Server Guaranteed Cap: An incentive mechanism for maximizing streaming quality in heterogeneous overlays

Ilias Chatzidrossos, György Dan, Viktoria Fodor

Work in Progress: Server Guaranteed Cap: An incentive mechanism for maximizing streaming quality in heterogeneous overlays

Ilias Chatzidrossos, György Dán, and Viktória Fodor

{iliasc,gyuri,vfodor}@kth.se
School of Electrical Engineering
KTH, Royal Institute of Technology
Osquldas väg 10, 100-44, Stockholm, Sweden

Abstract. We address the problem of maximizing the social welfare in a peer-to-peer streaming overlay given a fixed amount of server upload capacity. We show that peers' selfish behavior leads to an equilibrium that is suboptimal in terms of social welfare, because selfish peers are interested in forming clusters and exchanging data among themselves. In order to increase the social welfare we propose a novel incentive mechanism, Server Guaranteed Cap (*SGC*), that uses the server capacity as an incentive for high contributing peers to upload to low contributing ones. We prove that *SGC* is individually rational and incentive compatible. We also show that under very general conditions, there exists exactly one server capacity allocation that maximizes the social welfare under *SGC*, hence simple gradient based method can be used to find the optimal allocation.

Keywords: p2p streaming, incentive mechanisms, social welfare

1 Introduction

The goal of peer-to-peer (p2p) streaming systems is to achieve the maximum possible streaming quality using the upload capacities of the peers and the available server upload capacity. In general, the achievable streaming quality depends heavily on the aggregate upload capacity of the peers [1]. Hence, a key problem of p2p streaming systems is how to give incentives to selfish peers to contribute with all their upload capacity. Numerous schemes were proposed to solve this problem (e.g., [2, 3]). These schemes relate peers' contribution with the streaming quality they receive: the more a peer contributes, the better streaming quality it can potentially receive. The correlation of peer contribution to the quality it receives is based on the assumption that all peers are capable of contributing but refrain from doing so.

Nevertheless, peers might be unable to have a substantial contribution with respect to the stream rate because of their last-mile connection technology. Most DSL and cable Internet connections are asymmetric, hence peers may have sufficient capacity to, e.g., download high definition video but insufficient for forwarding it. Similarly, in high-speed mobile technologies, such as High Speed Downlink

Packet Access (HSDPA), the download rates are an order of magnitude higher than the upload rates [4]. Peers using assymetric access technologies would receive poor quality under incentive schemes that offer a quality proportional to the level of peer contribution.

Furthermore, using such incentive schemes, high contributing peers maximize their streaming quality if they prioritize other high contributing peers when uploading data. As a consequence, peers with similar contribution levels form clusters and exchange data primarily among themselves. While high contributing peers can achieve excellent streaming quality this way, the quality experienced by low contributing peers is low, and the average streaming quality in the p2p system is suboptimal.

In order to increase the average streaming quality in the system, we propose a mechanism that gives incentives to high contributing peers to upload to low contributing ones. The mechanism relies on reserving a portion of the server capacity and providing it as a safety resource for high contributing peers who meet certain criteria. We show that high contributing peers gain by following the rules set by the incentive mechanism, and they fare best when they follow the rules truthfully. We also show that due to some basic properties of p2p streaming systems our mechanism can easily be used to maximize the streaming quality.

The rest of the paper is organized as follows. In Section 2, we motivate our work by studying the effect of selfish peer behavior in a push-based p2p streaming overlay. In Section 3, we describe our incentive mechanism and provide analytical results. We show performance results in Section 4. In Section 5 we discuss previous works on incentives in peer-to-peer streaming systems. Finally, Section 6 concludes our paper.

2 Motivation

We consider mesh-based p2p streaming systems to evaluate the effect of selfish peer-behavior. Due to their flexibility and easy maintenance, mesh-based systems received significant attention in the research community [5–8], and are underlying the majority of commercial streaming systems (e.g., [9], [10]).

