Resource and Locality Awareness in an Incentive-based P2P Live Streaming System

Fabio Pianese
fabio.pianese@orange-ftgroup.com

Diego Perino
diego.perino@orange-ftgroup.com

France Telecom - Division R&D
F-92794, Issy-les-Moulineaux, France

ABSTRACT

One of the main challenges in P2P live streaming is the efficient allocation of the available resources. This paper presents an experimental evaluation of the effects of a local pairwise incentive mechanism applied to an unstructured mesh-based architecture. We focus on the relationship between resource availability in the system and the average quality of its data distribution paths, both in terms of bandwidth efficiency and awareness to network locality. We show via large scale testbed experiments based on the PULSE live streaming system that the introduction of appropriate incentive-based policies as the main peer selection mechanism can lead to a global content distribution mesh which has properties similar to tree-based structured systems.

1. INTRODUCTION

In the last few years, peer-to-peer networks for live streaming have attracted a lot of interest. It has been a common opinion for some time to date [1][2] that the current Internet infrastructure can support live streaming via an adequate application-layer protocol, without the need for a native multicast transport-layer infrastructure. Moreover, the main advantage peer-to-peer approaches can offer over the standard, multiple-unicast centralized solutions, is the ability to exploit the bandwidth capacity that is provided by the users. This feature alleviates the need for high-bandwidth access links at the streaming source, and theoretically allows the user population to scale to arbitrary sizes since both system capacity and bandwidth requirements increase at the same pace.

Today, while the speed of broadband commercial Internet access is largely sufficient for the downlink requirements of many current streaming applications, most typical uplink capacities are still not sufficient to support even a small number of unicast fixed-rate streams [3]. However, in practice, the bandwidth provided by each user can be lower than the stream rate, and a widespread lack of user contribution can disrupt the normal system operation. Also, users can join or leave the system at any time, and rapid changes in membership require a prompt reaction by the system to insure a steady streaming service to the remaining population.

While low uplink rates do not significantly affect the operation of bulk file distribution systems, e.g. BitTorrent [4], and can just increase the startup delay of video-on-demand (VoD) streaming systems, live streaming applications have more stringent capacity requirements, as they need to receive data at an average rate exactly equal to the media rate for the playback to be stable. Buffering helps to overcome brief capacity shortages in exchange for an increased initial playback delay, but, eventually, buffers get empty and playback stops. Unlike VoD, where the playback can be stopped and resumed without problems, the playback delay of a live stream monotonically increases without explicit bounds, until the data that have to be played are too outdated to be available any longer. The total available serving capacity of a live streaming system where all users receive a full service must thus be, in average, equal or greater than the capacity required to replicate the stream to all the nodes.

The main challenge for a live streaming system is exploiting the available bandwidth in a way that leads to timely data delivery. Thus, the available node capacity has to be carefully allocated: it is generally desirable that the path that data have to follow in the system be as short as possible, and that data exchanges be made preferably between nodes that are placed 'close by' in the underlying transport network, to reduce the redundant use of long-haul network links while also helping improve the delay of the data paths.

We designed and implemented PULSE [5] with the intent of building a peer-to-peer system capable to support live media streaming under realistic conditions. This goal guided us in our choice of *an unstructured mesh-based design where the interactions between the nodes are independently determined only using incentive-based local selection criteria iterated on a short timescale.* The resulting system is therefore dynamic, since the future state of the system is determined by the current data exchanges among the nodes: the progress of data reception is then at the same time an index of streaming performance (i.e. the average delay needed for a piece of data to get to the node, counted from the time it was generated at the source) and the main feedback variable upon which nodes can choose their exchange partners (i.e. a measure of the interest in the other peer for a possible future interaction). The presence of a delay-based feedback loop that is not restricted by structural constraints enables the PULSE mesh to adapt to the current global availability of resources in the system.

P2P-TV'07, August 31, 2007, Kyoto, Japan.

