Peer-to-Peer Receiver-driven Mesh-based Streaming

Nazanin Magharei, Amir H. Rasti, Daniel Stutzbach (Advisor: Reza Rejaie)
University of Oregon
{nazanin,amir,agthorr,reza}@cs.uoregon.edu

Peer-to-Peer (P2P) streaming has become an increasing popular approach for one-to-many multimedia streaming applications, mainly because it does not require any special support (*e.g.,* IP multicast or any content distribution infrastructure) from the network. A common theme in P2P streaming systems is that participating peers form an overlay where each peer receives content from one (or multiple) *parent* peer(s) in a session. Any P2P streaming system consists of two distinct but related components: *(i)* an *Overlay Construction* mechanism that organizes participating peers into an overlay structure with certain properties, and *(ii)* a *Content Delivery* mechanism that determines how the multimedia content should be streamed to each participating peer through the overlay. The design of these two components should accommodate the following two potentially conflicting goals: *(i) Session-level (Global) Goal*: The overall network load associated with content delivery should be minimized; *(ii) Peer-Level (Local) Goal*: The quality of the stream delivered to individual peers should be maximized within their fair share of available resources (mainly bandwidth). The session-level goal should be cooperatively pursued by all peers whereas each peer should separately maximize its own local goal. Note that the peer-level goal could easily be in conflict with the session-level goal. For example, selecting a parent peer that minimizes the overall network load may not maximize the delivered bandwidth to a receiver peer.

A few existing approaches to P2P streaming have adopted the idea of "application-level multicast" where participating peers often form a *topologically-aware tree structure* over which multimedia content is simply "pushed" from the source towards all peers. In essence, the primary (and often the only) goal in this class of solutions is to accommodate the session level goal by constructing an overlay with certain properties. This class of solutions is unable to maximize the quality delivered to individual peers for the following reasons:

First, available bandwidth to each peer in a tree-structure overlay is inherently limited. This problem is further aggravated by the following issues: *(i)* heterogeneity and asymmetry of access link bandwidth among participating peers, *(ii)* dynamic peer participation (*i.e.,* churn), and *(iii)* the competition among participating peers for available bandwidth from parent peers. We note that organizing peers into multiple, diverse trees (*e.g.,* CoopNet [2]) improves resiliency to churn but may not improve delivered bandwidth to individual peers due to shared bottlenecks among trees.

Second, existing solutions do not leverage the available flexibility in dynamic receiver-driven (*i.e.,* pull) content delivery to accommodate the peer-level goal. In these approaches, each peer simply pushes a specific subset of received packets to all its child peers. This *static* content-to-parent mapping can not cope with the heterogeneity and asymmetry of access-link bandwidths among participating peers. For example, a parent peer might be responsible for relaying a single layer (of a layer encoded stream, *e.g.,* [2]) to all of its child peers but it may not have sufficient out-going bandwidth.

Third, the rate of content delivery in existing solutions is limited since they do not incorporate a *swarm-like* mechanism. More specifically, there is a parent-child relationship between connected peers, since content always flows in one direction from parent to child peers. File swarming mechanisms (*e.g.,* BitTorrent) achieve higher rate of content delivery by allowing connected peers to exchange data in both directions. However, these mechanisms can not support *streaming* delivery.

This paper presents a new approach to large scale P2P streaming in live but non-interactive sessions, called Peer-to-Peer Receiver-drIven MEsh-based Streaming or PRIME. An example of such applications is TV-like video distribution from a single user to a large number of heterogeneous and dynamic users over the Internet. The primary design goal in PRIME is to maximize the quality delivered to each peer (*i.e.,* accommodating the peer-level goal) because it determines the observed performance by individual users. Toward this end, PRIME adopts a different design methodology. In PRIME, participating peers form a high-degree and randomly connected overlay mesh, but each peer employs a dynamic receiver-driven (*i.e.,* pull) content delivery mechanism to independently maximize its delivered quality. In other words, the overlay construction mechanism in PRIME is very simple and has low overhead. The complexity is shifted into the content delivery mechanism which has more flexibility and can be separately controlled by individual peers. For example, each peer may connect to 10 randomly selected peers in the session and then coordinate content delivery from all these peers. Since all pair-wise connections between peers are congestion controlled, PRIME enables participating peers to properly compete for available bandwidth. In essence, PRIME combines the strength of three key ideas into a coherent solution for P2P streaming: *(i)* the resiliency and biased connectivity of a high degree and randomly connected overlay, with *(ii)* the flexibility of dynamic receiver-driven coordination for pull content delivery [1, 5], and *(iii)* the high rate of *content diffusion* in swarm-like content delivery. In the following subsections, we present two key components of PRIME in further details. We assume that the video stream distributed by the source is multiple-description encoded to accommodate bandwidth heterogeneity among peers.