2.1 Case study: a mesh-based p2p streaming system

The streaming system we use as an example was proposed in [7] and was subsequently analyzed in [8, 11]. The system consists of a server and N peers. The upload capacity of the server is m_t times the stream rate. For simplicity we consider two types of peers: peers with high upload capacity, called contributors, and peers without upload capacity, called non-contributors. The upload capacity of the contributors is c times the stream rate, while that of non-contributors is zero. We denote by α the ratio of non-contributors in the overlay.

Each peer is connected to d other peers, called its neighbors. The server is neighbor to all peers. Every peer maintains a playout buffer of recent packets, and exchanges information about its playout buffer contents with its neighbors periodically, via so called buffer maps. The server sends a copy of every packet to m_t randomly chosen peers. The peers then distribute the packets among each other according to a forwarding algorithm. The algorithm takes as input the information about packet availability in neighboring peers (known from the buffer maps) and produces a forwarding decision, consisting of a neighbor and a packet sequence number. In this work we consider the Random Peer - Random Packet (*RpRp*) forwarding algorithm. This algorithm has been shown to have a good playback continuity - playback delay tradeoff ([7, 11]). According to this algorithm, a sending peer first chooses randomly a neighbor that is missing at least one of the packets the sending peer possesses, then it selects at random the missing packets to send. Peers play out data B time after they were generated by the server, and we refer to this as the playback delay.

To study the impact of peer cooperation in the overlay, we introduce the notion of the generosity factor, which we denote by β. This parameter shows how generous a peer is towards its non-contributing neighbors, and can be expressed as the ratio of the probability of uploading to a non-contributor over the ratio of a peer's non-contributing neighbors. The generosity factor takes values in the interval $[0, 1]$. When $\beta = 1$, the peers are completely generous and they upload to their neighbors regardless of whether they, on their turn, are uploading or not. When $\beta = 0$, peers are not generous at all, or equivalently completely selfish, and will only upload to peers that upload as well.

The generosity level affects the playout probabilities of the contributing and non-contributing peers. At $\beta = 1$ the two playout probabilities are equal. As β decreases, capacity is subtracted from the non-contributors and added to the contributors, and consequently the playout probability of the contributors increases, while that of non-contributors decreases.

2.2 Playout probability, individual utility and social welfare

The performance of p2p streaming systems is usually measured in terms of the playout probabilities of the peers, i.e., the probability p_i that peer i receives packets before their playout deadlines [7, 11, 8]. The impact of the playout probability p_i on the peers' satisfaction is, however, typically influenced by the loss resilience of the audiovisual encoding. To allow for a wide range of encodings, we use utility functions to map the playout probability to user satisfaction. Formally, the utility function is a mapping $u : [0, 1] \to [0, 1]$. We consider three kinds of utility functions.

Linear function: Utility function of the form $y = a \cdot p_i + b$. An improvement in the playout probability yields the same increase in utility regardless of the already achieved playout probability.

Concave function: Utility is a concave function of the playout probability, that is, the marginal utility is a non-increasing function of the playout probability.

Step function: There is an instantaneous transition from a zero utility to a utility of a unit upon reaching a threshold p_i^*. The peer is only satisfied above the threshold playout probability.

We measure the aggregate utility of the peers, called the social welfare, using the utilitarian welfare model. In the utilitarian welfare model the social welfare is the sum of the utilities, which is equivalent to the average peer utility

$$SWF = (1 - \alpha) \cdot u(p_c) + \alpha \cdot u(p_{nc}), \tag{1}$$

where p_c and p_{nc} denote the playout probability of contributors and non-contributors respectively.

