

EVOLUTIONARY GAMES FOR COOPERATIVE P2P VIDEO STREAMING

Yan Chen, Beibei Wang, W. Sabrina Lin, Yongle Wu and K. J. Ray Liu

Dept. ECE, University of Maryland, College Park.
E-mail: {yan, bebewang, wylin, wuy1, and kjrlu}@umd.edu.

ABSTRACT

The wide-spread use of P2P video streaming systems have introduced a large number of unnecessary traverse links leading to substantial network inefficiency. To address this problem and achieve better streaming performance, we propose to enable cooperation among *group peers*, which are geographically neighboring peers with large intra-group upload and download bandwidths. Considering the peers' selfish nature, we formulate the cooperative streaming problem as an evolutionary game and derive the evolutionarily stable strategy (ESS) for every peer. Moreover, we propose a simple and distributed learning algorithm for the peers to converge to the ESSs. Compared to the traditional non-cooperative P2P schemes, the proposed cooperative scheme achieves much better performance in terms of social welfare and probability of real-time streaming.

Index Terms— P2P, cooperative streaming, evolutionary, game theory, replicator dynamics, distributed learning.

1. INTRODUCTION

With the rapid development of signal processing and networking technologies, video-over-IP applications become more and more popular and have attracted millions of users over the Internet. One solution to video streaming over Internet is the Peer-to-Peer (P2P) service model, where a peer not only acts as a client to download data from the network, but also acts as a server to upload data for the other peers in the network. The upload bandwidth of the peers reduces the workload placed on the server dramatically, which makes large-scale video streaming possible.

While P2P video streaming systems have achieved promising results, they have several drawbacks. First, there is a large number of unnecessary traverse links within a provider's network. As observed in [1], each P2P bit on the Verizon network traverses 1000 miles and takes 5.5 metro-hops on average. Second, there is a huge number of cross Internet Service Provider (ISP) traffic. The studies in [2] showed that 50%-90% of the existing local pieces in active peers are downloaded externally. Third, most of the current P2P systems assume that all peers are willing to contribute their resources. However, this assumption may not be true since the P2P systems are self-organizing networks and the peers are selfish by nature [3].

In the literature, many approaches have been proposed to overcome these drawbacks. Karagiannis et al. [2] proposed to use locality-aware P2P schemes to reduce the unnecessary traverse links within and cross ISPs and thus reduce the download time. Xie et al. [1] proposed a P4P architecture that allows cooperative traffic control between applications and network providers. To stimulate selfish peers to contribute their resources, payment mechanism [4] and reputation schemes [5] are proposed. Game theoretical incentive mechanisms are also investigated in [3].

Most of the existing schemes treat every peer as an independent individual. However, in reality, every peer can have a large number of geographically neighboring peers with large intra-group upload and download bandwidths, e.g. the peers in the same lab, building, or campus. In this paper, we name those geographically neighboring peers with large intra-group upload and download bandwidths as *group peers*. To reduce the unnecessary traverse links and improve network efficiency, instead of considering each peer's strategy independently, we investigate possible cooperation among the *group peers* and answer the question of "how a group of selfish peers should cooperate with each other to achieve better streaming performance".

The rest of this paper is organized as follows. In Section 2, we describe the system model and the utility function. Then, we show in details how to select agents in Section 3. In Section 4, we propose a distributed learning algorithm for ESS. Finally, we show the simulation results in Section 5 and draw conclusions in Section 6.

2. THE SYSTEM MODEL AND UTILITY FUNCTION

2.1. System Model

As shown in Figure 1, there is a set of *group peers*¹ (three in this example) who want to view a real-time video streaming simultaneously. We assume that the upload and download bandwidth among the group is large and the peers in the same group will cooperate with each other and choose k representative peers, called **agents**, to download video streams from the agents in other groups. Then, the agents will distribute the video streams to the other peers within the group. In order to achieve good stream-

¹How to group the peers itself is an interesting problem. However, in this paper, we assume that the peers have already been grouped and mainly focus on how the *group peers* cooperate with each other to achieve better streaming performance.

