Rewarding Techniques in Peer-to-peer Video Streaming Systems with Tree and Forest Topology

P. Giacomazzi, and A. Poli
Department of Electronics and Information
Politecnico di Milano, Milan, Italy
giacomaz@elet.polimi.it, poli@elet.polimi.it

Abstract—**Peer-to-peer networks are an increasingly popular solution for the distribution of media content to a large number of users, with limited investments for network infrastructures. The distribution of a real time video stream imposes strict performance requirements such as small playback delays and few frame losses. However, performances are greatly affected by the upload bandwidth of the access networks of peers. In this paper we propose a set of rewarding techniques able to cleverly exploit the use of peers' upload bandwidth, and provide a sensitivity analysis through which we determine the conditions under which the rewarding techniques are beneficial. The study, carried out through simulation, considers a general peer-to-peer video streaming reference model with tree/forest topology. From our point of view, the proposed rewarding technique is simple enough to be used as a benchmark for the evaluation of more sophisticated approaches.**

Index Terms—**rewarding, peer-to-peer, video streaming, tree, performance analysis**

I. INTRODUCTION

PEER-TO-PEER video streaming systems are used to distribute live video content among large sets of users. The distributor is not required to have large network infrastructures, since the distribution of content relies mainly on users' resources. Users, referred to as peers, receive the video stream from other peers and forward it to one or multiple peers. This multicast-like paradigm can be achieved by creating an overlay network over which the content is exchanged. This overlay network could have various topologies, such as a tree, a forest – i.e. multiple trees – or a mesh. In the literature, examples of systems using a single tree topology are Narada [1], Scattercast [2], and Zig-zag [3]; systems with a forest distribution topology are VidTorrent [4], Coopnet [5]–[6], Nice [7], Overcast [8], and SplitStream [9]; systems with a mesh approach are CoolStreaming [10], GridMedia [11] and PPLive [12]. In peer-to-peer video streaming systems, the upload capacity is a critical resource, since it can greatly affect the system performances. To cope with this issue, rewarding techniques can be implemented in order to cleverly exploit peers' upload bandwidth. Most of the existing incentive mechanisms in peer-to-peer video streaming systems are inspired to the *tit-for-tat* policy used in BitTorrent [13]. When a tit-for-tat strategy is used, a peer rewards more to peers who contribute more to it. As a direct consequence, these rewarding techniques suits well on peer-to-peer video streaming systems with unstructured distribution schemes, referred to as mesh approaches, where peers can mutually exchange the media content. For example, PULSE [14] is an unstructured mesh-based system that constantly optimizes data connections among peers using a feedback-driven peer selection strategy based on pairwise incentives. In [15] a tit-for-tat strategy is used so that peers contributing more upload bandwidth receive more stream descriptions and consequently a better video quality. A tit-for-tat strategy is also used in [16], in addition to a mechanism able to dynamically replace poorly performing neighboring peers. In [17] the optimization of the existing capacity is achieved by means of virtual payments from the receivers to the senders. Many rewarding techniques are derived by implementing strategies computed by using game theory results; in [18], for example, strategic games are formulated to model the local exchanges of media content between neighboring peers.

9781-4244-3941-6/09/$25.00 ©2009 IEEE

In tree-based overlay networks, a mutual cooperation among peers is not straightforward, since when two peers in a tree are connected, one acts as a sender, and one acts as a receiver. As a consequence, a local *tit-for-tat* policy could not be beneficial, and rewards must be given on a system-wide basis. For example, [19] and [20] propose payment-based mechanisms where peers compete with each other for good parents (data suppliers) and earns points by forwarding data to others, while in [21] peers trade their upload bandwidth in order to receive a given incoming stream quality. Reference [22] proposes a peer selection protocol, where each peer selects appropriate sets of parents and children depending on their outgoing bandwidth, by means of a cooperative game.

Most of existing studies focus on the analysis of full systems *as-is*, without investigating the impact on performances of changes in their operational parameters. Moreover, the evaluation of the proposed rewarding techniques are performed by comparisons with systems where not any form of optimization is in place. In this paper, the simpler forms of rewarding in tree/forest-based peer-to-peer video streaming systems are presented and evaluated. These techniques entail the placement of peers with higher upload bandwidth at high levels in the tree topology (nearer to the source node); such reorganization allows to keep the trees short and to achieve better performances. From our point of view, such a basic rewarding technique must be used as a benchmark of more sophisticated approaches, involving virtual payments or game theory results. We provide a sensitivity analysis through which we determine the conditions under which the proposed rewarding techniques are beneficial, taking into account the dynamicity of the system, i.e. the existence of peers joining and leaving the system, and the time required by the rewarding protocol to reorganize the overlay topology. We carry out our study by means of a fine-grained simulative modeling of the peer-to-peer video streaming system. To the best of our knowledge, our simulative model of the system is significantly more accurate than similar extant models, and its description is fully disclosed in the paper.