2.3 The effects of selfish behavior

In the following we show the effects of selfish behavior on the social welfare. The numerical results we show, were obtained using an analytical model and via simulations. The analytical model is an extension of our previous work [11], where we developed a model of the playout probability in a push-based system with homogeneous peer upload capacities. We extended the model to incorporate two types of peers, contributors and non-contributors. The extended model can be found in [12]. The simulation results were obtained using the packet-level event-driven simulator used in [11]. In the simulations, nodes join the overlay at the beginning of the simulation, and are organized into a random d-regular graph. After the overlay with N peers is built, the data distribution starts according to the forwarding algorithm described in Section 2.1. The algorithm is executed in time slots in a way that contributors with capacity c make c forwarding decisions per slot. All results presented in the following refer to an overlay of one server with upload capacity $m_t = 11$ times the stream rate and $N = 500$ peers, where each peer is connected to $d = 30$ neighbors and contributors have an upload capacity of $c = 2$ times the stream bitrate.

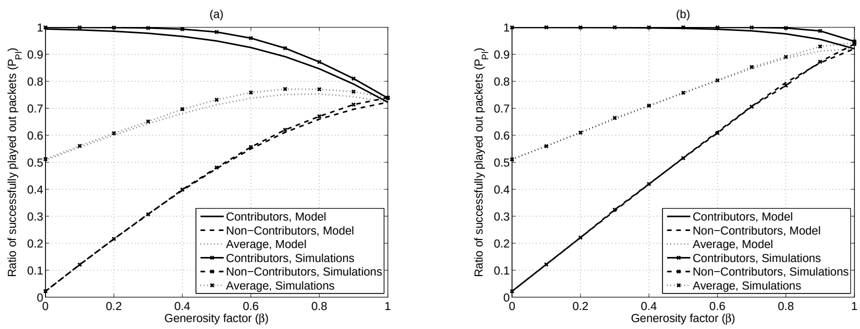

Fig. 1. Ratio of successfully played out packets vs generosity factor β for playback delay of (a) $B = 10$ and (b) $B = 40$. Analytical and simulation results.

Fig. 1a and 1b show the effect of the generosity factor on the playout probability of the contributors and the non-contributors obtained using the model and simulations. The figures also show the average playout probability in the overlay (dotted lines). The ratio of non-contributors is $\alpha = 0.5$.

Fig. 1a shows a system where the playback delay is small. Clearly, contributors maximize their playout probabilities for $\beta = 0$, but the average playout probability is suboptimal in this case. The average playout probability is suboptimal for $\beta = 1$ as well. For a larger playback delay (Fig. 1b) the average playout probability is maximized for $\beta = 1$.

From the figures we can conclude that the optimal generosity factor depends on the playback delay. At low playback delays high capacity is needed for on time delivery, and therefore contributors can receive and efficiently forward packets only at low values of β. At high playback delays though, increased capacity at contributors leads to marginal gains only, and therefore $\beta = 1$ is optimal. This inefficiency of the forwarding algorithm is due to the lack of coordination between the peers [13]. Under push-based algorithms, like the one considered here, the lack of coordination leads to duplicate packet receptions at peers, i.e., a peer receives the same data from more than one of its neighbors. Fig. 2 shows how the probability of receiving duplicate packets increases together with the capacity allocated for forwarding among contributors. This in turn leads to playout probabilities below 1 even when contributors are complectly selfish ($\beta = 0$). Similarly, in the case of pull-based systems the lack of coordination may lead to request collision at the peers, with the consequence that some of the requests can not be served and the packet miss ratio can become substantial [14]. Based on these findings, we argue that performance degradation due to peer clustering is intrinsic to uncoordinated p2p dissemination algorithms.

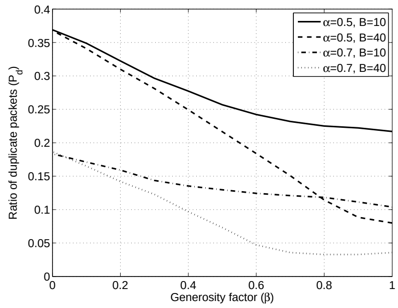

Fig. 2. Ratio of duplicate transmissions vs generosity factor (β) for $B = 10$ and $B = 40$. Simulations.

