Computer Networks 56 (2012) 3432–3445

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Hierarchically Clustered P2P Video Streaming: Design, implementation, and evaluation

Yang Guo [a,*,1], Chao Liang [a], Yong Liu [b]

[a] Alcatel-Lucent, Holmdel, NJ 07733, United States
[b] Polytechnic Institute of NYU, Brooklyn, NY 11201, United States

ARTICLE INFO

Article history:
Received 13 October 2010
Received in revised form 6 November 2011
Accepted 30 June 2012
Available online 17 July 2012

Keywords:
Peer-to-Peer
Streaming
Hierarchical clustering
Optimal scheduling

ABSTRACT

P2P based live streaming has been gaining popularity. The new generation P2P live streaming systems not only attract a large number of viewers, but also support better video quality by streaming the content at higher bit-rate. In this paper, we propose a novel P2P streaming framework, called *Hierarchically Clustered P2P Video Streaming, or HCPS*, that can support the streaming rate approaching the optimal upper bound while accommodating large viewer population. The scalability comes with the hierarchical overlay architecture by grouping peers into clusters and forming a hierarchy among them. Peers are assigned to appropriate cluster so as to balance the bandwidth resources across clusters and maximize the supportable streaming rate. Furthermore, individual peers perform *distributed queue-based scheduling algorithms* to determine how to retrieve data chunks from source and neighboring peers, and how to utilize its uplink bandwidth to serve data chunks to other peers. We show that queue-based scheduling algorithms allow to fully utilize peers' uplink bandwidths, and HCPS supports the streaming rate close to the optimum in practical network environment. The prototype of HCPS is implemented and various design issues/tradeoffs are investigated. Experiments over the PlanetLab further demonstrate the effectiveness of HCPS design.

1. Introduction

Video over Internet has been gaining popularity due to the fast penetration of high-speed Internet access, expanding group of savvy broadband users, and rich video contents available over the Internet. ISPs are aggressively rolling out new infrastructure enhanced with advanced protocols to offer customers Internet TV services. Video traffic is expected to dominate the Internet in near future. Traditionally, video content is streamed to end users either directly from video source servers, or indirectly from edge servers belong to Content Distribution Network (CDN). Peer-to-Peer video streaming has emerged as an alternative with low infrastructure cost. P2P streaming systems, such as CoolStreaming [1], PPLive [2], and SopCast [3], attract millions of users to watch live or on-demand video programs. The existing P2P based systems show remarkable capability of handling large viewer population, and being robust against peer churns and dynamic network environment.

P2P design philosophy seeks to utilize peers' upload bandwidth to reduce server's workload. The maximum video bit-rate that can be serviced over a P2P system is determined by the server's upload capacity and peers' average upload capacity [4]. The recent study [5] suggests that simple scheduling algorithm employed by many P2P systems is unable to fully utilize peers' upload capacity. Higher streaming rate equates better video quality. The capability to support high bit-rate streaming also provides more cushion to absorb the bandwidth variations in case

* Corresponding author.
E-mail addresses: yang.guo@alcatel-lucent.com (Y. Guo), chao_c.liang@alcatel-lucent.com (C. Liang), yongliu@poly.edu (Y. Liu).

[1] The work in this paper was conducted when Yang Guo was with Technicolor Corporate Research.

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.06.020

the constant-bit-rate (CBR) video is broadcasted. A new P2P streaming framework that is capable of supporting high video bit-rate and accommodating a large number of users is highly desirable.

Most existing P2P streaming solutions maintain a loosely connected mesh to accommodate a large number of users. Individual peers share data with a small set of neighboring peers. While such random mesh is scalable, the study in [6] suggests that the supportable video bit-rate in a random mesh is throttled by the *content bottleneck*, i.e., a peer's upload bandwidth cannot be utilized if it does not have *fresh* video data for its neighbors. More recently, several intelligent scheduling algorithms capable of fully utilizing peers' upload capacities have been developed [4,7,8]. These scheduling algorithms can achieve the maximum streaming rate if peers are connected in a full mesh. The requirement of fully connected mesh, however, confines the system scalability. It is unrealistic to maintain hundreds or thousands of peering connections on a peer.

