Computer Communications 35 (2012) 1893–1901

Contents lists available at SciVerse ScienceDirect

Computer Communications

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

Understanding the impact of neighboring strategy in peer-to-peer multimedia streaming applications

Xiangyang Zhang *, Hossam Hassanein

School of Computing, Queen's University, Kingston, Canada K7L 3N6

ARTICLE INFO

Article history:
Available online 30 May 2012

Keywords:
Multimedia streaming
Peer-to-peer networks
Network traffic

ABSTRACT

Peer-to-peer (P2P) multimedia streaming applications need to reduce network traffic to address ISPs' concerns without sacrificing the quality of users' viewing experience. Several studies on P2P file sharing applications propose that a peer only neighbors with nearby peers to reduce network traffic, but whether this strategy is applicable to P2P multimedia streaming applications remains an open issue. In this paper, we study packet propagation behavior and the impact of neighboring strategies on system performance in P2P multimedia streaming applications. We identify two "typical" schemes that capture the essential aspects of the swarm-based and tree-based P2P multimedia streaming schemes, respectively, and compare their performance on two types of neighboring overlays: a random overlay where a peer selects neighbors without considering their network locations, and a nearby overlay where a peer only neighbors with nearby peers. We first conduct simulation study and then provide models to analyze packet propagation behavior on a given overlay in the two typical schemes and the impact of the neighbor-with-nearby-peers strategy on system performance. We find that in the swarm-based scheme, packets propagate along short paths (in terms of hops) on the neighboring overlay, while in the tree-based scheme, peers select parents randomly with respect to their hop counts to the source peer. Applying the neighbor-with-nearby-peers strategy reduces network traffic but results in more lost packets because the nearby overlay has a larger diameter and clustering coefficient. This problem is more severe in the tree-based scheme than in the swarm-based scheme due to their different packet propagation behavior.[1]

1. Introduction

Peer-to-peer (P2P) multimedia streaming applications such as Internet television, live event broadcast, and remote education have gained great popularity during the past few years. While users are enjoying these applications (mostly for free), their enormous traffic causes Internet service providers (ISPs) great financial expenditure and threatens the quality of other Internet services. To avoid consequent unpleasant traffic throttling or blocking, P2P multimedia streaming applications need to address ISPs' concerns, i.e., to reduce inter- and intra-autonomous system (AS) traffic on the Internet.

In a P2P network, in order to reduce communication and processing overhead, a peer only maintains relationship with a limited number of peers, which we call the *neighbors* of the peer.[2] All the neighboring relationships form an overlay network, which we call the *neighboring overlay*. A peer exchanges packets only with its neighbors. The strategy by which a peer selects peers to neighbor with and the strategy by which a peer selects neighbors to exchange packets determine the network traffic. Several measurement studies, such as [2], on real-world P2P multimedia streaming deployments show that a peer does not consider other peers' network locations in the two strategies, and hence enormous traffic ensues.

Several studies [3–5] on P2P file sharing applications propose that a peer only neighbors with nearby peers in the construction of the neighboring overlay to reduce network traffic. However, compared with file sharing applications, live streaming applications have rigid delay constraints. Packets that arrive after playback deadlines are considered lost, and peers will not attempt to fetch packets that have missed or are about to miss playback deadlines. Whether the neighbor-with-nearby-peers strategy is applicable in the context of P2P multimedia streaming remains an open question in the literature.

In this paper, we study packet propagation behavior and the impact of neighboring strategies on system performance in P2P multimedia streaming applications. We attempt to answer the following questions. (1) Can we use the neighbor-with-nearby-peers

* Corresponding author. Tel.: +1 6135336052.
E-mail addresses: xiang@cs.queensu.ca (X. Zhang), hossam@cs.queensu.ca (H. Hassanein).

[1] An abbreviated version of this paper that includes the typical swarm-based and tree-based schemes and part of the simulation results appeared in [1].

[2] In some schemes, a peer maintains a membership table and selects some members to construct a neighbor table. In other schemes, a peer constructs the neighbor table directly.

0140-3664/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.05.012

strategy in P2P multimedia streaming applications to reduce network traffic without causing unfavorable side-effects? (2) Given a neighboring overlay, will packets propagate along low-cost paths?

Since there exist a large number of P2P multimedia streaming schemes, we first identify "typical" schemes that capture the essential aspects of these schemes, especially those in actual deployment on the Internet. P2P multimedia streaming schemes can be classified into two categories: swarm-based schemes and tree-based schemes [6]. Swarm-based schemes employ the *swarm* technique used in BitTorrent [7]. A stream is split into chunks and peers pull missing chunks from neighbors. Our typical swarm-based scheme (called TS hereafter) employs the rarest-first chunk scheduling policy, which is used in most swarm-based schemes. In tree-based schemes, peers form parent–child relationship to build a tree along which a parent pushes packets to children. Our typical tree-based scheme (called TT hereafter) constructs the tree in a straight-forward manner—each peer subscribes to the neighbor with the latest position in the stream. We then compare system performance of the two typical schemes on two types of neighboring overlays: *random overlay* and *nearby overlay*. When constructing a random overlay, peers select neighbors without considering their network locations. When constructing a nearby overlay, peers use the neighbor-with-nearby-peers strategy and only select nearby peers as neighbors.

