Survey and Problem Statement of P2P Streaming

Ning Zong

Huawei Technologies

1 Survey of Some Popular P2P Streaming Systems

1.1 PPLive

System Architecture and Working Flows of PPLive

The PPLive software, running in user computers (peers), has two major communication protocols: (i) a registration and peer discovery protocol; and (ii) a P2P chunk distribution protocol. Figure 1 depicts an overview of the registration and peer discovery protocol [5]. When an end-user starts the PPLive software, it joins the PPLive network and becomes a PPLive peer node. The first action (step 1) is an exchange with the PPLiveWeb site to retrieve a list of channels distributed by PPLive. Once the user selects a channel, the peer node registers with the bootstrap root servers (trackers) and requests a list of peers that are currently watching the channel (step 2). The peer node then communicates with the peers in the list to obtain additional lists (step 3), which it aggregates with its existing list. In this manner, each peer maintains a list of other peers watching the channel.

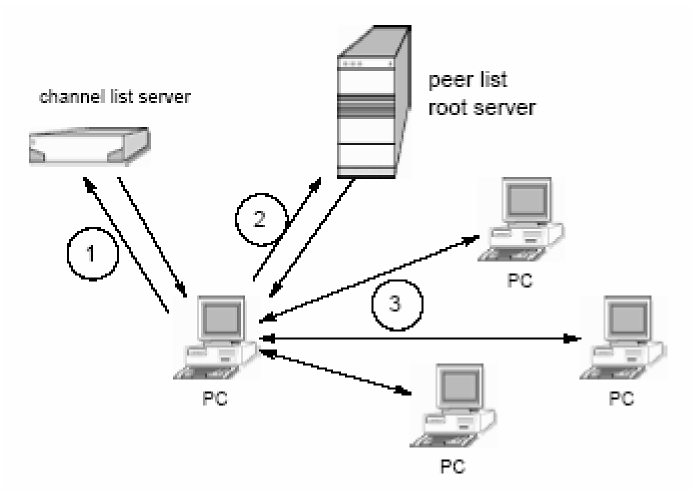

Figure 1, Registration and Peer Discovery of PPLive

At the video-on-demand (VoD) instant, because different peers watch different parts of the channel, a peer buffers up to a few minutes worth of chunks within a sliding window to share with each other. Some of these chunks may be chunks that have been recently played; the remaining chunks are chunks scheduled to be played in the next few minutes. Peers upload chunks to each other. To this end, peers send to each other "buffer map" messages; a buffer map message indicates which chunks a peer currently has buffered and can share. The buffer map message includes the offset (the ID of the first chunk), the length of the buffer map, and a string of zeroes and ones indicating which chunks are available (starting with the chunk designated by the offset).

Figure 2 illustrates a buffer map [5]. A peer can request, over a TCP connection, a buffer map from any peer in its current list of peers. After a peer A receives a buffer map from peer B, peer A can request one or more chunks that peer B has advertised in the buffer map. A peer may download chunks from tens of other peers simultaneously. PPLive continually searches for new partners from which it can download chunks.

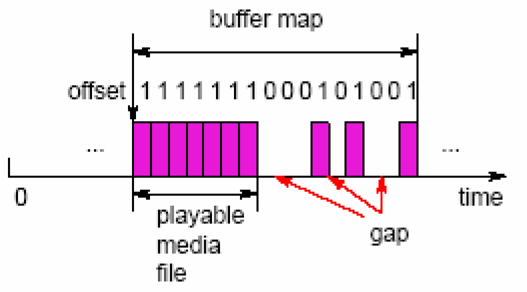

Figure 2, Peer's Buffer Map of PPLive

PPLive works in conjunction with a media player (either Windows Media Player or RealPlayer). Figure 3 illustrates the interaction between the PPLive peer software and the media player [5]. The PPLive engine, once having buffered a certain amount of contiguous chunks, launches the media player. The media player then makes a request to the PPLive engine, and the PPLive engine responds by sending video to the media player. The media player buffers the received video; when it has buffered a sufficient amount of video content, it begins to render the video.

