at the cost of the increased investment, which is quite tough for the new players. P2P tree structures provides the great flexibility to deploy the system, but as the peers are dynamic and with the varying network connectivity, smoothness of stream is difficult to obtain, thus stream timeout may occur and quality is compromised.

4) **FIREWALL OR ISP THROTTLING:** P2P is the major source of the traffic over the internet. As shown in the figure below, P2P plays major role for utilization of bandwidth. As very heavy files are shared by these P2P software, ISP try to limit the traffic from them by using speed throttling which has very negative impact on the Live Stream and also the VOD suffers slow video download rate [7]. Moreover while we are in LAN, Network administrators block these services so that bandwidth abuse should be prevented, which result in poor quality of stream and video services.

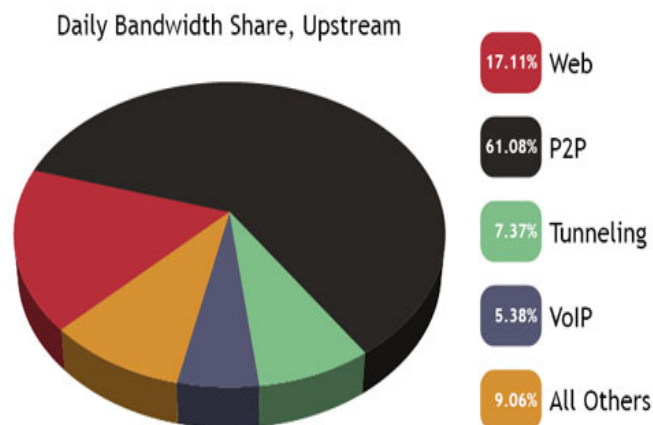


Fig. 4 Chart depicting daily upstream bandwidth usage by various services.

5. ISSUES IN P2P VOD (VIDEO ON DEMAND)

P2P VOD is quite similar to the Live Streaming because it also streams the content, but it provides the extra features by enabling the end user to seek forward reverse or pause the stream. Users may also enjoy the video of their choice. Peers share the data with their neighbouring peers. P2P also suffer from various challenges which are discussed as follow:

1) **UTMOST REQUIREMENTS:** To detect the best connection such as according to the lowest latency special components and features may be needed such as trackers and log of peers, their location and speed information. Transit servers are needed for the discovery of the peers on the network behind several networks. Special program or application, which may be provided by the VOD service provider in order to view Videos. Moreover running these services in the

browsers is a tough task due to the limited functionality, varying protocol and restriction's imposed by web browsers.

2) **SIZE OF THE SLICE:** Instead of streaming full video file, P2P VOD works by dividing the large files into small chunks which may also be referred as segments. These segments are divided in very small sizes so that transmitting and scheduling of the slices should easily take place [8]. Large slices are usually better as they provide smooth playback and less overload on peers, but these may also lead to timeout errors as large slices take more time to transmit and player may be waiting for the data and result into the error loading the stream.

3) **STORAGE PROBLEMS:** In order to serve VOD streams to the peers. Peers must have the videos or stream buffer stored on their machine memory. As the selection the video is entirely based on the user's choice. So it leads to the wastage of the memory of the peers. While Replication of the chunk some problems may occur while fulfilling the users demand as multiple movie streams may be cached by system when there is more space and different movies get upload from the peers side. It is quite challenging to handling replication task. It leads to the bandwidth wastage of the peers. In the case of Client-Server all data is stored on the server's storage devices, thus there is less overhead on user's storage devices and network.

4) **DISCOVERY OF REQUIRED DATA:** Discovery of the required content is very challenging task in the case of P2P VOD systems. As we don't know where the required content is located in the Peer Network and which peer has that content which is required by the requesting peer. Same is in the case of file sharing P2P systems. It requires the following data in order to fetch the content which is required by the requesting peer without creating much overhead:

- Trackers: They are used to maintain the list of peers and chunks which they are holding.
- DHT (Distributed Hash table): DHT provides the look up service and helps in reducing load as it describes what segments are to be provided to the trackers.
- Seed Servers: Seed servers usually hold all the data, which is provided to the peers. Even if the no peer is holding any data. Seed servers may fulfil the request. Seeding is must necessary in P2P VOD which may lead to the upload bandwidth wastage on certain seeding peers.

5) **PIECE SELECTION PRIORITY:** In multimedia data such as Videos and audios timely delivery of the segments is very critical, failed to do so will result in useless content. So selection of the piece i.e. Segment is very necessary and it is achieved by the taking care of the various steps such as in proper sequence which is nearest from the peer [9]. Selecting the rarest piece first and selecting the fixed points. There are various algorithms available. Despite following algorithms. Piece may skip due to low availability, which can result in slightly degraded performance.