Next, we proceed with our utility based analysis of the overlay. For linear utility function we use $u(p_i) = p_i$, so the utility curve coincides with the curve presented in Fig. 1a and 1b. For concave utility we use a logarithmic function, $u(p_i) = log_{10}(1 + 9p_i)$. For the step function we set the threshold to $p_i^* = 0.95$. Our conclusions do not depend on the the particular values of the parameters and the choice of the logarithmic function.

Fig. 3a and 3b show the social welfare versus the generosity factor for the three kinds of utility functions and for playback delays of $B = 10$ and $B = 40$ respectively. In the case of small playback delay ($B = 10$) the social welfare for the linear and the concave utility functions attains its maximum for $\beta < 1$. For the step function the social welfare equals 0 for high values of β, when contributors are not able to receive at least with probability $p_i^* = 0.95$. As β decreases, there is a transition in utility, but the contributors do not gain

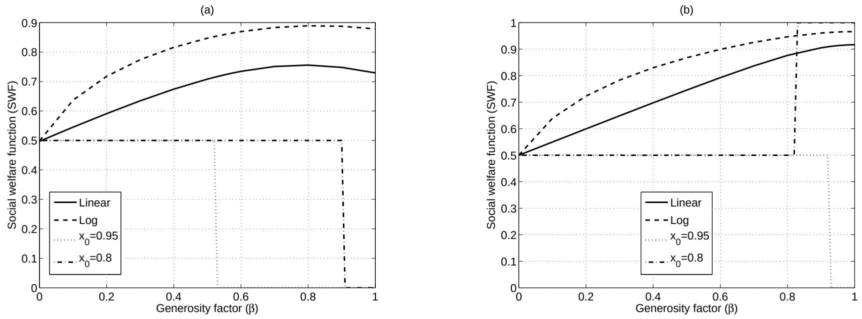

Fig. 3. Social welfare vs. generosity factor β for playback delays of (a) $B = 10$ and (b) $B = 40$. Overlay with $\alpha = 0.5$. Analytical results.

anything by becoming more selfish after the transition, and the social welfare remains constant. In the case of large playback delay ($B = 40$), we see that the social welfare for linear and concave utility functions attains its maximum for $\beta = 1$. For the step function we observe a similar transition of the social welfare as for $B = 10$, but at a higher value of the generosity factor β. The transition occurs where the contributors achieve a playout probability of $p_i^* = 0.95$. To understand the importance of the threshold value, let us consider $p_i^* = 0.8$. We see in Fig. 1b that in this case, the social welfare becomes maximal for $\beta \geq 0.8$, as both contributors and non-contributors achieve playout probabilities above the threshold. To summarize, we draw two conclusions from these figures. First, for the linear and concave utility functions the value of β that maximizes the social welfare is a function of the playback delay, but in general $\beta = 0$ is far from optimal. Second, for the step function the threshold value p_i^* plays an important role in whether $\beta = 0$ is optimal.

3 The SGC incentive mechanism

Our work is motivated by the observation that the peers' selfish behavior leads to a loss of social welfare. In our solution, we exploit the inability of contributors to achieve the maximum playout probability by being selfish and offer them seamless streaming if they increase their generosity, that is if they serve non-contributors as well. In the following, we describe our incentive mechanism, called Server Guaranteed Cap (SGC).

Under SGC, there are two types of data delivery: *p2p dissemination* and *direct delivery* from the server. Fresh data is distributed in the overlay using *p2p dissemination*: the peers forward the data according to some forwarding algorithm. Contributors can also request data *directly from the server* if they do not exceed a threshold playout probability of T_p via p2p dissemination. In our scheme the server ensures that by combining p2p dissemination and direct delivery the contributors possess all data with probability 1. In order to be able to serve the requests for direct delivery, the server reserves m_r of its total upload capacity m_t for direct delivery. m_r has to be large enough to cap the gap between the threshold probability T_p and 1. Given the number of contributors in the overlay and the reserved capacity m_r, the server can calculate the threshold value of the playout probability below which it would not be able to cap all contributors

$$T_p = 1 - \frac{m_r}{(1-\alpha) \cdot N}. \tag{2}$$

The server advertises the threshold value T_p as the maximum playout probability that contributors should attain through p2p dissemination. In turn the peers report their playout probabilities p_i achieved via p2p dissemination to the server. Based on these reports, the server knows which are the contributors with $p_i \leq T_p$, that is, which contributors are entitled for direct delivery.

3.1 Properties of SGC

In the following we show two important properties of the proposed mechanism: ex-post individual rationality and incentive compatibility [15].

Ex-post individual rationality means that a contributing peer does *not* achieve *inferior* performance by following the rules of the mechanism irrespective of the playout probability it would achieve without following the mechanism.

Proposition 1. *The SGC mechanism is ex-post individually rational.*

Proof. Consider that the server advertises a threshold probability of T_p. All contributors that receive up to $p_i \leq T_p$ via p2p dissemination are entitled to pull the remaining $1 - T_p$ directly from the server. Hence a peer with $p_i = T_p$ receives data with probability $P_i = p_i + (1 - T_p) = 1$, which is at least as much as it would achieve by not following the rules of the mechanism. □

Since *SGC* relies on peers reporting their playout probabilities to the server, it is important that peers do not have an incentive to mis-report their playout probabilities. In the following we show that *SGC* satifies this criterion, i.e., it is incentive compatible.

Proposition 2. *The SGC mechanism is incentive compatible.*

Proof. Let us denote the playout probability of peer i by p_i and the probability it reports by $\overline{p}_i$. As before T_p is the threshold probability that the contributors must not exceed in order to be directly served by the server, and m_r is the corresponding reserved capacity at the server. Contributors can receive $m_r/(1-\alpha)N = 1-T_p$ share of the stream from the server directly if they report $\overline{p}_i \leq T_p$. Consequently, if peer i achieves $p_i \leq T_p$ and reports it truthfully ($\overline{p}_i = p_i$), it receives data with probability $P_i = min(1, p_i + (1 - T_p))$. If $p_i > T_p$ and peer i reports truthfully, it receives with probability $P_i = p_i$.

Clearly, peer i can not benefit from over-reporting its playout probability, so we only have to show that it has no incentive for under-reporting it either. In order to show this we distinguish between three cases.

– $\overline{p_i} < p_i \leq T_p$: the playout probability that the peer will finally receive will be $P_i = min(1, p_i + 1 - T_p) \leq 1$, which is the same that it would receive if it were telling the truth.
– $\overline{p_i} \leq T_p < p_i$: the playout probability that the peer will finally receive will be $P_i = min(1, p_i + 1 - T_p) = 1$. The peer could achieve the same by having $p_i = T_p$ and reporting $\overline{p} = p$.
– $T_p < \overline{p_i} < p_i$: the peer is not entitled to direct delivery, so $P_i = p_i$. □

3.2 Optimal server capacity allocation

A key question for the implementation of the mechanism is how to determine the advertised probability threshold T_p, that is, how to find the reserved capacity m_r, that maximizes the social welfare. Since the server capacity is fixed, the choice of m_r affects the server capacity available for the p2p dissemination, and hence the efficiency of the data delivery through p2p dissemination.

In the following we show that for a wide class of p2p streaming systems there is a unique value of m_r that maximizes the social welfare, and this class is characterized by the fact that the marginal gain of increasing the upload capacity in the system is non-increasing.

Let us express the playout probability achieved through p2p dissemination as a function of the overlay size N and the p2p upload capacity. We denote the p2p upload capacity by C, and it is the sum of $m_t - m_r$ and the aggregate upload capacity of the contributors. We define the mapping $f : (\mathbb{N}, \mathbb{R}) \rightarrow [0, 1]$ of number of peers in an overlay and the p2p upload capacity, to the average playout probability of the peers. Clearly, f depends on the implemented forwarding algorithm.

Definition 1. *A p2p streaming system is called efficient if the playout probability of the peers is a concave function of the p2p upload capacity C.*

We only consider *linearly scalable* systems, where the efficiency of the forwarding algorithm does not depend on the overlay size for a given ratio of peers over p2p upload capacity, that is, $f(k \cdot N, k \cdot C) = f(N, C), \forall k \in \mathbb{R}$. Given that, we formulate the following proposition.

Proposition 3. *The construction of an efficient p2p streaming system is always possible regardless of the characteristics of the forwarding algorithm used.*

Proof. Suppose that f is strictly convex in an interval in its domain. Formally, there exist p2p upload capacity values C_1, C_2, with $C_1 < C_2$, for which it holds

$$\lambda f(N, C_1) + (1 - \lambda) f(N, C_2) > f(N, \lambda C_1 + (1 - \lambda) C_2), \forall \lambda \in (0, 1). \tag{3}$$

Let us consider a system with N peers and p2p upload capacity C, $C_1 < C < C_2$. We split the overlay into two partitions, one with size λN and server capacity λC_1 and the other with $(1 - \lambda)N$ peers and $(1 - \lambda)C_2$ server capacity, such that $C = \lambda C_1 + (1 - \lambda)C_2$. For the two overlays we have that

$$f(\lambda N, \lambda C_1) = f(N, C_1) \tag{4}$$
$$f((1-\lambda)N, (1-\lambda)C_2) = f(N, C_2). \tag{5}$$

Consequently, for the original overlay we have $f(N, C) = f(N, \lambda C_1 + (1 - \lambda)C_2) = \lambda f(N, C_1) + (1-\lambda)f(N, C_2)$, which contradicts (3). That is, by splitting the overlay and applying the same forwarding algorithm in the two parts independently, we can create an efficient p2p streaming system. □

For a given server capacity, m_t, *SGC* requires that the server caps the gap between the playout probability p_i achieved via p2p dissemination and 1. Therefore, the value of m_r should be such that contributors can achieve T_p for some $\beta \in [0, 1]$.

Definition 2. *The feasible range for the implementation of* SGC *is then defined as* $\mathbb{M}_r = \{m_r \in (0, m_t) : m_r \geq (1 - f\big((1-\alpha) \cdot N, C\big)) \cdot (1-\alpha) \cdot N\}$*, where* C *is the p2p upload capacity.*

Proposition 4. *For an efficient system, a feasible range of server upload capacities* $\mathbb{M}_r$ *and a concave utility function, the social welfare is a concave function of the reserved server capacity* m_r*.*

Proof. The social welfare of the system is given as

$$SWF = (1-\alpha) \cdot u\Big(f((1-\alpha) \cdot N, C_c) + \frac{m_r}{(1-\alpha) \cdot N}\Big) + \alpha \cdot u\Big(f(\alpha \cdot N, C_{nc})\Big), \tag{6}$$

where C_c and C_{nc} are the upload capacities allocated to contributors and to non-contributors respectively (as a function of β), and $C = C_c + C_{nc}$. Under the *SGC* mechanism, the contributors receive the stream with probability 1, so the above equation becomes

$$SWF = (1-\alpha) \cdot u(1) + \alpha \cdot u\Big(f(\alpha \cdot N, C_{nc})\Big). \tag{7}$$

The first part of the sum is constant in m_r, so we have to show that the second part of the sum is a concave function of m_r. First we evaluate C_{nc}. Since our system is efficient the playout probability p_i is concave with respect to C_c, or equivalently C_c is convex in p_i=T_p. Since $C_{nc} = C_t - C_c$, C_{nc} is concave in $1 - T_p$, which in turn is linear in m_r. Therefore, C_{nc} is a concave function with respect to m_r. Consequently the composite function $f(C_{nc})$ is concave as well with respect to m_r, as a composition of non-decreasing concave functions [16]. For the same reason $u \circ f$ is concave with respect to m_r, which proves the proposition. □

A consequence of *Proposition 4* is that the social welfare function SWF has exactly one, global, maximum on $\mathbb{M}_r$. Hence, the server can discover the optimal amount of reserved capacity m_r by using a gradient based method starting from any $m_r \in \mathbb{M}_r$.

