Computer Networks 57 (2013) 2856–2868

Contents lists available at SciVerse ScienceDirect

Computer Networks

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

An efficient scheduling algorithm for scalable video streaming over P2P networks ☆

Kai-Lung Hua [a,*], Ge-Ming Chiu [a], Hsing-Kuo Pao [a], Yi-Chi Cheng [b]

[a] Dept. of Computer Science & Information Engineering, National Taiwan University of Science & Technology, Taipei 106, Taiwan
[b] Apexx Technology Corp, Taoyuan 325, Taiwan

ARTICLE INFO

Article history:
Received 29 February 2012
Received in revised form 15 June 2013
Accepted 18 June 2013
Available online 6 July 2013

Keywords:
P2P streaming
Scalable video coding
Block scheduling algorithm

ABSTRACT

During recent years, the Internet has witnessed rapid advancement in peer-to-peer (P2P) media streaming. In these applications, an important issue has been the block scheduling problem, which deals with how each node requests the media data blocks from its neighbors. In most streaming systems, peers are likely to have heterogeneous upload/download bandwidths, leading to the fact that different peers probably perceive different streaming quality. Layered (or scalable) streaming in P2P networks has recently been proposed to address the heterogeneity of the network environment. In this paper, we propose a novel block scheduling scheme that is aimed to address the P2P layered video streaming. We define a soft priority function for each block to be requested by a node in accordance with the block's significance for video playback. The priority function is unique in that it strikes good balance between different factors, which makes the priority of a block well represent the relative importance of the block over a wide variation of block size between different layers. The block scheduling problem is then transformed to an optimization problem that maximizes the priority sum of the delivered video blocks. We develop both centralized and distributed scheduling algorithms for the problem. Simulation of two popular scalability types has been conducted to evaluate the performance of the algorithms. The simulation results show that the proposed algorithm is effective in terms of bandwidth utilization and video quality.

1. Introduction

With the widespread deployment of broadband technology, multimedia service has attracted a large number of users from the Internet. Many multimedia platforms over the Internet, such as YouTube and NetTV, have been presented to the public in recent years. One of the major challenges faced by multimedia streaming services is serving a massive number of concurrent users online. In this respect, IP multicast seems to be one of the most promising approaches. The deployment of IP multicast, however, turns out to be limited as a result of a variety of technical and market reasons, such as the difficulty of deploying inter-ISP multicast, and unwillingness of users to install multicast-capable routers. Recently, the idea of the application-layer technique in peer-to-peer (P2P) environment has been proposed as an alternative solution. In a P2P streaming system, an unstructured overlay network is first constructed for a collection of nodes (or users). Through the overlay network, the participant nodes cooperate to exchange multimedia contents to meet their respective streaming demands. Several P2P streaming systems, such as CoolStreaming [1], Bittos [2], PPTV [3], and Mol's system [4], have already been deployed in practice and are considered as promising solutions for serving large-scale

☆ This work was supported by National Science Council of Taiwan via 101-2221-E-011-138.
* Corresponding author. Tel.: +886 2 2730 3664.
E-mail addresses: hua@mail.ntust.edu.tw (K.-L. Hua), chiu@mail.ntust.edu.tw (G.-M. Chiu), pao@mail.ntust.edu.tw (H.-K. Pao), poca.cheng@apexx.com.tw (Y.-C. Cheng).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.06.014

users owing to the self-scalability property of the P2P technique.

These P2P streaming systems operate in a similar way as the BitTorrent [5], with each node being assigned a set of neighboring peers over an overlay network. The streaming media contents are divided into blocks of data for exchanges among the peer nodes. At any given instant of time, each peer may possess some subset of the data blocks with different peers having different content subsets. A node periodically requests each of its neighboring peers for the list of blocks in cache. With this information, the node then issues requests for blocks of interest from its neighbors. On the other hand, each responding node in the network typically allocates only limited upstream bandwidth for answering data requests issued independently by several peer nodes that arrive at the same time. Therefore, it remains a challenge for a node to determine the specific data blocks to retrieve from each of the neighboring nodes under the constraint of the allocated bandwidth. The decision of course directly affects the performance of the system and is generally known as the *block scheduling problem*.

