MICHAEL PIATEK AND
ARVIND KRISHNAMURTHY

improving the performance and robustness of P2P live streaming with Contracts

Michael Piatek is a graduate student at the University of Washington. After spending his undergraduate years working on differential geometry, his research interests now include incentive design in distributed systems, network measurement, and large-scale systems building.

piatek@cs.washington.edu

Arvind Krishnamurthy is an associate research professor at the University of Washington, Seattle. His research interests are primarily at the boundary between the theory and practice of distributed systems, in the topics of peer-to-peer systems, network measurements, and network protocols. His recent projects include iPlane, a distributed service that performs network measurements, BitTyrant, an optimized content distribution system, Botlab, a pervasive monitoring infrastructure for Botnets, and OneSwarm, a privacy-preserving peer-to-peer system.

arvind@cs.washington.edu

THE INCREASING POPULARITY OF ONLINE streaming video is defining a major shift in the Internet's workload. To cope with the demands of distributing video, many popular services take a peer-to-peer approach, relying on users to redistribute video data after receiving it. PPLive is one such system, used daily for live streaming by millions of people worldwide. As with any P2P design, the scalability of PPLive depends on users contributing capacity to the system. But, currently, these contributions are neither verified nor rewarded. This article describes Contracts, an extension of the PPLive protocol, that improves performance by recognizing and rewarding users who contribute. For example, in our experiments, the fraction of PPLive clients using Contracts experiencing loss-free playback is more than four times that of native PPLive.

Live Streaming

The Internet is rapidly becoming one of the main distribution channels for video. Hulu, Netflix, and the BBC's iPlayer are just a few examples of well-known video streaming services. Increasingly, video distribution services support live content as well, e.g., sporting events.

The popularity of video streaming creates scalability challenges. Publishers would prefer that playback start quickly, continue without interruption, and have high quality. But achieving these goals is difficult given the realities of transient congestion, flash crowds, and network bottlenecks at the broadcast source.

To achieve scalability, many Internet video services use peer-to-peer (P2P) content distribution. P2P designs rely on users who have already received part of the video stream to redistribute that data to others. This decreases load on the broadcaster and provides more redundant data sources, increasing robustness.

PPLive is one of the most widely deployed live streaming services on the Internet today, serving more than 20 million active users spread across the globe. As with any P2P design, PPLive's scalability depends on its users contributing their capacity by redistributing video data. But the current PPLive

design neither verifies nor rewards contributions. All users are treated equally by the protocol, even if they contribute nothing whatsoever.

In this article, we examine how best to provide incentives for users to contribute resources to P2P live streaming systems. Our goal is to structure the system so that rational users will *want* to contribute resources because doing so will improve their performance. We use PPLive as a concrete example; it is one of the most popular P2P live streaming systems available today, and its developers were willing to work with us to provide modified binaries and workload data.

The goal of motivating contributions is common to many P2P designs, notably the popular BitTorrent file-sharing protocol. BitTorrent and PPLive take a similar approach to distribution: data is broken up into blocks and distributed by a source, with peers redistributing individual blocks after receiving them. Unlike PPLive, however, BitTorrent incorporates contribution incentives into its design. BitTorrent's policy is "tit-for-tat"—clients *reciprocate* by providing data to peers that give data in return. As a result, users interested in faster downloads should increase their upload contribution.

The similarity of BitTorrent and PPLive raises the question: *will tit-for-tat work for live streaming?* While intuitive and simple, we find that tit-for-tat is much less effective for live streaming than file sharing. We consider three challenges to applying tit-for-tat that motivate the design of our protocol, Contracts.

CAPACITY HETEROGENEITY

The picture quality of a video stream is determined by its data rate. For a P2P system, choosing the data rate depends on the total capacity of all the users in the system, including any seed capacity provided by the source. The maximum rate is the average capacity-beyond the average, bandwidth demand exceeds supply, and a rate lower than the average wastes capacity that could be used to increase quality.

Although streaming at the average capacity maximizes quality, doing so is incompatible with incentive strategies based on *strict reciprocation*; i.e., users trade one block sent for one block received. Strict reciprocation creates a strong incentive to contribute—to view the stream, users must contribute as much as they receive. But such a policy would exclude users with a capacity less than the average. In other words, for strict tit-for-tat to work well in live streaming systems, all users should have roughly the same capacity.

In practice, a defining feature of P2P workloads is that users' upload capacities vary significantly, and PPLive is no exception. Capacity measurements of more than 90,000 clients show that the top 10% of PPLive users contribute 58% of total capacity. This yields a discouraging trade-off. On one hand, streaming at the average rate maximizes quality, but would exclude 86% of PPLive clients when insisting on strict reciprocation. Alternatively, supporting 95% of users reduces capacity utilization to just 15%. In short, when applying strict reciprocation to live streaming, we can have high quality or robust incentives, but not both.

