Computer Communications 32 (2009) 1410–1417

Contents lists available at ScienceDirect

Computer Communications

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

Request-driven swarming scheme for P2P data streaming ☆

Jialing Xu, Victor O.K. Li *

Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong, China

ARTICLE INFO

Article history:
Available online 12 February 2009

Keywords:
Peer-to-peer
Streaming
Swarming
Request-driven

ABSTRACT

Data streaming by swarming over peer-to-peer (P2P) overlay networks has attracted much attention in recent years and initially the swarming solution is based on data-driven schemes. This paper presents a new request-driven swarming scheme. The scheme offers several advantages: high efficiency in data delivery, low control overhead, and flexibility in streaming control. Some key issues on the design of the relevant scheduling and forwarding mechanisms are discussed. In addition, we present a scheduling model for this scheme and propose some heuristics. Our evaluation of these heuristics reveals that there is a tradeoff among performance metrics such as delivery reliability, loading on the underlay network, and on the streaming overlay network. Comparing our proposed request-driven scheme with an existing data-driven scheme using computer simulation, we find that our request-driven scheme incurs much lower control overhead while providing comparable or better efficiency in data delivery.

1. Introduction

With advances in broadband access, the Internet is increasingly being used as a new medium to provide multimedia service to the public. Multimedia service requires efficient and scalable delivery techniques for streaming data. Presently, IP Multicast is probably the first choice for this type of data delivery service due to its sophisticated design [5,7–9] and high performance [6]. Its deployment, however, is limited due to high cost, excessive management workload, and redeployment difficulties when network topology and user distribution change. Thus researchers pursue application-level solutions, which attempt to build a logical overlay network among cooperative nodes on IP Unicast networks.

Peer-to-peer (P2P) technology has been suggested as the ideal platform to carry out application-level streaming. Numerous P2P-based streaming systems and theoretical studies have been completed. These can be broadly classified into two categories: tree-based and mesh-based. The former adopts similar design principles as the traditional IP Multicast which builds and maintains an explicit tree-like structure. This structure, however, is mismatched with the application-level overlay environment with dynamic nodes, and may offer poor streaming performance [1]. On the other hand, in the mesh-based approach, autonomous nodes exchange data with others spontaneously and there is no prescribed structure. This is the reason why it is also called swarming. It is simple and robust.

The initial swarming design is driven by the streaming data and involves heavy control overhead, rendering it inefficient and unscalable. We propose a lightweight, flexible, request-centric P2P streaming scheme named request-driven swarming. In this scheme, requests are utilized to construct the forwarding relationship before actual data delivery and to adjust the relationship during delivery. The simulation results and comparison with existing work demonstrate that the request-driven swarming scheme can greatly reduce the control overhead and delivery latency, while enjoying high efficiency.

In this paper, challenges in realizing the request-driven swarming scheme are identified. We discuss the issues of how to use requests to guide data streaming among nodes in the system, including streaming abstraction, request definition, and data forwarding. In particular, this paper presents a scheduling model for this scheme and proposes some heuristics. Our evaluation of these heuristics reveals that there is a tradeoff among performance metrics such as delivery reliability, loading on the underlay network, and on the streaming overlay network. Comparing our proposed request-driven scheme with an existing data-driven scheme using computer simulation, we find that our request-driven scheme incurs much lower control overhead while providing comparable or better efficiency in data delivery.

The rest of the paper is organized as follows. After a brief introduction of related work in multicast and in data-driven swarming over P2P in Section 2, Section 3 presents the design of the request-driven swarming scheme, including the design principle and the request mechanism based on describing streams in abstract form. Section 4 models the upstream scheduling for the request-driven scheme, and proposes four heuristics. Evaluation of these scheduling heuristics and a prototype implementation based on simula-

☆ A preliminary version of this article is published at ICST QShine, Hong Kong, China, July 2008.

* Corresponding author. Tel.: +852 28578425; fax: +852 25598738.
E-mail address: vli@eee.hku.hk (V.O.K. Li).

0140-3664/$ - see front matter
doi:10.1016/j.comcom.2009.02.005

tion experiments and comparisons with existing work are included in Section 5. Finally, conclusion is given in Section 6.

2. Related work

In the past, IP Multicast has been considered the most efficient vehicle for data streaming. Recently, however, systems implementing multimedia streaming applications on P2P overlay platforms have achieved much success. These systems can generally be classified into two categories according to their organizational characteristics: tree-based and mesh-based. The tree-based schemes organize the streaming relationship among all nodes following a prescribed tree-like structure. In the mesh-based systems, autonomous nodes exchange data with others spontaneously. Our work, the request-driven streaming scheme, belongs to the second category. In this section we give a brief review of the existing P2P streaming schemes under these two categories.

