An evolutionary game approach to P2P video streaming

L. Chisci, F. Papi
Dpt. di Sistemi e Informatica
Università di Firenze, Via S. Marta 3
50139 Firenze, Italy.
Email: chisci@dsi.unifi.it
Email: papi_francesco@yahoo.it

T. Pecorella, R. Fantacci
Dpt. di Elettronica e Telecomunicazioni
Università di Firenze, Via S. Marta 3
50139 Firenze, Italy.
Email: tommaso.pecorella@unifi.it
Email: romano.fantacci@unifi.it

Abstract—This paper deals with the allocation of resources and the creation of distribution trees for P2P video streaming. Assuming that the video distribution is based on a Multiple Description Codec (MDC), the proposed approach performs jointly the two tasks, in a completely distributed client-based fashion, by means of an evolutionary (socially inspired) game played by all peers. The key idea of the evolutionary game is that each peer continuously measures its own utility (i.e., video quality) and, by periodical comparison with other randomly chosen peers, tries to mimic the behavior (i.e., resource allocation and video distributors) of peers with higher utility. Extensive simulation experiments by PeerSim have been carried out in order to assess the performance of the proposed technique. The obtained results have highlighted remarkable benefits in terms of scalability, fast adaptation to churning, high degree of cooperation among peers and self-organization of the distribution multtree network.

I. INTRODUCTION

P2P video streaming of live events is certainly a challenging and strategic real-world application of the P2P paradigm. Many experiments have been carried out over the last few years with promising results. A still open problem is the system scalability, primarily determined by the video distribution management and resource allocation algorithms.

The main issue of today’s P2P video distribution systems is the lack of functional separation between the DHT (Distributed Hash Table), the P2P network and the video distribution architecture. In fact, the video distribution network, unlike the general P2P network, should not be an overlay network completely independent of the underlying physical network.

As a matter of fact, the Internet structure is not flat (see [1], [2]). The network graph turns out to be a complex tree-based structure with multiple cross-links between tree areas, each area corresponding to an Autonomous System (AS). It must be pointed out that the access ASs usually operate in limited geographical areas, with a geographical coverage trough Points of Presence (PoP) at continental level at best, and more usually at national or regional level. Even when the Internet Service Provider (ISP) spans multiple countries, the local ASs might be connected through a core AS.

When a video P2P architecture has to be realized, it is of paramount importance to focus on what kind of events the P2P video system will broadcast. In this respect, video broadcasts can be roughly classified as:

- global coverage,
- local coverage,
- special events.

An example of global coverage event is the CEO keynote speech at some important conference. Potential users are a large number, but scattered around the globe so that there is no strong correlation between geographical position and event interest. On the other hand there is little or no interest for high quality video if not for a limited number of users. Data flows will have to span multiple ASs and, probably, there is also little interest to consider those events as a source of income from the ISPs.

The exact opposite is for the local coverage events. A classical example is the football match. The majority of the users will be located in a rather small geographical area, often at regional or national level. A small percentage of users can be located far away. In such cases, the geographic structure of the underlying network can and should be used to improve performance. Moreover, it is possible to have multiple ISPs offering access in the same geographical areas. The data flows between them should be minimized in order to increase both users’ performances and ISPs’ revenues. The same considerations applies to films and TV IP broadcasting.

Special events, like Olympic Games, can be considered as a mix between the global coverage and the local coverage kind, as some sub-events have special local interest due to athlete nationality or such.

The video distribution structure should, therefore, enforce locality. This means that there should be as many as possible exchanges between close P2P users. A problem is that the distance between two nodes cannot be measured in geographical terms, but should rather be interpreted as a network distance. It has been demonstrated that the distance in a P2P network is not a simple distance, due to the massive use of NAT (Network Address Translation) and active network elements. The XOR distance between IP numbers is not feasible either, as NAT can destroy any logical meaning of IP number; it could be used in a full IPv6 system but this is not yet a realistic assumption. The distance measurement issue, however, will
not be further addressed in this work and will be the subject of future investigation, although this issue needs clearly to be considered for a real implementation.

A further point to be addressed in video distribution systems is the scalability. In our assumptions, any source-based tree management system will lead sooner or later to scalability issues. In order to avoid this problem, a possible way is to leave all the decisions to the clients, i.e. the video receivers (remind that, in a P2P network, the client is also a source). Self-organization of the network should be enforced.