II. THE REFERENCE SYSTEM MODEL

In this paper, we focus on tree-based peer-to-peer video streaming systems. Our model is inspired to VidTorrent [4], a tree-based peer-to-peer video streaming system developed at the Massachusetts Institute of Technology. However, our model is more general and it accounts for peer-to-peer video streaming systems with the following properties: (1) a unique content distribution source is responsible for the provisioning of the video stream to the whole system, (2) the structure of the distribution is a tree or a forest, (3) at the application level, in the overlay peer-to-peer network, the content is organized into chunks of video frames referred to as *segments*, (4) a single *frame* can be split into a fixed number (≥1) of *sub-frames* of variable length, (5) users can join and leave the peer-to-peer system dynamically, even during the distribution of a video.

In the following sections the model of the reference system is explained in detail.

A. The Video Stream

The video stream provided by the source is a sequence of m ordered frames. We identify a single *frame* f_i by its frame number $i = 1, 2, \ldots, m$. For each frame we know the start time $f_i.start$ and the end time $f_i.end$ in seconds, identifying the time interval covered by the frame with respect to the entire video stream. Every frame is split into n *sub-frames*, where a sub-frame represents a part of the whole frame, such as, for example, a specific layer in a layered coding, or a single description in a Multiple Description Coding (MDC). A subframe sf_{ij} is identified by the frame number i and the sub-frame offset $j = 1, 2, \ldots, n$. When coping with single description/single layer codings, a frame comprises of one sub-frame ($n = 1$). For each sub-frame we know the length $sf_{ij}.length$ in bytes. Sub-frames can have either variable or fixed size.

B. Segments

At the application layer, chunks of k sub-frames are organized into segments. A segment s_i is assembled by grouping sub-frames having the same offset in consecutive frames (see Fig. 1). For example, when $n = 4$ and $k = 3$, the first segment of the video stream s_1 is made of the sub-frames sf_{11}, sf_{21}, and sf_{31}, while s_2 is made of the sub-frames sf_{12}, sf_{22}, and sf_{32}.

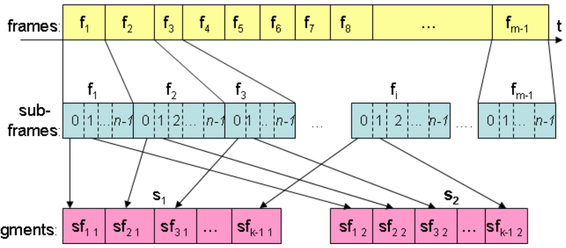

Figure 1. Frames, sub-frames and segments.

C. Source and Trees

The video content is distributed among all peers through a set of q independent trees. The source provides the video stream and is placed at the root of each tree. It sequentially sends segments to its children at the rate determined by frame start and end times. The source has a limited amount of bandwidth Sup, measured in bit/s.

The number of trees q is a multiple of the number of sub-frames per frame n, such that only the segments composed by the sub-frames with the same j-th sub-frame offset are forwarded in the same tree. A variable number d of trees is allowed to transport segments with the same sub-frame offset (*tree diversity* property). The total amount of trees is thus $q = d \cdot n$. Every segment is sent through the appropriate tree, alternating among the d available trees.

D. Trees and Peers

A client in the peer-to-peer video streaming system is called *peer*. A peer, identified by p_i, in order to receive the video content, must be a node of the trees carrying the content. A peer is not required to be part of all trees. For example, it could be a node of the d trees transporting the segments made up of sub-frames with the same single sub-frame offset. With a Multiple Description Coding this would translate into the reception of a single description, causing the display of a degraded version of the video stream.

All peers, for each tree they are part of, receive segments from their parents and then send them to their children. A peer can be placed in different positions in different trees, and different trees can have a different topology.

E. Peers' Operations

Each peer is responsible for the distribution of the media. It operates both at the overlay (application) and the underlay (network) levels. The access bandwidth of peer p_i, expressed in bit/s, is referred to as $p_i.Cup$ and $p_i.Cdown$, representing the upload and the download capacity of the access network, respectively. The peer is provided with a transmission buffer (in the upload direction) and a reception buffer (in the download direction).

Each peer performs the following actions (see Fig. 2):

1) The download queue, where the packets from parent peers are received, is emptied at a rate determined by $p_i.Cdown$.
2) All sub-frames received from the download queue are stored in an overlay buffer, named playout buffer, that reassembles the stream, according to frame numbers and sub-frame offsets.
3) As soon as all the sub-frames forming a segment are received, the segment is divided into packets and sent multiple times to all the children peers, through the appropriate trees. These packets are transmitted through the upload link.
4) The frames stored in the playout buffer are sequentially extracted by the client's player at the rate determined by the video stream.

