Epidemic Live Streaming: Optimal Performance Trade-Offs

Thomas Bonald†, Laurent Massoulié⋆, Fabien Mathieu†, Diego Perino†, Andrew Twigg⋆

†Orange Labs
Issy-les-Moulineaux, France
{thomas.bonald,fabien.mathieu,diego.perino}
@orange-ftgroup.com

⋆Thomson Technology Paris Laboratory
Boulogne-Billancourt, France
{laurent.massoulié,andrew.twigg}
@thomson.net

ABSTRACT

Several peer-to-peer systems for live streaming have been recently deployed (e.g. CoolStreaming, PPLive, SopCast). These all rely on distributed, epidemic-style dissemination mechanisms. Despite their popularity, the fundamental performance trade-offs of such mechanisms are still poorly understood. In this paper we propose several results that contribute to the understanding of such trade-offs.

Specifically, we prove that the so-called *random peer, latest useful chunk* mechanism can achieve dissemination at an optimal rate and within an optimal delay, up to an additive constant term. This qualitative result suggests that epidemic live streaming algorithms can achieve near-unbeatable rates and delays. Using mean-field approximations, we also derive recursive formulas for the diffusion function of two schemes referred to as *latest blind chunk, random peer* and *latest blind chunk, random useful peer*.

Finally, we provide simulation results that validate the above theoretical results and allow us to compare the performance of various practically interesting diffusion schemes in terms of delay, rate, and control overhead. In particular, we identify several peer/chunk selection algorithms that achieve near-optimal performance trade-offs. Moreover, we show that the control overhead needed to implement these algorithms may be reduced by restricting the neighborhood of each peer without substantial performance degradation.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*
; C.4 [**Performance of Systems**]: Design Studies

General Terms

Algorithms, performance

Keywords

P2P live streaming, epidemic diffusion, delay optimality.

SIGMETRICS'08, June 2–6, 2008, Annapolis, Maryland, USA.

1. INTRODUCTION

The diffusion of live streaming through peer-to-peer (P2P) overlays has become increasingly popular in the past few years, as shown by the success of commercial systems like CoolStreaming [23], PPLive [9], SopCast [2], TVants [1] and UUSee [3]. These all rely on distributed, epidemic-style dissemination mechanisms: the stream is divided into small parts, so-called chunks, that follow random, independent paths in the peer population.

This is in contrast with *structured* systems like SplitStream that consist in building a multicast overlay by means of one or several static spanning trees [5]. The inherent scalability and robustness of *unstructured* systems make them more suitable for the heterogeneous, dynamic environment of the Internet where peers have different upload and download capacities and may join or leave the system at random [13]. The price to pay is a random, hardly predictable performance. It is the objective of the present paper to better understand the performance trade-offs that can be achieved by unstructured, epidemic live streaming systems.

1.1 Related work

The performance of unstructured P2P systems critically depends on the peer/chunk selection algorithm used for the chunk transmission between any two peers. These algorithms may be broadly categorized as *push* or *pull*, depending on whether it is the *sender* or the *receiver* that does the selection, respectively. Push-based schemes are more suitable for upload-constrained systems, which is representative of peers connected through ADSL or cable for instance, since the dissemination of chunks is then regulated by the sender, as a function of its upload capacity. Pull-based schemes, on the other hand, are more appropriate for download-constrained systems, since the rate of chunk requests adapts to the download capacity of each peer. These allow peers to request those chunks that are closest to their playback deadline, for instance.

It turns out that unstructured P2P systems are hard to analyze, due to the strong interaction between peers imposed by the peer/chunk selection algorithm. Most performance studies rely either on measurements [4, 8], simulations [19] or experiments [11, 12, 16, 22]. Analytical results, that are key to the design of efficient, robust diffusion schemes, concern either structured systems [7, 15, 18, 10] or unstructured pull-based systems [20, 21, 24]. Very few theoretical results exist in the practically interesting case of unstructured push-based systems.

Notable exceptions are the paper by Massoulié et. al. [14], showing the rate optimality of the so-called *most deprived peer, random useful chunk* algorithm, and that by Sanghavi, Hajek and Massoulié [17], showing the delay optimality of the *random peer, latest blind chunk* algorithm. 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 poor due to the random peer selection. To our knowledge, no unstructured P2P diffusion scheme has yet been proved both rate and delay optimal.

1.2 Contributions

The main contribution of the paper is to prove that the so-called *random peer, latest useful chunk* algorithm can achieve dissemination at an optimal rate and within an optimal delay, up to an additive constant term. This qualitative result suggests that epidemic live streaming algorithms can achieve near-unbeatable rates and delays.

Another key result of the paper is the delay optimality of the *random peer, latest blind chunk* algorithm when combined with source coding. Such a diffusion scheme is known to achieve a diffusion rate of only $1 - e^{-1}$ in the critical regime where the source speed is equal to the upload speed [17]. It is thus necessary to add some redundancy to the original signal to allow the peers to recover from chunk losses. We show that the additional delay due to the coding/decoding scheme can be controlled (that is, made be equal to O(1)) by bounding the correlation of successive missing chunks.

Using mean-field approximations, we also derive recursive formulas for the diffusion function of two schemes, the above *random peer, latest blink chunk* algorithm as well as the so-called *latest blind chunk, random useful peer* algorithm.

Finally, we provide simulation results that validate the above theoretical results and allow us to compare the performance of various practically interesting diffusion schemes in terms of delay, rate, and control overhead. In particular, we identify several peer/chunk selection algorithms that achieve near-optimal rate/delay performance trade-offs. Moreover, we show that the control overhead needed to implement these algorithms may be reduced by restricting the neighborhood of each peer without substantial performance degradation.