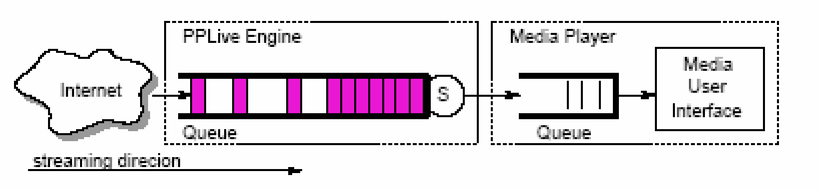

Figure 3, PPLive Engine and Media Player

Traffic Pattern of PPLive

(1) PPLive transfer data mainly using UDP, a few TCP;

Video Download Policy of PPLive

(2) Top ten peers contribute to a major part of the download traffic. The top peer session is quite short comparing to the trace duration. This would suggest that PPLive gets the video from only a few peers at the same time and switches periodically from one peer to another;

(3) PPLive can send multiple chunk requests for different chunks to one peer at one time;

Peer List Maintenance of PPLive

(4) PPLive maintains a constant peer list with relatively small number of peers. [7]

1.2 PPStream

System Architecture and Working Flows of PPStream

The system architecture and working flows of PPStream is similar to PPLive.

Traffic Pattern of PPStream

(1) PPStream transfer data mainly using TCP, a few UDP;

Video Download Policy of PPStream

(2) Top ten peers do not contribute to a large part of the download traffic. This would suggest that PPStream gets the video from many peers at the same time and its peers have long session duration;

(3) PPStream does not send multiple chunk requests for different chunks to one peer at one time;

Peer List Maintenance of PPStream

(4) PPStream maintains a constant peer list with relatively large number of peers. [7]

1.3 SOPCast

Traffic Pattern of SOPCast

(1) SOPCast transfer data mainly using UDP, a few TCP;

Video Download Policy of SOPCast

(2) Top ten peers contribute to about half the total download traffic. SOPCast download policy looks like PPLive policy in that it switches periodically from provider peer. However, SOPCast seems to always need more than a peer to get the video compare to PPLive where a single peer could be the only video provider;

Peer List Maintenance of SOPCast

(3) SOPCast's peer list can be as large as PPStream's one but fluctuates. [7]

1.4 TVAnts

Traffic Pattern of TVAnts

(1) TVAnts is more balanced between TCP and UDP in data transmission;

Video Download Policy of TVAnts

(2) Top ten peers contribute to about half the total download traffic (like SOPCast), but top peer does not contribute to a large amount of the total traffic (like PPStream). TVAnts top peer does not contribute as few as PPStream's one but does not stay as long as PPStream top peer;

Peer List Maintenance of TVAnts

(3) TVAnts' peer list is also large and fluctuates. [7]

2 Common Features of P2P Streaming Systems

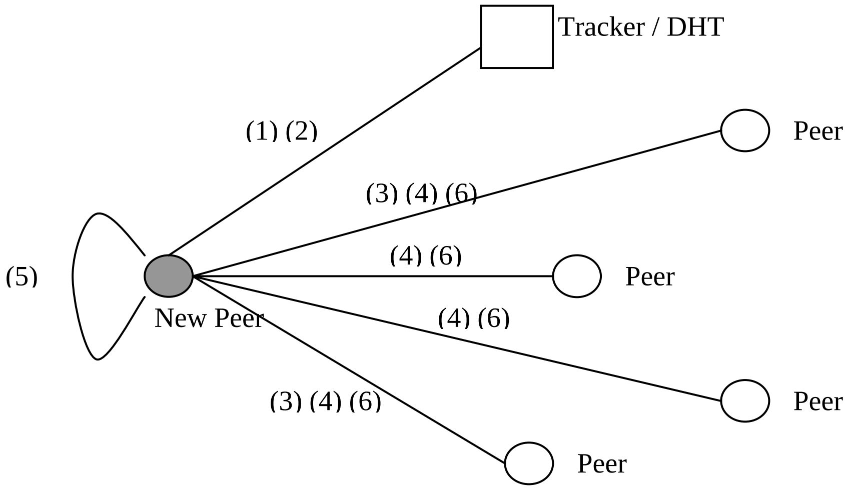

Figure 4, Main Components and Steps of P2P Streaming System

The main components of P2P streaming system consist of tracker and peers.

The main steps of P2P streaming system are:

(1) A new peer registers with tracker / dynamic hash table (DHT) to join the peer group which shares a same channel / media content;

(2) Tracker / DHT returns an initial peer list to the new peer;

(3) The new peer harvest peer list by gossiping (i.e. exchange peer list) with the peers in the initial peer list to aggregate more peers sharing the channel / media content;

(4) The new peer randomly (or with some guide) selects some peers from its peer list to connect and exchange peer information (e.g. buffer map, peer status, etc) with connected peers to know where to get which data;

(5) The new peer decides which data should be requested in which order / priority using scheduling algorithm and the peer information obtained in Step (4);

(6) The new peer requests the data from some connected peers.