6. OTHER ISSUES IN LIVE STREAMING

P2P systems are capable of handling large traffic and moreover they can be deployed on very large scale. Peers participate in this network and thus help to reduce the need of resources on the server side and moreover as the peer or viewers grow, availability and scalability of P2P streaming network also grows. But somehow dynamic nature of the P2P streaming network leads to the continuous maintenance of the system and removing any stability occurring in the network, which may lead to the increase overhead. Due to its distributed design, handling the whole network and maintaining it is a really challenging task. There are certain more problems which are discussed as follows:

1) **QOS (QUALITY OF SERVICE):** Quality of service is the parameter which defines the quality of the network. It helps to obtain the better performance of the stream, by taking care of the following things:

- Reduced throughput: Due to the dynamic nature of the peer's going and exiting the system and when the network is shared among various users, bitrate may get so much reduced that it is unable to stream the live content.
- Missed packets: When there is error in the packet header, packet may get dropped which require the retransmit of that packet, which may lead to the delay in the stream.
- Errors: Noise and fluctuation may cause the error in the packets which may lead to the missing frames in the videos or skipping audio.
- Latency: Distance from the source leads to the delay in packet arrivals and long waiting lists may increase the waiting time that leads to the reduced quality of the stream
- Jitter: Some packets may reach the users with great delays and not in proper sequence, this sequence may not be known property thus creates variation in the stream thus may adversely impact the quality of the stream.

2) **SECURITY:** Security is tough task to implement in the Live Streaming environment and P2P networks. Due to the open nature of the P2P networks, it is easy target for the attackers to attack network and inject malicious content. P2P networks are fully depended on the participating peers, using the bandwidth. If certain content demand is made from very high speed network, thus utilizing all the bandwidth, then it can result to Denial of Service. Malicious scripts may result in generating artificial queries which may result in bandwidth shortage on peer sides, if not configured properly and huge amount of traffic peak can be generated which result in leeching of the source data.

QOS is very critical in Live streaming environment. And failed to provide it so will degrade the performance and fluctuations in network traffic made by the malicious piece of code will result in the dissatisfaction of the users due to the poor quality of stream received.

Other types of attacks may include:

- Free Riders: These are those who only request the data required and do not participate in sharing them or

providing them to other [10][11]. It leads to the reduced network performance in P2P networks. These are difficult to detect in the network.

- Fake Packets: In this attacker may send messages to other peers that it has all the chunks available and at great bandwidth. Neighbouring peers may try to obtain data from this peer, attacker then send the bogus data in the P2P network, thus creating the pollution and scrambling the stream quality. P2P is improving in detecting the attacks by calculating checksum of decoded blocks and comparing the blocks with other majority of peers, and if corrupted block is detected, it is rejected and origin of the corrupted block is known. It makes a proper system to gather the list of peers who spread the defective and polluted blocks and they are blocked to share multimedia data in future to reduce the problem. This has proven to be effective in handling the pollution but leads to the huge overhead.

CONCLUSION

Live Streaming VOIP and Video On Demand (VOD) are the trending applications these days over the internet. User base of this application is increasing very rapidly and bandwidth consumption is also increasing with the use of these services. Despite the high usage of bandwidth as more number of users are joining these application day by day, Live streaming provides the method to view the video or audio in such a way that, user don't have to download the full video in order to view it instead it can be streamed to the user by encoding the videos in different qualities so as to match the requirement of the users and provide them with the great video quality. There are various architectures used to deliver this service. Client-Server architecture is one of these approaches in which there is centralized architecture. There is one source i.e. Server, whole data is stored in server and clients make the request to the server. Server fulfils the request by providing them with required stream, but it creates bottleneck at the server side, as there is huge utilization of bandwidth and resource of servers, thus leading to degradation of resources. Even the CDN has certain deployment limits as it increases the cost.

So to overcome these problems, P2P services has proven to be the great alternative, as it require less resources and there is the participation of the peer or nodes, which have sufficient resources in order to extend the network, P2P makes better use of resources of Peers and networks, thus avoiding the bottleneck problems on server side, as stream is shared among various peers. Peers can communicate and share data with other peers thus reducing the need of central server. As the peers in networks grow, it leads to high availability of the stream and better quality. Despite providing various features it still lacks in some areas, such as security which is very tough challenge to handle. Due to its open nature it is more vulnerable to attacks. And somehow due to varying network, quality of stream may not be up to mark. In this paper we have addresses various architectural and security issues and the necessary measures which can be followed to solve the problems. P2P live

streaming is still the cheapest way to share stream or files among large number of audiences.

REFERENCES

- [1]. Bittorent, *Bittorent Homepage*, <http://bittorent.com>
- [2]. Vuze, *Vuze Homepage*, <http://vuze.com>
- [3]. Peercast, *Peercast Homepage*, <http://peercast.sourceforge.net>
- [4]. Video Streaming Protocols, Cohu HD, http://www.cohuhd.com/Files/white_papers/CohuHDVideoStreamingProtocolsWhitePaper.pdf
- [5]. Internet Protocol Multicast, Ip Multicast White Papers, http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.pdf.
- [6]. The CDN, [Papers]. Available: http://www.us.ntt.net/downloads/papers/The_Value_of_a_CDN_2010.pdf
- [7]. ISP THROTTLING, <http://www.infoworld.com/article/2940538/internet/isp-s-do-throttle-traffic-and-the-fcc-cant-stop-it.html>
- [8]. Segmentation playback, Chia-Cheng Hu, Naval Academy, Kaohsiung, <https://www.computer.org/csdl/trans/td/preprint/06573953.pdf>
- [9]. Piece Selection in Bittorent, David Erman, Karel, Adrian, <http://www.diva-portal.org/smash/get/diva2:835742/FULLTEXT01.pdf>
- [10]. Free Riders in P2P, www.sciencedirect.com/science/article/pii/S1389128611003367A
- [11]. E. Adar and B.A. Huberman, "Free Riding on Gnutella," 2000, www.firstmonday.dk/issues/issue5.10/adar.pdf