P2P based Approaches for Real-Time Streaming

Ashish Kumar Yadav (07CS3028)
Naveen Kumar Sharma (07CS3027)
Rishav Agarwal (07CS1029)

1 Introduction

Peer-to-Peer (P2P) overlays offer a promising approach to stream live video from a single source to a large number of receivers (or peers) over the Internet without any special support from the network. This approach is called P2P streaming. The goal of P2P streaming mechanisms is to maximize delivered quality to individual peers in a scalable fashion while accommodating the heterogeneity and asymmetry of access link bandwidth and churn among participating peers. To effectively scale with the number of participating peers in a session, a P2P streaming mechanism should be able to utilize the contributed resources (namely outgoing bandwidth) by individual peers[1]. An emerging technology which use P2P based streaming is the IPTV. Internet Protocol television (IPTV) is a system through which Internet television services are delivered using the architecture and networking methods of the Internet Protocol Suite over a packet-switched network infrastructure[2].

2 Design Issues & Approaches

There exist two major design issues for constructing a P2P streaming network: (i) How to form an overlay topology between peers? (ii) How to deliver video content efficiently?

2.1 Tree based Approach

In this approach, the participating peers are organised into multiple, diverse tree-shaped overlays where each specific sub-stream of the live content is pushed through a particular tree from source to all interested peers. This approach has the following potential limitations: (i) In the presence of churn, maintaining multiple diverse trees could be very challenging. (ii) The rate of content delivery to each peer through individual trees is limited by the minimum throughput among the upstream connections. (iii) The outgoing bandwidth of those peers that do not have a sufficient number of child peers or an adequate amount of new content can not be effectively utilized. This in turn limits the scalability of the tree-based approaches.

2.2 Mesh based Approach

In this approach, the peers form a mesh-shaped overlay and they pull video from each other for content delivery, namely, the mesh-pull (swarming) approach. File swarming mechanisms leverage the elastic nature and the availability of the entire file at the source to distribute different pieces of a file among participating peers, enabling them to actively contribute their outgoing bandwidth through swarming. But this has two major challenges: (i) Accommodating the streaming constraint of in-time de- livery for individual packets is difficult, and (ii) Since the content is progressively generated by a live source, the limited availability of future content limits the diversity of available pieces.

Over years, many tree-push systems have been proposed and evaluated in academia and achieved some successes. However, they have never taken off commercially. Nevertheless, mesh-pull IPTV systems have enjoyed a number of successful deployments to date, such as CoolStreaming and PPLive.

3 Mesh-Pull P2P Streaming Architecture

Almost all mesh-pull P2P architectures have the following characteristics.

- The video is divided into media chunks and is made available from an origin server for broadcast. All the video information is accessible for users at the channel server.
- A host, interested in viewing the video, requests from the channel server for the available video streams.
- The tracker server maintains the list of the hosts who are interested in watching the same video.
- After a host selects its interested video, it retrieves a list of hosts currently watching the same video. The host then establishes partner relationships (TCP/UDP connections) with a subset of hosts on the list.
- These peers help each other and deliver video traffic cooperatively.

There are two major software components of a peer in a mesh-pull system. (i) Streaming Engine, and (ii) Media Player.

3.1 P2P Streaming Engine

The streaming engine has the job of:

1. Retrieving chunks from partner peers and (possibly) from the origin server.
2. Storing the retrieved chunks in a cache.
3. Sharing media chunks stored in its cache with its partners.
4. Sending a copy (of the data) of each chunk, which it receives, to the media player.

Peers download chunks from each other by sending each other buffer map messages; a buffer map message indicates which chunks a peer currently has buffered and can share. A buffer map message includes the offset (the ID of the first chunk), the width of the buffer map, and a string of zeroes and ones indicating which chunks are available (starting with the chunk designated by the offset).