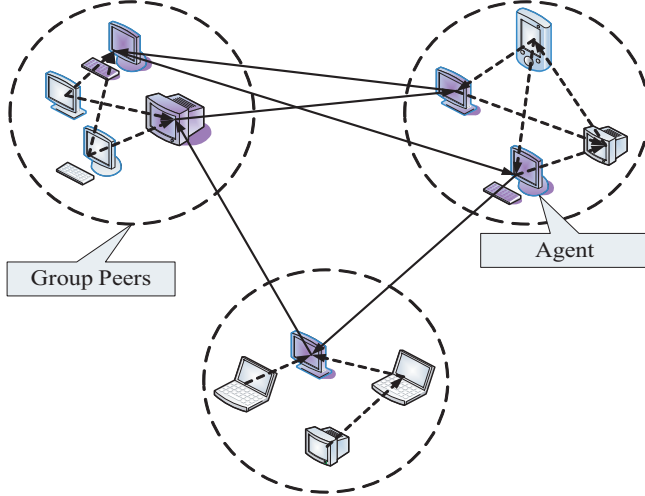


Fig. 1. In this example, there are three *group peers* who want to view a real-time video streaming simultaneously. The peers within a group cooperate with each other and choose k agents to download video streams from the agents in other groups, where the agents are marked in purple. Then, the agents distribute the video streams to the other peers within the group. Here, the solid line stands for the inter-group connection while the dashed line stands for the intra-group connection. In this paper, we address the following two questions: given a group of peers, how many agents should be chosen and which peers should be chosen as agents?

ing performance through cooperation, two questions need to be addressed: given a group of peers, how many agents should be chosen and which peers should be chosen as agents.

2.2. Utility Function

In a P2P network, a peer not only acts as a client to download video data from the other peers but also acts as a server to upload video data for the other peers. Therefore, while a peer can benefit from downloading data from the other peers, he/she also incurs a cost in uploading data for the other peers, where the cost can be resource spending on uploading data, e.g. bandwidth, buffer size.

Given the *group peers*, u_1, u_2, \dots, u_N , we assume that k agents are chosen to download multimedia data from the peers outside the group. Suppose that the download rates of the k agents are r_1, r_2, \dots, r_k , then the total download rate of the *group peers* is given by $y_k = \sum_{i=1}^k r_i$.

Since the agents randomly and independently select peers outside the group for downloading data, the download rate r_i 's are random variables. According to [6], the Cumulative Distribution Function (CDF) of a peer's download bandwidth can be modelled as a linear function, which means that the PDF of a peer's download bandwidth can be modelled as a uniform distribution, i.e., r_i 's are uniformly distributed.

Obviously, if the total download rate y_k is not smaller than the source rate r , then the group peers can have a real-time streaming, and all the group peers can obtain a certain gain G . Otherwise, there will be some delay, and in this case we assume the gain is

zero. Therefore, given the total download rate y_k and the source rate r , if peer u_i chooses to be an agent, then the utility function of u_i is given by

$$U_{A,i}(k) = P(y_k \geq r)G - C_i, \forall k \in [1, N], \quad (1)$$

where C_i is the cost of u_i when he/she serves as an agent, and $P(y_k \geq r)$ is the probability of achieving a real-time streaming.

Since the upload and download bandwidths within the group is large, the cost of uploading data to the other peers within the group can be negligible. In such a case, if peer u_i chooses not to be an agent, then there is no cost for u_i and the utility function becomes

$$U_{N,i}(k) = \begin{cases} P(y_k \geq r)G, & \text{if } k \in [1, N-1]; \\ 0, & \text{if } k = 0. \end{cases} \quad (2)$$

3. AGENTS SELECTION

In this section, we will discuss how to select agents within a homogeneous group where the cost of all peers serving as an agent is assumed to be the same.²

3.1. Evolutionary Cooperative Streaming Game

Since peers are selfish by nature, they are not as cooperative as a system designer/controller desires. To provide a robust equilibrium strategy for the selfish peers, we adopt the concept of Evolutionarily Stable Strategy (ESS) [7].

