Computer Networks 69 (2014) 101–120

Contents lists available at ScienceDirect

Computer Networks

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

A performance comparison of hose rate controller approaches for P2P-TV applications ☆

S. Traverso [b,*], C. Kiraly [a], E. Leonardi [b], M. Mellia [b]

[a] Bruno Kessler Foundation, Via Sommarive 18, I-38123 POVO (TN), Italy
[b] Department of Electronics and Telecommunications (DET), Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

ARTICLE INFO

Article history:
Received 19 November 2013
Received in revised form 24 March 2014
Accepted 16 April 2014
Available online 26 April 2014

Keywords:
P2P
Streaming
Rate control
P2P-TV

ABSTRACT

The goal of this paper is to investigate rate control mechanisms for unstructured P2P-TV applications adopting UDP as transport protocol. We focus on a novel class of Hose Rate Controllers (HRC), which aim at regulating the aggregate upload rate of each peer. This choice is motivated by the peculiar P2P-TV needs: video content is not elastic but it is subject to real-time constraints, so that the epidemic chunk exchange mechanism is much more bursty for P2P-TV than file sharing applications. Furthermore, the peer up-link (e.g., ADSL/Cable) is typically the shared for flows in real scenarios. We compare two classes of aggregate rate control mechanisms: Delay Based (DB) less-than-best-effort mechanisms, which aim at tightly controlling the chunk transfer delay, and loss-based Additive Increase Multiplicative Decrease (AIMD) rate controllers, which are designed to be more aggressive and can compete with other AIMD congestion controls, i.e., TCP.

Both families of mechanisms are implemented in a full-fledged P2P-TV application that we use to collect performance results. Only actual experiments – conducted both in a controlled test-bed and over the wild Internet, and involving up to 1800 peers – are presented to assess performance in realistic scenarios.

Results show that DB-HRC tends to outperform AIMD-HRC when tight buffering time constraints are imposed to the application, while AIMD-HRC tends to be preferable in severely congested scenarios, especially when the buffering time constraints are relaxed.

1. Introduction

P2P-TV applications are designed to offer real-time video streaming services exploiting the Peer-to-Peer (P2P) paradigm. In these systems, peers are arranged in a generic meshed overlay topology which is dynamically adapted and optimized by a distributed algorithm, and neighboring peers are enabled to exchange data. The content to be delivered is chopped in segments, called *chunks* which are distributed among peers exploiting an epidemic approach. Because of this characteristics, unstructured P2P-TV systems may closely resemble P2P applications for file sharing, e.g., BitTorrent.

However, three peculiar aspects of P2P-TV applications mark a significant difference with respect to P2P file sharing applications like BitTorrent:

(1) The content is not available in advance, but it is generated in real time at a single point (the source). As a consequence no seeds are available in the system.

☆ This work was funded by the European Commission under the FP7-Strep Project NAPA-WINE.
* Corresponding author. Tel.: +39 011 090 4247.
E-mail addresses: traverso@tlc.polito.it (S. Traverso), kiraly@fbk.eu (C. Kiraly), leonardi@tlc.polito.it (E. Leonardi), mellia@tlc.polito.it (M. Mellia).

http://dx.doi.org/10.1016/j.comnet.2014.04.010
1389-1286/

(2) New generated chunks must be delivered to all peers, within a deadline, the *total offset delay*, i.e., the lag between the chunk generation time at the source and the time the chunk has to be played at the peer. For live content distribution, it is highly desirable to keep this delay as small as possible so that chunks delivery times are minimized and peers can view the show in almost real-time. Total offset delay is controlled by limiting the number of chunks that peers can trade with other peers at any time.
(3) As a consequence of the previous requirements, receiving data rate at peers cannot be adapted, i.e., the application is not elastic: either the peer is in the condition of receiving chunks at the same rate they are generated at the source or it is not able to watch the show with an acceptable Quality of Experience (QoE). This implies that peers are not greedy (pre-fetching policies are indeed not possible, since content is generated in real-time), and the goal of the transmission rate controller is not to maximize the download/upload rate, but to guarantee that the total demand is sustained. That is, the system-wide total upload/download rate cannot exceed N times the video rate (where N is the number of peers in the system).