The main purpose of this paper is to explore the effect of the simple incentive-based local peer selection policy used in PULSE on the global organization of its data-exchange mesh. Our analysis is based on experimental traces obtained by running the PULSE prototype node on few hundreds PlanetLab nodes, and is integrated with references to results from a testbed platform [10] when more controlled conditions are required. Because of space constraints, in this paper we only deal with scenarios where peer membership is constant and the system is operating at steady state: we argue that, as the PULSE mesh lacks a fixed structure and constantly evolves over time, the use of scenarios where more instability is introduced by node churn does not add new insights relevant to the main subject of this work.

We will focus our analysis on the twin themes of resource and locality awareness in a peer-to-peer live streaming system: a) we study the correlation between a node's upload bandwidth contribution and its reception performance, b) we measure the average hop-count properties of the data paths observed during stable system operation, and c) we test the global effects of introducing a simple latency bias parameter on the local peer selection policy.

The contributions we make in this paper are the following:

1) We empirically confirm the widespread belief that the use of incentives as a peer selection policy can lead to the creation of clusters of nodes with similar performance. Incentives, as used in the PULSE system, produce a dynamic mesh with nodes ordered by bandwidth which can meet the constraints of live streaming.

2) We show one fundamental advantage of unstructured mesh-based systems: flexibility. The lack of structural constraints in PULSE allows us to perform global optimizations in a distributed, recursive way, by way of small modifications in the peer selection criteria.

This paper is organized as follows. In Section 2, we present a short background on the use of incentive techniques in the context of peer-to-peer live streaming systems. Then, in Section 3, we briefly introduce the definitions and details required to support the subsequent evaluation of PULSE: our analysis, which is based on data traces from instrumented nodes running on testbed environments, is divided into two parts: a study of the adaptiveness of the system to resource availability, in Section 4, and of its awareness to network locality, in Section 5. Finally, we conclude in Section 6.

2. INCENTIVES AND RELATED WORK

The use of incentive mechanisms to encourage contribution in peer-to-peer systems was introduced in relatively recent times. The first remarkable example of incentive-based peer-to-peer system, BitTorrent [4], has clearly shown the potential of this class of mechanisms to support the operation of distributed systems. BitTorrent is a mesh-based bulk content distribution system where the data-exchange partners of a node are decided using a local *altruistic tit-for-tat* (TFT) policy. This policy amounts to choosing a fixed number of nodes which contributed the most during the previous few seconds, plus one random node, as the designated targets of data exchange for the next few seconds.

Until 2004, the debate in the field of live media streaming had been revolving mainly around the evaluation of the properties of different structured approaches in cooperative environments under node churn. The achievements of BitTorrent had quickly a noticeable impact: it became soon evident [6] that encouraging cooperation and altruism between nodes were key aspects of a successful streaming architecture. At the same time, early examples of mesh-based data-driven systems began to make their appearance: DONET/Coolstreaming [7] first advanced the claim that an unstructured system could perform in a way comparable to structured single-tree based approaches. Moreover, the advantages offered by mesh-based systems were shown to be an increased resilience to churn and the possibility of recursively optimizing the overlay mesh over time without any noticeable playback disruption.

Incentives to share have recently been adapted to work with structured and unstructured streaming systems. The use of strict pairwise incentives on structured architectures has shown results that are somewhat disappointing [8]. We still believe that incentives can have a fundamental role in a P2P streaming context: however, rather than employing them to enforce a fair retribution policy, we argue that they can be exploited more effectively as a criterion to optimize the structure of the streaming overlay. The use of pairwise incentives as the base peer-selection mechanism is the main original aspect of PULSE [5], but also appears as an important component of other unstructured systems, such as Chunkyspread [2], whose stated goal is to support node populations with heterogeneous resources under node churn.

