

Maximizing Streaming Quality in Heterogeneous Overlays Through Incentives

Ilias Chatzidrossos
School of Electrical Engineering
KTH, Royal Institute of Technology, Sweden
iliasc@kth.se

ABSTRACT

The upload contribution of peers in a peer-to-peer streaming system depends on their willingness to contribute as well as their physical limitation. Several incentive schemes have been proposed to enforce non-willing peers to cooperate. But we find it of great interest to see how physically constrained, with respect to resources, peers can be supported by a streaming application. In this paper we investigate how free-riders, namely non-contributing peers, can be served in a peer-to-peer streaming system. We examine different prioritization schemes that are used by high contributing peers to prioritize other contributing peers over free-riders and show that as the level of prioritization increases, contributing peers receive higher quality but the average quality drops. To avoid this, we propose an incentive mechanism that encourages contributing peers to upload to free-riders so that the average quality experienced by the peers in the overlay is maximized.

Categories and Subject Descriptors

C.4 [Performance of systems]: Design studies

General Terms

Design, Performance

1. INTRODUCTION

Mesh-based systems have nowadays become the predominant solution for streaming applications. One of the major reasons is their flexibility and thus high utilization under heterogeneous and dynamically changing upload capacities. The upload capacity that each peer contributes with depends on two factors, its willingness to cooperate and the physical limitations.

There have been a number of schemes proposed to give incentives to selfish peers to cooperate (e.g. [1, 2]). These approaches relate a peer's contribution level with the quality it receives. Peers that contribute with much resources, receive higher quality. These schemes constitute a reasonable solution when peers can contribute but refrain from doing so. There are, however, peers that are incapable of contributing to the overlay or can only contribute with low resources, e.g., mobile hand-held devices. These peers would not be able to receive good quality under such incentive schemes,

since they cannot increase their cooperation level. Furthermore, if peers are selfish they prefer to become neighbors and exchange data with high contributing peers, because they can receive the data faster and they do not have to upload much. This makes high contributing peers to form clusters excluding low contributing ones.

Nevertheless, for a streaming services provider it is of great interest to be able to have as large an overlay as possible. This means that peers that cannot contribute as much, should not be denied to be served as long as there is available capacity to serve them. To this end, the scope of our current work is to achieve the best possible streaming for all peers in the overlay by introducing priority schemes and giving incentives to the high contributing peers to upload to low contributing ones.

2. SYSTEM DESCRIPTION

In order to understand the effect of priority schemes on the streaming quality, we consider the simplified scenario of an overlay consisting of contributing and free-riding peers. Contributing peers have an upload capacity of c times the stream rate. We denote the ratio of free-riders in the overlay by α . Peers use a push algorithm to exchange data among themselves. Packet forwarding is based on information about packet availability. This information is exchanged in the form of buffer maps, which contain information about the already received packets. We consider priority schemes that contributors could use in order to tune the service that free-riders receive. The first scheme is based on altering the forwarding logic of a peer, while the latter relies on altering the structure of the overlay, keeping the forwarding logic intact. These schemes are:

Probabilistic forwarding (PF): A peer splits its neighbors into two groups, that of contributors and that of free-riders. Whenever it sends something it randomly chooses which of the groups it should pick the recipient from. We denote by p the probability that it chooses a peer from the free-rider group and define the priority factor β as the ratio p/α .

Control maximum number of neighbors of free-riders (CF): A free-rider has a maximum number of connections that is a fraction of the maximum number of connections of a contributor. We denote this ratio by ρ .

Throughout our analysis we assume that a peer can recognize which of its neighbors are free-riders. The mathematical model constitutes an extension of our previous work [3], where we studied the data propagation in a homogeneous overlay with peers contributing with upload capacity equal to the stream rate.

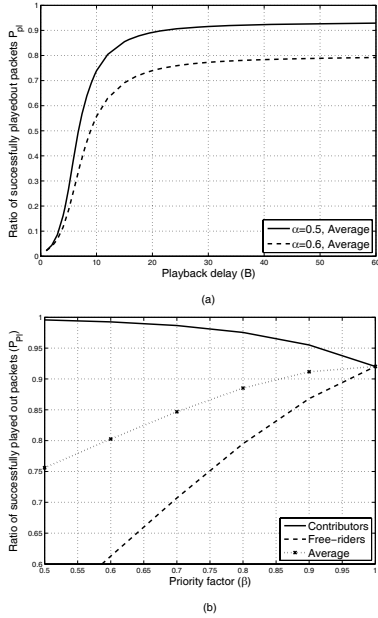


Figure 1: PF priority scheme. Ratio of successfully played out packets vs playback delay and priority factor for different values of α . Model results.