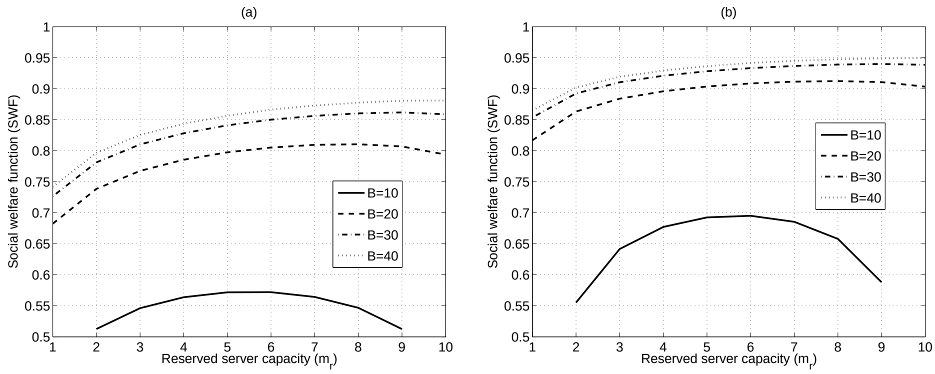

Fig. 4. Social welfare versus reserved server capacity for different playback delays. Linear (a) and logarithmic (b) utility functions. Overlay with $N = 500$, $d = 30$ and $m = 11$.

4 Numerical results

In the following we present numerical results that quantify the gain of the proposed incentive mechanism. The social welfare with respect to the reserved capacity by the server is shown in Fig. 4. The total capacity of the server is $m_t = 11$. We can see that the feasible region of *SGC* depends on the playback delay for the system. For B=10, it holds that $m_r \in [2, 9]$, while for larger playback delays $m_r \in [1, 10]$. The increase of m_r triggers two contradicting effects. On one side, it increases the playout probability of contributors through the direct delivery. On the other side, it decreases the efficiency of the p2p dissemination phase, since the amount of server capacity dedicated to that type of dissemination is decreased. The social optimum is at the allocation where the rate of decrease of the efficiency of p2p dissemination becomes equal to that of the increase achieved through the direct delivery.

Finally, we note that even using *SGC* there is a *loss of social welfare* compared to the *hypothetical case* when β is optimized by generous peers to maximize the social welfare. We can observe this loss by comparing the maximum social welfare obtained in Figs. 4 to that in Fig. 3 (for $B = 10$ and $B = 40$). This loss of social welfare is the *social cost of the selfishness of peers.*

5 Related work

A large number of incentive mechanisms was proposed in recent years to solve the problem of free-riding in p2p streaming systems. These mechanisms are either based on pairwise incentives or on global incentives.

Pairwise incentive schemes were inspired by the tit-for-tat mechanism used in the BitTorrent protocol [3, 17]. However, tit-for-tat, as used in BitTorrent, was shown not to work well in live streaming with random neighbor selection [3, 17]. The authors in [17] proposed an incentive mechanism for neighbor selection based on playback lag and latency among peers, achieving thus better pairwise performance. In [18], the video was encoded in layers and supplier peers favored neighbors that uploaded back to them, achieving thus service differentiation as well as robustness against free-riders.

Global incentive schemes take into account the total contribution of a peer to its neighbors. In [2], a rank-based tournament was proposed, where peers are ranked according to their total upload contribution and each peer can choose as neighbor any peer that is below itself in the ranked list. Thus, peers that have high contribution have also higher flexibility in selecting their neighbors. In [19], the authors proposed a payment-based incentive mechanism, where peers earn points by uploading to other peers. The supplier peer selection is performed through first price auctions, that is, the supplier chooses to serve the peer that offers her the most points.