In this work, it is assumed that the video distribution is based on a Multiple Description Codec (MDC) [3]. As well known, MDC splits the video stream into multiple substreams allowing the final user to reconstruct the video from an arbitrary number of substreams with an increasing quality for each additional substream. The paper will focus on the two related tasks of allocation of peer resources and routing, i.e. the construction of multiple trees for distributing the video content (MDC substreams) among the peers. In order to accomplish such tasks in a fully distributed and client-based fashion, an evolutionary (socially inspired) game approach has been undertaken along the lines of [4], [5]. The key idea of the evolutionary game is that each peer continuously measures its own utility (i.e., video quality) and, by periodical comparison with other randomly selected peers, tries to mimic the behavior (i.e., resource allocation and video distributors) of peers experiencing higher utility. The effectiveness of this approach in terms of scalability, fast responsiveness to churning phenomena, self-organization and high degree of cooperation among peers will be demonstrated by extensive Peersim simulation.

The rest of the paper is organized as follows. Section II shows how a socially inspired (evolutionary) game theory approach can be a promising solution to enforce cooperation in P2P networks. Section III describes the proposed approach to joint resource allocation and routing for P2P video-streaming. Next, in section IV the performance of the proposed method is evaluated via computer simulations. Finally, V ends the paper with some concluding remarks and perspectives for future work.

II. EVOLUTIONARY GAME APPROACH TO PEER COOPERATION

The aim of this section is to explain how a socially inspired (evolutionary) game theory approach [4], [5] can be beneficial for enforcing cooperation in a P2P network. It is well known that in all P2P applications of both file-sharing and real-time types, a major problem is to develop mechanisms that discourage selfish behavior, e.g. peers downloading without uploading, and encourage altruistic behavior, e.g. peers offering available resources to the service. These situations are analogous to the so called commons tragedy, since all individuals would have benefits if all acted in an altruistic way but each has a natural inclination toward a selfish behavior. In P2P systems, this problem is evident in many applications, both of the “file sharing” type (e.g. the sharing of processing power or storage, the passing of messages and performing remote operations) and beyond “file sharing” such as in real-time video streaming of live events. Hence, techniques that can address the commons tragedy would appear to have wide application and to yield significant performance improvements within P2P systems.

Classical game theory generally assumes that each individual acts so as to maximize its utility, assuming that the other do the same and understanding completely the possible utility payoffs of their interactions. An alternative perspective from evolutionary game theory assumes that individuals will copy the behavior of others who obtain a higher utility.

In a P2P network, peers can selfishly increase their own performance in a greedy and adaptive way by changing their decisions (e.g., routing and resource allocation). They do this by copying nodes that appear to be performing better and by making randomized changes with low probability. A possible solution to avoid the commons tragedy is briefly described hereafter.

Assume that to measure the quality of service (e.g., video quality in the video-streaming application) an appropriate utility $U$ is defined. Hence, each peer $i$ engaged in the application must periodically compute its utility $U_i$ and compare it against another peer $j$, randomly selected from the population. If $U_i < U_j$, peer $i$ copies the decisions of peer $j$ (evolution) and, with low probability $p_{m}$, undergoes a random mutation of such decisions. Notice that the mutation mechanism is introduced, like in genetic algorithms, to provide an enrichment of research directions. To understand how this evolutionary game approach can help in enforcing cooperation in a P2P network, the so called “Prisoner’s Dilemma” will be discussed hereafter.

A. The Prisoner’s Dilemma

The Prisoner’s Dilemma is a minimal test that captures a range of possible application tasks in which nodes must establish cooperation and trust with their neighbors without central authority or external mechanisms that enforce it. The two-player, single-round, Prisoner’s Dilemma game captures a situation where a contradiction exists between self-benefit and the collective benefit. Two players interact by choosing either to cooperate ($C$) or to defect ($D$. For the game’s four possible outcomes, players receive specified payoffs. Both players receive a reward payoff ($R$) for mutual cooperation and a punishment payoff ($P$) for mutual defection. However, when individuals select different moves, the defector receives a temptation payoff ($T$) and the cooperator receives a sucker payoff ($S$).

It is assumed that neither player know in advance which move the other will make and that both players wish to maximize their own payoff. The dilemma is evident in the payoff ranking $T > R > P > S$ as well as in the constraint $2R > T + S$. Although both players would prefer $T$, the highest payoff, only one can attain it in a single game. No player wants $S$ because it is the lowest payoff. No matter what the other player does, a player always gets a higher score by defecting rather than cooperating. $D$ is,
therefore, the dominant strategy; an ideally rational player would, in fact, always choose \( D \). So, the dilemma is that if both players cooperate, they are jointly better off than if they both defect; however, selfish players have incentives to select mutual defection.

In [4], [5] it is shown how it is possible to establish cooperation in a P2P network where each peer application-level behavior involves nodes playing the Prisoner’s Dilemma with randomly selected neighbors.

Starting from this point, let us now specialize the P2P scenario by considering a P2P video streaming application: in particular, the objective is to increase nodes’ video quality using a socially inspired algorithm.

III. JOINT RESOURCE ALLOCATION AND ROUTING FOR P2P VIDEO-STREAMING

Let us now consider a video streaming application based on unstructured P2P overlay networks. Such networks have a population of nodes that maintain links to other nodes (called source neighbors) in order to provide video services.

Each node can be, at the same time, both a client and a server for the P2P video streaming application. It is characterized by a total bandwidth \( B_{tot} \) available for remote connections and a bandwidth \( B_{p2p} \) allocated to the P2P video-streaming application; the latter bandwidth is, in turn, divided into download bandwidth \( B_d \) and upload bandwidth \( B_u \) such that \( B_{p2p} = B_d + B_u \). In this specific context, the resource allocation policy of an individual peer consists of appropriately choosing the band parameters (which means changing the peer’s behavior at the application level)

\[
\alpha = \frac{B_u}{B_{p2p}}, \quad \beta = \frac{B_{p2p}}{B_{tot}}
\]

which specify the fraction of P2P bandwidth allocated to uploading/downloads and, respectively, the fraction of overall bandwidth reserved for the video-streaming P2P service. Clearly \( \alpha, \beta \in [0, 1] \). Notice that, depending on the type of application, other resource parameters concerning, e.g., memory or processing power could also be used. Further, each peer must also select a set of distributors (i.e. peers from which it receives the video MDC substreams).

It is assumed that, periodically, each peer can get some measure of a suitably defined video utility \( U \) possibly depending on various parameters including, for instance, the number of received substreams, the band parameters \( \alpha \) and \( \beta \), delay, jitter, etc. It is worth pointing out that the definition of utility is completely general and can also account for incentives to cooperation. It is also assumed that a peer can discover other peers randomly out of the entire population, compare its performance against such peers and copy their source links (e.g. distributors) and resource parameters.

A dynamic resource allocation algorithm for P2P video streaming based on the Commons Tragedy’s solution, is presented in detail hereafter.

Video Streaming Joint Resource Allocation and Routing Algorithm

Initialization - Let us consider a P2P network consisting of \( N \) peers, with random initial strategy \( (\alpha_i, \beta_i) \) for peer \( i = 1, 2, \ldots, N \). A list of source nodes \( L_i \) is initialized for each peer \( i \).

First Stage: Get the Video and Measure Utility - Each peer \( i \) performs, at each time step, the following actions:

- receives video flows from its current distributors;
- computes the video utility \( U_i \).

Second Stage: Evolution/Mutation - Once every \( T_e \) time steps, the generic peer \( i \) proceeds with the following actions:

- picks up a random peer \( h \); if \( U_h > U_i \):
  1) sets \( \alpha_i = \alpha_h \) and \( \beta_i = \beta_h \);
  2) for each peer \( k \in L_i \), removes \( k \) from \( L_i \) with high probability \( p_k \);
  3) adds to \( L_i \) the peers in \( L_h \), possibly removing the oldest peers in \( L_i \) if there is no available space;
  4) with very low probability \( p_m \), randomly selects the band parameters \( \alpha_i \) and \( \beta_i \) in \( [0, 1] \);
  5) for each peer \( k \in L_i \), removes \( k \) from \( L_i \) with probability \( 10p_m \) and replaces it with a new peer, randomly selected, with available resources;
- resets to zero the utility \( U_i \).

The step 3 of the algorithm is obviously conditioned to the resources availability of the nodes in the set \( L_h \). Moreover, the substream quality (delay, jitter) experienced by node \( i \) should be high enough in order to prevent an utility degradation. To this end, it is necessary to have a probe phase for each one of the nodes in the set \( L_h \). If the peer does not have any resource or the resulting quality would be too low, a new node must be chosen, much like for step 5 of the algorithm. Hence, it is necessary to be able to find the nodes with available resources.

The structure suitable for this task is very similar to the Spare Capacity Group (SCG) defined by SplitStream [6]. A complete decentralized and self-organizing version of the SCG has been defined, but its operational details are beyond the scope of this paper.

IV. PERFORMANCE EVALUATION

In order to demonstrate the potential benefits of a socially inspired allocation algorithm for P2P video streaming, simulation experiments have been carried out using the PeerSim simulator [7]. At the moment the considered simulation are performed in the cycle-driven mode so that time is measured in cycles. A local utility function defined as

\[
U_i = \frac{n_i}{n_{sb}}
\]

where \( n_{sb} \) is the total number of different substreams in the MDC description and \( n_i \) is the number of substreams received by peer \( i \); note that this utility is nothing but the fraction of substreams received by peer \( i \) and measures the perceived video quality. Clearly more complicated utility functions could be considered. The parameters adopted in the simulations take the values reported in table I.
It is pointed out that, in the performed simulations, a single source of the video stream is considered and that such a source has the same total bandwidth of all the peers, as it should be according to a pure P2P paradigm.

Figs. 1 and 2 plot the time behavior of the average peer utility for a total peer bandwidth $B_{tot} = 450$ kbps and, respectively, $B_{tot} = 300$ kbps as well as for different numbers $N$ of peers, starting from random and independent initializations of the band parameters $\alpha_i$ and $\beta_i$ for all peers $i = 1, 2, \ldots, N$. It is worth noting that in fig. 2 the total peer bandwidth is less than $2B_{st}$, i.e. twice the video stream bandwidth. This means that the overall P2P network will not be able to ensure a perfect video quality for all peers. Conversely, in fig. 1 the available bandwidth is greater than $2B_{st}$, thus allowing good performance even for suboptimal tree configurations. As a general remark, it can be observed that the curves are more or less similar for the different sizes of the P2P network thus demonstrating the high scalability of the proposed approach.

To confirm the results previously shown on a single realization, fig. 3 displays the results averaged over 50 independent Monte Carlo simulations obtained by randomly generating the initial band parameters $\alpha_i$ and $\beta_i$ for $N = 2000$ peers and a peer bandwidth $B_{tot} = 300$ kbps. Notice the good convergence toward a high average utility and the high scalability of the proposed algorithm; actually, the higher is the network’s size the greater is the quality achievable in steady-state.

In order to test the behavior of the proposed technique in presence of churning, the following experiment has been carried out: every 300 cycles $N_c$ out of $N = 2000$ leaving peers are replaced by as many randomly initialized joining peers. The results, reported in fig. 4 for various values of $N_c$, exhibit a fast recovery especially when the average quality is already high.

Fig. 5 shows how the behavior of the allocation-routing evolutionary algorithm is affected by the choice of the evolution period $T_e$, i.e. the time between two consecutive evolution steps. The plots highlight the tradeoff between convergence speed (faster for lower $T_e$) and quality (slightly better and less oscillatory for higher $T_e$); clearly a higher $T_e$ implies a reduced computational burden and, most importantly, a reduced signaling throughout the P2P network. Motivated by the above considerations, an adaptive version of the algorithm has been implemented in which the evolution time is online selected by each peer on the basis of its current utility. Fig. 6 compares the two versions with fixed and, respectively, adaptive $T_e$; it can be seen that the adaptation of $T_e$ provides indeed remarkable improvements.

Fig. 7 shows the percentage of the network traffic due to the proposed protocol over the normal signaling traffic of the P2P network with the adaptive algorithm. Due to the simulation environment, the numbers are the count of the exchanged packets in the network. The packet size is not taken into account, nor is the real video traffic bandwidth. This is a rough index of the network load generated by the algorithm. It is worth noting that the network overhead is strictly related to the video quality dynamics. When there is a substantial uniformity in the network, the signaling drops considerably.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>evolution period</td>
<td>$T_e$</td>
<td>3</td>
</tr>
<tr>
<td>evolution probability</td>
<td>$p_e$</td>
<td>0.8</td>
</tr>
<tr>
<td>mutation probability</td>
<td>$p_m$</td>
<td>0.0005</td>
</tr>
<tr>
<td>stream bandwidth</td>
<td>$B_{st}$</td>
<td>200 kbps</td>
</tr>
<tr>
<td>substream number</td>
<td>$n_{sb}$</td>
<td>10</td>
</tr>
</tbody>
</table>

**Fig. 1.** Average peer quality, $B_{tot} = 450$

**Fig. 2.** Average peer quality, $B_{tot} = 300$

**Fig. 3.** Average peer quality over 50 Monte Carlo runs
thus confirming the adaptive algorithm benefits.

V. CONCLUSIONS

The paper has addressed the dynamic allocation of resources and construction of the distribution multiple trees for P2P video-streaming based on the Multiple Description Codec (MDC). A novel approach based on an evolutionary (socially inspired) game played by all peers has been proposed in order to jointly accomplish the two tasks of resource allocation and routing in a fully distributed, self-organizing and client-based fashion. The resulting video-streaming system, thoroughly tested by Peersim simulations, has exhibited promising features in terms of scalability, fast adaptation to churning phenomena and high degree of cooperation among peers, provided that appropriate design parameters of the evolutionary game are suitably tuned. Motivated by the successful results, future work will address a theoretical analysis on the properties of the proposed techniques also to better understand the influence on such properties of the choice of the parameters of the evolution-mutation strategy. A further objective will be to extend the approach to other P2P applications.

ACKNOWLEDGEMENTS

This paper has been partially funded by the Italian PRIN project “PROFILES (PeeR-to-peer beyOnD FILE Sharing)”.

The authors would like to thank the whole team of PROFILES for valuable comments and suggestions. They also would like to thank the students Andrea Biancalani and Felix Basile Nguifor Oumbe for the development and running of simulation software.

REFERENCES