3 Problem Statement of P2P Streaming Protocol

3.1 Need for P2P Streaming Protocol

P2P streaming applications attract more and more users to consume video content online [1][2][3][4]. At the instant of VoD, different users watch different parts of the video content so that a peer normally maintains a buffer to share video content with other peers, as introduced in Section 1. At the instant of live content, because all peers are only interested in what is actually happens "now", one possible role of a peer is to request video content from live source and then forward the content to more peers, hence reduce the work load of the live source [10].

P2P streaming applications adopt decentralized streaming architecture where the media content is shared among peers who not only download but also upload media content to each other. The advantages of this decentralized streaming architecture include less workload (hence reduced cost) on streaming servers, and better streaming scalability on large number of users. However, most current P2P streaming applications make use of proprietary protocols, which is impossible for various applications (e.g. web services, IPTV, content distribution, etc) to reuse all or part of their components to implement decentralized streaming. Therefore, an open and standard protocol for P2P streaming (PPSP) defined in IETF would greatly benefit more and more applications through decentralized streaming architecture which enables reduced cost on infrastructure (e.g. media servers) and better scalability on increased number of users. A possible scenario is that in content distribution network (CDN) deployed by content providers, PPSP can be used to reduce the streaming load of edge servers and improve the streaming scalability by sharing the media among the users (i.e. peers) as well as the edge servers.

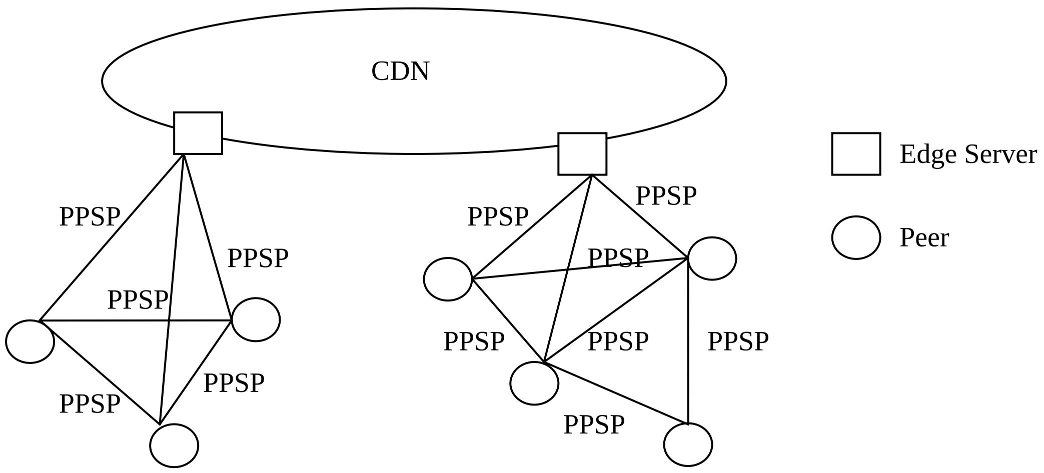

Figure 5, PPSP Scenario – Hybrid CDN and P2P Streaming

Why not P2PSIP? P2PSIP is IETF WG focusing on the protocols for distributed resource location [12]. However, in P2P streaming, the content information of each peer (e.g. buffer map) is highly dynamic and real-time, which means that simply maintaining these highly dynamic information in the P2PSIP network may cause overload of the latter. Therefore, in P2P streaming, it is better to keep the content information locally in distributed peers and use PPSP to discover which peer has which content. Nevertheless, P2PSIP network can be reused to replace the tracker to implement distributed peer registration.

3.2 Scope of P2P Streaming Protocol

The basic role of PPSP is to discover distributed real-time content in peer-to-peer streaming scenario, i.e. the key of PPSP is to find which peer has which content. Based on the survey of P2P streaming systems, the core part of PPSP can be a set of signaling protocol to implement the negotiation between peers about: (1) the content information of each peer (e.g. buffer map); (2) any other peer information related to service provisioning. The core part of PPSP corresponds to Step (3) and (4) in Section 2 and Step (3) of harvesting peer list could be optional operation.