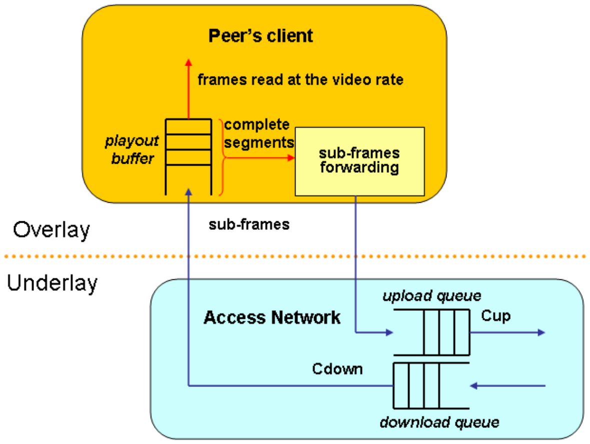

Figure 2. Peer's client and access network.

F. Playout Buffer

The playout buffer is responsible for the re-assembly of the video stream (segments are carried by multiple packets in the underlay network). The playout buffer has a finite length, $PBlength$, measured in segments. When a sub-frame is received, it is placed in the correct position in the playout buffer. When a complete segment is reassembled in the playout buffer, it is sent to the children, as described in the previous section. When all the frames of a segment have been read by the player, a position in the playout buffer is freed.

A peer starts playing the video stream as soon as a playback threshold $PBTh$, measured in seconds, is reached. The reaching of the threshold is computed independently for every sub-frame offset. The playback starts as soon as, for at least one sub-frame offset (corresponding to a layer or a description in a layered or in a multiple description coding), the threshold has been exceeded.

G. Standard Join

When a new peer wants to receive the video stream, it must become part of one or multiple distribution trees. In order to play the video stream, every peer must join at least the d trees transporting the segments composed of sub-frames with the same sub-frame offset. In the following, the *standard join* procedure is described.

The *standard join* procedure connects the new peers at the highest levels in the trees, without any optimization based on peers' access network capacity. In details, for peer p_i (see the example of Fig. 3):

1) Depending on the free download bandwidth (equal to $p_i.Cdown$ if the peer is not already receiving any other sub-stream), the maximum number of sub-frames per frame to be received is computed.
2) The sub-frame offsets are chosen randomly.
3) For every chosen sub-frame offset, the peer selects a parent in all the d trees for that offset (a parent can be either another peer or the source). For each tree, the parent is randomly chosen among the peers at the highest level in the tree (nearer to the source) with a sufficient amount of free upload bandwidth.
4) After a time interval t_{join}, measured in seconds, the new node starts receiving the sub-streams from its new parents. This interval models the time required for the identification and the selection of a parent.

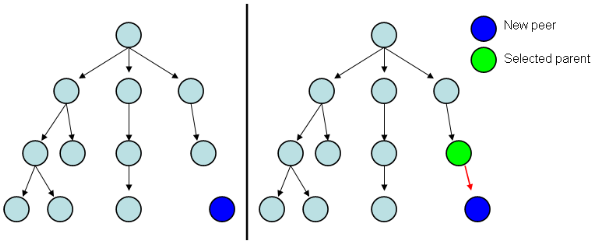

Figure 3. Standard join of a new peer.

H. Standard Leave

A peer can leave the system unpredictably and without notification. Whenever this happens, its children – and, iteratively, all their grandchildren in the same tree – cease to receive the sub-stream. In the following, the *standard leave* procedure is described.

When a peer leaves the system, the orphan peers start the join procedure for each tree they have been disconnected from, and start receiving the sub-stream from their new parents after a time interval t_{rejoin_leave}. This time interval can represent, for example, the time required by a keep-alive failure detection mechanism for the identification of the leave of a parent. The join procedure for an orphan peer is identical to the procedure followed by a new peer, as described above. Only direct children of the dead peer try to rejoin the trees in new positions, while all the isolated trees move together with their ancestors.

III. REWARDING TECHNIQUES

The rewarding techniques analyzed in this paper aim at moving the peers with higher upload bandwidth to high levels in the tree topology (nearer to the source node); such a reorganization allows to keep the trees short, since the nodes with higher outdegree – i.e. peers with more upload bandwidth and the option to accept a larger number of children – are placed in the first levels. Peers rewarded with positions nearer to the source node can benefit higher stability of the video stream and shorter time elapsing from the instant in which the source provides the content to the instant in which the peer receives it.

The reorganization of the system topology is performed when a new peer starts the join procedure or when a peer leaves the system. In this section we present the optimized join and leave procedures used to reward peers with high upload bandwidth.

A. Join with Base Optimization

When the *base optimization* of the join procedure is enabled, new peers entering the system are placed at the highest position in each tree such that no peers with lower upload bandwidth are at higher levels. The new peer replaces an already existing peer with lower upload capacity. The existing peer is disconnected from the initial position and it starts a new join procedure. We refer to this rewarding technique as *base optimization*. In details, for peer p_i:

1) Depending on the free download bandwidth (equal to $p_i.Cdown$ if the peer is not already receiving any other sub-stream), the maximum number of sub-frames per frame to be received is computed.
2) The sub-frame offsets are chosen randomly.
3) For every chosen sub-frame offset, in all the d trees for that offset, a parent is randomly chosen among those at the highest level in the tree having a sufficient amount of free upload bandwidth or at least one child with an upload capacity lower than the upload capacity $p_i.Cup$ of the current peer (a parent can be either another peer or the source).
4) If, for each tree, the chosen parent peer has not a sufficient amount of free upload bandwidth, its child with the lowest upload capacity is disconnected from the parent (if more than one peer have the same lowest bandwidth, the peer is randomly chosen among these).
5) After a time interval t_{join}, the new node starts receiving the sub-streams from its new parents. This interval models the time required for the identification and the selection of a parent.
6) All the disconnected peers replaced by the new peer because of the rewarding optimization starts a new join procedure (join with *base optimization*) for the tree they no more belong to. The nodes start receiving the sub-stream from their new parents after a time interval t_{rejoin_rew}. This interval models the time required for the identification and the selection of a new parent, when the rewarding procedure disconnects a peer.

B. Join with Advanced Optimization

The *advanced optimization* is similar to the *base optimization*, except for the behavior towards the peer disconnected from a tree and replaced by the joining peer. The *advanced optimization* requires that not only the disconnected peer starts a join procedure, but that the join procedure is also started by its direct children. Otherwise, the new positions assigned to the children of the disconnected peer would be at least one level deeper with respect to their initial position, since they would move together with their parent (which is supposed to rejoin the tree in a position at least one level deeper with respect to its initial position). The use of the *advanced optimization* allows these peers to search for a position in the tree at a depth level equal to their initial depth. The disconnected peer's children are consequently disconnected from the tree and, in the same way as their parent, start a new join procedure (join with *advanced optimization*) and start receiving the sub-stream from their new parents after a time interval t_{rejoin_rew}. This interval models the time required for the identification and the selection of a new parent.

C. Optimized Leave

The *optimized leave*, in addition to the activities prescribed by the *standard leave* procedure (see Section II.H), identifies an already existing peer for replacing the peer that has left the system. The leaving peer, at the depth level l in the tree, is replaced by considering all the peers connected to the same tree at the depth level $l+1$. Among all the peers at the depth level $l+1$ in the considered tree, the one with the maximum upload capacity is chosen (if more than one peer have the same highest bandwidth, the peer is randomly chosen among these), disconnected from its parent, and connected to the parent of the leaving peer. Then, the peer starts receiving the sub-stream from its new parent after a time interval t_{rejoin_rew}. This optimization avoids the occupation, by new joining peers, of the free positions left by the leaving peers, when peers with higher upload capacity are already in the system.

IV. PERFORMANCE ANALYSIS

In this paper, we analyze a peer-to-peer video streaming system where peers dynamically join and leave. The system, in the steady state, has a number N of simultaneously active peers. The time spent by a client in the system is exponentially distributed with average duration equal to $1/\mu$ s. Joins are independent Poisson events with a total average rate of joins equal to $\Lambda\ s^{-1}$, such that $N = \Lambda/\mu$.

A. Analysis Parameters and Indexes

In this paper we study the impact of three critical system parameters on the performance of the peer-to-peer video streaming system described in Sections II and III, when different versions of the join procedures – *standard*, with *base optimization*, or with *advanced optimization* – and of the leave procedure – *standard* or *optimized* – are used.

The selected performance parameters are:

1) *average number of peers*, N, that are concurrently in the peer-to-peer video streaming system;
2) *rewarding rejoin time*, t_{rejoin_rew}, required by a peer to rejoin a tree when it is disconnected from its current parent and reassigned to a new parent by the rewarding techniques implemented by the optimized join procedures and the optimized leave procedure;
3) *average permanence time of peers* in the system, $\frac{1}{\mu}$, measured from peer's first join to its leave.

The first parameter represents the number of simultaneously active peers, and is related to the actual popularity of video stream for the users. The *rewarding rejoin time* strictly depends on the coordination protocol implemented by the particular peer-to-peer video streaming system, in order to put into action the reorganization of the system topology when a peer is disconnected from its initial position and needs to be reassigned to a new parent. The third parameter represents the average time spent by a peer in the system, and mainly depends on users' behavior.

Other system parameters may be equally important. In this work we have concentrated our attention on system features with a significant impact on performance, not covered in depth by extant literature, and critical for the evaluation of the advantages that can be gained by implementing a rewarding technique in a peer-to-peer video streaming system with tree topology.