Since all peers are selfish, they will cheat if cheating can improve their payoffs, which means that all peers are uncertain of other peers' actions and utilities. In such a case, to improve their utilities, peers will try different strategies in every play and learn from the strategic interactions using the methodology of understanding-by-building. During the process, the percentage of peers using a certain pure strategy may change. Such a population evolution can be modelled by replicator dynamics. Specifically, let x_a stand for the probability of a peer using pure strategy $a \in \mathcal{A}$, where $\mathcal{A} = \{A, N\}$ is the set of pure strategies including being an agent (A) and not being an agent (N). By replicator dynamics, the evolution dynamics of x_a are given by the following differential equation

$$\dot{x}_a = \eta[\bar{U}(a, x_{-a}) - \bar{U}(x_a)]x_a, \quad (3)$$

where $\bar{U}(a, x_{-a})$ is the average payoff of the peers using pure strategy a , x_{-a} is the set of peers who use pure strategies other than a , $\bar{U}(x_a)$ is the average payoff of all peers, and η is a positive scale factor.

From (3), we can see that if adopting pure strategy a can lead to a higher payoff than the average level, the probability of a peer using a will grow and the growth rate \dot{x}_a/x_a is proportional to the difference between the average payoff of using strategy a (i.e., $\bar{U}(a, x_{-a})$) and the average payoff of all peers (i.e., $\bar{U}(x_a)$).

²Due to the page limitation, we will not show the analysis for the heterogeneous scenario in this paper. However, we would like to point out that the proposed distributed learning algorithm in Section 4 is applicable to both the homogeneous and heterogeneous scenarios.

3.2. Analysis of the Cooperative Streaming Game

According to (1) and (2), the average payoff of a peer if he/she choose to be an agent can be computed by

$$\bar{U}_A(x) = \sum_{i=0}^{N-1} \binom{N-1}{i} x^i (1-x)^{N-1-i} [P(y_{i+1} \geq r)G - C], \quad (4)$$

where x is the probability of a peer being an agent, and $\binom{N-1}{i} x^i (1-x)^{N-1-i}$ is the probability that there are i agents out of $N-1$ other peers.

Similarly, the average payoff of a peer if he/she chooses not to be an agent is given by

$$\bar{U}_N(x) = \sum_{i=1}^{N-1} \binom{N-1}{i} x^i (1-x)^{N-1-i} P(y_i \geq r)G. \quad (5)$$

According to (4) and (5), the average payoff of a peer is

$$\bar{U}(x) = x\bar{U}_A(x) + (1-x)\bar{U}_N(x). \quad (6)$$

Substituting (6) back to (3), we have

$$\dot{x} = \eta x(1-x)[\bar{U}_A(x) - \bar{U}_N(x)]. \quad (7)$$

At equilibrium x^* , no player will deviate from the optimal strategy, which means $\dot{x}^* = 0$, and we can obtain $x^* = 0, 1$, or the solutions to $\bar{U}_A(x) = \bar{U}_N(x)$. However, since $\dot{x}^* = 0$ is only the necessary condition for x^* to be ESS, we examine the sufficient condition for each ESS candidate and draw the following conclusions.

- $x^* = 0$ is an ESS only when $P(y_1 \geq r)G - C \leq 0$.
- $x^* = 1$ is an ESS only when $P(y_N \geq r)G - P(y_{N-1} \geq r)G \geq C$.
- Let x^* be the solution to $\bar{U}_A(x) = \bar{U}_N(x)$, and $x^* \in (0, 1)$. Then, x^* is an ESS.

4. A DISTRIBUTED LEARNING ALGORITHM FOR ESS

From the previous section, we can see that the ESS can be found by solving the replicator dynamics equations. However, solving the replicator dynamics equations require the exchange of private information and strategies adopted by other peers. In this section, we will present a distributed learning algorithm that can gradually converge to ESS without information exchange.

We first discretize the replicator dynamics equation as

$$x_i(t+1) = x_i(t) + \eta [\bar{U}_i(A, x_{-i}(t)) - \bar{U}_i(x_i(t))] x_i(t), \quad (8)$$

where t is the slot index and $x_i(t)$ is the probability of u_i being an agent during slot t . Here, we assume that each slot can be further divided into M subslots and each peer can choose to be an agent or not at the beginning of each subslot.

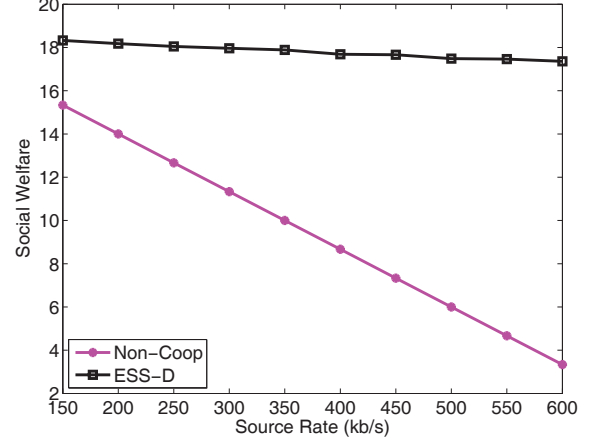


Fig. 2. The social welfare comparison.

From (8), we can see that in order to update $x_i(t+1)$, we need to first compute $\bar{U}_i(A, x_{-i}(t))$ and $\bar{U}_i(x_i(t))$. Let us define an indicator function $\mathbf{1}_i(t, k)$ as

$$\mathbf{1}_i(t, q) = \begin{cases} 1, & \text{if } u_i \text{ is an agent at subslot } q \text{ in slot } t, \\ 0, & \text{else,} \end{cases} \quad (9)$$

where q is the subslot index.

The immediate utility of u_i at subslot q in slot t can be computed by

$$U_i(t, q) = \begin{cases} G - C_i, & \text{if } u_i \text{ is an agent and } r^t \geq r, \\ -C_i, & \text{if } u_i \text{ is an agent and } r^t < r, \\ G, & \text{if } u_i \text{ is not an agent and } r^t \geq r, \\ 0, & \text{if } u_i \text{ is not an agent and } r^t < r, \end{cases} \quad (10)$$

where r^t is the total download rate of the agents and r is the source rate.

Then, $\bar{U}_i(A, x_{-i}(t))$ can be approximated using

$$\bar{U}_i(A, x_{-i}(t)) = \frac{\sum_{q=1}^M U_i(t, q) \mathbf{1}_i(t, q)}{\sum_{q=1}^M \mathbf{1}_i(t, q)}, \quad (11)$$

Similarly $\bar{U}_i(x_i(t))$ can be approximated as

$$\bar{U}_i(x_i(t)) = \frac{1}{M} \sum_{q=1}^M U_i(t, q). \quad (12)$$

Based on (8-12), u_i can gradually learn the ESS.

5. SIMULATION RESULTS

In all simulations, the parameters G , r^L , and r^U are set to be 1, 50, and 800, respectively. For convenience, in the rest of this paper,

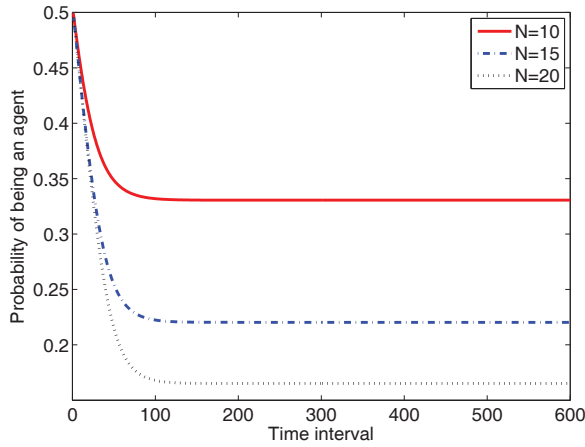


Fig. 3. Behavior dynamic of a homogeneous group of peers.

we denote the proposed ESS-based approach as **ESS-D**. We compare the proposed method with traditional P2P non-cooperation method, which is denoted as **Non-Coop**.

In the first simulation, we show the social welfare comparison, where we assume that there are 20 homogenous peers and the cost C is 0.1. As show in Fig. 2, **ESS-D** performs much better than the **Non-Coop** method since the social welfare performance of **Non-Coop** decreases linearly in terms of the source rate. With cooperation and adaptively selecting the proper number of agents, the proposed method can preserve a high social welfare performance even with a large source rate.

In the second simulation, we evaluate the convergence property of the **ESS-D**. In Fig. 3, we show the replicator dynamic of the cooperation streaming game with homogeneous peers with $r = 500$. We can see that starting from a high initial value, all peers gradually reduce their probabilities of being an agent since being a free-rider more often can bring a higher payoff. However, since too low a probability of being an agent increases the chance of having no peer be an agent, the probability of being an agent will finally converge to a certain value which is determined by the number of peers.

In the third simulation, we compare the probability of real-time streaming performance between **Non-Coop** and **ESS-D**. The simulation results are shown in Fig. 4. We can see that with cooperation, the probability of real-time streaming can be significantly improved especially at the high source rate region. We also find that at the high source rate region, the probability of real-time streaming increases as N increases. Note that to give more insight into the proposed algorithms, we assume that there is no buffering effect in this paper. However, the analysis and conclusion can be extended to the case when buffering effect is considered.

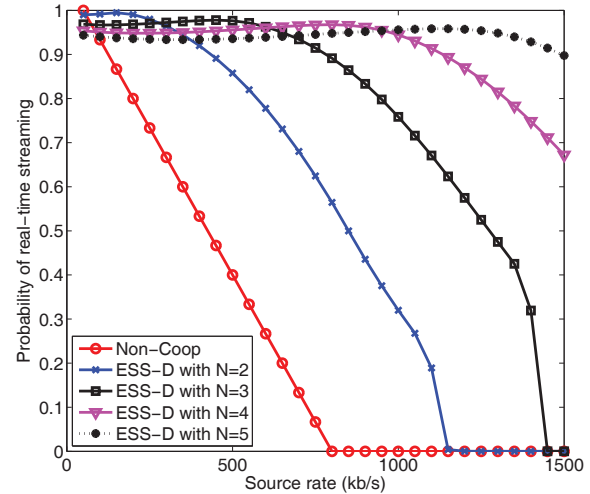


Fig. 4. The probability of real-time streaming comparison.

6. CONCLUSION

In this paper, we propose a cooperative streaming scheme to address the network inefficiency problem encountered by the traditional non-cooperative P2P schemes. We answer the question of “how a group of selfish peers with large intra-group upload and download bandwidths cooperate with each other to achieve better streaming performance” by formulating the problem as an evolutionary game and deriving the ESS for every peer. We further propose a distributed learning algorithm for each peer to converge to the ESS by learning from his/her own past payoff history. From the simulation results, we can see that compared with the traditional non-cooperative P2P schemes, the proposed algorithm achieves much better social welfare and higher probability of real-time streaming.

7. REFERENCES

- [1] Haiyong Xie, Y. Richard Yang, Arvind Krishnamurthy, Yanbin Liu, and Avi Siberschatz, “P4p: Provider portal for applications,” in *ACM SIGCOMM*, 2008, pp. 351–362.
- [2] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, “Should internet service providers fear peer-assisted content distribution?,” in *Proceedings of the Internet Measurement Conference*, 2005.
- [3] Min Xiao and Debao Xiao, “Understanding peer behavior and designing incentive mechanism in peer-to-peer networks: An analytical model based on game theory,” in *Proc. of International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, 2007.
- [4] V. Vishumurthy, S. Chandrakumar, and E.G. Sirer, “Karma: A secure economic framework for peer-to-peer resource sharing,” in *Proc. of the 2003 Workshop on Economics of Peer-to-Peer Systems*, 2003.
- [5] S. Marti and H. Garcia-Molina, “Limited reputation sharing in p2p systems,” in *Proc. of the 5th ACM conference on Electronic commerce*, 2004.
- [6] Cheng Huang, Jin Li, and Keith W. Ross, “Can internet video-on-demand be profitable?,” in *ACM SIGCOMM*, 2007, pp. 133–144.
- [7] B. Wang, K. J. R. Liu, and T. C. Clancy, “Evolutionary game framework for behavior dynamics in cooperative spectrum sensing,” in *Proc. IEEE Globecom*, 2008.