In this paper, we propose *Hierarchically Clustered P2P Streaming system (HCPS)* that supports the streaming rate approaching the optimum yet scales to host a large number of peers. P2P overlay topology and distributed chunk scheduling algorithms running at individual peers collectively determine the performance of a P2P streaming system. Accordingly, we address the design challenges following a two-step approach. First, we propose a hierarchically clustered P2P overlay that scales to host a large number of users/peers. In addition, the overlay is designed as such that its *maximum supportable streaming rate*, defined as the maximum streaming rate allowed by a P2P overlay, approaches the optimum upper bound. Second, we develop *distributed queue-based chunk scheduling algorithms* that actually achieves the maximum supportable streaming rate allowed by the HCPS overlay in spite of the large number of users/peers. The main contributions of this paper can be summarized as follows:

- We propose a novel P2P streaming framework that is scalable and supports the streaming rate approaching the optimum upper bound. The optimality is proved analytically and evaluated through experiments.
- The full-fledged prototype is implemented. Various design considerations are explored to handle dynamics in realistic network environments, including peer churns, peer bandwidth variations, and inside network congestion.
- The performance of the prototype system is examined through experiments over PlanetLab [9]. Both the optimality and the adaptiveness of the proposed chunk scheduling method are demonstrated.

The remaining paper is organized as follows. Related work is presented in Section 2. In Section 3, overlay construction of Hierarchically Clustered P2P Video Streaming (HCPS) is presented. In Section 4, distributed queue-based chunk scheduling algorithms are described. Section 5 discusses the design issues in implementing HCPS prototype. Section 6 presents the experiment results of HCPS over PlanetLab. Finally, conclusions and discussions are included in Section 7.

2. Related work

P2P technology has become an effective paradigm for building distributed networked applications. P2P based file sharing [10], voice-over-IP [11], and video streaming services [12,1–3] all achieve admirable success, attracting a large number of users and changing the way *digital goods* are delivered over the Internet. According to the overlay structure, P2P systems can be broadly classified into two categories: tree-based systems and mesh-based systems. The tree-based systems, such as ESM [12], have well-organized overlay structures and typically distribute video by actively pushing data from a peer to its children peers. In contrast, a mesh-based system is not confined to a static topology. A peer dynamically connects to a subset of random peers in the system based on the content and bandwidth availability on peers. Video chunk is pulled by a peer from its neighbor who has already obtained the chunk. The study in [13] shows that the mesh-based scheme is superior over the tree-based scheme thanks to the dynamic mapping of content to the delivery paths.

Despite the success of mesh-based P2P streaming systems, the *quality of experience* perceived at end users requires further improvement in terms of video quality, startup delay, and playback smoothness. Measurement study on PPLive [14,15] showed that most programs have bit rates around 400 kbps and the start-up delays for a channel range from a few seconds to a few minutes. Performance comparison study [5] further sheds lights on the potential root cause of limited streaming rate currently supportable over the Internet. It turns out a naieve mesh-based P2P scheme is not able to efficiently utilize the available bandwidth resources available in the P2P system.

Several efforts have been made to improve the resource utilization. [6] proposes a two phase swarming scheme where the fresh content is diffused to the entire system in the first phase, and peers exchange available content in the second phase. Network coding has been applied to P2P streaming and a reality check is done in [16]. [17] further proposes a P2P live streaming protocol that takes full advantage of random network coding to improve the overall performance. Neither above approaches, however, can be proved to optimally utilize the bandwidth resources. [4] develops a centralized solution that fully utilizes peer uploading bandwidth and achieves the streaming rate upper bound. [7] designs an optimal randomized distributed algorithm. [8] further expands the result in [7] and designs a set of pushed-based schemes. [18] analyzes pull-based streaming protocol and shows that the streaming rate can be near optimal if the long delay and large signaling overhead are tolerable. A hybrid push–pull based scheme is proposed mitigating some of the issues. The aforementioned studies, nevertheless, always require the assumption of the fully connected mesh in their optimality proofs. HCPS overcomes the challenge by introducing hierarchically clustered P2P overlay and distributed queue-based scheduling algorithms, which are provably optimal with small startup delay. Full-fledge prototype further allows us to address realistic implementation issues, and evaluate the system over the real network.