We measure system performance by means of the following indexes:

1) *playback delay*, defined as the time elapsing from the instant in which the source provides the content to the instant in which a client reads it from the peers' playout buffer;
2) *received frames and sub-frames ratios* for each peer, measured by considering the presence or absence of sub-frames in the playout buffer at their playback time.

Additional information is also registered, such as the number of total joins and leaves, and the topological characteristics of the distribution trees.

B. Simulation Parameters

The system is analyzed through a simulation tool expressly implemented for this purpose. We have carried out an extensive simulation campaign by considering a peer-to-peer video streaming system fed with a real video trace of a soccer match with a duration of 36 minutes, coded with a Multiple Description Coding (MDC) with 9 descriptions per frame ($n = 9$). The average rate of the video stream is 943.213 kbps, and every frame has a fixed duration of 33.3 ms. Each segment comprises $k = 20$ sub-frames. A single description is sent through one tree ($d = 1$), such that a total number of $q = 9$ trees/sub-streams are used. We have selected the source bandwidth Sup in such a way that it can provide up to 45 sub-streams (i.e. up to 5 complete streams) simultaneously. The playout buffer length has been set equal to 133.3 s ($PBlength = 1800$), and we have used a playback threshold $PBTh$ of 3.33 s. The join time t_{join} has been considered equal to 500 ms, and the rejoin time t_{rejoin_leave} equal to 100 s. When not otherwise explicitly notified, the average number N of simultaneously active peers has been considered 250, the *rewarding rejoin time*, t_{rejoin_rew}, has been set to 500 ms, and the *average permanence time of peers* has been considered equal to 15 minutes.

The upload and download access bandwidth ($p_i.Cup$ and $p_i.Cdown$) for joining peers, have been set according to the following probability distribution:

1) 50% – $Cdown = 7$ Mbps , $Cup = 1$ Mbps ;
2) 30% – $Cdown = 20$ Mbps , $Cup = 1$ Mbps ;
3) 10% – $Cdown = 8$ Mbps , $Cup = 1$ Mbps ;
4) 10% – $Cdown = 10$ Mbps , $Cup = 10$ Mbps .

The capacity of the transmission/reception buffers in the access network is infinite, so that frame losses are not caused by buffer overflows.

In order to evaluate the benefits that can be achieved by the utilization of rewarding techniques of peers, simulations have been carried out in 6 different scenarios, considering the enabling of the following versions of the join and leave procedures:

1) *standard* join, *standard* leave (**S,S** scenario);
2) *base optimization* join, *standard* leave (**BO,S** scenario);
3) *advanced optimization* join, *standard* leave (**AO,S** scenario);
4) *standard* join, *optimized* leave (**S,O** scenario);
5) *base optimization* join, *optimized* leave (**BO,O** scenario);
6) *advanced optimization* join, *optimized* leave (**AO,O** scenario).

C. Results

In this section, the performance indexes presented in Section IV.A are used to compare the behavior of the system by using the different rewarding techniques presented in Section III, as the average number of peers, rewarding rejoin time, and average peer duration vary. The reported numerical values are referred to averages on all the peers observed in the system, and have been obtained with a grand average over a set of independent simulations using different seeds for random number generation. The number of simulations for each point is variable and it has been chosen in such a way that the 95%-confidence intervals are smaller than 10% of the average values.

1) Average Number of Peers

The use of the rewarding techniques presented in Section III allows gaining a remarkable increase in the quality of the video stream received by the users.

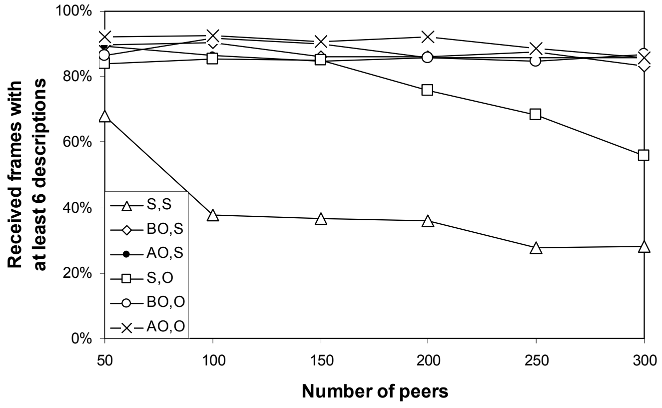

Figure 4. Ratio of received frames with at least 6 descriptions, with varying average number of peers.

In Fig. 4, we plot the average percentage of frames that have been received with at least 6 sub-frames (over a maximum of 9 descriptions) versus the average number of peers. When not any rewarding technique is used (S,S scenario) the quality of the video stream decreases significantly when the number of users increases. With a number of 50 peers, the use of the standard join and standard leave (S,S scenario) allows to reach a 68% of high quality frames, while with the availability of at least one of the presented rewarding techniques the percentage exceeds 84%. As the number of peers in the system increases, in the standard scenario (S,S) the quality of the video stream degrades, with a percentage of 28% of frames with at least 6 descriptions when 300 peers are in the system. Conversely, the use of one of the rewarding techniques for the join procedure allows keeping the system scalable, i.e. the quality of the video stream is not affected by an increase in the number of users. The use of the rewarding technique for the leave procedure allows to increase the quality of the video stream if compared to the non-optimized solution, but is not sufficient to assure scalability by itself. The graph also shows that the *join with advanced optimization* (AO,S and AO,O scenarios) does not offer significant increases in the performances with respect to the *join with standard optimization* (SO,S scenario); moreover, the use of the optimized leave, in addition to the optimization of the join (BO,O and AO,O scenarios), does not offer significant advantages.

In our simulations, we have found that the average playback delay is not significantly affected by the number of peers in the system. With respect to the scenario when not any optimization is enabled (S,S), the average playback delay decreases from 4.7 s to 3.6 s when the optimization of the join – both in the *base* and in the *advanced* versions – is used. The use of the *optimized leave* only (S,O scenario), allows reaching an average delay of 4.3 s. The use of the optimization of both the join and the leave (SO,O or AO,O scenarios) does not allow to improve the results, since the resulting average delay is steadily 3.6 s.

The observed improvement, offered by the rewarding techniques, allows building trees where the nodes with higher outdegree (i.e. a larger number of possible children) are placed in the higher levels (nearer to the source). This allows creating shorter trees; for example, with an average number of 300 peers in our scenarios, the maximum depth of the trees is about 10 on average when no optimization is used, and 4 when the *join with advanced optimization* and the *optimized leave* are concurrently used.

2) Rewarding Rejoin Time

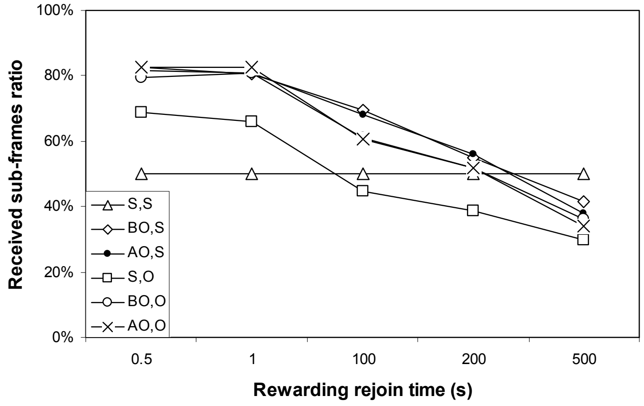

Figure 5. Ratio of received sub-frames, with varying rewarding rejoin time.

Fig. 5 shows the average percentage of sub-frames received by peers versus the rewarding rejoin time. The rewarding rejoin time is the time required by a peer to rejoin a tree when it is disconnected from its current parent and reassigned to a new parent because of the rewarding techniques implemented by the optimized join procedures and the optimized leave procedure. The non-optimized scenario (S,S) is clearly not affected by this variable, since not any disconnection is put into action. Results show that the use of a rewarding technique is beneficial only if the time required by a peer in order to identify a new parent when it is disconnected does not exceed a maximum threshold. In our scenarios, this value is about 200 s when one of the optimized versions of the join (BO,S – AO,S – BO,O – AO,O scenarios) is used. This threshold is critical, since above this level the frame losses because of the time spent in a disconnected status are not compensated by the rewarding technique. The coordination protocol implementing the rewarding technique must be designed in such a way to quickly reassign the disconnected peers to new parents, in order to be beneficial for the system's performance. Results also show that playback delay is not affected by variations of the rewarding rejoin time.

3) *Average Permanence Time of Peers*

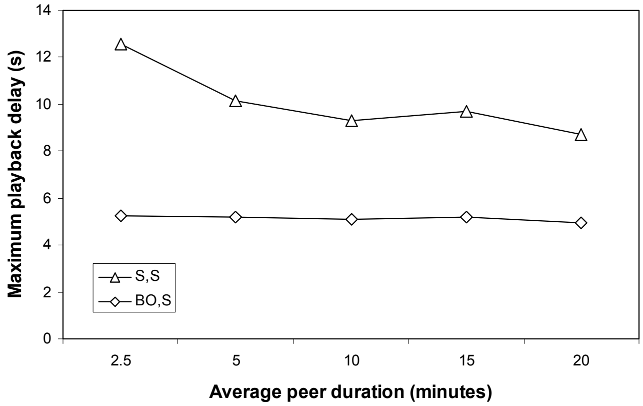

Figure 6. Maximum playback delay, with varying average permanence time of peers.