Most of existing P2P schemes adopt ad hoc-like strategies such as random selection [6–9], rarest first [5,1], or round-robin [10] for block scheduling. Note that these ad hoc scheduling methods are not effective for video streaming because when a node requests the media data blocks from its neighbors using these methods, it does not consider the different importance of its desired blocks. Furthermore, to handle the heterogeneity of users' network environment, the layered encoding technique [11] has been employed to encode the video contents into multiple non-overlapping layers. The fundamental data of the video contents is contained in the *base layer*. The rest of the layers, called *enhancement layers*, are used to enhance the quality of video playback in a progressive manner. To decode a high layer of data, all lower layers, including the base layer, must be available to support the decoding function. These layers offer the flexibility of meeting distinct requirements for the streaming quality demanded by different users. But the design also raises new challenges for P2P streaming applications. In particular, it complicates the block scheduling problem.

Recently, a number of works have employed the layered coding technique in their P2P streaming systems. Rejaie and Ortega [12] presented a framework for layered P2P streaming, where the round-robin mechanism is employed for requesting video layers for each peer. Lan et al. [13] proposed a scheduling algorithm, which enables scalable streaming for low- to high-bandwidth contents, for peers to request data from the senders. Zhang et al. [14,15] specifically addressed the block scheduling problem from the theoretical perspective of optimizing network performance. The work modeled the problem as a min-cost network flow problem and proposed a centralized optimal solution accompanied by a distributed heuristic version. All of the mentioned works confirm the viability of layered streaming in different aspects.

With layered encoding, the layers can represent temporal scalability (frame rate), spatial scalability (picture size), or quality scalability for video [11]. Spatial scalability modality generally makes block sizes of different layers non-uniform, and thus, requiring distinct bandwidth to transport a block of data in each layer. In addition, some existing techniques, such as Scalable Video Coding (SVC), combine different scalability modalities to enhance video quality. These coding techniques often yield multiple layers with non-uniform block sizes. The non-uniformity of block size introduced by the layered coding techniques leads to further challenges for the block scheduling problem, and renders existing block scheduling algorithms either inapplicable or inadequate in offering satisfactory performance for users. Hence, there is a need to investigate block scheduling for P2P streaming from the perspective of *non-uniform* block size, in order to cope with a more realistic operating paradigm.

The block scheduling algorithms are generally classified into two categories: *push-based* [16,17] and *pull-based* [14], according to either it is the sender or the receiver that decides which blocks to transmit, respectively. In this paper, we present a novel solution to the pull-based block scheduling problem. A priority-based model is introduced to determine the weight of a desired block. The priority of a block is designed as a function of block rarity, playback urgency, the layer of the block, and the block size. The priority function is unique in that it strikes good balance between different factors, which makes the priority of a block well represent the relative importance of the block over a wide variation of block size between different layers. More importantly, the priority function employs no weighting parameters and hence requires no time-consuming try-and-error or training to obtain a proper set of these weighting parameters. In general, the weighting parameter settings (if exist) are essential to the system performance; however it is not only troublesome to adjust the optimal parameter setting whenever a system condition changes, but also difficult to tune these weighting parameters to optimize the performance given a system setting. The system can significantly enhance the quality of video playback by maximizing the sum of priorities of the blocks that are exchanged between the peers. To this end, we have first studied the block scheduling problem from a centralized perspective, which is then extended to a distributed version of the block scheduling scheme for pragmatic purpose. Simulation-based evaluation has been conducted to evaluate the performance of the proposed algorithm with respect to various performance metrics, including layer delivery ratio, network goodput, useless throughput, and peak signal-to-noise ratio (PSNR). The simulation results show that the proposed algorithm successfully avoids delivering undecodable blocks, and hence better utilizes the bandwidth to improve the visual quality.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes the network model used in this paper, followed by the proposed block scheduling algorithm. In Section 4, we conduct simulation to evaluate the performance of our algorithm and compare the results with existing solutions. Finally, we conclude the paper in Section 5.

2. Related work

P2P video streaming systems comprise of two major operations: overlay construction and block scheduling. For