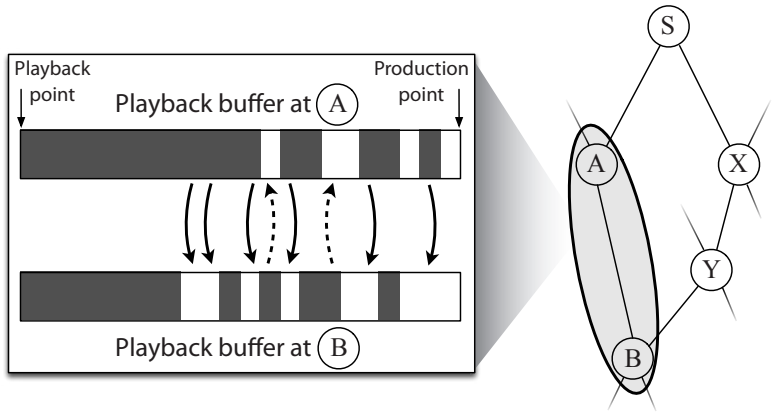

FIGURE 1: BLOCK TRADING OPPORTUNITIES IN A LIVE STREAMING MESH. DISTANT CLIENTS HAVE FEW OPPORTUNITIES FOR RECIPROCATION WITH PEERS CLOSER TO THE SOURCE.

LIMITED TRADING OPPORTUNITIES

An alternative to strict reciprocation is to allow imbalanced exchange. Rather than insisting on one block received for one block sent, imbalanced reciprocation schemes simply prioritize peers that contribute the most. But imbalanced exchange still depends on trading opportunities arising between peers—i.e., two peers exchanging blocks of mutual interest.

Unfortunately, live streaming provides clients with few opportunities for mutually beneficial trading between peers. The key property is that unlike bulk data distribution, where blocks have roughly equal value over time and among clients, *the value of blocks in live streaming varies over time and client*. A block has little value if it is received after the playback point at a client. Thus, the data useful to an individual client is limited to a narrow range between the production point and the local playback point.

This effect is shown in Figure 1. In this case, a snippet of the overlay mesh is shown. A source S sends blocks to directly connected peers A and X, who forward it to other peers in turn. Consider the trading opportunities between A and B. A is directly connected to the source—it receives most data immediately after it is produced. B is more distant, receiving data only after it has been forwarded by others. This reduces the trading opportunities between A and B, since virtually all of B's data blocks have already been received by A.

This example illustrates a general property: peers close to the source enjoy a near monopoly on new blocks, creating a trade imbalance that puts more distant peers at a perpetual disadvantage under any incentive scheme based on pairwise reciprocation between peers.

NO COMPELLING REWARD

The final obstacle we consider is the lack of a *compelling reward* for increasing contribution in live streaming. For bulk data distribution, the incentive to increase upload rate is a corresponding increase in download rate. But, live streaming is *inelastic*. To watch the stream, each user needs to download video data at the production rate. Even a small loss rate can quickly degrade the quality of a live video stream. On the other hand, neither can we increase download speeds to reward users—for live streams, additional video data has not yet been produced.

Streaming Incentives with Contracts

The challenges of using reciprocation lead us to take a different approach to providing contribution incentives in live streaming. Our scheme, Contracts, is based on two key design choices, which we summarize here. More details are available in our paper [1].

- Contracts evaluates contributions according to both their *amount* and *effectiveness*. As in any P2P system, PPLive benefits from users contributing as much as possible. But, in live streaming, contribution alone is not sufficient. Because data blocks must arrive in time to meet playback deadlines, Contracts prioritizes requests from peers with the greatest capacity—i.e., the peers that can replicate new data most quickly.

In Contracts, evaluating contributions and effectiveness must be *verifiable*. While we could rely on honest reporting of these values, a strategic client could easily misrepresent its contributions to game the system. To prevent this, each Contracts client bases its calculations of a peer's value on cryptographically strong *receipts* of contribution. Each receipt acknowledges that one user has sent some data blocks to another. Receipts are gossiped among peers, allowing clients to evaluate nearby peers in the mesh.

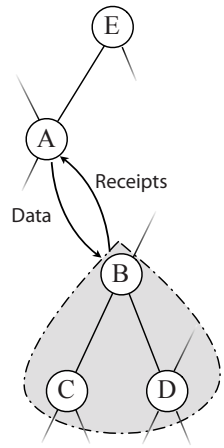

FIGURE 2: EVALUATING PEERS IN CONTRACTS. CLIENT *A* RECEIVES CRYPTOGRAPHIC RECEIPTS FROM *B* ACKNOWLEDGING DATA TRANSFER, AND FORWARDS THESE TO *E* TO DEMONSTRATE ITS CONTRIBUTION AMOUNT. TO DEMONSTRATE EFFECTIVENESS, *E* ALSO INCLUDES RECEIPTS FROM ITS ONE HOP NEIGHBORHOOD OF PEERS (SHADED).