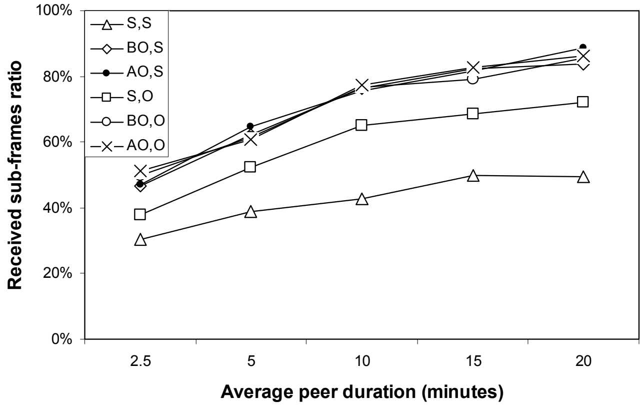

Figure 7. Ratio of received sub-frames, with varying average permanence time of peers.

The average playback delay does not vary significantly as a function of the average permanence time of peers. However, as shown in Fig. 6, in the non-optimized scenario (S,S) the maximum playback delay noticeably increases when peers stay in the system for a short time; the system can be very unfair, as in some cases a small number of peers can experiment a playback delay much higher than the average (the average playback delay ranges from 4 to 5 s in the considered interval, for all the 6 scenario variations). This behavior is greatly reduced by using an optimized version of the join procedure; as shown in the graph, the maximum delay is kept low even when peers frequently join and leave the system.

The quality of the received stream is alike influenced by the average permanence time of peers, as shown in Fig. 7. When the system is less stable, i.e. peers frequently join and leave, frame losses increase. This behavior is not blocked by any of the proposed rewarding techniques that, anyway, allow significant increases in the overall performances versus the non-optimized case.

V. CONCLUSIONS

We have carried out through simulation a detailed analysis of a set of rewarding techniques in a peer-to-peer video streaming system with a forest topology. Our simulation model is based on the VidTorrent system, but it is significantly general and it can be applied to a large class of peer-to-peer video streaming systems, based on a tree or forest distribution structure. The simulation model is very detailed and it accounts for many important operations and features of the real system.

In our analysis we have studied the impact of the use of rewarding techniques aiming at moving the peers with higher upload bandwidth to high levels in the tree topology (nearer to the source node). The proposed rewarding techniques are implemented by modifying the behavior of the procedure followed in the system for the join of a new peer, and the behavior of the procedure followed when a peer leaves the system.

Results show that the use of the proposed rewarding techniques allows gaining a remarkable increase in the quality of the video stream received by the users. Among all the proposed options, we have discovered that the performances that can be achieved by the use of the *base* version of the *join optimization* are not improved by the use of more complex techniques. When the *base optimization* of the join procedure is enabled, new peers entering the system are placed at the highest position in each tree such that no peers with lower upload bandwidth are at higher levels; the new peer replaces already existing peers with lower upload capacity, and the existing peers are disconnected from the initial positions and are required to start a new join procedure.

We have studied how the performances are affected by the *average number of peers*, the *average permanence time of peers*, and the *rewarding rejoin time*, i.e. the time required by a peer to rejoin a tree when it is disconnected from its current parent because of the rewarding procedure.

We have discovered that the use of rewarding techniques allows keeping the system scalable, i.e. the quality of the video stream is not affected by an increase in the number of users. Conversely, when no optimization is used, the quality of the video stream degrades as the number of peers in the system increases.

By studying the effects of the rewarding rejoin time, we have found that the use of a rewarding technique is beneficial only if the time required by a peer, in order to identify a new parent when it is disconnected, does not exceed a maximum threshold. This threshold (around 200 s in our scenarios) is critical, since above this level, the frame losses because of the time spent in a disconnected status are not compensated by the rewarding technique. The coordination protocol implementing the rewarding technique must be designed in such a way to quickly reassign the disconnected peers to new parents, in order to be beneficial for the system's performance.

As far as playback delay is concerned, in a non-optimized scenario the systems exhibits a sharply unfair behavior when peers stay in the system for a short time, as in some cases a small number of peers can experience a playback delay much higher than the average. For real-time events this feature is critical. This use of rewarding techniques greatly reduces this problem, since the maximum delay is kept low even when peers frequently join and leave the system.

In this work we have used a simulation framework for the analysis of peer-to-peer video streaming systems with tree or forest topology, for evaluating the effects on performances of rewarding techniques based on the actual transmission capacity of peers. The upload capacity is a critical resource since it can greatly affect the system topology. We have proposed a set of rewarding techniques able to cleverly exploit the use of peers' upload bandwidth, and studied the conditions under which these mechanisms are beneficial.