All the aforementioned incentive mechanisms assume that peers are always capable of contributing but, due to selfishness, refrain from doing so. We, on the contrary, consider peers that are unable to contribute because of their access technologies. Associating streaming quality with contribution unnecessarily punishes these weak peers. Therefore our goal is to maximize the social welfare in the system, by convincing high contributing peers to upload to low contributing peers as well. In this aspect, our work is closely related to [20], where a taxation scheme was proposed, based on which high contributing peers subsidize low contributing ones so that the social welfare is maximized. However, in contrast to [20], where it is assumed that peers voluntarily obey to the taxation scheme and they can only react to it by tuning their contribution level, we prove that our mechanism is individually rational and incentive compatible. To the best of our knowledge our incentive scheme is unique in these two important aspects.

6 Conclusion

In this paper we addressed the issue of maximizing the social welfare in a p2p streaming system through an incentive mechanism. We considered a system consisting of contributing and non-contributing peers and studied the playout probability for the two groups of peers. We showed that when contributing peers are selfish the system operates in a state that is suboptimal in terms of social welfare. We proposed an incentive mechanism to maximize the social welfare, which uses the server's capacity as an incentive for contributors to upload to non-contributing peers as well. We proved that our mechanism is both individually rational and incentive compatible. We introduced the notion of efficient p2p systems and proved that for any efficient system there exists exactly one server resource allocation that maximizes the social welfare. An extension of our scheme to several classes of contribution levels is subject of our future work.

References

1. R. Kumar, Y. Liu, and K. Ross. Stochastic fluid theory for p2p streaming systems. In *IEEE INFOCOM*, 2007.

2. 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.
3. T. Silverston, O. Fourmaux, and J. Crowcroft. Towards an incentive mechanism for peer-to-peer multimedia live streaming systems. In *IEEE International Conference on Peer-to-Peer Computing*, 2008.
4. T. Chahed, E. Altman, and S.E. Elayoubi. Joint uplink and downlink admission control to both streaming and elastic flows in CDMA/HSDPA systems. *Performance Evaluation*, 65:869–882, November 2008.
5. R. Rejaie N. Magharei. Prime: Peer-to-peer receiver-driven mesh-based streaming. In *Proc. of IEEE INFOCOM*, 2007.
6. Y. Tang M. Zhang, L. Zhao, J-G. Luo, and S-Q. Yang. Large-scale live media streaming over peer-to-peer networks through the global internet. In *Proc. ACM Workshop on Advances in peer-to-peer multimedia streaming (P2PMMS)*, 2005.
7. T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg. Epidemic live streaming: Optimal performance trade-offs. In *Proc. of ACM SIGMETRICS*, 2008.
8. C. Liang, Y. Guo, and Y: Liu. Investigating the scheduling sensitivity of p2p video streaming: An experimental study. *IEEE Transactions on Multimedia*, 11:348–360, April 2009.
9. PPLive. http://www.pplive.com/en/about.html.
10. UUSee. http://www.uusee.com.
11. 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*, May 2009.
12. I. Chatzidrossos, G. Dán, and V. Fodor. Server Guaranteed Cap: An incentive mechanism for maximizing streaming quality in heterogeneous overlays. Technical Report TRITA-EE 2009:059, KTH, Royal Institue of Technology, Dec 2009.
13. F. Picconi and L. Massoulié. Is there a future for mesh-based live video streaming? In *IEEE International Conference on Peer-to-Peer Computing*, 2008.
14. C. Liang, Y. Guo, and Y. Liu. Is random scheduling sufficient in p2p video streaming? In *Proc. of the 28th International Conference on Distributed Computing Systems (ICDCS)*, 2008.
15. D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.
16. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
17. F. Pianese and D. Perino. Resource and locality awareness in an incentive-based P2P live streaming system. In *Proc. of the Workshop on Peer-to-peer Streaming and IP-TV*, 2007.
18. Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang. Using layered video to provide incentives in p2p live streaming. In *Proc. of Workshop on Peer-to-Peer Streaming and IP-TV*, 2007.
19. G. Tan and S. A. Jarvis. A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast. In *Proc. of 14th International Workshop on Quality of Service (IWQoS)*, 2006.
20. Y. Chu, J. Chuang, and H. Zhang. A case for taxation in peer-to-peer streaming broadcast. In *Proc. of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, 2004.