3.2 Media Player

Once the media player is initialized, it sends (typically) an HTTP request to the P2P streaming engine. After having received the request, the P2P streaming engine assembles its chunks and header information into a media file and delivers the file to the media player. Because new chunks continually arrive to the streaming engine, the streaming engine continually adds data to the file. Because some chunks may not arrive before the playback deadline, there may be gaps in the received media file.

3.3 System Scalability

In mesh-pull streaming systems, participating peers are very heterogeneous, particularly in terms of the amount of upload bandwidth they contribute [2]. In addition, peers may randomly join the system, watch the video for a random period of time, and then leave the system. These two factors, peer heterogeneity and churn, bring forth the major challenges in provisioning the P2P IPTV services so that all participating peers can continuously playback the video (without freezing or skipping) with a small playback dela

4 Quality Metrics

The users viewing experience in IPTV is crucial for a successful service deployment. If users experience frequent freezes in the video playback, significant delays for startup after switching channels, or significant time lags among users for the same video frame, then the users may abandon the service.

- **Start-up Delay**: *Start-up delay* is the time interval from when one video is selected by a user until actual playback starts on his/her screen.
- **Video Switching Delay**: End-users may also switch to watch another video from the current one. Before users are able to watch the new video, the buffering in mesh-pull systems often incur *video switching delay.* These delays are, of course, significantly longer than what are provided by traditional television.
- **Playback Time Lags**: Another unfortunate characteristic of a mesh-pull P2P streaming system is the possibility of *playback time lags* among peers due to the deployment of the buffering mechanisms. Specifically, some peers watch frames in a video minutes behind other peers.
- **Frame Freezes and Skips**: When a chunk does not arrive before its playback deadline, the peer has two options: it can *freeze* the playback with the most recently displayed video frame and wait for the missing chunk to arrive; or it can *skip* the playback of the frames in the chunk and advance the deadlines for the subsequent chunks accordingly.
- **Engine Reboot**: In many P2P live streaming systems, when the playback freezes for an extended period of time, the engine terminates the connection with the player and reinializes the entire streaming process; we refer to this impairment as *rebooting.*

Various parties are interested in monitoring service quality of IPTV applications. Service providers would like to detect when service quality degrades, so that they can add additional uploading capacity for maintaining a satisfied service level. This information could be provided to users as an aid in selecting P2P video providers. It could also be provided to advertisers who wish to advertise in IPTV systems.

5 Security Concerns

The distributed P2P architecture of mesh-pull streaming systems makes them prone to various security threats.

- A malicious peer in the system may mix video stream with bogus chunks, which may significantly degrade the quality of the rendered media at the receivers.
- A peer may also advertise a large number of non-existing peers who are interested in the same channel; therefore, a legitimate peer may find it difficult to identify other legitimate peers to download video chunks.
- If malicious peers advertise that one victim host has abundant video chunks, other peers may send chunk requests to this victim host, consuming the CPU power and network bandwidth of this host. As a result, this victim host may undergo Denial-of-Service (DoS) attacks.

Due to the real-time communication in IPTV, the potential attacks on mesh-pull system can be devastating.

5.1 Defence

In defending attacks from bogus chunks or advertisements, chunk signing is an effective mechanism. In chunk signing techniques, the so-called authentication information, or signature, needs to be transmitted to the receivers along with the chunks. This authentication information can either be provided by the source (in which case the load on the source might be high) or could be distributed through the P2P system itself, in the form of a separate stream or be piggybacked with video chunks. A peer receives each chunk and its corresponding signature one by one, verifies its integrity and plays back (and forwards) only if the chunk is valid, otherwise rejects the chunk as being polluted.

The current practice of mesh-pull P2P streaming systems demonstrate the feasibility of large-scale application layer multicast on top of the best-effort Internet.

References

1. IPTV over P2P Streaming Networks: the Mesh-pull Approach, Xiaojun Hei, Yong Liu and Keith W. Ross
2. PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming, Nazanin Magharei, Reza Rejaie