Previous literature proposes particularly sophisticated approaches, involving virtual payments or game theory results. However, results are compared to systems where not any form of optimization is in place. In our work, a basic form of rewarding has been presented and accurately evaluated. From our point of view, such a basic rewarding technique must be used as a benchmark of more sophisticated approaches.

Our current research aims at devising and assessing the performance of peer-to-peer video streaming systems with different distribution schemes, and the evaluation of the overall impact of the introduction of nodes with higher stability.

REFERENCES

[1] Y.H. Chu, S.G. Rao, and H. Zhang, "A Case for End System Multicast," *Proc. ACM Sigmetrics Int'l. Conf. Meas. Modeling of Comp. Sys.*, 2000.

[2] Y.D. Chawathe, "Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service", *Un. of California*, Berkeley, 2000.

[3] D.A. Tran, and K.A. Hua, T.T. Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming," *22nd Annual Joint Conf. of the INFOCOM*, 2003.

[4] I. Mirkin, "Reliable Real-time Stream Distribution Using an Internet Multicast Overlay," http://viral.media.mit.edu/index.php?page=vidtorrent, 2006.

[5] V. N. Padmanabhan, and K. Sripanidkulchai, "The Case for Cooperative Networking," *Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems*, 2002.

[6] V.N. Padmanabhan, H.J. Wang, P.A. Chou, and K. Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking," *ACM NOSSDAV*, 2002.

[7] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," *Proc. 2002 Conf. Apps., Technologies, Architectures, and Protocols for Comp. Commun.*, 2002.

[8] J. Jannotti, D.K. Gifford, K.L. Johnson, M.F. Kaashoek, and J.W. O'Toole, "Overcast: Reliable multicasting with an overlay network," *Proc. OSDI*, 2000.

[9] M. Castro, P. Druschel, A.M. Kermarrec, A. Nandi, *et al.*, "SplitStream: high-bandwidth multicast in cooperative environments," *Proc. of the 19th ACM symposium on Operating systems principles*, 2003.

[10] X. Zhang, J. Liu, B. Li, and T. S. P. Yum, "CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming," *IEEE Infocom*, 2005.

[11] L. Zhao, J.G. Luo, M. Zhang, W. J. Fu, J. Luo, Y.F. Zhang, *et al.*, "Gridmedia: A Practical Peer-to-Peer Based Live Video Streaming System," *IEEE 7th Workshop on Multimedia Signal Processing*, 2005.

[12] PPLive, http://www.pplive.com/.

[13] B. Cohen, "Incentives build robustness in BitTorrent," *1st Workshop on Economics of Peer-to-Peer Systems*, 2003.

[14] F. Pianese, D. Perino, J. Keller, and E. W. Biersack, "PULSE: An adaptive, incentive-based, unstructured P2P live streaming system," *IEEE Transactions on Multimedia*, vol. 9, pp. 1645-1660, 2007.

[15] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang, "P2P video live streaming with MDC: Providing incentives for redistribution," in *IEEE International Conference onMultimedia and Expo, ICME 2007*, July 02,2007 - July 05, 2007, pp. 48-51.

[16] P. Shah, J. Rasheed, and J. Pâris, "Performance study of unstructured P2P overlay streaming systems," in *17th International Conference on Computer Communications and Networks*, ICCCN 2008, August 03,2008 - August 07, 2008, pp. 608-613.

[17] E. Mykoniati, R. Landa, S. Spirou, R. G. Clegg, L. Latif, D. Griffin, and M. Rio, "Scalable peer-to-peer streaming for live entertainment content," *IEEE Communications Magazine*, vol. 46, pp. 40-46, 2008.

[18] M. K. H. Yeung, and Y. Kwok, "Game-theoretic scalable peer-to-peer media streaming," in *IPDPS 2008 - 22nd IEEE International Parallel and Distributed Processing Symposium*, April 14,2008 - April 18, 2008, pp. IEEE Computer Society Technical Committee on Parallel Processing.

[19] G. Tan, and S. A. Jarvis, "A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast," in *2006 14th IEEE International Workshop on Quality of Service, IWQoS 2006*, June 19,2006 - June 21, 2006, pp. 41-50.

[20] A. Habib, and J. Chuang, "Service differentiated peer selection: An incentive mechanism for peer-to-peer media streaming," *IEEE Transactions on Multimedia*, vol. 8, pp. 610-621, 2006.

[21] D. A. G. Manzato, and N. L. S. Da Fonseca, "An incentive mechanism for peer-to-peer networks with live streaming," in *IEEE GLOBECOM 2006 - 2006 Global Telecommunications Conference*, November 27,2006 - December 01, 2007.

[22] M. K. H. Yeung, and Y. Kwok, "Game theoretic peer selection for resilient peer-to-peer media streaming systems," in *28th International Conference on Distributed Computing Systems, ICDCS 2008*, July 17,2008 - July 20, 2008, pp. 817-824.