As an example of how Contracts evaluates contributions, consider Figure 2. In this case, A is being evaluated at E, and E uses receipts from peers in the shaded region to perform its calculation. As A sends data to B, it receives receipts attesting to its contributions. A forwards these receipts to E. Receipts from B → A allow E to compute the *amount* of A's contribution. To compute *effectiveness*, E needs receipts from both A and B. Receipts from C, D → B show that A has made *effective* contributions to a peer (B) that replicated the data to others. In general, A would forward receipts from all of its peers (and peers of peers) to E. Contracts restricts the propagation of receipts to one hop, however, to limit the overhead of the protocol.

In addition to prioritizing service, receipts are used to update the mesh topology. Contracts restructures connections to move high capacity peers towards the source, promoting efficiency. In Figure 2, for example, suppose peer E is the closest to the broadcaster, and B has higher capacity than A.

During its evaluation of peers, E can recognize the mismatch while inspecting B's receipts and connect to B directly. This moves B closer to the source.

Performance and Incentives

Our evaluation of Contracts shows two main results: (1) PPLive with Contracts significantly outperforms both unmodified PPLive and PPLive modified to support tit-for-tat (TFT); (2) Contracts provides our intended contribution incentives; when the system is bandwidth constrained, increasing contribution improves performance. PPLive has adopted some of our techniques in their production code, and we continue to work towards full support in the public client. In this section, we report performance measurements of our Contracts PPLive prototype.

PERFORMANCE

We define performance as the fraction of data blocks received by their playback deadlines, and compare the performance of PPLive and PPLive using Contracts. We conducted a test broadcast of 100 users with arrivals and departures and upload capacities based on measurements of PPLive users. Crucially, in order to provide a meaningful comparison, we use a video data rate which exercises capacity constraints. (It would be easy to achieve good performance by simply over-provisioning the system substantially.)

We find that Contracts significantly improves performance relative to unmodified PPLive: 62% of Contracts clients experience loss-free playback compared with just 13% when using unmodified PPLive. In other words, the fraction of PPLive/Contracts clients experiencing loss-free playback is more than four times that of unmodified PPLive.

INCENTIVES

Contracts rewards contribution with increased robustness. We evaluate this by comparing the performance of PPLive using Contracts with that of PPLive using tit-for-tat. In both cases, the system is bandwidth constrained.

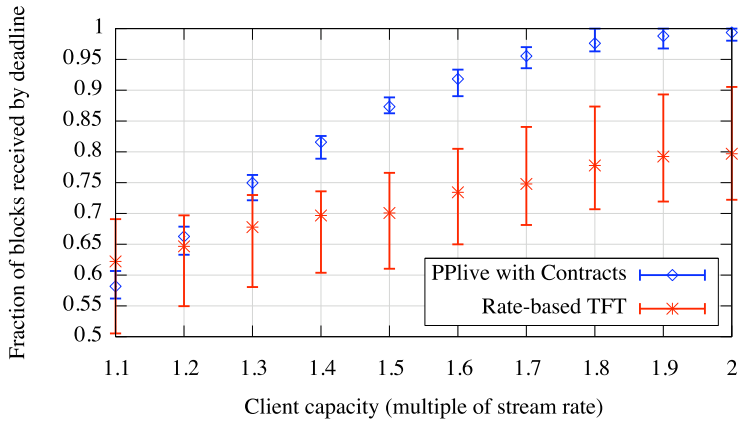

FIGURE 3: DELIVERY RATE AS A FUNCTION OF CONTRIBUTION. CONTRACTS PROVIDES LARGER AND MORE CONSISTENT REWARDS FOR INCREASING CONTRIBUTIONS.

In Figure 3, averages are shown with error bars giving the full range of block delivery rates for clients with a given capacity. While tit-for-tat does provide some correlation between contribution and performance, the amount of

improvement varies significantly because tit-for-tat does not update the topology. In contrast, Contracts combines both topology updates and priority service for contributors to provide a consistent improvement in performance, strengthening incentives.

Summary

P2P technology has emerged as a powerful technique for achieving scalability in live streaming systems. But scalability depends on users contributing their capacity, and in many systems, contributions are neither verified nor rewarded. The unique features of P2P live streaming limit the effectiveness of widely used incentive strategies based on reciprocation, e.g., tit-for-tat. Contracts provides an alternative: a new incentive strategy that rewards contribution with quality of service by evolving the overlay topology. Experiments using our prototype PPLive implementation show that Contracts both improves performance relative to PPLive and strengthens contribution incentives relative to existing approaches.

REFERENCES

[1] Michael Piatek, Arvind Krishnamurthy, Arun Venkataramani, Richard Yang, David Zhang, and Alexander Jaffe, "Contracts: Practical Contribution Incentives for P2P Live Streaming," *Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI '10)*, 2010.