Today, practical peer-to-peer systems for distributing live media over the Internet are still in their infancy, as only few actual applications have been deployed on a scale sufficient to perform meaningful evaluation: Coolstreaming and PPLive seem to be the two more relevant examples. The algorithms of Coolstreaming are public, and a limited performance study has been undertaken by the authors [7]: the drawback of this system seems to be its lack of awareness to network locality. The source code of PPLive, on the other hand, is not available, so the details of the algorithms are unknown. However, it is currently unclear even whether PPLive has mechanisms in place to detect and adapt to the underlying network conditions, as all we know about this system was deduced from measurement studies [9].

3. PULSE AND ITS INCENTIVES

The main incentive mechanism used in PULSE closely follows the peer selection mechanism used in BitTorrent, with the changes required to support the strict timing constraints of live streaming. For practical reasons, the core incentive has been coupled with additional mechanisms that increase the robustness of the system to temporary resource shortages and that encourage an efficient altruistic contribution by resource-rich peers[1]. In this section, we will introduce the appropriate terminology to deal with PULSE and describe in detail how the incentive-based peer selection works.

3.1 Definitions

In PULSE, all nodes (or peers) are identical in role, except the source, that is the node distributing the stream for the first time. The source divides the original media

[1]As our analysis is principally focused on the effect of incentives on live streaming performance, we will describe in depth the main selection technique used in PULSE (originally called MISSING selection) while leaving out the details about all other mechanisms. Interested readers can refer to [5] for a description of the whole system, which lies out of the scope of this paper.

Parameter	Value	Description
W	32	Length of buffer sliding window (chunks)
TW	64	Total length of trading window (chunks)
LR_{max}	20%	FEC tolerance to chunk losses/window
T_D	250	Min lag to trigger buffer reset (chunks)
EPOCH	2	Time b/w subsequent peer selections (s)
N_{TFT}	4	Peers chosen w/ optimistic TFT incentive
CR	8	Rate of chunk generation @source (s^{-1})
SBR	256	FEC-encoded stream bit rate ($Kbit/s$)
R_{TO}	0.5	Timeout of chunk request messages (s)
R_{max}	2	Max outstanding requests to same peer

Table 1: PULSE Protocol Parameters

stream into a series of FEC-encoded pieces (called *chunks*) and sends them to the other peers with a constant rate. Peers must then exchange pieces in order to retrieve a complete series of chunks and recover the original media. Every peer has a buffer, where it stores the chunks it receives prior to playback. Table 1 lists the main buffer parameters.

The buffer of a PULSE node uses a mechanism based on a *sliding window* of W chunks to regulate the data reception process. The purpose of this mechanism is to allow chunk losses to happen when they can be compensated (using FEC, with a maximum allowed loss rate of LR_{max} that is chosen by the source) without any loss of quality, and to adapt the buffer progression to the actual state of chunk reception. In any moment, a peer is only interested in obtaining chunks from a limited contiguous region, named *Trading Window* (TW), which includes the sliding window. The TW region can then move forward to include more recent chunks from the stream only when the sliding window inside it contains enough data to allow the recovery of the media from its FEC coding. We will call the delay between the source's TW (i.e. where new chunks are introduced in the system) and a node's TW as *node lag*, and represent its running average value as T_B.

The main difference between live streaming and bulk data distribution (where all peers share an interest in a same set of data, i.e. the entire file) is the dependence of the interest for data on the current stream reception status of a peer, which can change over time. This difference has to be considered when peer selection is performed. The main incentive mechanism in PULSE adapts the optimistic TFT strategy from BitTorrent to support the strict timing constraints of live streaming.

3.2 The Incentive Mechanism

Live streaming requires timely reception of recent data. Thus, the T_B value of a remote node is a fundamental piece of information for any peer to determine *whether the remote node can provide data chunks* that are useful to progress, and *whether the remote node is interested* in the benefits of a pairing. Reciprocal interest in the stream data can be represented as the integral of the product between the areas of interest of the two nodes. As the TW length is a fixed, system-wide parameter, we can estimate the reciprocal interest I of two peers a and b as a function of the difference in their lag: $I(a, b) = TW - |T_B(a) - T_B(b)| = TW - \Delta T_B$.