We study the impact of the neighbor-with-nearby-peers strategy by both simulation and analysis. We first develop a discrete-event simulator to study system performance of the two typical schemes on the two overlays. Based on the findings in the simulation, we then provide packet propagation models to analyze packet propagation paths on a given overlay and overlay models to characterize the random and nearby overlays and analyze their impact on system performance. We find that packets propagate in distinct manners in TS and TT although both schemes are *data-driven* (i.e., the propagation paths are determined by availability of packets at peers rather than network metrics). In TS, peers have a high probability to pull from neighbors that have fewer hops to the source peer; the set of paths a chunk traverses from the source to peers (called *propagation tree*) is comparable to a degree-bounded shortest path tree (in term of hops) on the neighboring overlay. In TT, peers select parents almost randomly with respect to their hop counts to the source, resulting in significantly taller propagation trees compared with TS. The neighbor-with-nearby-peers strategy reduces network traffic significantly, but also results in more lost packets in the presence of peer churn and substrate network errors because of the large diameter and clustering coefficient of the nearby overlay. This problem is more severe in TT than in TS because they have different packet propagation behaviors, which cause chunk to be lost in different ways.

The remainder of this paper is organized as follows. Section 2 introduces background and related work. Section 3 describes TS and TT. Section 4 studies TS and TT on the random overlay and nearby overlay by simulation. Section 5 presents packet propagation models in TS and TT and analyzes the impact of propagation behavior on system performance. Section 6 presents overlay models and analyzes the impact of the neighbor-with-nearby-peers strategy on system performance. Section 7 concludes this paper.

2. Background and related work

There are two basic approaches to P2P multimedia streaming. The first approach is inspired by the *swarm* technique used in the BitTorrent file sharing application [7]. A stream is split into chunks of a fixed size. A peer maintains a sliding window of recently received chunks. A peer advertises its buffer map, which describes the chunks the peer has, to neighbors, and exchanges chunks with neighbors. The second approach is inspired by IP multicast. A peer subscribes to a neighbor to form explicit parent–child relationship. All the parent–child relationships form a tree on the neighboring overlay. (A stream may be interleaved into multiple substreams and peers build a separate tree for each substream.) Upon receiving a packet, a peer immediately forwards the packet to its children. A P2P multimedia streaming scheme may use either approach or combine them together. Most deployments on the Internet use the first approach or both approaches.

In swarm-based schemes [8–12], every interval of length T, a peer selects the chunks to pull in the next interval and selects the neighbors to pull these chunks from. A peer may select chunks randomly [8], but in most schemes, a peer employs a *latest-first* or *rarest-first* policy [9–11]. (In multimedia streaming applications, latest chunks are also rarest chunks.) A peer may select neighbors in several ways to pull the selected chunks, such as randomly [8,10], according to neighbors' workload [9,11], according to the bandwidths and delays to neighbors, and according to data exchange history. A peer may implement an incentive mechanism similar to the *tit-for-tat* policy of BitTorrent to encourage peers to contribute upload bandwidth [11]. However, since multimedia streaming applications have stringent time constraints and a peer only buffers recently received chunks (compared with the whole file in BitTorrent), the efficacy of this policy is limited. In swarm-based schemes, propagation trees are determined by the availability of chunks at peers rather than by network metrics such as cost, delay, and bandwidth, hence swarm-based schemes are said to be "data-driven" [9].

In tree-based schemes, a peer may consider multiple factors when selecting neighbors to subscribe to. According to whether the main factors under consideration are network metrics or neighbors' buffer maps and data exchange history, we can classify tree-based schemes into two types: network-driven, such as [13–16], and data-driven, such as [17–20]. Early schemes [13,14] are usually network-driven with the objective of minimizing the tree cost. Recent schemes include both types, but data-driven schemes are more prominent in real-world deployments [17,18]. Data-driven schemes combine the swarm and tree-building techniques together. In data-driven schemes, a peer advertises its buffer map to neighbors, establishes parent-child relationships with neighbors, and pulls missing chunks from neighbors. Because packets (or chunks) are both pushed and pulled, data-driven tree-based schemes are also called push-pull hybrid schemes [17,18]. The main factors a peer considers when selecting parents are neighbors' proceeding positions in the stream and the data exchange history. For example, in [18], a peer tries to maintain that it advances at similar pace in each substream and its parent advances at similar pace as its neighbors. In [17], a peer subscribes to a neighbor that it has received more packets in the last subscription interval with a higher probability. In [20], a peer subscribes to neighbors with the latest positions in the stream and short packet delivery delays.

In most swarm-based schemes and data-driven tree-based schemes, like in P2P file sharing schemes, peers' network locations are not considered in the construction of the neighboring overlay. Several ISP-friendly schemes for P2P file sharing applications, such as [3–5], propose that a peer only neighbors with nearby peers to reduce inter-AS traffic. References [4,5] report reduced inter-AS hops and shorter file downloading time. We remark that a study of using the neighbor-with-nearby-strategy in P2P multimedia streaming applications does not exist in the literature.

We also remark that there are only a few analytical studies on P2P multimedia streaming applications and none of them studies the neighboring strategy or packet propagation behavior. Zhou et al. [21] compare the chunk delivery rate and delivery delay of