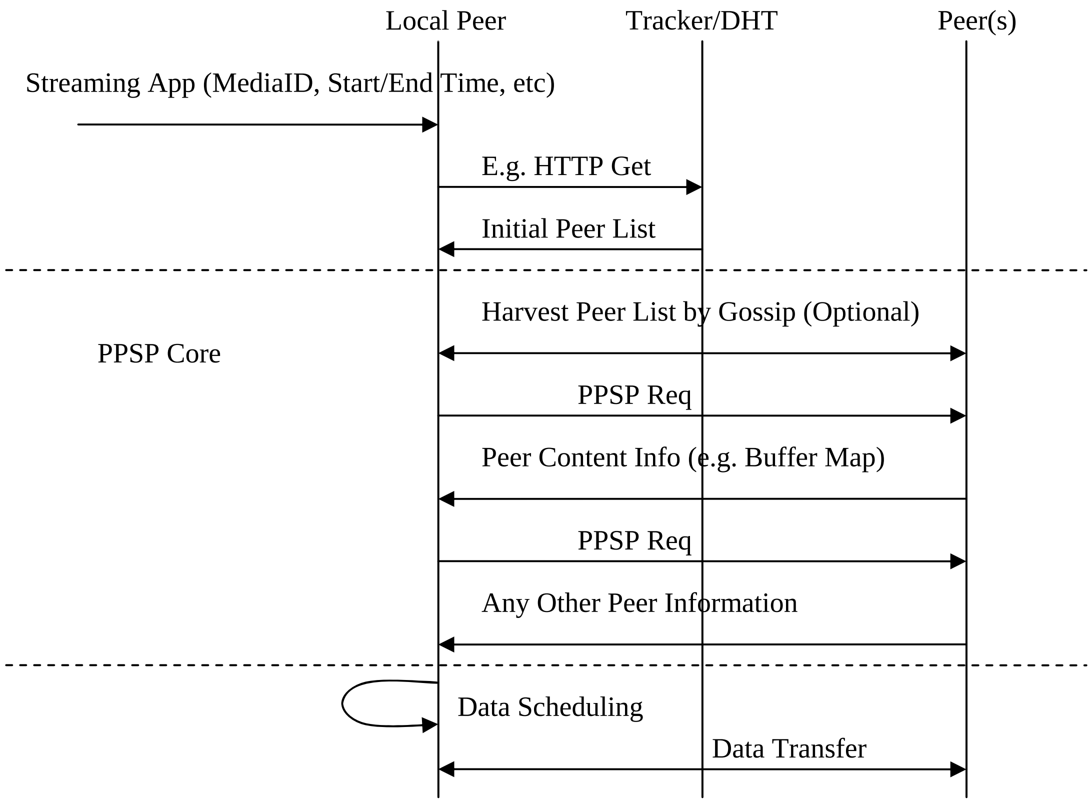

Figure 6, PPSP Core

4, References

[1] www.pplive.com

[2] www.ppstream.com

[3] www.sopcast.com

[4] www.tvants.com

[5] A Measurement Study of a Large-Scale P2P IPTV System. Xiaojun Hei, Chao Liang, Jian Liang, Yong Liu, Keith Ross. Polytechnic University, NY, USA.

[6] A Survey on Peer-to-Peer Video Streaming Systems. Yong Liu, Yang Guo, Chao Liang. Polytechnic University, NY, USA.

[7] Measuring P2P IPTV Systems. Thomas Silverston, Olivier Fourmaux. Universit ´e Pierre et Marie Curie, Paris, France.

[8] Challenges, Design and Analysis of a Large-Scale P2P-VoD System. Yan Huang, Tom Z. J. Fu, Dah-Ming Chiu, John C. S. Lui, Cheng Huang. Shanghai Synacast Media Tech., The Chinese University of HongKong, China.

[9] Understanding User Behavior in Large-Scale Video-on-Demand Systems. Hongliang Yu, Dongdong Zheng, Ben Y. Zhao, Weimin Zheng. Tsinghua University, China, U.C. Santa Barbara, USA.

[10] Adaptive Queue-based Chunk Scheduling for P2P Live Streaming. Yang Guo, Chao Liang, Yong Liu. Thomson Lab, NJ, USA, Polytechnic University, NY, USA.

[11] Real Time Streaming Protocol 2.0 (RTSP). Draft-ietf-mmusic-rfc2326bis-18.txt.

[12] Resource Location and Discovery (RELOAD). Draft-ietf-p2psip-reload-00.txt.