Information Sciences 298 (2015) 425–446

Contents lists available at ScienceDirect

Information Sciences

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

Peer-assisted multimedia delivery using periodic multicast

Younghoon Kim [a], Yusung Kim [b], Hyunsoo Yoon [a], Ikjun Yeom [b,*]

[a] Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, South Korea
[b] Department of Computer Engineering, Sungkyunkwan University, Suwon, South Korea

ARTICLE INFO

Article history:
Received 25 April 2014
Received in revised form 20 October 2014
Accepted 22 November 2014
Available online 11 December 2014

Keywords:
P2P
VoD
Multicast
Multimedia delivery

ABSTRACT

Multimedia content distribution is one of the most popular services available on the Internet, and several approaches have been developed to satisfy user requirements and reduce server load. IP multicast is one efficient mechanism for distributing content to multiple users with minimal server and network loads. Because its single stream of data is transmitted simultaneously to multiple users, IP multicast is well-suited for real-time broadcasting. However, it is difficult to offer on-demand multimedia services with IP multicast in which users request content without synchronization. In this paper, we propose an on-demand based multimedia delivery system using IP multicast. In our system, the multimedia server starts multicasting content upon receiving the first content request. Subsequent users then join the multicast session at any time during the transmission. The content that subsequent users have missed is provided by the users (peers) that joined earlier, or by server unicasts. To minimize server load, we quantitatively analyzed the optimal multicast starting point. Through simulation, we show that our proposed system outperforms prior related systems based on peer-to-peer delivery and multicasting.

1. Introduction

Video and audio content services such as YouTube and podcasts are among the most popular services available on the current Internet, where increasing amounts of content are being digitized and released. Two delivery approaches are currently used to serve this content: client–server (CS) and peer-to-peer (P2P). It is widely accepted that the P2P approach is an efficient way to serve content because it reduces server management costs. Using P2P, however, it is difficult to secure sufficient resources for content streaming services because a peer dynamically joins and leaves a system, and the uplink bandwidth of most home users is much less than the downlink bandwidth. Hence, P2P is typically used to share stored content, such as in BitTorrent [2]. Recently, there have been several attempts by [17,6,21,33] to implement streaming services using the P2P approach, but these have mostly targeted non-commercial services.

Most commercial streaming services use a CS approach. In CS, content is directly delivered from a server, and the service quality is dependent on server-side resources. Compared to the approaches using P2P, considerable server management costs are needed to be added, but it is relatively simple to secure the required resources by adding more servers. Recently, however, the number of users of popular services has increased exponentially, and it is almost impossible to serve the huge number of users with a single set of servers. For example, the number of requests for popular podcasts may at times exceed one million, while the number of concurrent users is half a million.

* Corresponding author.
E-mail addresses: kyhoon@gmail.com (Y. Kim), yskim525@gmail.com (Y. Kim), hyoon@cs.kaist.ac.kr (H. Yoon), ijyeom@gmail.com (I. Yeom).

http://dx.doi.org/10.1016/j.ins.2014.11.033
0020-0255/

To handle the large number of content users, popular commercial streaming services use a content distribution network (CDN) [1,22,26,28]. In a CDN, content distribution servers are located at the edges of the Internet, and user requests are handled by a nearby server. By repeated delivery of popular content, CDN enables the main server overhead to be reduced along with the Internet backbone bandwidth. The reduced distance to the content server also reduces service response time. However, CDN management costs are very high because the content distribution servers are globally deployed. Commercially successful services can afford these costs, but the costs are prohibitive for emerging services.

IP multicast [13] is a technique in which a sender distributes a set of packets in a single stream to multiple receivers without multiple transmissions. In an IP multicast, a set of packets are simultaneously delivered to multiple receivers, which is useful for real-time services such as video conferencing and broadcasting services. However, it is challenging to use IP multicast for video-on-demand (VoD) services because requests for content are not synchronized. For popular content, requests can be concentrated in a short period of time, but there is still a time gap between requests. In [16], an attempt called *Cooperative Peer Assists and Multicast* (CPM) was made to utilize IP multicast for VoD services. In this scheme, however, requests are delayed for a certain period of time to serve them by IP multicast. The server bandwidth can be decreased by increasing the delay, because more requests can be served by one multicast session, but user start-up delays likewise increases. The other limitation of CPM is that the number of receivers for an IP multicast session is fixed when the session starts. Once a session begins, the subsequent requests for the same content need to be handled by the next session. As a consequence, multiple IP multicast sessions with different start-up times are necessary for a content.

In this paper, we propose a novel system for efficient multimedia distribution. To reduce the server load, the proposed system utilizes IP multicast. The main difference between our system and that of [16] is that ours does not require additional start-up delays for request aggregation; the server can immediately start a multicast session upon receiving the first request for content.[1] Subsequent users can receive the content by multicast from the moment they join. The missing part of the content can be delivered preferentially by other users who joined earlier. If user upload bandwidth is not sufficient to deliver the content immediately, it can then be served by unicast from the server. With the passage of time, the server load due to unicast increases, because the number of late users who are missing content increases. The server then starts a new multicast session to reduce that unicast load. To minimize server load by either multicast or unicast, we quantitatively investigated the optimal starting point of a new multicast session. Through analysis and simulation, we show that our proposed system can drastically reduce server load. We also show that the proposed scheme outperforms the multicast-based scheme described in [16] and peer-to-peer based schemes. The advantages of our proposed system are as follows.

(a) The server load in our proposed system is lower than in other conventional CS systems. By leveraging IP multicast, the load for sending common data can be reduced, while optimally chosen starting points of IP multicast can minimize the server load. With an assist of P2P, as a result, the server load of our proposed system can remain low through varying circumstances.
(b) Start-up delays for most users are minimal in the proposed system. When the capacity of P2P in our system is sufficient to handle user requests, users are immediately served by P2P. If capacity is insufficient, users are served by the server unicast. In either case, start-up delays can be achieved in a minimum level.
(c) The proposed system is mathematically analyzed to minimize the server load. It is also evaluated through extensive simulation, and the results show that the server load of our system is significantly lower than those of other systems with reasonable quality of experience (QoE). For more close investigation on QoE related metrics, we perform emulation based evaluation, and emulation results confirm our analysis and simulation.

The rest of this paper is organized as follows. In Section 2, we introduce and summarize prior studies related to our proposed system. In Section 3, we describe how our system works, and in Section 4, we mathematically analyze the system to minimize server load. In Section 5, we validate our system with quantitative simulations. Finally, we conclude our paper in Section 6.

2. Related work

VoD schemes using multicast have been developed and implemented for more than two decades. Skyscraper [18] and Pyramid [32] used multicast streams starting at periodic intervals to serve popular videos. User requests, however, would arrive at different times, and users' QoE can be degraded due to long start-up delay. To reduce the start-up delay, a VoD system using both multicast and unicast was proposed in [15]. In this system, it was attempted to minimize the server load by optimal multicast session scheduling. However, the server load is still high since the video delivery is fully relied on the server.

More recently, P2P paradigms have been proposed for VoD services to reduce the server load. By utilizing P2P in the VoD service area, service providers can reduce server overhead and deployment costs to compensate for peer loads and a lower QoE for users, which includes relatively long start-up delays and less seamless playing. P2Cast [17] builds multiple multicast trees among peers for load balancing and resiliency against failures. Swarming protocols such as those discussed in [6,21,33] divide content into fixed-size chunks and build an overlay mesh with central or distributed directory servers.

[1] In our system, it is possible to additionally reduce the server load by delaying requests.