Another factor that is also relevant to the good choice of exchange partners is the pairwise link latency $L(a, b)$. In a data-driven system, nodes have to actively negotiate before data transfers are performed, to avoid unnecessary chunk duplication. The latency first comes into play when a chunk request is propagated to a partner node, and adds once again for the partner's reply to be received. Also, an estimate of this value is easy to obtain at any peer using simple techniques. It is thus perfectly reasonable to treat latency as a factor that directly affects the amount reciprocal interest, as more latency causes a reduction in the range of data that can be requested by either peer to their exchange partner (if we suppose that both trading windows are moving forward at the same speed). We then define a *latency-biased interest function* $BI(a, b) = I(a, b) - C \cdot L(a, b)$, where the C parameter is called the *latency weight*.

At each EPOCH, the N_{TFT} data connections opened by all nodes are renegotiated. At every peer except the source, $N_{TFT} - 1$ connections are allocated to the nodes that have contributed the most during the previous EPOCH. For the remaining slot, all known nodes are ordered by decreasing latency-biased interest BI, and the highest-ranked is selected. At the source, which does not receive data from the other nodes and thus cannot rely on the pairwise incentive, peer selection is performed by randomly choosing partners among the nodes with the lowest T_B.

Classes \ Scenario	HH-LB	LH-LB
Very Rich (VR)	4%, 4*CR (2Mbps)	-
Rich (R)	20%, 2*CR (1Mbps)	20%, 2*CR (1Mbps)
Normal (N)	21%, CR (512Kbps)	80%, CR (512Kbps)
Poor (P)	55%, CR/2 (256Kbps)	-

Table 2: Upload Bandwidth Distribution Scenarios

4. RESOURCE AWARENESS

We performed experiments with our PULSE system to evaluate the effect of its incentive on the organization of its data-exchange mesh. Our first goal is to evaluate the correlations between bandwidth availability at the nodes and their behavior, in terms of their reception lag and their choice of exchange partners. For this reason, in all the following experiments, the latency weight is always $C = 0$.

4.1 Correlating Node Lag to Node Upload

We defined two *bandwidth scenarios* that approximate realistic worst-case operating conditions. In these scenarios, nodes belong to different *bandwidth classes*: the nodes from each class have had their upload contribution limited to a certain rate, while all download speeds are unrestricted. We run the emulations on our testbed, and then collect and aggregate the results, trying to establish correlations between node behavior and affiliation to a particular bandwidth class. For this experience, we used a population of 800 testbed nodes. The most important system parameters were set in a similar way as we did in [5] (see Table 1) to allow for an easy direct comparison with results from preliminary simulations. The scenarios are defined in Table 2:

HH-LB (High Heterogeneity, Low Bandwidth): this scenario corresponds to a very pessimistic bandwidth distribution: not only the upload capacities are heavily asymmetric, but more than half of the nodes can contribute no more than one half of the original stream bitrate. The percentages are loosely inspired by the results of the study by Sariou et al. on Gnutella peers [3] that showed an approximative power-

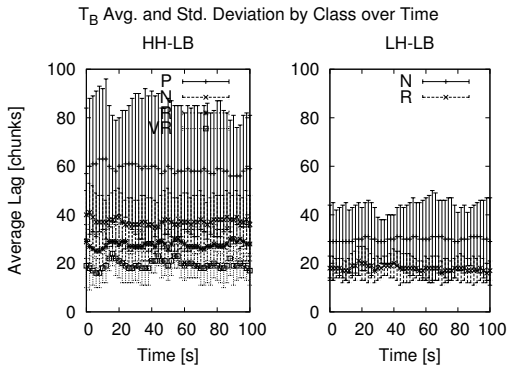

Figure 1: Average Class Lag over Time in HH-LB (l) and LH-LB (r)

