Computer Networks 74 (2014) 53–63

Contents lists available at ScienceDirect

Computer Networks

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

Overlay live video streaming with heterogeneous bitrate requirements

Dongni Ren *, Wang Kit Wong, S.-H. Gary Chan

Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong, China

ARTICLE INFO

Article history:
Received 24 February 2014
Received in revised form 30 July 2014
Accepted 30 August 2014
Available online 16 September 2014

Keywords:
Live streaming
Heterogeneous bitrate
Parent-disjoint trees

ABSTRACT

We study a streaming cloud formed by distributed proxies providing live video service to diverse users (e.g., smart TVs, PCs, tablets, mobile phones, etc.). The proxies form a push-based overlay network, with each proxy serving a certain video bitrate for users to join. To form a proxy overlay serving heterogeneous bitrates, we consider that the video is encoded into multiple MDC (Multiple-Description Coding) streams with the serving bitrate of proxy i being k_i description streams. In order to effectively mitigate stream disruption due to node churns, proxy i also joins an additional r_i redundant MDC streams ($r_i \geqslant 0$) in such a way that all the $(k_i + r_i)$ streams are supplied by *distinct* parents. For live streaming, the critical issue is how to construct the *parent-disjoint* trees minimizing the assembly delay of the proxies.

We present a realistic delay model capturing important system parameters and delay components, formulate the optimization problem and show that it is NP-hard. We propose a centralized algorithm which is useful for a centrally-managed network and serves as a benchmark for comparison (PADTrees-Centralized). For large network, we propose a simple and distributed algorithm which continuously reduces delay through overlay adaptation (PADTrees-Distributed). Through extensive simulation on real Internet topologies, we show that high stream continuity can be achieved with push-based trees in the presence of node churns. Our algorithms are simple and effective, achieving low loss and low delay.

1. Introduction

We have witnessed in recent years the proliferation and penetration of Internet-ready multimedia-capable smart devices such as tablets, notebooks, PCs, smart TVs, set-top boxes, etc. These devices have different screen resolution and processing capabilities. In order to serve these devices, a streaming application has to meet heterogeneous bandwidth requirements ranging from, for example, 500 kbps for tablets to several Mbps for smart TV.

There has been increasing interest in providing live streaming services (such as Internet TV) offering heterogeneous bitrate to these smart devices. To achieve scalability and low delay, we consider a streaming cloud formed by distributed proxies placed close to user pools.[1] The proxies are light-weight content servers deployed by the content provider in the public Internet across multiple ISPs. They form an overlay, and each serves streams of a certain bitrate. Users on different devices join the proxies in their proximity of corresponding bitrate requirements to be served directly.

---

* Corresponding author.
E-mail addresses: tonyren@cse.ust.hk (D. Ren), wwongaa@cse.ust.hk (W.K. Wong), gchan@cse.ust.hk (S.-H.G. Chan).

[1] In this paper, we use “proxy” and “node” interchangeably.

http://dx.doi.org/10.1016/j.comnet.2014.08.019
1389-1286/

In this work we focus on the cloud formed by the proxy overlay, and address its design and optimization issues. In order to efficiently support different bitrates of the proxies, the live video is encoded with Multiple-Description Coding (MDC) into K streams of similar bandwidth [1–7]. All the description streams are first generated at the source, and then pushed down to proxies via multiple delivery trees. Each description is distributed by a unique delivery tree. Note that these trees may not span all the proxies in the network. Proxy i receives some subset of the description streams k_i ($k_i < K, k_i \in Z^+$) according to its bitrate requirement. After receiving k_i description streams, the proxy re-assembles the descriptions into a full video, and then serves it to its users.

We consider the realistic case that the overlay network may be dynamic, i.e., the nodes may churn at any time (i.e., be introduced, removed or fail). Whenever there is a churn, the video at the downstream nodes will be disrupted. In order to offer high continuity, node i receives an *additional* r_i description streams as redundancy ($r_i \geqslant 0$). In this way, the requirement can be met as long as node i receives k_i streams or more from the streams delivered. In other words, in case of packet loss due to node churn, the redundancy streams are used to meet the rate requirement. It is clear that the total number of MDC streams encoded is the highest proxy requirement, i.e., $K = \max_i(k_i + r_i)$. Due to the use of MDC and redundancy, video quality would only be gracefully degraded if fewer than k_i streams are received at node i. The cost due to node churn is an increase in bandwidth because of the extra r_i streams a node receives.