2.1. Tree-based structure

Derived from IP Multicast, the tree structure is employed in many early overlay streaming schemes [2,10,14]. In this type of overlay streaming, efficient algorithms for multicast tree construction and maintenance are the key to system efficiency and robustness.

SplitStream [14] is a representative example of this type. In SplitStream, the streaming content is striped across a forest of interior-node-disjoint multicast trees that distribute the forwarding load among all participating peers [4]. Such a multi-tree based forest brings some guarantee of efficiency and failure-tolerance, but is still vulnerable to churning and requires assistance from content coding to combat topology changes of the trees. Subsequent researchers [2,10] have addressed the issue of reliability by more complicated algorithms. However, they are still not satisfactory.

Essentially, the tree structure is mismatched with the application-level overlay environment with dynamic nodes [3]. As the autonomous overlay nodes join and leave at will, the tree-like prescribed relationship is highly vulnerable, especially for streaming applications that require high bandwidth and time-stringent delivery.

2.2. Unstructured swarming

Swarming schemes have recently become popular since they address the issue of robustness better than the tree-based schemes. Swarming design is also known as data-driven swarming, since it is the design principle of early swarming schemes.

Coolstreaming [14] is the first significant swarming system design based on the data-driven principle. In Coolstreaming, each node periodically advertises data availability information to a set of nodes called partners,[1] retrieves locally unavailable data from them and supplies available data to them. Thus, data arrival drives the dissemination of availability information, then the information triggers the request for data and finally data are delivered. The data act as the initiator of the procedure; hence the name data-driven.

Data-driven swarming, however, involves large control overhead leading to low efficiency and scalability. In multimedia applications, streaming must be timely to ensure playback quality. Control information, including data availability and data request, must be exchanged following the pace of data delivery. Therefore, the intensive data volume in multimedia applications leads to intensive volume of these control messages. As a result, large control overhead is incurred in data-driven swarming schemes. To address these problems, we propose a new swarming scheme for P2P streaming, called request-driven swarming, in which requests instead of data are used to guide the streaming.

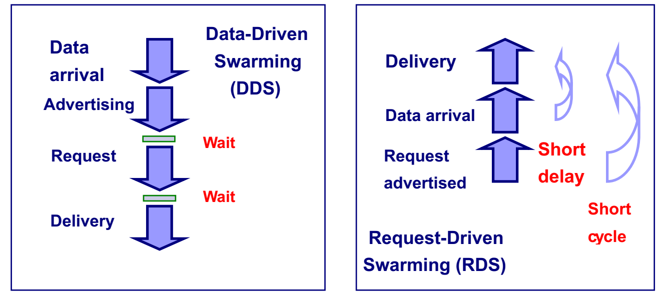

Fig. 1. Comparison of data-driven and request-driven schemes.

3. Request-driven streaming

3.1. Introduction

We propose the request-driven swarming scheme. In comparison with data-driven swarming, our design offers three advantages: efficiency in data delivery, low overhead traffic, and flexibility in streaming control.

Fig. 1 compares the processing cycles of data-driven and request-driven swarming. The data-driven scheme contains four stages and two waiting periods. First, data arrives and then its availability is advertised periodically. When other nodes receive the advertised availability information, they then periodically send out requests for these data. Finally, these requests are served by the delivery of the required data. In this process, nodes wait for the periodical advertising of availability information and the periodical requests for data before data delivery. As a result, the four process steps and two waiting periods lead to long latency.

On the contrary, the processing in the request-driven scheme includes only three stages but no notable waiting. In Fig. 1, requests for streaming are sent to those nodes which have attached to the stream already.[2] If these requests can be accepted according to current available bandwidth, they will be recorded. When these nodes receive data, they can immediately forward the data according to the previously accepted requests. Obviously, from data arrival to data delivery, only two steps and no notable waiting are required. The latency is thus reduced significantly.

Low control overhead is another merit of the request-driven scheme. In the data-driven scheme, high data volume leads to large control overhead. In this scheme, the data availability information is based on the description of data segments rather than the stream, and hard to compress. The stringent time constraint on data delivery requires data availability information to be advertised frequently, further intensifying the volume of control overhead. On the contrary, our request-driven scheme focuses on describing the stream. In most streaming application scenarios, a few attributes like rate, start time, and stop time, can characterize the basic properties of a stream. Therefore, describing a stream in a request by stream attributes reduces the message size and sending frequency, and thus greatly decreases the total control overhead.

The flexibility in streaming control is an extra attraction of the request-driven scheme. In the data-driven scheme, the receiver nodes request data passively based on the data availability

[1] A partner in Coolstreaming is a node that cooperates with the given node in swarming for the same stream.

[2] It is assumed that these attached nodes can be discovered through existing P2P search and location mechanisms.