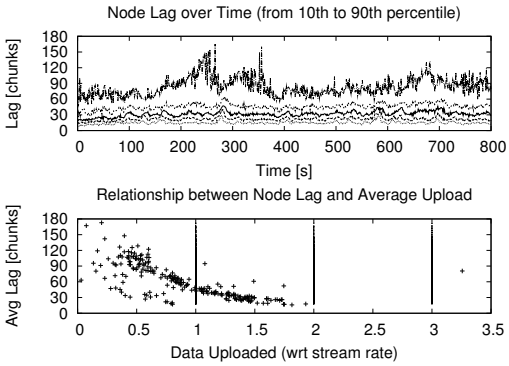

Figure 2: Results of an Uncontrolled PULSE Run on PlanetLab (200 nodes)

law distribution of upload capacities higher than 10 Kbps. The Resource Index of this configuration (as defined in [1]) is about 1.04, meaning that there is barely enough capacity to serve a complete stream to the whole node population.

LH-LB (Low Heterogeneity, Low Bandwidth): this scenario portrays a system with a Resource Index of about 1.20, where the bandwidth excess is evenly split among a minority of the population. The challenge in this scenario is given by the small difference of capacity between the two classes, and the relative scarcity of bandwidth.

All nodes join the session at roughly the same time (about 2 sec are required to launch the 800 peers in our setup), and they keep running indefinitely. Data are collected long enough for the system to reach and operate at steady state.

In Figure 1, we plot the per-class average over time of a node's T_B for a typical run, under HH-LB and LH-LB scenarios respectively. The most striking result conveyed by Figure 1 is the strong relationship between the available upload bandwidth of a class and the average lag of its members: peers with the highest bandwidth contribution reach in both cases a steady-state lag of about 20 chunks (that is, less than 3 sec) from the media source. On the other hand, the less a class contributes, the worse its average lag: the POOR class in HH-LB gets the highest average lag among the four, at nearly 60 chunks (slightly more than 7 sec). Visually, the plot for HH-LB is especially telling, as the four classes appear *sorted by resources and layered one after the other*, with a meaningful difference between the average lag performance of each class: also, we notice that *the lag difference becomes higher when a class' available upload capacity is smaller than the stream bandwidth.* The same remarks can be made about the LH-LB scenario, as the lag difference of the two classes is smaller but still evident (18 chunks or 2.2 sec for RICH, 30 chunks or about 4 sec for NORMAL).

Figure 1 also depicts the lag variance as vertical lines. Interestingly, the variance for all classes also quickly converges to a stable value, which is again related to the upload contribution: the variance of T_B becomes higher as the upload bandwidth available to each class decreases. This can be seen both in the HH-LB and in the LH-LB scenarios: the correlation between upload capacity and lag stability is evident, resulting in a much lower standard deviation of lag performance for the resourceful classes (4-10 chunks for VERY RICH and RICH, vs. 10-25 chunks for NORMAL and POOR in HH-LB; 4 chunks for RICH vs. 11 chunks for NORMAL in LH-LB). This suggests that the fact of *having more bandwidth not only reduces the average lag, but also tends to give nodes a more stable performance in the system.* The analysis of upload and download bandwidth utilization data (not shown) also confirms the system's stability: all classes contribute an average total bandwidth which is almost constant over time, and receive chunks at rates that are always sufficient to reconstruct the original media stream from its FEC encoding.

The two previous observations are important, as they represent a tangible benefit that can appeal to rational agents. It is indeed in the best interest of any node to provide at least as much bandwidth as it demands, as the typical consequences of doing so result in a better performance, especially in terms of stability, achieved by nodes that provide sufficient resources. Providing less is allowed, but higher lag and temporary disconnections should be expected, especially if the resources in the system become scarce.

4.2 Average Lag vs. Bandwidth Contribution

Next, we performed experiments on PlanetLab in order to confirm the behavior of the incentive-based selection under uncontrolled network conditions. In fact, PlanetLab offers its users little control on the node resources, not only in terms of an unknown available bandwidth, but also because of the high CPU load of machines. This problem makes it especially difficult to test a time-sensitive streaming application that requires low response times. For this reason, we just managed to obtain about 200 hosts with semi-acceptable CPU load conditions, while we had to lower the chunk rate (CR) to 4 chunks per second and the stream bitrate (SBR) to 128 Kbps. We did not limit the node bandwidth but chose to leave it naturally limited by the resources available at each host, since the high CPU load in most PlanetLab nodes would also slow down the execution of the software in unpredictable ways.

The results show that the use of an incentive-based selection allows PULSE to behave reasonably well even on this difficult environment, proving a high level of resilience and adaptiveness. Looking at Figure 2, it can be noticed that 90% of peers manage to regularly obtain a T_B lower than about 100 chunks (25 seconds), and that 50% present a node lag lower than 30 chunks (less than 10 seconds). The

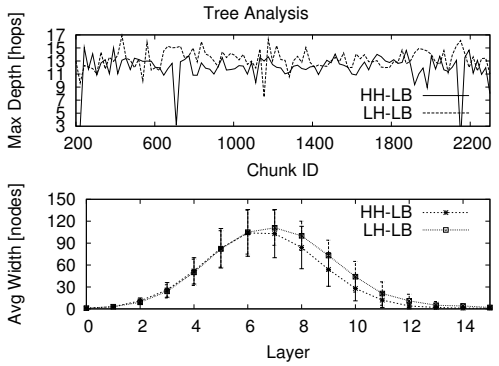

Figure 3: Analysis of Average Chunk Distribution Tree Properties

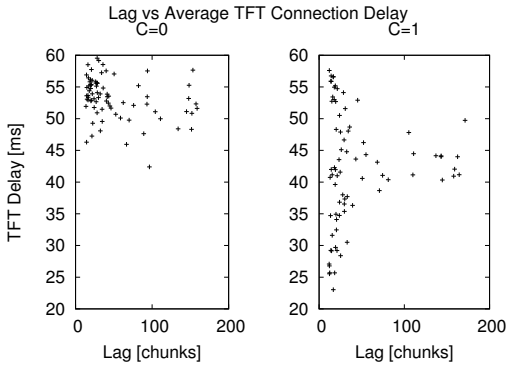

Figure 4: Effect of Latency Bias on Cumulative Connection Latency

Layer	1	2	3	4	5	6	7	8
% VR	3	20	21	15	7	2	1	0
% R	15	35	40	41	34	18	6	2
% N	19	21	20	20	24	27	18	8
% P	62	24	19	24	35	53	76	90

Table 3: Average Observed Composition of Distribution Tree Layers by Bandwidth Class (HH-LB)

T_B distribution is a consequence of the upload bandwidth distribution, as about 60% of peers offer less than the full stream rate while the other 40% upload at a rate lower than twice the stream rate. By correlating the total bandwidth contribution with the average lag of the nodes, we can obtain in Figure 2 a clear inverse relationship between the two variables: the more a peer uploads, the lower is its lag.

4.3 Bandwidth Classes and Data Paths

We have seen above that the position of the nodes in the incentive-generated data exchange mesh is related to their bandwidth contribution: we are now interested in analyzing what is the specific impact of this global node placement on the distribution process of individual data chunks. To this end, we will study the paths taken by data chunks as they are replicated by the nodes. As no duplicate chunks are allowed, the resulting directed distribution graphs for each chunk are free of cycles (i.e. single trees). The average properties of these trees (width, depth) can provide precious insights to complete our observations on node lag.

We show the analysis of the average properties of chunk distribution trees from our testbed emulations in Figure 3. We notice that the maximum tree depth[2] in hops for individual chunks is short and quite stable over time. In our system with 800 nodes, maximum tree depths are in average between 11 and 14 hops, for both bandwidth scenarios. Without any explicit structural guidance, the paths taken by the chunks were consistently good, even under a widespread bandwidth scarcity and while the data connections between nodes were being continuously renegotiated. More information can be gathered by the statistics on tree width: from Figure 3 we can appreciate the fact that the first few layers of the trees are in average very wide, and that the average percentage of nodes that find themselves placed in the last few layers of the trees is low (<20%). Also, the variance of layer width is high for the first few layers, with a standard deviation in the order of up to 30% of the average value.

Several intriguing details can be also seen in Figure 3, if we look more closely: we can notice how HH-LB trees are in average a little *shorter* than LH-LB trees, despite the fact that the Resource Index for the HH-LB scenario is lower than for LH-LB. Average tree widths are also very similar, especially in the first few layers, where HH-LB tops LH-LB by a small margin. These counter-intuitive observations can be explained if we take into account the effects of the incentive-based peer selection mechanism: as we remember from Section 1, the streaming source randomly selects its targets for data exchange among the nodes with lowest lag; we then observed node clustering by upload capacity, with the richest classes constituting a large fraction of the peers with low lag values. The net effect of these two combined mechanisms is that the chunk distribution trees from scenarios with high levels of heterogeneity can have very wide initial layers, due to the richest peers being on top. Wide initial layers are very important in the context of live media distribution, since they reduce considerably the maximum number of hops a chunk has to traverse to reach all the nodes. To verify our conjecture, we analyze in Table 3 the average placement of nodes that belong to each bandwidth class in the chunk distribution trees (the data we use are taken from a simulated HH-LB scenario). We can observe that there is indeed a preponderant presence of peers from the richest classes in the few first layers, especially between layer two and five, where roughly half of the peers have the necessary resources to replicate each chunk more than once and up to four times.

5. LATENCY AWARENESS

After examining the macroscopic effect of a loose TFT incentive on the global system organization, we now study the impact of a weighted latency bias on the system in terms of locality awareness. These experiments were performed

[2]Chunk distribution trees obviously only include nodes that receive a certain data chunk, thus the number of nodes in a tree can change on a chunk-by-chunk basis. However, the protocol mechanisms guarantee that connected nodes can never lose in percentage more than the the FEC rate (in our case, 20%), and bandwidth traces show that they actually lose much less.

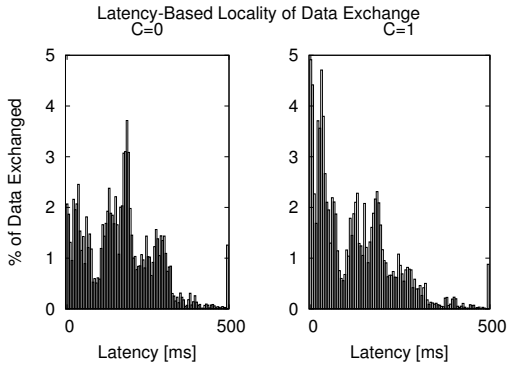

Figure 5: Effect of Latency Bias on Overall Data Exchange Locality

on Planetlab using a population of 100 peers without any artificial upload limitation. We observed the behavior of the PULSE system as the C latency weight parameter was set to 0 and 1. To define the locality metric, we measured the pairwise node latencies during our experiments using exponentially-spaced *ping* probes ($\lambda = 10s$).

Figure 4 shows the *cumulative latency* of the incentive-driven connections in function of the average lag of each peer. Cumulative latency is computed for each single node by adding together the latencies of the four connections that it established using the biased TFT incentive, and averaging this value over time. It is possible to notice how the introduction of the latency bias can sharply reduce the average TFT delay, especially for those peers whose lag is low. We can see that, when $C = 0$ (i.e with no latency bias), all peers show an average cumulative latency uniformly distributed between 45 ms and 60 ms, regardless of their average lag. With the addition of a latency bias $C = 1$, the minimum average cumulative latency goes down to 22 ms, while just few nodes maintain a cumulative latency of about 60 ms. Also, the average cumulative latency of all nodes becomes lower for non-zero values of C.

In Figure 5 we correlate the percentage of data being uploaded by each peer with the latencies of the connections that it is using, again averaged over the time. The histograms clearly show that locality of data exchange definitely increases if we add a latency bias: when $C = 0$, the data is sent to other peers in an almost uniform way (we remember that the latency distribution of the peers is not uniform). On the contrary, when $C = 1$, the amount that has to travel on shorter distances is much higher: the stream data are prevalently exchanged between peers with pairwise latencies lower than 125 ms. Finally, Table 4 shows the effects of latency awareness on the global performance of data reception in the system, in terms of percentile node lag. As we expected, with the latency bias peers achieve a slightly lower reception delay, thanks to the fact that chunks are sent more often to peers which are *closer* locality-wise. The extent of this reduction is quite small, however, as the skew in the node latency distribution is quite limited. We expect that, by introducing the latency bias in a scenario with larger difference between pairwise node latencies, the reception delay reduction would also be more significant.

Lag Percentile	10%	30%	50%	70%	90%
C=0	12.31	18.10	26.18	37.70	61.08
C=1	10.53	14.47	18.89	27.22	49.39

Table 4: Effect of Latency Bias on Average Node Lag (in chunks)

6. CONCLUSIONS

In this paper, we analyzed the effects of a simple pairwise incentive on the organization of an unstructured peer-to-peer system for live streaming using experiments. Our results confirm that incentive-based peer selection policies are able to support heterogeneous resource availability in a distributed system. We also showed that an appropriate use of incentives allows to build unstructured systems that are comparable to tree-based systems in terms of data distribution performance. The main benefit of an incentive-based system lies in its higher flexibility and adaptiveness to network conditions: here we focused our attention on resource and locality awareness only, but other desirable properties, such as high resilience to churn, can also be obtained with the same technique.

The analysis we performed in this paper was based on PULSE, a dynamical mesh-based approach to live streaming which relies on a feedback-driven incentive mechanism as its main peer selection strategy. We are currently working to the deployment of PULSE on the Internet, in order to gather data traces from user activity and to further explore the performance of this system in a production environment.

7. REFERENCES

[1] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, "The feasibility of supporting large-scale live streaming applications with dynamic application end-points", in *Proc. of ACM SIGCOMM 2004*

[2] V. Venkataraman, K. Yoshida, P. Francis, "Chunkyspread: Heterogeneous Unstructured End System Multicast", in *Proc. of IEEE ICNP'06*

[3] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems", in *Proc. of MMCN 2002*

[4] B. Cohen, "Incentives Build Robustness in BitTorrent", in *Proc. of P2PECON 2003*

[5] F. Pianese, J. Keller, and E. W. Biersack, "PULSE, a Flexible P2P Live Streaming System", in *Proc. of IEEE Global Internet Symposium 2006*

[6] Y. Chu, and H. Zhang, "Considering Altruism in Peer-to-Peer Internet Streaming Broadcast" in *Proc. of ACM NOSSDAV 2004*

[7] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: A Data-driven Overlay Network for Live Media Streaming", in *Proc. of IEEE INFOCOM 2005*

[8] Y.-W. Sung, M. Bishop, and S. Rao, "Enabling Contribution Awareness in an Overlay Broadcasting System", in *Proc. of ACM SIGCOMM 2006*

[9] X. Hei, C. Liang, J. Liang, Y. Liu, and K.W. Ross, "Insights into PPLive: A Measurement Study of a Large-Scale P2P IPTV System", in *IPTV 2006*

[10] Grid'5000 testbed platform - *http://www.grid5000.org*