To further protect stream continuity against unexpected node churns, all the $(k_i + r_i)$ description streams of proxy i are supplied by *distinct* parents. The challenge is hence how to construct the K *parent-disjoint* description trees in order to minimize source-to-end delay due to stream assembly while meeting the heterogeneous rate requirements of the proxies. We tackle this problem by presenting a realistic node model on delay, formulating the optimization problem, analyzing its complexity, and designing effective optimization algorithms (centralized and distributed).

Push-based overlay live streaming has been shown to achieve substantially lower delay than pull-based approach [8,9]. However, there has not been work on the design and *optimization* of a *push-based* overlay with proxies of *heterogeneous* rates. Our approach is shown to achieve low delay with high continuity. Previous approaches are often based on a random pull-based mesh, where a node continuously searches for neighbors (using gossip) and pulls content from them. This rather uncoordinated *ad-hoc* connectivity clearly is not bandwidth-efficient and may not even meet heterogeneous requirements. Furthermore, as the major objective of pull-based approach is to aggregate a full video, it seldom optimizes source-to-end delay, leading to unsatisfactory delay and resource (bandwidth) utilization. On the other hand, much of the previous tree-based overlay work has not sufficiently considered MDC with redundant streams to achieve stream continuity. We propose and optimize a push-based overlay structure composed of multiple trees and redundancy streams to mitigate node churns. Our overlay meets heterogeneous bitrate requirements of proxies with high stream continuity.

Fig. 1 shows an example of our streaming overlay with three MDC streams, i.e., $K = 3$. Nodes A to F serve videos with different rate requirements: $k_i = 2$ for nodes A to D, and $k_i = 1$ for nodes E and F. They all receive one more description stream as redundancy. Nodes A, B and C are directly connected to the streaming server where they receive all their streams. Because the streaming server is stable, they do not need any redundant stream. On the other hand, because node D is not served by the streaming source, it connects to *distinct* parents in order to achieve fault-tolerance in streaming, i.e., D receives from three parents three descriptions with one as redundant stream. Nodes E and F both receive two descriptions from distinct parents while they require only one for viewing.

Because a proxy can decode and serve the video only after it assembles all the required streams, its delay from the source is the *slowest path* out of all the $(k_i + r_i)$ trees (i.e., the maximum-delay path). Such delay increases quickly if the trees are not constructed properly. The challenge is how to construct the *parent-disjoint* trees to achieve minimum delay. We propose algorithms, termed PADTrees (**Pa**rent-**D**isjoint Trees), to construct highly efficient trees achieving low delay and high continuity while meeting heterogeneous bitrate requirements.

The contributions of our study are:

- *Delay model, problem formulation and its complexity analysis:* Given K MDC streams, we present a rather realistic and comprehensive delay model for a node capturing all the major network and delay components such as scheduling delay (due to fanout of a node), edge bandwidth, end-to-end bandwidth, propagation delay, etc. With the model, we formulate the delay optimization problem which is to design MDC trees that minimize the diameter (i.e., worst-case delay) of the overlay

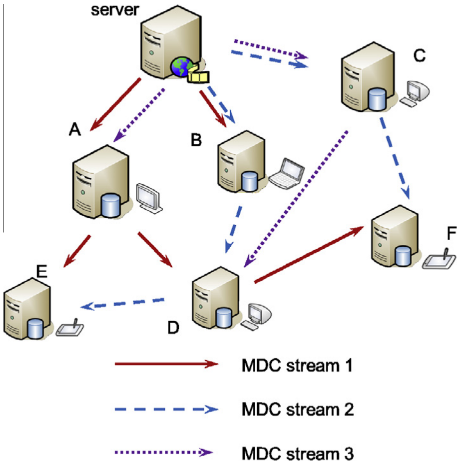

Fig. 1. An example of live overlay streaming with heterogeneous requirements, showing the constituent underlying delivery trees.