Overlay Construction: Our goal is to organize participating peers into a high degree and randomly connected mesh. The idea of using such an overlay for P2P streaming was inspired by a property recently observed in other unstructured P2P overlays (*e.g.,* Gnutella) [4]. We discovered that the connectivity of each peer in the overlay is biased towards peers with higher uptime. More specifically, each peer tends to be more connected to other peers that have been in the overlay for an equal or longer period of time. This means that the longer a peer remains in the overlay, the more likely it establishes connection to other long-lived peers, and thus the less churn (and thus better quality) it observes in its connections due to neighbor departures. Interestingly, such a desired property is achieved because of only two reasons: *(i)* a high degree and random connectivity of participating peers, and *(ii)* a heavy-tailed distribution of peer uptime which is common in streaming sessions (*e.g.,* [3]).

Besides its simplicity and biased connectivity, using a high-degree and randomly connected mesh structure has several other advantages as follows: *(i)* graceful accommodation of bandwidth heterogeneity and asymmetry by allowing each peer to receive content from multiple neighbors, *(ii)* a significant increase in the probability of path diversity (*i.e.,* reducing the possibility of shared bottleneck) between connections from different neighbors (whenever such diversity is feasible), and *(iii)* a smooth diffusion of content among participating peers since each peer provides *shortcuts* for content diffusion among its neighbors.

Content Delivery: Content delivery is the key component of PRIME that combines *push* content reporting with *pull* content delivery. Toward this end, each peer reports its available content and its playout time to its neighbors either periodically or in an event-driven fashion (*e.g.,* after receiving a certain number of new packets). This simple and efficient reporting scheme ensures that each peer quickly becomes aware of available content among all its neighbors. In PRIME, participating peers maintain a close playout time which remains sufficiently behind the source's playout time in order to provide an opportunity for data buffering at each peer. Maintaining close playout time maximizes the overlap between buffers at different peers and thus increases the opportunity for swarming.

Each peer passively monitors available bandwidth from its neighbors during content delivery. Given the information about available content and available bandwidth for all neighbors, the coordination mechanism at each peer periodically takes several steps in the following order: *(i)* it determines the proper number of descriptions that can be delivered from all neighbors (*i.e.,* performs quality adaptation), *(ii)* it estimates the number of packets that can be collectively delivered by all neighbors and then determines what specific packets are required, and *(iii)* it sends a request to each neighbor containing an ordered list of required packets based on the neighbor's available content and bandwidth. When a peer receives a request from its neighbor, it delivers requested packets in the given order through a congestion controlled connection. In a nutshell, the rate of delivery is determined by a congestion control mechanism whereas delivered content is controlled by the receiving peer.

By controlling the delivered content from all neighbors, each peer effectively determines the evolution of its buffer state. The requested packets should evolve the buffer state at each peer such that the following three rather conflicting conditions are satisfied at any point of time. First, each peer should request and receive a proper number of unique packets (from different descriptions) for each timestamp before their playout time in order to ensure stability of delivered quality. Second, each peer should request a proper number of new packets (*i.e.,* packets with higher timestamps) from its neighbors in order to facilitate rapid diffusion and thus availability of new packets throughout the overlay. Third, buffer states at any pair of neighbors should remain diverse as they evolve so that they can exchange data and effectively utilize their pairwise bandwidth in both directions. Note that there is a tradeoff between these conditions. For example, if the available bandwidth is mostly used for the delivery of packets with lower timestamps to ensure in-time delivery, the receiver peer can not request many new packets, and the degree of diversity between buffers will be low. The content delivery mechanism in PRIME leverages this tradeoff in a balanced fashion, and randomizes requested packets from each peer to ensure diverse buffer states across neighbor peers. In summary, the key design questions are:
(i) How the available bandwidth is divided between requesting packets with close playout time versus new packets?
(ii) How the requested packets should be ordered to ensure diversity of buffers?

We are currently conducting simulation-based evaluation of PRIME to investigate several key issues including: *(i)* the effect of the packet request strategy at each peer on the global pattern of content diffusion throughout the overlay; *(ii)* the impact of overlay properties such as the distribution of node degree, population size, overlay dynamics, and the degree of bandwidth heterogeneity and asymmetry on delivered quality to individual peers; *(iii)* identifying whether the limiting factor on delivered quality to each peer is the *bandwidth bottleneck* or the *content bottleneck*. Further information on this project is available at *http://mirage.cs.uoregon.edu/PRIME*.

1. REFERENCES

[1] V. Agarwal and R. Rejaie. Adaptive Multi-source Streaming in Heterogeneous Peer-to-Peer Networks. In *MMCN*, 2005.

[2] V. N. Padmanabhan, H. J. Wang, and P. A. Chou. Resilient Peer-to-Peer Streaming. In *ICNP*, Nov. 2003.

[3] K. Sripanidkulchai, B. Maggs, and H. Zhang. An Analysis of Live Streaming Workloads on the Internet. In *IMC*, 2004.

[4] D. Stutzbach and R. Rejaie. Characterizing Two-Tier Overlay Topologies in Modern P2P File-Sharing Systems. Technical Report CIS-TR-2005-01, University of Oregon, Feb. 2005.

[5] X. Zhang, J. Liu, B. Li, and T. P. Yum. DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming. In *INFOCOM*, Mar. 2005.