In Fig. 1 we present results obtained for the PF priority scheme as an illustration of how these priority schemes work. These results correspond to an overlay of 500 peers where each peer has 30 neighbors and the contributing peers can forward 2 times the stream rate ($c = 2$). Fig. 1a shows the ratio of successfully played out packets vs the playback delay for two different ratios of free-riders, without any priorities ($\beta = 1$). When there is available capacity in the overlay to support all peers ($\alpha = 0.5$), the playout probability asymptotically reaches 1. In the case where the ratio of free-riders makes it impossible to stream to all peers ($\alpha = 0.6$), the playout probability converges to a value less than 1.

Fig. 1b shows the playout probabilities of contributing and free-riding peers versus the priority factor (β) for an overlay with $\alpha = 0.5$ and buffer size of 40 packets ($B = 40$). When no prioritization is used ($\beta = 1$), contributors and free-riders have the same playout probability. Let us also note that at that point the average playout probability attains its maximum, as shown graphically by the marked dotted line. As the contributors are prioritized over free-riders ($\beta < 1$), their playout probability converges to one, while that of free-riders drops significantly. Furthermore, the average playout probability is reduced. Namely, the gain for the contributors is lower than the loss for the free-riders. This can be explained by the increase in the ratio of the duplicate packets that emerges when data is exchanged among contributors.

If peers in the overlay are not selfish, namely they do not discriminate free-riders, then there is no prioritization ($\beta = 1$). If peers are selfish, though, then the contributors will try to form a cluster and exchange data among themselves, which means a move towards the left in Fig. 1b. Although, this yields a higher quality to them, it decreases the average quality of all peers.

3. SERVER ASSISTED STREAMING WITH INCENTIVES

In order to make the system operate at a point that is close to the optimal one ($\beta = 1$), contributors must be encouraged to forward data to free-riders. This can be achieved through an incentive mechanism, that offers contributors a playout probability at least as high as the one they can achieve by forming a cluster. Such a mechanism cannot be enforced by the peers themselves, since no contributor would, by itself, find it rational to upload to free-riders. Thus, a distributed incentive mechanism cannot be implemented.

Nevertheless, the allocation of server's available upload capacity can be used as an incentive for contributors to upload to free-riders as well. In the following, we describe our incentive mechanism and prove its validity by showing that under this mechanism contributors are at least as well off by uploading to free-riders.

If all contributors in the overlay team up, the playout probability for them is $1 - \epsilon$, where $\epsilon > 0$. However, with the use of a push-pull system with incentives, the server can provide the extra capacity required for the contributors to achieve a playout probability of 1. The condition for providing this extra capacity would be for the contributors to keep some free-rider neighbors as well. Given its available capacity, the server asks the contributors to reach a certain level of playout probability, using any of the prioritizing schemes, and if they reach that level, it can provide the remaining portion. Let us denote by m_r the reserved capacity of the server to serve pull requests from contributors. The server calculates the threshold playout probability below which it would not be able to serve all contributors with the remaining part of the stream as $P_t = 1 - \frac{m_r}{(1-\alpha)N}$, where N is the total number of peers. If contributors exceed that threshold by forming clusters, they are not entitled to the rest of the stream, and end up with a playout probability of $1 - \epsilon$.

The server verifies that contributors serve free-riders by having peers report their playout probability as well as that of their neighbors, where the playout probability of a neighboring peer is estimated based on the received buffer maps.

4. CONCLUSIONS AND FUTURE WORK

In this work, we presented a scheme that maximizes the quality of peers in a heterogeneous p2p streaming system using the server capacity as an incentive for the high contributors to serve low contributing peers as well. We would like to make a detailed comparison between priority schemes based on the overlay structure and those based on altering the forwarding algorithm. Finally, we intend to study the equilibria that emerge under these prioritization schemes and various scenarios with respect to the composition of the overlay and the utility functions of the participating peers.

5. REFERENCES

- [1] A. Habib and J. Chuang. Service differentiated peer selection: An incentive mechanism for peer-to-peer media streaming. *IEEE Transactions on Multimedia*, 8:610–621, June 2009.
- [2] Thomas Silverston, Olivier Fourmaux, and Jon Crowcroft. Towards an incentive mechanism for peer-to-peer multimedia live streaming systems. In *IEEE International Conference on Peer-to-Peer Computing*, 2008.
- [3] I. Chatzidrossos, G. Dán, and V. Fodor. Delay and playout probability trade-off in mesh-based peer-to-peer streaming with delayed buffer map updates. *P2P Networking and Applications*, 2009.