The above three characteristics heavily impact the overall design of P2P-TV applications. First, to control the delivery time, chunk size must be kept small. A common choice made in many P2P-TV systems is to deliver one video frame in a chunk. This causes the size of each chunks to be highly variable, due to the video encoding process nature [1–3].

Second, UDP is largely preferred to TCP, to avoid long retransmission delay and accelerate sending of chunks. Packets of the same chunk are transmitted back-to-back without sharing of the up-link capacity among different chunks [2–5].

Third, to increase the ability of peers to retrieve chunks within a limited delay, every peer should rely on a sufficient large set of neighbors. Hence, the size of peers' neighborhoods to exchange data with is typically much larger than in BitTorrent (order of several tens versus five).

At last, scheduling algorithms must be designed having in mind the goal of distributing chunks to all peers within the limit imposed by the total offset delay [6–11].

The literature about P2P-TV systems is mostly focused on the overlay topology design and on the modeling of scheduling algorithms to improve performance, and to the best of our knowledge, only a few works focus on transmission rate controllers for mesh-based P2P-TV systems (see Section 7 for a discussion). Indeed, most existing P2P live video systems do not consider the sender rate control problem explicitly, and they simply adopt a best-effort approach, where the senders try their best to serve the receivers by using as much up-link bandwidth as possible, without any rate control. To the best of our knowledge, the only works that explicitly deal with the issue of regulating the transmission rate of peers are [12,13]. In particular [13] compares a delay-based transmission rate controller against a simple scheme, where the chunk offering rate is fixed.

Considering instead the industrial side, the many commercial P2P-TV clients available in the Internet, e.g., PPLive, SopCast, etc., do not disclose their source code, making it impossible to understand which kind of rate controller they implement, and making unfeasible any experimental comparison with them.

In a nutshell, only little attention has been devoted to the problem of how to efficiently exploit the peer upload capacity by controlling the sending rate of peers. Considering the typical nowadays scenario faced by P2P applications, it is natural to assume that the bottleneck is provided by the peer up-link, e.g., it is located at the ADSL/cable up-link. In order to guess the right chunk sending rate most of P2P applications ask the user to manually set the available bandwidth, but bandwidth is known to vary in time due to background traffic condition (take the case of a shared wireless local network), and any static (mis) configuration often leads to an inefficient use of the available resources possibly inducing congestion and packet loss. Automatically tuning the transmission rate is thus essential (i) to keep low chunk delivery delays, (ii) to guarantee that all users fairly participate to the chunk dissemination process in a way proportional to their available resources, and (iii) to let the system be more robust when faced with (abrupt) changes of the network conditions.

For instance, Fig. 1 describes by means of a simple example the benefits brought by an adaptive approach able to match the transmission rate of chunks to the available bandwidth at the peer. A few peers form a small mesh network where S represents the source peer: in the top part of the figure the sending rate of the chunks at each peer is fixed and does not change in time. Moving from Time 1 to Time 2, peer A undergoes a bandwidth reduction from 5 Mb/s to 1 Mb/s, but its chunk transmission rate does not adapt to the new condition, introducing delays in the delivery of the chunks and, thus, losses. Notice, that peers B and C could actually replace peer A in the chunk distribution process (they have enough bandwidth), but they do not, since their transmission rate is fixed too. The bottom part of the figure depicts what happens when an adaptive regulation is available: in Time 2 peer A undergoes a bandwidth reduction, but its chunk transmission rate is nicely reduced to match its new upload capacity. Moreover, peers B and C increase their transmission rate to replace peer A in the chunk distribution. Observe that by doing so, no delivery delays nor losses are introduced.

For these reasons, embedding an application level rate controller algorithm that is able to dynamically adapt the rate at which chunks are transmitted by peers becomes a crucial task in P2P-TV applications design. Observe that the same motivations drove to the development of Ledbat [14] in the context of P2P file sharing, and it is now embedded into the majority of BitTorrent clients.

Results presented in [13] show that in P2P-TV systems it is preferable to adopt transmission rate controllers rather than best-effort and static schemes. This paper goes further, and investigates two families of rate control algorithms for P2P-TV systems. We present, discuss and validate possible alternatives by means of thorough experimental campaigns. In particular, we address the following key points: