Exploiting Heterogeneity in P2P Video Streaming

Ana Paula Couto da Silva, Emilio Leonardi, *Senior Member*, *IEEE*, Marco Mellia, *Senior Member*, *IEEE*, and Michela Meo

Abstract—In this paper, we investigate the impact of peer bandwidth heterogeneity on the performance of a mesh-based P2P system for live streaming. We show that bandwidth heterogeneity constitutes an important resource for P2P live streaming systems. Indeed, by effectively exploiting it, the overall performance of the system is significantly improved. This requires the adoption of smart schemes for both the overlay topology construction and chunk scheduling mechanisms that discriminate among peers based on their bandwidth.

Index Terms—P2P streaming systems, diffusion algorithms, bandwidth aware scheduling, fluid models.

1 Introduction

The transformation of PCs into multifunctional terminals that provide services, such as telephony, social networking, gaming, and e-commerce, is now turning our PCs into televisions. Live TV distribution over the Internet is already a successful business: systems such as PPLive [1], PPStream [2], SOPcast [3], TVAnts [4], just to mention a few, have thousands or even hundreds of thousands of users watching TV at the same time.

To cope with the need for high scalability and the impossibility of controlling resources and mechanisms of the Internet, most of these TV distribution systems are based on the P2P paradigm (P2P-TV systems). According to the P2P approach, the users, i.e., the *peers* of the system, contribute to the distribution of the video by delivering the content they have already received to other peers. To do so, peers are in contact with each other forming an *overlay* network. Two peers that can communicate are called *neighbors*. For swarm-based systems, a *source* divides the video stream into small *chunks* of data which are separately and independently distributed on the overlay. Basically, every chunk is delivered to the peers using a potentially different *distribution tree*.

The main design objectives for a P2P-TV system for live broadcasting are: delivering all the chunks to all the peers (so as to avoid video quality degradation) and reducing the delivery delay as much as possible. A number of reasons call for the reduction of the delivery delay. First, reduction of delay translates into shorter start-up times, both when the application starts and when the user switches channel. Second, live P2P-TV is expected to provide a video stream to all the interested users with (roughly) the same delay. The need for tight delay delivery is one of the main fundamental difference between these systems and video on demand systems. Finally, reducing delays means that all the peers are receiving the same chunks at roughly the same time, making redistribution of chunks easier under real-time constraint. Practically, the design of a P2P-TV system should address the following two issues: 1) how to create the overlay topology assigning the set of neighbors to every peer; and 2) how to distribute chunks among neighboring peers. The first issue concerns the *overlay topology design*; the second one concerns the *chunk and neighbor scheduling algorithms*.

In this paper, we contribute to the understanding and investigation of these issues by analyzing the fundamental role that peer upload bandwidth plays. The main finding is that, in order to optimize delay performance, the information about peer upload bandwidth must be carefully exploited by the chunk scheduling algorithm. Indeed, highest priority must be given to peers with the highest upload bandwidth, i.e., to those peers which can provide the largest contribution to the chunk distribution process. Moreover, peer upload bandwidth information must be carefully considered when constructing the overlay topology, i.e., largest bandwidth peers must be pushed closer to the source.

2 System Assumptions and Notation

We make a number of assumptions about the system behavior.

We assume that peers have infinite download bandwidth and finite upload bandwidth. The network has infinite capacity, i.e., the bottleneck is at the access network. These assumptions are common to most of the papers in this field.

We also assume that peers know the upload bandwidth of their neighbors. This requires that some measurement is performed; however, since peers are actively exchanging a lot of data and signaling information, setting up a measurement technique for the upload bandwidth does not add a significant overhead. In addition, as we show in the paper, the benefits of taking into account bandwidth are

- *A.P.C. da Silva is with the Computer Science Department, Instituto de Ciências Exatas (ICE), Universidade Federal de Juiz de Fora, Rua José Lourenço Kelmer, s/n—Campus Universitário, Bairro São Pedro—CEP—Juiz de Fora—Minas Gerais 36036-330, Brazil. E-mail: anapaula.silva@ufjf.edu.br.*
- *E. Leonardi, M. Mellia, and M. Meo are with the Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10129, Italy. Email: {Emilio.Leonardi, Marco.Mellia, Michela.Meo}@polito.it.*

so significant that even an inaccurate estimate of upload bandwidth is valuable. At last, we assume that the overlay maintenance mechanisms allow each peer to keep a perfectly updated knowledge about which chunks are needed by its neighbors.

Since the observed scheduling phenomena and the role of heterogeneity in the chunk distribution process are not directly related to the dynamic behavior of peers joining and leaving the system (peer churning), we neglect this aspect, as do all the related papers described in Section 3.

We consider a chunk-based system in which a single source encodes and seeds the content. Let $\mathcal{P}$ be the set of peers, with cardinality N. The overlay is represented by an undirected graph $G(V, E)$, where $(p, q) \in E$ if and only if $p \in \mathcal{P}$ and $q \in \mathcal{P}$ are neighbors to each other. The information to be transmitted is organized into small chunks each of fixed size of L bits. The source generates chunks at rate λ and sends them to its neighbors; the chunk diffusion in the overlay is then accomplished by the peers themselves, according to a scheduling policy. Each peer transmits at most one chunk at a time to one of its neighbors. The transmission speed of a chunk is determined only by the upload bandwidth of the sender. We denote by B_p the upload bandwidth of peer $p \in \mathcal{P}$.

A simple upper bound λ_{sup} to the video stream rate that can be successfully distributed is provided by bandwidth conservation law:

$$\lambda_{\text{sup}} = \min\left\{ B_s, \frac{B_s + \sum_{p \in \mathcal{P}} B_p}{N} \right\},$$

where B_s is the source upload bandwidth. We say that a P2P system achieves bandwidth efficiency $\rho \leq 1$, when it can successfully deliver a stream whose rate is $\lambda < \rho\lambda_{\text{sup}}$.

We focus on *push-based* diffusion systems where the transmission of a chunk between any two peers is initiated by the sender, which is a natural choice in systems that are constrained by peer upload bandwidth. Push-based schemes can be categorized into two classes depending on whether the destination peer or the chunk is selected first. For each peer p, let $C(p)$ be the collection of chunks that p has received; and let $\mathcal{N}(c, p)$ be the set of neighboring peers of p which are still missing chunk c. In this paper, we restrict the analysis to the following peer and chunk selection algorithms:

Random Peer, Latest Useful chunk (RP/LU). The destination peer q is selected first by uniformly choosing it from the set of neighboring peers. Once the destination is selected, the delivered chunk c is the most recent chunk in $C(p)$ which is not in $C(q)$.

Latest Useful chunk, Random Peer (LU/RP). The chunk c is selected first by choosing the most recently generated chunk in the collection $C(p)$ such that there exists a neighbor of p not having c, i.e., the set $\mathcal{N}(c, p)$ is not empty. Then, the destination peer q is randomly selected in $\mathcal{N}(c, p)$.

Latest Useful chunk, Weighted Priority peer (LU/WP). The sender peer p selects the most recent chunk in the collection $C(p)$ such that the set $\mathcal{N}(c, p)$ is not empty. The destination peer $q \in \mathcal{N}(c, p)$ is selected according to probability $p_q = \frac{w_q}{\sum_{r \in \mathcal{N}(c,p)} w_r}$ where $w_q = f(B_q)$ is a function of B_q, the upload bandwidth of q. The function that maps peer bandwidth B_p into weight w_p is assumed nondecreasing.

Latest Useful chunk, Strict-Priority Peer (LU/PP). The sender peer p selects the most recent chunk in the collection $C(p)$ such that the set $\mathcal{N}(c, p)$ is not empty. The destination peer is the highest weighted neighbor peer in $\mathcal{N}(c, p)$, where w_q is a function of B_q. The function that maps peer bandwidth B_p into weight w_p is assumed nondecreasing. Ties are randomly solved.

3 Previous Work and Paper Contribution

This paper focuses on the overlay topology design and chunk scheduling algorithm. Only few previous works have considered these issues. To the best of our knowledge, the problem of building an efficient overlay topology *in the context of unstructured P2P-TV systems* has been addressed only in [5], for a scenario in which the stream delivery delay is mainly due to the transport network latency. This is usually not the case in chunk-based architectures.

Several chunk scheduling algorithms have been proposed and analyzed in [6], [7], and [8]. In [6], the authors prove the rate optimality of the so-called *most deprived peer, random useful chunk* algorithm, i.e., a policy in which the peer chooses to distribute the chunk to the neighbor with the largest number of missing chunks, under the assumption that the overlay topology is a full mesh. In [7], delay optimality (for $N \to \infty$) of the *random peer, latest blind chunk* algorithm is proven assuming that all peers are characterized by the same upload bandwidth (latest blind refers to the fact that the latest chunk is selected regardless of whether or not the selected neighbor needs it), under the assumption that the overlay topology is a full mesh. It turns out, however, that the delay performance of the former is poor due to the random chunk selection, while the rate performance of the latter is rather poor due to the blind nature of peer/chunk selection. More recently, in [8], it has been shown that joint optimal rate and asymptotic delay performance can be achieved using a *Random peer, latest useful chunk* (i.e., RP/LU) scheduling algorithm; this result also applies only to the case of peers with the same upload bandwidth and under the assumption that the overlay topology is a full mesh. In addition, the authors show by simulation that RP/LU and LU/RP have similar good performance also for more general overlays.

The goal of this paper is to provide general guidelines for the choice of the chunk scheduling policies in the much more challenging scenario in which peers have heterogeneous upload bandwidth. Preliminary investigations on heterogeneous scenarios have already been carried out in [9], [10], [11], and [12]. In [9], lower bounds to the chunk diffusion delay have been obtained. These results show that heterogeneity may be exploited to reduce the chunk delivery delay. Both the lower bounds in [9] ignore possible contentions at peers among different chunks. Furthermore, they are not constructive, providing no guidelines to devise efficient scheduling policies in peer-to-peer streaming systems. In [10], the authors propose an asyntotic analysis of the effect of the server bandwidth, number of peers, and number of hops in a heterogeneous scenario with general distribution of the peers' bandwidth. Close to our work is [11], in which the authors find delay bounds for the chunk distribution in both

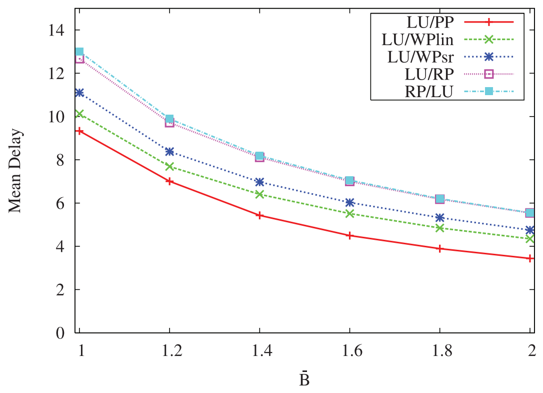

Fig. 1. Uniform scenario. Mean chunk delivery delay versus mean peer bandwidth.

homogeneous and heterogeneous cases. They also prove that these bounds can be achieved by properly constructing distribution trees that favor high-bandwidth peers over low bandwidth ones. We emphasize, however, that the algorithms proposed in [11], require a tight centralized control of the system. While our findings are in line with these results, we specifically consider and compare policies that can be implemented in a distributed fashion without requiring any form of global coordination among peers. Moreover, Liu [11] does not deal with overlay topology construction, as if a full-mesh topology is assumed.

In our previous work [12], the LU/WP policy with weights proportional to peer upload bandwidth has been first proposed, and its performance has been compared against LU/WP by simulation in some heterogeneous scenarios. This paper generalizes the ideas in [12] under several aspects: first, it proposes a new class of policies LU/PP; second, it provides a much wider and systematic performance comparison between different scheduling policies in heterogeneous scenarios; and third, it provides theoretical foundations to the empirical intuitions proposed in [12], showing that the chunk distribution delay is minimized by policies that give priority to large-bandwidth peers.

Finally, it is worth mentioning that, while we focus on unstructured systems, other approaches were proposed in the literature. For example, some solutions that exploit Multidescription Codecs for video encoding are based on the construction of multitrees, [13], [14], [15]. In particular, Kobayashi et al. [15] also deal with the problem of heterogeneous peers, but considering a multitree topology only.

4 A First Discussion

We start our discussion presenting some results obtained with an event-driven simulator. A simulation is organized in two phases. First, a random overlay topology is generated according to the desired graph specification and system parameters. During the second phase, the chunk distribution dynamics are simulated until all nodes have completed the content download.

Chunks are generated periodically by the source at a rate $\lambda = 1$; the chunk length, L, is normalized to 1 so that the interchunk emission time is $T = 1$. The bandwidth of the video source is $B_s = 10$. The system comprises $N = 10{,}000$ peers, all with infinite download and finite upload

TABLE 1
Percentage of Losses in the Uniform Scenario

	LU/PP	LU/WPlin	LU/WPsr	LU/RP	RP/LU
$\bar{B}$ =1.0	17.66	0.38	0.52	0.92	0.09
$\bar{B}$ =1.2	7.38	0	0	0	0

bandwidth. The size of the chunk collection is $W = 40$ chunks. This assumption reflects the fact that P2P streaming systems usually adopt a window mechanism to avoid the distribution of chunks that are not useful any more, because their delay exceeds the play-out buffer. The overlay is built as a *Quasi Regular Random Graph* in which arcs are randomly placed. Each peer selects $d^o = 100$ random neighbors, called *out-neighbors*. If peer q chooses peer p as an out-neighbor, we say that peer q is an *in-neighbor* of p. The union of the *in-neighbor* and *out-neighbor* sets of p is the neighborhood of p. Since links are bidirectional, chunks are exchanged between a peer p and the peers in its neighborhood. Thus, the size of the neighborhood of p is, on average, $\bar{d} = 2d^o = 200$. For each scenario, we simulate the distribution of 500 chunks.

In what follows, we analyze the system performance. We consider two sets of scenarios: the uniform and the heterogeneous bandwidth scenarios.

4.1 Uniform Scenario

In the first scenario, peer upload bandwidths are uniformly distributed in $[0, B_{\max}]$, the mean bandwidth is $\bar{B} = B_{\max}/2$ and we consider the cases: $\bar{B} = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$. By varying $\bar{B}$, the system load varies.

The mean chunk distribution delay is reported in Fig. 1. We consider five policies: RP/LU, LU/RP, LU/PP with weights proportional to peer upload bandwidth, and two versions of LU/WP, one (denoted by LU/WPlin) in which peer weights are proportional to peer upload bandwidths, i.e., $w_q = B_q$, and one (LP/WPsr) in which weights are proportional to the square root of the bandwidth, i.e., $w_q = \sqrt{B_q}$. In particular, with the choice of the LP/WPlin and LP/WPsr policies, we intend to test different degrees of dependence on the bandwidth; the LP/WPlin having a stronger dependence on the bandwidth than the LP/WPsr. Actually, this is an arbitrary choice; the use of other functions of the bandwidth in the scheduling process would lead to similar considerations.

We assume that the video source adopts the same scheduling policy of the other peers.

As expected, the performance of all the policies improves when $\bar{B}$ increases since the ratio $\rho = \lambda/\lambda_{\text{sup}}$ decreases. Interestingly, LU/RP and RP/LU achieve comparable performance (the curves are almost indistinguishable), in accordance with what was observed in [8], but their performance is the worst among the set of considered policies. Chunk delivery delay can be decreased by favoring large-bandwidth peers in the distribution process. The best performing policy turns out to be LU/PP, whose performance gain with respect to LU/RP and RP/LU is about 30 percent.

Table 1 reports the chunk loss ratio defined as the number of chunks that have never been received by peers over the total number of chunks; for $\bar{B} > 1.2$, no losses are observed. In the case $\bar{B} = 1$, that corresponds to the critical

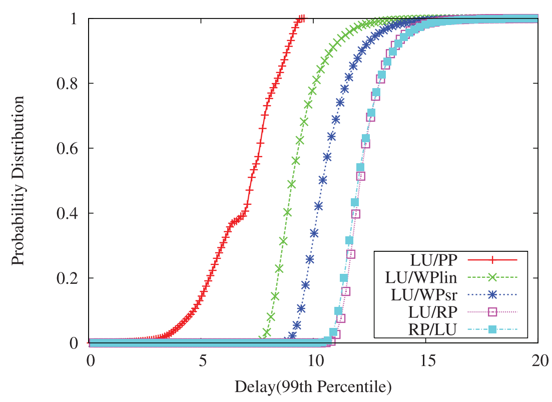

Fig. 2. Uniform scenario. Distribution of the 99th percentile of chunk delivery delay, with $\bar{B} = 1.4$.

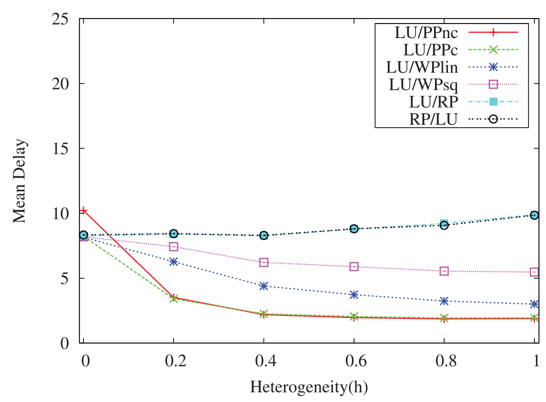

Fig. 4. Heterogeneous scenario. Mean delay versus heterogeneity coefficient h, with $\bar{B} = 1.4$.

condition $\rho = 1$, the use of a deterministic peer selection mechanism in the LU/PP case worsens performance in terms of losses: the percentage of losses is about 17 percent for the LU/PP scheme, and less than one percent in all the other cases. Indeed, since all peers have a different value of the bandwidth and the scheduling applies *strict* priority based on the bandwidth, chunks are typically distributed along a single spanning tree embedded into the overlay topology; the spatial diversity offered by the mesh approach is not effectively exploited.

Let us now focus on a case in which there is enough bandwidth, e.g., $\bar{B} = 1.4$. Fig. 2 reports the cumulative distribution function of the 99th percentile of the chunk distribution delay perceived by a peer; it shows that, even for this metric, the policies are ordered according to the importance they give to upload bandwidth. We remark that the adoption of policies favoring large-bandwidth peers results into a uniform reduction of the chunk delivery delay. Note that delay percentiles are important since they are closely related to the play-out delay and loss probability and loss probability.

These results suggest that the weighted peer approaches are at the same time efficient in terms of delay and throughput, and thus seem to offer the best trade-offs between delay and throughput.

4.2 Heterogeneous Scenario

In the scenarios discussed so far, peer upload bandwidths are homogeneous. Let us now focus on the impact of bandwidth heterogeneity on the system performance. Peers are partitioned into three classes according to their upload bandwidth: class 1 peers have bandwidth uniformly distributed between $[0.8B^{(1)}, 1.2B^{(1)}]$, with $B^{(1)} = 8.0\bar{B}$; class 2 peers have bandwidth uniformly distributed between $[0.8B^{(2)}, 1.2B^{(2)}]$, with $B^{(2)} = \bar{B}$; and class 3 peers have bandwidth uniformly distributed between $[0.8B^{(3)}, 1.2B^{(3)}]$, with $B^{(3)} = 0.5\bar{B}$. Based on the value of parameter $h \in [0, 1]$, that represents the *degree of bandwidth heterogeneity*, peers are distributed over the three classes according to the following ratios: $h/15$ peers fall in class 1; $1 - h$ in class 2; and $14h/15$ in class 3, so that the average peer upload bandwidth is equal to $\bar{B}$ for any value of h. The bandwidth variance monotonically increases with h.

In this scenario, we consider two versions of LU/PP: in the first one (denoted by LU/PP-nc), peer weight is set equal to peer upload bandwidth, as before; in the second one (LP/PP-c), all the class i peers have the same weight which is equal to the average bandwidth of the considered class $B^{(i)}$, i.e., $w_p = B^{(i)}$ if p belongs to class i. In other words, in LU/PP-nc, there are N different levels of priority, one *per peer*, while in LU/PP-c, there are only three, one *per class*.

Figs. 3 and 4 report the mean distribution delay for the different policies, respectively, for $\bar{B} = 1$ and $\bar{B} = 1.4$.

Again, RP/LU and LU/RP exhibit similar average delay in all cases. In addition, their absolute performance tends to slightly deteriorate with increasing values of the heterogeneity coefficient h. On the contrary, the policies that exploit bandwidth improve their performance with h. For $h = 1$, the relative gain of LU/PP (both versions) with respect to LU/RP and RP/LU approaches a factor of five.

In the case of $\bar{B} = 1.4$, no losses were observed for all the policies with the exception of LU/PP-nc, which experiences four percent of losses for $h = 0$. On the contrary, as shown in Table 2, when $\bar{B} = 1$, all policies experience some losses. In

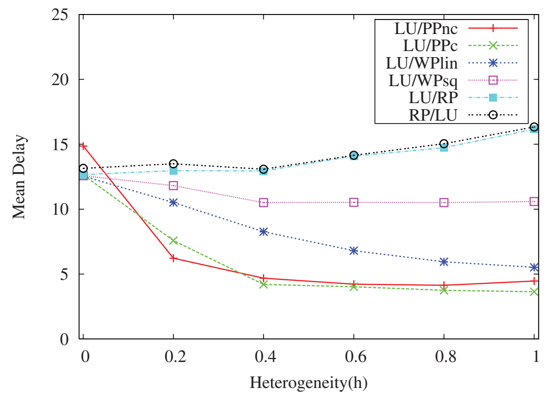

Fig. 3. Heterogeneous scenario. Mean delay versus heterogeneity coefficient h, with $\bar{B} = 1$.

TABLE 2
Percentage of Losses in the Heterogeneous Scenario for $\bar{B} = 1$

h	LU/PP-nc	LU/PP-c	LU/WPlin	LU/WPsr	LU/RP	RP/LU
0	54.4	0.3	0.32	0.25	0.30	0.20
0.2	22.4	0.12	0.88	0.66	0.09	0.65
0.4	2.0	0.06	0.53	0.12	0.02	0.04
0.6	3.2	0.08	1.5	1.4	0.72	1.0
0.8	5.2	2.2	3.0	3.3	2.3	3.0
1.0	7.3	4.4	4.6	5.0	4.2	4,5

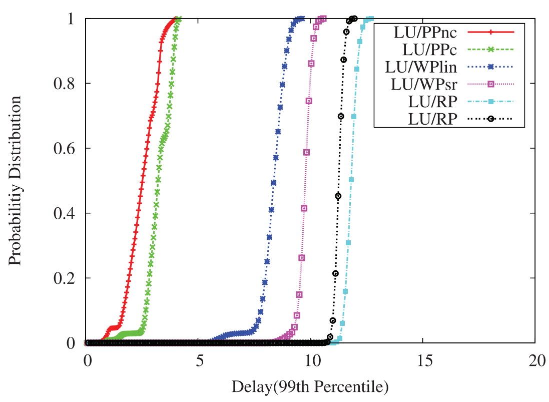

Fig. 5. Heterogeneous scenario. Distribution of the 99th delay percentile, with heterogeneity coefficient $h = 0.4$ and $\bar{B} = 1.4$.

particular, LU/PP-nc experiences very high losses for small h, losses exceed 50 percent for $h = 0$; for all the other policies, the number of losses increases with h, remaining, however, limited to five percent for $h = 1$; there is no evidence that some policy is more efficient than the others in terms of losses.

Observe that LU/PP-nc and LU/PP-c achieve very different loss performance especially for small h. Indeed, LU/PP-nc selects peers with different bandwidth in a deterministic order; thus, chunks are pushed over the same distribution tree. The bandwidth of low-weighted peers is inefficiently used and heavy throughput degradation is experienced. On the contrary, with LU/PP-c, the peers belonging to the same class have the same weight; in this case, the LU/PP-c selects the class in a deterministic way but chooses the peer within a class at random, thus guaranteeing a large degree of diversity among the distribution trees of different chunks. As a result, good throughput performance is achieved by LU/PP-c.

The slight increase of the mean delay that can be observed under the LU/RP and RP/LU policies for large values of the heterogeneity coefficient is due to the following phenomenon. When h is large, the network contains many low-bandwidth peers and only a few large-bandwidth peers. The RP policy suffers a higher probability of selecting a low-bandwidth peer, that, once selected, introduces large transmission delays that propagate to the receiving peers.

Let us now consider the case of $h = 0.4$ and $\bar{B} = 1.4$. Fig. 5 shows the distribution of the 99th delay percentile. As before, the policies are "ordered" according to the importance they give to bandwidth in the peer selection mechanism. Fig. 5 makes clear that all the peers, even those belonging to the third class, significantly benefit from the adoption of bandwidth aware policies.

The visible steps in the distributions of the LU/PP-nc, LU/PP-c, and LU/WPlin policies are due to the fact that peers with high upload bandwidth are favored in the selection mechanism and, thus perceive smaller delay than the other peers.

Summarizing, the following lessons can be learned from the presented results:

- By favoring peers that contribute the most to the chunk distribution process, bandwidth aware schemes improve delay performance; improvements increase with the degree of bandwidth heterogeneity.
- Under critical load conditions, the deterministic peer selection leads to throughput degradation (losses), due to a nonefficient use of spatial diversity. To avoid these effects, in LU/PP strategies, a sufficiently large degree of randomness must be guaranteed, for example, by assigning the same weight to peers with similar bandwidth (remember that ties are randomly broken).
- Delay-throughput trade-offs may arise in critical load conditions only; when conditions are not critical, performance is determined by delay and, by reducing delays, bandwidth aware schemes are preferable.

Given the above considerations, in all subsequent use of the LU/PP policy, we consider only the case in which peers are partitioned in classes and the same weight is assigned to peers in the same class.

5 Understanding the Role of Heterogeneity

To get some insights on previous results, in this section, we investigate the chunk diffusion process. The objective is to answer a few fundamental questions: 1) we previously observed that bandwidth aware schemes achieve significant performance gain; is this gain preserved when the system becomes big or huge? 2) in which way is bandwidth heterogeneity helping the chunk diffusion process? and finally, 3) can we generalize our observations and state that, in the presence of heterogeneity, an optimal scheduling algorithm must be bandwidth aware? We organize the discussion in three sections, each one devoted to one of the above questions. In this section, for ease of tractability, we consider a full-mesh overlay topology.

5.1 Analysis for Large Networks

To investigate the performance of the scheduling policies in large systems, we use a set of differential equations. In the equations, the variables that represent number of peers are continuous. Due to the large size of these systems, this relaxation does not introduce significant inaccuracy. The most difficult step in building a general representation of the chunks diffusion process is to account for possible bandwidth contention among different chunks. To the best of our knowledge, this is the first attempt to incorporate such contention effect in an equation-based model with the exception of the recursive equation derived in [8] for the latest blind chunk random peer. We emphasize, however, that the independent assumption made in [8] strongly relies on both the assumption that peers are indistinguishable (homogeneity) and the fact that peers are randomly selected by the scheduling policy; thus, that model cannot be used in our context.

In our model, peers are partitioned in I classes according to their upload bandwidth. Peers belonging to class i have the same upload bandwidth B_i. Without loss of generality, we assume $B_i > B_{i+1}$. Let N_i be the number of class i peers, and let N denote the total number of peers: $N = \sum_i N_i$. We focus on a chunk c that becomes available at the source at time $t = 0$. For any $t \in \mathbb{R}^+$, we represent with $n_i(t) \in \mathbb{R}^+$ the number of class i peers that have received the chunk by

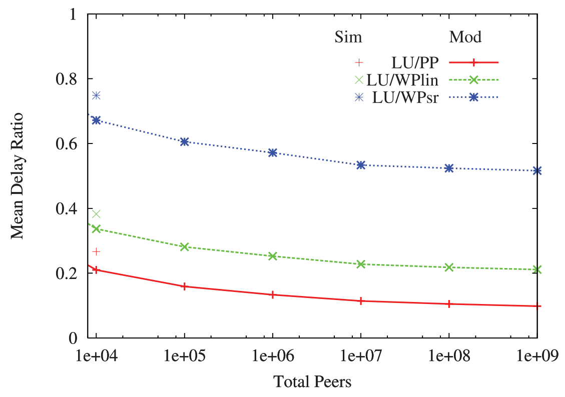

Fig. 6. Ratio of the mean delay in the LU/WP and LU/PP policies over the mean delay of the LU/RP policy versus the number of peers.

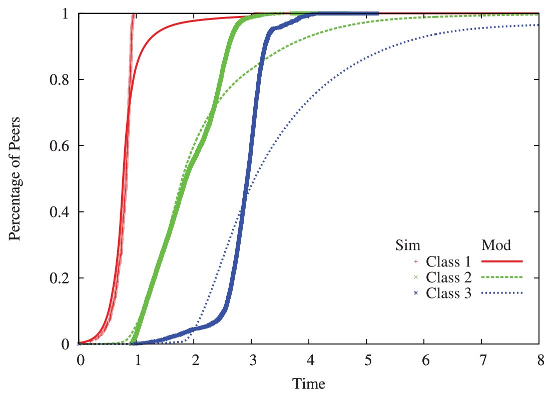

Fig. 7. Evolution of the number of class i peers holding a chunk for LU/PP policy, heterogeneous scenario with $\bar{B} = 1.4$ and $h = 0.4$.

time t, and with $n(t) = \sum_i n_i(t)$. Let $\mathcal{N}_i(t)$ be the set of class i peers that received chunk c by time t and $\bar{\mathcal{N}_i}(t)$ the set of class i peers that have not yet received chunk c. Lastly, let $\mathcal{N}(t) = \cup_i \mathcal{N}_i(t)$ and $\bar{\mathcal{N}}(t) = \cup_i \bar{\mathcal{N}_i}(t)$.

According to a *latest useful chunk* strategy, peers preferentially offer to neighbors the latest chunk they hold; other chunks are offered only when it is not possible to find neighbors that are still waiting for the latest chunk. The probability that a node offers any chunk but the latest one is negligible ($o(1/n)$). Thus, we neglect in the model this possibility. As a consequence, at time t, only the peers in $\mathcal{N}_i(t)$ that have not yet received any chunk fresher than c are offering c to other peers. Let us denote by $\mathcal{F}_i(t)$ this set of peers.

Assuming that the system is stable, the average time between the reception of two consecutive chunks by the same peer must be equal to the intergeneration time between chunks, T. Therefore, $n_i(t) - n_i(t-T)$ can be regarded as a first-order estimate of the cardinality of $\mathcal{F}_i(t)$, i.e., an estimate of the number of nodes that at time t are offering c to other peers. Depending on the specific scheduling strategy, peers in $\mathcal{F}_i(t)$ offer c to peers in class j, namely, peers in $\bar{\mathcal{N}_j}(t)$, with probability $p_{ij}(t)$. Thus, the total number of potential chunk transfers issued by nodes in $\mathcal{F}_i(t)$ toward peers in $\bar{\mathcal{N}_j}(t)$ is given by $p_{ij}(t)[n_i(t) - n_i(t-T)]$. These chunk transfers can be fully granted only when there is a sufficiently large number of nodes in $\bar{\mathcal{N}_j}(t)$ that can receive the chunk. The fraction of chunk offers directed to class j peers that can be granted is

$$r_j(t) = \min\left(1, \frac{N_j - n_j(t)}{\sum_i p_{ij}(t)[n_i(t) - n_i(t-T)]}\right). \qquad (1)$$

The number $n_j(t)$ of nodes which possess chunk c evolves with time according to the following differential equation:

$$\frac{dn_j(t)}{dt} = \ln(2)\frac{1}{L}\sum_i B_i r_j(t) p_{ij}(t)[n_i(t) - n_i(t-T)], \qquad (2)$$

where L is the chunk size and $\ln(2)$ is a correcting factor necessary to compensate the artificial speedup in the distribution process introduced by the fluidification of epidemic dynamics (note, indeed, that the model represents discrete quantities, the number of copies of a chunk fully available in the network, with continuous variables).

Probability $p_{ij}(t)$ of offering a chunk to peers in class j depends on the adopted peer selection strategy. For the random selection strategy,

$$p_{ij}(t) = \frac{N_j - n_j(t)}{N - n(t)}.$$

When strict priority is given to higher bandwidth peers with respect to lower bandwidth peers:

$$p_{ij}(t) = \begin{cases} 1, & \text{for } j = 1, \\ p_{i,j-1}(t)(1 - r_{j-1}(t)), & \text{for } j > 1. \end{cases}$$

In case a weighted priority w_i is given to peers based on bandwidth peers:

$$p_{ij}(t) = \frac{w_j \frac{N_j - n_j(t)}{N - n(t)}}{w_j \frac{N_j - n_j(t)}{N - n(t)} + \sum_{k \neq j} w_k \frac{N_k - n_k(t)}{N - n(t)} r_k(t)}.$$

Note that by construction $\sum_i p_{ij}(t) = 1$ when $r_i(t) = 1$ (for every i since all the transfer offers are granted); on the contrary, $\sum_i p_{ij}(t) \geq 1$ when some $r_i(t)$ becomes less than 1, because peers whose transfer offer was not granted redirect the request to peers of other classes.

This system of differential equations is numerically solved using the standard Runge-Kutta method [16]. By normalizing $n_i(t)$ over N_i, we obtain the cumulative distribution function of the chunk delay for class i peers. From this, we can obtain the mean delay for class i peers as well as the overall mean delay.

Fig. 6 shows the ratio of the mean delay achieved with the bandwidth aware policies over the delay of the LU/RP policy for increasing values of the number of peers. The plot confirms that the advantages of using bandwidth aware policies are significant and it also indicates that improvements increase with the size of the network. For large systems, the LU/WPsr achieves half the mean delay of the LU/RP policy; the LU/PP is one order of magnitude better. The plot also reports for $N = 10{,}000$ the results obtained with the simulator (clearly, the other cases, with larger values of N, cannot be simulated). The model slightly overestimates the gains achieved with bandwidth aware policies. However, note that the errors are in all cases within 20 percent. A more direct comparison between model and simulation, reported in Fig. 7, shows the evolution of the number of class i peers that received a specific chunk for the

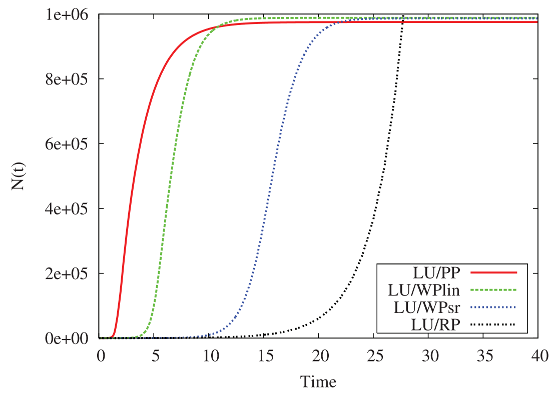

Fig. 8. Evolution of the peers holding a chunk, heterogeneous scenario with $\bar{B} = 1.4$ and $h = 0.4$.

LU/PP policy. The model is accurate in describing the evolution of the number of class 1 and class 2 peers holding the chunk; some error is introduced in the chunk distribution final phase for class 3 peers. The model is slightly pessimistic; however, it can be effectively used to provide any estimate of the number of peers holding the chunk.

5.2 Initial Phase Speedup

We now use the model to investigate the phenomena that make bandwidth aware policies so effective. In particular, we focus on the initial phase of the chunk diffusion process. Indeed, due to its epidemic nature, the chunk diffusion process can be divided into two parts: 1) an initial (slow) phase during which the chunk is distributed exploiting a limited number of peers, and 2) an explosive (fast) phase during which the number of peers holding the chunk is so large that the chunk distribution is completed in a very short time. The two phases are clearly visible in Fig. 8 that reports the evolution of the number of peers holding the chunk for the various policies for $N = 10^6$. Since the first phase is the slowest one, it dominates the delay performance.

In case of an LU/PP scheme, under the initial assumption $n(0) = n_1(0) = 1$, and defining with $t_i^* = \inf\{\tau : r_i(\tau) < 1\}$, we obtain that by construction $t_i^* < t_{i+1}^*$ and $n_i(t) = 0$ for $i > 1$ and $t < t_{i-1}^*$.

From the above considerations, the dynamics of $n_1(t)$ in (2) are given by

$$\frac{dn_j(t)}{dt} = \ln(2)\frac{1}{L}B_1[n_1(t) - n_1(t-T)] \quad \text{for } t < t_1^*. \tag{3}$$

This equation can be solved iteratively by standard techniques. We start during the first interval of length T with the term $n_1(t-T) = 0$ and the initial value $n_1(0) = 1$. In the interval $t \in [T, 2T]$, the right hand side of the equation contains the forcing term $n_1(t-T)$ that is derived from the solution of the equation in the previous interval. At the kth iteration step, the solution of (3) is in the form:

$$n(t) = n_1(t) = \sum_{m=0}^{k} \beta_m^k t^m 2^{\frac{B_1}{L}t} \qquad kT \le t < (k+1)T, \tag{4}$$

where coefficients β_m^k are derived by forcing (4) to satisfy (3) at every interval $t \in [kT, (k+1)T)$ as well as to be continuous at points kT. After some standard mathematical passages, it can be shown that the coefficients obey the following recursion equations:

$$\beta_{m+1}^k = -\frac{B_1}{L}2^{-\frac{B_1}{L}} \sum_{h=m}^{k-1} \binom{h}{m} \beta_h^{k-1} \tag{5}$$

and

$$\sum_{m=0}^{k-1} \beta_m^{k-1} k^m = \sum_{m=0}^{k} \beta_m^k k^m,$$

under the starting condition $\beta_0^0 = 1$.

In the case of the LU/RP scheme, it can be easily shown that in the initial phase (i.e., when all $r^i(t) = 1$), by construction

$$\frac{n_i(t)}{n_j(t)} \approx \frac{n_i(t) - n_i(0)}{n_j(t) - n_j(0)} = \frac{N_i}{N_j}, \tag{6}$$

and thus for any i and j, $t_i^* = t_j^*$. As a consequence, it turns out that

$$n(t) = \sum_i n_i(t) \asymp \sum_{m=0}^{k} \beta_m^k t^m 2^{\frac{\bar{B}}{L}t} \qquad kT \le t < (k+1)T \tag{7}$$

with $\bar{B} = \sum_i B_i \frac{N_i}{N}$, and the coefficients β_m^k are related by the same recursion as (5), replacing B_1 with $\bar{B}$.

At last, for the LU/WP scheme, assuming $w_i \ge w_{i+1}$, it turns out that for sufficiently small t:

$$\frac{n_i(t)}{n_j(t)} \approx \frac{n_i(t) - n_i(0)}{n_j(t) - n_j(0)} = \frac{N_i w_i}{N_j w_j}.$$

More generally, in the initial phase (i.e., when all $r^i(t) = 1$), by construction

$$\frac{N_i}{N_{i+1}} \le \frac{n_i(t) - n_i(0)}{n_{i+1}(t) - n_{i+1}(0)} = \frac{N_i w_i}{N_{i+1} w_{i+1}},$$

thus, $t_i^* \le t_{i+1}^*$. As a consequence

$$n(t) = \sum_i n_i(t) \le \sum_{m=0}^{k} \beta_m^k t^m 2^{\frac{B_w}{L}t} \quad \text{for} \quad kT \le t < (k+1)T \tag{8}$$

with

$$B_w = B_i \frac{N_i w_i}{\sum_j w_j N_j}$$

and the coefficients β_m^k are related by the same recursion as (5) replacing B_1 with B_w. The above bound is tight for $t \ll t_0$.

Note that according to (4), (7), and (8), the initial evolution of $n(t)$ is driven by different time constants: B_1/L for LU/PP, B_w/L for LU/WP, and $\bar{B}/L$ for LU/RP. This shows the advantage in terms of delay of schemes that favor higher bandwidth peers. Fig. 9 reports the initial evolution of $n(t)$ for the different policies for previously described heterogeneous scenario with $\bar{B} = 1.4$ and $h = 0.4$ (notice the logarithmic scale of the y-axis). These results show that the initial phase of the chunk distribution is much faster in the bandwidth aware schemes: the slopes of the curves are remarkably different.

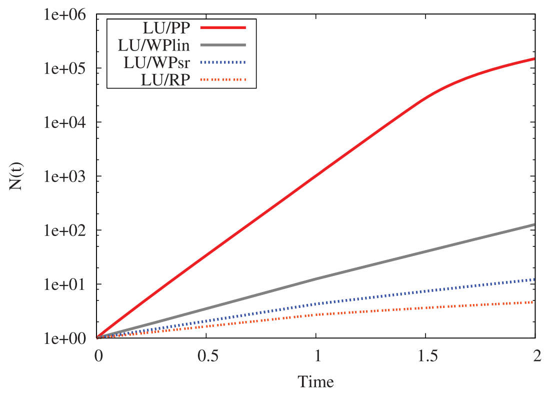

Fig. 9. Evolution of the number of peers holding a chunk during the initial phase of the chunk distribution process for $N = 10^6$.

5.3 The Optimal Scheduling

Further hints to the delay optimality of the policy that assigns priority to higher bandwidth peers can be gathered focusing on the *inherent distribution average delay* (i.e., the minimum average delay) encountered by a tagged chunk c while being distributed along a spanning tree embedded in the overlay topology. Our analysis aims at identifying structural properties of the distribution tree that minimize the chunk distribution delay. To capture the effect of competition for peer upload bandwidth, we constrain the structure of the distribution trees imposing a maximum out-degree constraint at every peer.

Peer degree constraints, indeed, express in graph theoretical terms the fact that the maximum amount of time that each peer can devote to the transmission of a given chunk c is *on average* bounded by T since all chunks emitted by the source are sharing peer bandwidth. In the case of the latest chunk policy, for example, the time T is given by the chunk intergeneration time $1/\lambda$. This results in a constraint on the average number of other peers to which peer p can deliver chunks. This constraint depends on T and the peer upload bandwidth B_p, so that typically it can be assumed to be equal to $B_p \frac{T}{L}$. In general, the higher the bandwidth is, the higher the number of neighbors the chunk is sent to.

Our main finding is stated in the following theorem:

Theorem 1. *Among all the possible distribution trees X satisfying the following degree constraint:*

$$NC^X(p) \le \lfloor g(B_p) \rfloor \tag{9}$$

being $NC^X(p)$ the number of children of peer p in tree X and $g()$ an arbitrary nondecreasing function of the bandwidth, a distribution tree O minimizing the mean inherent delay can be found in which chunk reception times of peers are in reverse order of their upload bandwidth.

Proof. The proof is provided in the Appendix. □

In [10], [11], the authors have proved simpler versions of this result, assuming that there is no constraint on the number of children a peer can have in the distribution tree. In our formulation, the constraint of (9) comes from the fact that peers adopt the latest chunk first policy.

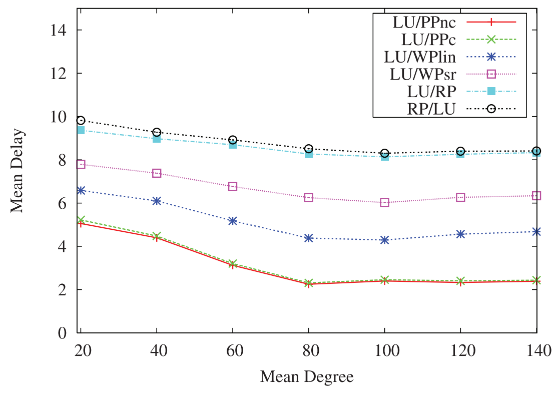

Fig. 10. Heterogeneous scenario. Mean delay versus mean degree, with heterogeneity coefficient $h = 0.4$ and $\bar{B} = 1.4$.

6 Overlay Topologies with Limited Degree

Theorem 1 states that chunks should be distributed to large upload bandwidth peers first so that in the resulting optimal chunk distribution tree, the distance of a peer from the source depends on the peer upload bandwidth: the larger the bandwidth is, the closer the peer is to the source. In a full-mesh topology overlay, scheduling policies that take into account peer bandwidth tend to actually implement distribution trees according to this rule. However, when the degree of the overlay is small, a random selection of the neighbors may lead to overlays in which a significant number of high-bandwidth peers are placed far away from the source. In these situations, the effectiveness of the scheduling algorithm reduces.

Fig. 10 shows the mean delay versus the mean degree for the considered scheduling policies in the simulation scenario of Section 4.2 where $\bar{B} = 1.4$ and $h = 0.4$. Basically, for values of the mean degree larger than 80, the mean delay is constant, suggesting that 80 neighbors, on average, are already enough to fully exploit the bandwidth of high-bandwidth peers at the beginning of the chunk diffusion process. For values of the mean degree smaller than 80, some high-bandwidth peers might receive the chunk and start contributing to its diffusion with some additional delay.

For the LU/PP case, Fig. 11 shows the distribution of the 99th delay percentile for several randomly generated overlay topologies with different average degree. The

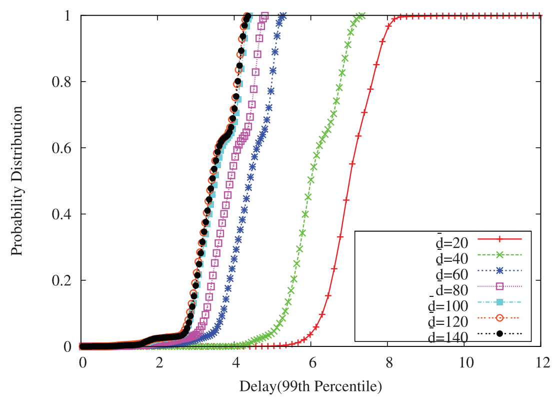

Fig. 11. Distribution of the 99th delay percentile for the LU/PP policy, with random overlay topology and different values of the average peer degree.

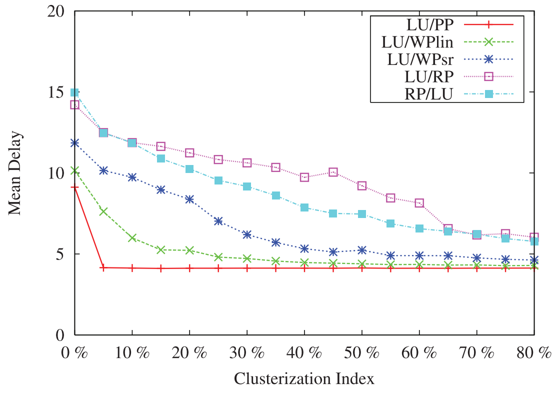

Fig. 12. Mean delay versus the clusterization index, defined as the percentage of high-bandwidth peers contacted by an high-bandwidth peer; number of contacted peer, $d^o = 20$, heterogeneous scenario with $h = 0.4$ and $\bar{B} = 1.0$.

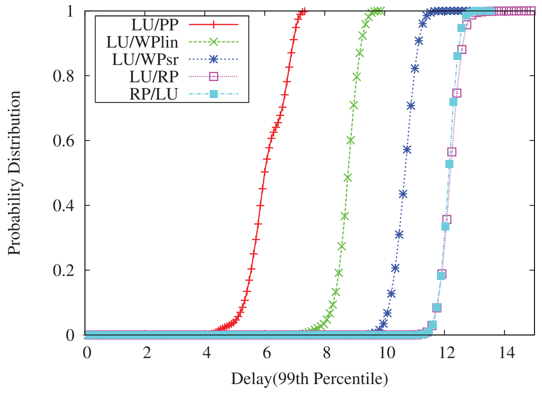

Fig. 14. Distribution of the 99th delay percentile with random topology construction (clusterization index 0), number of contacted peer, $d^o = 20$.

performance degradation for small values of $\bar{d}$ is even more visible focusing on the 99th delay percentile. However, while improving performance, maintaining large values of $\bar{d}$ may have significant costs in terms of algorithm complexity and signaling overhead since neighbor status (such as the collection of chunks held by a neighbor q, $C(q)$) must be periodically exchanged among peers.

The performance degradation can be explained by means of the equations in Section 5. The limited degree affects the evolution of the number of peers that possess the chunk that is expressed by (2) in two ways. First, the number of peers that can contribute to the chunk diffusion process is not anymore given by $[n_i(t) - n_i(t-T)]$, but it reduces as the degree decreases. Indeed, the chance that a peer is connected to neighbors that still need the chunk decreases with the degree. Second, the reduction of the degree reduces also the effectiveness of the scheduling policies. A contributing peer can choose which peer to serve within the set of its neighbors only, so that a smaller degree translates into a more restricted choice. The number of all class j potential receivers of a chunk, which in (1) is given by the numerator, $N_j - n_j(t)$, reduces to the number of class j peers that have a contributing neighbor. As a consequence, the fraction of chunk offers that can be granted, $r_j(t)$, becomes smaller than 1 sooner and sooner as the degree decreases. This implies that the policies that favor high-bandwidth peers are less effective, due to the topological constraints. As an extreme example, consider the case in which no high-bandwidth peer has a neighbor which is distributing the chunk. In this case, regardless the scheduling policy, no high-bandwidth peer can receive the chunk.

To avoid performance degradation, the neighborhood selection mechanism should take into account peer upload bandwidth, making high-bandwidth peers highly connected with each other so that the topological constraints have limited effect on $r_1(t)$. For example, to show how clusterization can have positive impact, let us consider the following scheme. High-bandwidth peers choose other high-bandwidth peers as out-neighbors with probability α, named *clusterization index*, and with probability $1-\alpha$ they randomly select a peer. The other peers randomly choose out-neighbors. Notice that when $\alpha = 0$, all out-neighbors are randomly chosen. Figs. 12 and 13 report the average delay as a function of the clusterization index in the heterogeneous scenario with $h = 0.4$ and out-degree $d^o = 20$; $\bar{B} = 1.0$ in Fig. 12 and $\bar{B} = 1.4$ in Fig. 13. The number of observed losses is negligible in the case $\bar{B} = 1.0$ (it is about one percent for clustering index equal to 0, and even smaller in the other cases) and no losses at all were observed when $\bar{B} = 1.4$. Figs. 14 and 15, instead, report the 99th percentile of the delay for clusterization index equal to 0 and 80 percent, respectively; $\bar{B} = 1.4$.

The results show that all policies benefit from the clusterization, the larger benefits are observed when

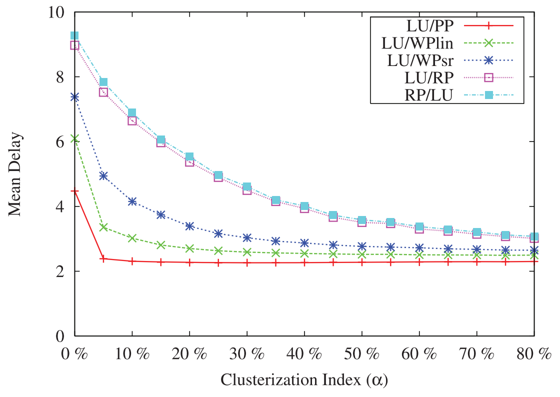

Fig. 13. Mean delay versus the clusterization index, defined as the percentage of high-bandwidth peers contacted by an high-bandwidth peer; number of contacted peer, $d^o = 20$, heterogeneous scenario with $h = 0.4$ and $\bar{B} = 1.4$.

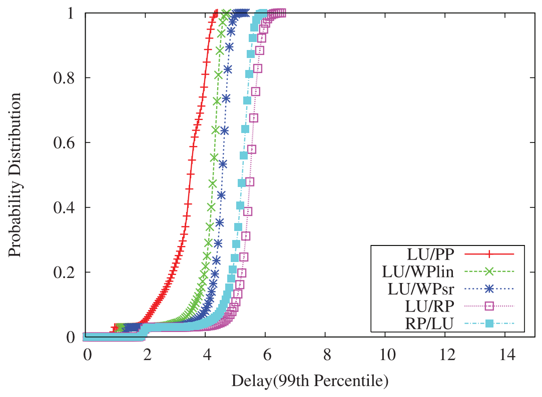

Fig. 15. Distribution of the 99th delay percentile with random topology construction (clusterization index 80 percent), number of contacted peer, $d^o = 20$.

LU/RP and RP/LU policies are used. Note that the gap among the performance of different policies decreases when the clusterization parameter increases. This phenomenon is due to the bias that the clusterization induces in the peer selection performed by the random choice. As the fraction of high-bandwidth peers in the neighborhood of peer p increases, the probability for p of selecting a high bandwidth increases as well. Thus, LU/RP and RP/LU over a highly clustered topology tend to indirectly favor high-bandwidth peers, behaving more similarly to bandwidth aware policies.

As a final remark, we emphasize that our investigation is just devoted to showing the potential advantages of considering the peer attributes such as upload bandwidth while building an overlay topology; however, in this paper, we have not tackled the delicate issue of how to achieve clusterization through simple distributed schemes.

7 Conclusions

In scenarios where peer upload bandwidths exhibit a significant degree of heterogeneity, the resources of large-bandwidth peers are very precious and can be exploited to speed up the chunk spreading process at an early stage of the chunk epidemic distribution process. Our results show both chunk scheduling policies and overlay topology design mechanisms aimed at favoring an early distribution of chunks among high-bandwidth peers are very effective in reducing the chunk distribution delay.

In particular, those schemes according to which the chunk reception times of peers are sorted in reverse order of their upload bandwidth can be proved to minimize the average inherent distribution delay. Unfortunately, the deterministic peer selection may lead to some undesired throughput degradation due to a nonefficient use of the spatial diversity provided by the mesh approach. These undesired effects can be easily avoided by introducing some degree of randomness in the order in which chunks are distributed among peers.

Appendix

Proof of Theorem 1. Let X be a given distribution tree. Under distribution tree X, we define the following notation:

- D_p^X: peer p delay in receiving the chunk.
- $l^X(p)$: peer p level along the tree; i.e., the number of peers along path connecting p to the root.
- $f^X(p)$: peer p father.
- $f_i^X(p)$: ith level ancestor of p; note that by construction $f_{i-1}^X(p) = f^X(f_i^X(p))$ and $f_{l^X(p)-1}^X(p) = f^X(p)$ and $f_{l^X(p)}^X(p) = p$.
- $i^X(p)$: chunk reception ordering within the set of brothers; i.e., $i^X(p) = k$ means that p is the kth child of $f^X(p)$ receiving the chunk.
- $C^X(p,i)$: child of p that is served as the ith one.
- $NC^X(p)$: number of peer p children.
- $ND^X(p)$: number of peer p descendants comprising p itself.
- $h(B)$ is a chunk transmission time for a peer with upload bandwidth B. Function $h()$ is assumed decreasing with respect to its argument.

□

Theorem. *Among all the possible distribution trees X satisfying the following degree constraint:*

$$NC^X(p) \leq \lfloor g(B_p) \rfloor \tag{10}$$

being $NC^X(p)$, the number of children of peer p in tree X and $g()$ an arbitrary nondecreasing function of the bandwidth, a distribution tree O minimizing the mean inherent delay can be found in which chunk reception times of peers are in reverse order of their upload bandwidth.

Proof. The proof is articulated in two steps.

As first step, we consider a distribution tree X and we show that if there are two peers p and q such that $D_p^X < D_q^X$ and $B_p < B_q$; i.e., peer p, whose bandwidth is smaller than q's one, receives the chunk before q, then it is possible to build a distribution tree Y such that $E[D^Y] \leq E[D^X]$ in which the order in which p and q receive the chunk is reversed, i.e., $D_q^Y = D_p^X$ and $D_q^Y = D_p^X$.

As second step, iterating the previous argument we show that it is always possible to find a delay optimal distribution tree O in which peer reception time D_p^O is in reverse order of peer upload bandwidth B_p, that is, $E[D^O] \leq E[D^X]$ for any possible distribution tree X.

We start by proving the first step.

Consider a chunk c which is distributed over the distribution tree X. The mean delay in receiving c can be expressed as

$$E[D^X] = \frac{1}{N}\sum_{p\in\mathcal{P}} D_p^X, \tag{11}$$

where D_p^X is

$$\begin{aligned} D_p^X &= D_{f(p)}^X + i^X(p)h(B_{f^X(p)}) \\ &= D_{f_m(p)}^X + \sum_{l=m+1}^{l^X(p)} i^X(f_l^X(p))h(B_{f_{l-1}^X(p)}) \\ &\quad \forall m < l^X(p). \end{aligned} \tag{12}$$

When, as assumed in the rest of this paper, latencies and signaling overhead are negligible, it results $h(B) = L/B$ and $g(B) = BT/L$ (with T the average chunk interchunk generation time). Observe that D_p^X depends on the bandwidth of all the ancestors of p while it is independent on B_p.

Now, given the distribution tree X in which a pair of peers p and q can be found with $D_p^X < D_q^X$ and $B_p < B_q$, it is possible to build a new tree Y with $D_q^Y < D_p^Y$ such that $E[D^Y] \leq E[D^X]$. Tree Y can be obtained from X according to the following procedure.

Let us start by considering the case in which p and q are not descendant of each other.

The children of q are scanned sequentially. Consider the ith child of q, namely, $C^X(q,i)$. The number of descendants of $C^X(q,i)$ is compared with the number of descendants of $C^X(p,i)$. Whenever $ND^X(C^X(q,i)) \geq ND^X(C^X(p,i))$, the subtree composed of $C^X(q,i)$ with

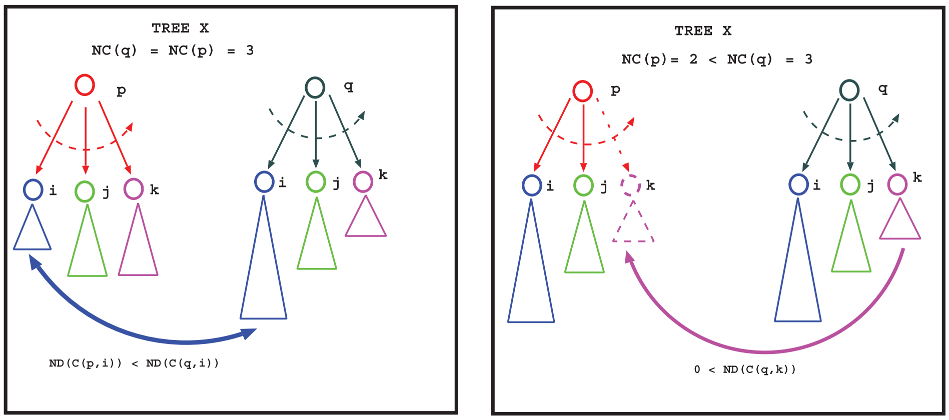

Fig. 16. Sketch of the proof. On the left, subtrees in position i are swapped; subtrees in positions j and k are unswapped. On the right, the case of a number of children of p smaller than that of q swaps with a null subtree.

all its descendants is exchanged with the subtree composed of $C^X(p,i)$ with all its descendants; we say that the two subtrees are *swapped*, and call *unswapped* the subtrees that do not undergo this swapping. This case is made on left part of Fig. 16 that graphically represents the transformation operated on tree X.

If the number of children of p is smaller than those of q, not all the comparisons are possible. In particular, for all the ith children of q, with $i > NC^X(p)$, the comparison between $ND^X(C^X(q,i))$ and $ND^X(C^X(p,i))$ is not possible. In these cases, we swap the subtree formed by $C^X(q,i)$ with all its descendants with the null subtree that conventionally occupies the position of the ith child of p (child that actually does not exist); or, equivalently, it is conventionally assumed that $ND^X(C^X(p,i)) = 0$ (see right part of Fig. 16).

Finally, swap p and q.

Notice that, at the end of the procedure, the peer that is the root of a swapped subtree conserves its father and its ordering; if it is the ith child of p in X, it is also the ith child of p in Y. On the contrary, the root of an unswapped subtree changes its father but not its ordering as a child, e.g., if it is the ith child of p in X, it becomes the ith child of q in Y.

Observe that, by construction, in tree Y, the number of children of q is equal to the maximum between the number of children of p and q in X: $NC^Y(q) = \max(NC^X(p), NC^X(q))$. In addition, the number of children of any other peer is unchanged. Since by assumption $B_q > B_p$ (the degree constraint of p is stricter than the one of q), Y always satisfies the degree constraints (10) whenever X satisfies them.

Now, in order to verify that $E[D^Y] < E[D^Y]$, notice that $E[D^Y]$ can be related to $E[D^X]$ by summing up any delay variation experienced by any peer. The only peers in $\mathcal{P}$ with delay variations are: p, q, or their descendants. For each peer in a swapped subtree with father p, the delay variation is equal to

$$\Delta D_s = \left(D_p^Y - D_p^X\right), \tag{13}$$

since the subtree is served by p in the same order as before, with the same bandwidth, but p itself was swapped with q; that is, p receives in distribution tree Y the chunk with the same delay with which q used to receive the chunk in tree X:

$$D_p^X = D_q^Y \quad \text{and} \quad D_q^X = D_p^Y. \tag{14}$$

For each peer in a swapped subtree with father q, the delay variation is equal to $-\Delta D_s$

$$\Delta D_s^- = \left(D_q^Y - D_q^X\right) = -\Delta D_s. \tag{15}$$

Each peer in an unswapped subtree of order i with ancestor p (the ancestor changed from q to p, the serving ordering i is unchanged), the delay variation is

$$\Delta D_u(i) = i\big[h(B_p) - h(B_q)\big], \tag{16}$$

where $h(B_p)$ is a nonincreasing function of the bandwidth that represents the time needed to deliver the chunk to the neighbor. Similarly, peers in an unswapped subtree of order i with ancestor q perceive delay variation

$$\Delta D_u^-(i) = i\big[h(B_q) - h(B_p)\big] = -\Delta D_u(i). \tag{17}$$

By summing up all the delay variations, i.e., taking into account that each peer in a subtree contributes to the delay variation according to the expressions above, we get

$$\begin{aligned} E[D^X] - E[D^Y] = \frac{1}{N} \cdot \Bigg(& \sum_{i=1}^{NC^X(p)} \Delta D_u(i) \cdot \\ & \left[ND^X(C^X(p,i)) - ND^X(C^X(q,i))\right]^+ + \\ & + \sum_{i=1}^{NC^X(q)} \Delta D_s \cdot \\ & \left[ND^X(C^X(q,i)) - ND^X(C^X(p,i))\right]^+ \Bigg), \end{aligned} \tag{18}$$

where $[x]^+ = \max(0,x)$. Thus, $E[D^Y] \le E[D^X]$.

The case in which p and q are descendant of each other corresponds to p being an ancestor of q since $D_p^X < D_q^X$. We proceed to the construction of B in a similar way as before. If $NC^X(p) \ge NC^X(q)$, we just exchange p and q. It results that q is ancestor of p, all subtrees of q in X become subtrees of p in Y, and vice versa. If, on the contrary, $NC^X(p) < NC^X(q)$, we swap p and q together with some subtrees of q. In particular, we also move $NC^X(q) - NC^X(p)$ children of q together with their subtrees; these children of q in X remain children of q in Y. The reduction of average delay can be computed similarly to (18).

Summarizing, the advantage of Y over X in terms of average delay comes from the combination of the following aspects: First, the ith child of q is favored over the ith child of p because q is served first and q has larger bandwidth. Then, since this advantage propagates to all the descendants, we force the subtree below the ith child of q to be larger than the one below the ith child of p.

Proof of the second step.

Starting from a general distribution tree X, we implement the following algorithm:

```
STEP 0. Set $i = 0$, $X_i = X$, $C_0 = \infty$.
STEP 1. Scan all the peers in $X_i$ and select, if any, the
        pair of peers $p$ and $q$ that maximizes the sum
        $C_i = B_p + B_q$, under the constraint that
        $D_p^X < D_q^X$ and $B_p < B_q$.
STEP 2. If no such pair of peers is found, then exit.
        Otherwise, if a pair of peers $(p, q)$ is found,
        then build a new tree $Y$ in which $p$ and $q$ are
        exchanged according to the procedure
        defined above.
STEP 3. Increase $i \to i + 1$, set $X_i = Y$, set
        $C_i = B_p + B_q$.
STEP 4. Go back to STEP 1.
```

Note that by construction:

1. $E[D^{X_{i+1}}] \leq E[D^{X_i}]$;
2. necessarily, the algorithm terminates only when peers receive the chunk in reverse order of their upload bandwidth, i.e., X is the optimal distribution tree, $X = O$;
3. costs C_i are monotonic, $C_i \geq C_{i+1}$; and
4. monotonicity of costs C_i implies that algorithm will necessarily terminate after at most $N(N-1)$ iterations.

As a result, the average delay $E[D^O] \leq E[D^X]$ for any possible distribution tree X. □

Acknowledgments

This work has been supported by the European Commission through NAPA-WINE Project (Network-Aware P2P-TV Application over Wise Network), ICT Call 1 FP7-ICT-2007-1.

References

[1] PPLive, http://www.pplive.com, 2009.

[2] PPStream, http://www.PPStream.com, 2009.

[3] SOPCast, http://www.sopcast.com, 2011.

[4] TVAnts, http://www.tvants.com, 2009.

[5] D. Ren, Y.T. Hillman Li, and S.H. Gary Chan, "On Reducing Mesh Delay for Peer-to-Peer Live Streaming," *Proc. IEEE INFOCOM '08,* Apr. 2008.

[6] L. Massoulie, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized Decentralized Broadcasting Algorithms," *Proc. IEEE INFOCOM '07,* May 2007.

[7] S. Sanghavi, B. Hajek, and L. Massoulie, "Gossiping with Multiple Messages," *Proc. IEEE INFOCOM '07,* May 2007.

[8] T. Bonald, L. Massoulie, F. Mathieu, D. Perino, and A. Twigg, "Epidemic Live Streaming: Optimal Performance Trade-Offs," *Proc. ACM Sigmetrics '08,* June 2008.

[9] F. Picconi and L. Massoulie, "Is There a Future for Mesh-Based Live Video Streaming?," *Proc. IEEE Int'l Conf. Peer-to-Peer (P2P) Computing '08,* Sept. 2008.

[10] T. Small, B. Liang, and B. Li, "Scaling Laws and Tradeoffs in Peer-to-Peer Live Multimedia Streaming," *Proc. Ann. ACM Int'l Conf. Multimedia,* Oct. 2006.

[11] Y. Liu, "On the Minimum Delay Peer-to-Peer Video Streaming: How Realtime Can It Be?," *Proc. ACM Int'l Conf. Multimedia,* Sept. 2007.

[12] A. Couto da Silva, E. Leonardi, M. Mellia, and M. Meo, "A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems," *Proc. IEEE Int'l Conf. Peer-to-Peer (P2P) Computing '08,* Sept. 2008.

[13] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-Bandwdith Content Distribution in Cooperative Environments," *Proc. Int'l Workshop Peer-to-Peer Systems (IPTPS),* Oct. 2003.

[14] V. Padmanabhan, H.J. Wang, and P.A. Chou, "Resilient Peer-to-Peer Streaming," *Proc. IEEE Int'l Conf. Network Protocols (ICNP),* Nov. 2003.

[15] M. Kobayashi, H. Nakayama, N. Ansari, and N. Kato, "Robust and Efficient Stream Delivery for Application Layer Multicasting in Heterogeneous Networks," *IEEE Trans. Multimedia,* vol. 11, no. 1, pp. 166-176, Jan. 2009.

[16] S.A. Teukolsky and W.T. Vettering, *Numerical Recipes in C: The Art of Scientific Computing,* p. 710. W.H. Press Cambridge Univ. Press, 1992.

Ana Paula Couto da Silva received the BSc degree in computer science from Federal University of Juiz de Fora, Brazil, in 1999, and the MSc and DSc degrees in computer and system engineering from Federal University of Rio de Janeiro, Brazil, in 2001 and 2006, respectively. She was with IRISA-Rennes, France, in 2005 and 2007 and with Politecnico di Torino, Italy, in 2008. Since February 2009, she has been with the Computer Science Department at Federal University of Juiz de Fora, as an assistant professor. Her areas of interest are in the fields of modeling and analysis of computer systems, reliability analysis, and P2P application analysis.

Emilio Leonardi received the Dr-Ing degree in electronics engineering in 1991 and the PhD degree in telecommunications engineering in 1995 both from Politecnico di Torino. He is currently an associate professor at the Dipartimento di Elettronica of Politecnico di Torino. In 1995, he visited the Computer Science Department of the University of California, Los Angeles (UCLA), in summer 1999, he joined the High Speed Networks Research Group, at Bell Laboratories/Lucent Technologies, Holmdel (New Jersey); in summer 2001, the Electrical Engineering Department of the Stanford University, and finally in summer 2003, the IP Group at Sprint, Advanced Technologies Laboratories, Burlingame CA. He is the scientific coordinator of the European Seventh FP STREP project "NAPA-WINE" on P2P streaming applications, involving 11 European research institutions, operators, and manufacturers. His research interests are in the fields of performance evaluation of wireless networks, P2P systems, and packet switching. He is a senior member of the IEEE.

Marco Mellia (S'08) received the PhD degree in telecommunications engineering in 2001 from Politecnico di Torino. During 1999, he was with the Computer Science Department at Carnegie Mellon University, Pittsburgh, Pennsylvania, as a visiting scholar. During 2002, he visited the Sprint Advanced Technology Laboratories Burlingame, California. Since April 2001, he has been with the Electronics Department of Politecnico di Torino as an assistant professor. He has coauthored more than 140 papers published in international journals and presented in leading international conferences, all of them in the area of telecommunication networks, and he participated in the program committees of several conferences including IEEE Infocom and ACM Sigcomm. His research interests are in the fields of traffic measurement and modeling, P2P application analysis, and energy aware network design. He is a senior member of the IEEE.

Michela Meo received the Laurea degree in electronics engineering in 1993 and the PhD degree in electronic and telecommunications engineering in 1997, both from the Politecnico di Torino, Italy. Since November 1999, she has been an assistant professor at Politecnico di Torino. She coauthored more than 120 papers, about 40 of which are in international journals. She edited six special issues of international journals, including *ACM Monet*, *Performance Evaluation*, and *Journal and Computer Networks*. She was program cochair of two editions of ACM MSWiM, general chair of another edition of ACM MSWiM, program cochair of IEEE QoS-IP, IEEE MoVeNet '07, and IEEE ISCC '09, and she was in the program committee of about 50 international conferences, including Sigmetrics, Infocom, ICC, and Globecom. Her research interests are in the fields of performance evaluation and modeling, traffic classification and characterization, P2P, and green networking.