1.3 Outline

The paper is structured as follows. In Section 2 we define the live streaming model and the schemes that we consider in the paper. Section 3 is devoted to the theoretical results on delay optimality. The recursive formulas are described in Section 4. In Section 5 we give the simulation results comparing the performance of the considered schemes under various network conditions. Section 6 concludes the paper.

2. MODEL

2.1 Source rate and capacity constraints

Consider a P2P live streaming system consisting of one source and N peers. The source creates a sequence of chunks, numbered $1, 2, 3, \ldots$, at rate λ, and sends each chunk to one of the N peers, chosen uniformly at random. The dissemination of each chunk to the N peers is then achieved by the peers themselves.

Let V be the set of peers. For any $u \in V$, we denote by $s(u)$ the upload speed of peer u. This is the maximum number of chunks that u can *send* per time unit. For simplicity, we assume that there is no constraint on the number of chunks that each peer can *receive* per time unit.

By convention, the average upload speed corresponds to the transmission of one chunk by time unit, so that:

$$\frac{1}{N} \sum_{u \in V} s(u) = 1.$$

We say that the system is in *underload* regime if $\lambda < 1$, in *critical* regime if $\lambda = 1$ and in *overload* regime if $\lambda > 1$. Clearly, some peers receive only a fraction of the chunks sent by the source in the overload regime. Nevertheless, peers may successfully decode the original audio or video streaming signal if some redundancy has been added to this signal and is included in the chunks sent by the source. Thus all three regimes are of practical interest.

2.2 Push-based diffusion schemes

We shall focus on *push-based* diffusion schemes where the transmission of a chunk between two peers is initiated by the sender, which is the natural choice for the considered upload capacity constrained system. We assume each peer has only a partial knowledge of the overall system. This is represented as a directed graph $G = (V, E)$ where $(u, v) \in E$ if and only if u knows v, for all $u, v \in V$ (we say that v is a neighbor of u). Each peer can only send chunks to one of its neighbors.

For any $u \in V$, let $C(u)$ be the collection of chunks that peer u has received. We denote by $\mathcal{C}$ the set of possible collections of chunks owned by a peer. A push-based scheme is formally described as a (possibly random) mapping from $V \times \mathcal{C}^N$ to $V \times \mathcal{C}$ that gives for any sender peer u, as a function of the collections of chunks $C(v)$ of its neighbors v, the destination peer and the chunk $c \in C(u)$ to be sent.

Push-based schemes may be broadly categorized into two classes depending on whether the destination peer or the chunk is selected first. In this paper, we shall restrict the analysis to the following peer and chunk selection schemes:

Random peer: The destination peer is chosen uniformly at random among the neighbors of u;

Random useful peer: The destination peer is chosen uniformly at random among those neighbors v of u such that $C(u) \setminus C(v) \neq \emptyset$. When the chunk c is selected first, the choice of the destination peer is restricted to those neighbors v of u such that $c \notin C(v)$;

Most deprived peer: The destination peer is chosen uniformly at random among those neighbors v of u for which $|C(u) \setminus C(v)|$ is maximum. When the chunk c is selected first, the choice of the destination peer is restricted to those neighbors v of u such that $c \notin C(v)$;

Latest blind chunk: The sender peer u chooses the most *recent* chunk (that is, the chunk of highest index) in its collection $C(u)$;

Latest useful chunk: The sender peer u chooses the most recent chunk c in its collection $C(u)$ such that $c \notin C(v)$ for at least one of its neighbors v. When the destination peer v is selected first, c is the most recent chunk in the set $C(u) \setminus C(v)$.

Random useful chunk: The sender peer u chooses uniformly at random a chunk c in its collection $C(u)$ such that $c \notin C(v)$ for at least one of its neighbors v. When the destination peer v is selected first, c is chosen uniformly at random in the set $C(u) \setminus C(v)$.

A rich class of push-based schemes follows from the combination of these peer/chunk selection algorithms. Those considered in the paper are summarized in Table 1 below. Figure 1 gives an example of peer/chunk selection under these schemes. Note that, for this particular example, the latest chunk of the sender peer has already been received by all its neighbors. The transmission capacity of the sender peer is then wasted in this state under the *lb/up* and *lb/rp* schemes, since a peer will receive two or more copies of the same chunk.

Table 1: Some push-based diffusion schemes.

Notation	Scheme
rp/lb	random peer, latest blind chunk
rp/lu	random peer, latest useful chunk
dp/lu	most deprived peer, latest useful chunk
dp/ru	most deprived peer, random useful chunk
lb/rp (= *rp/lb*)	latest blind chunk, random peer
lb/up	latest blind chunk, useful peer
lu/up	latest useful chunk, useful peer
lu/dp	latest useful chunk, most deprived peer

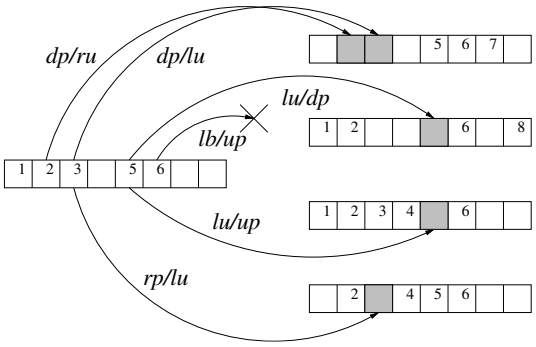


Figure 1: Peer/chunk selection of a sender peer (left) under the considered push-based schemes.

In the following, we assume that time is slotted so that the transfer of any chunk by any peer takes exactly one time slot. The source sends $\lfloor \lambda \rfloor$ chunks per time slot, plus one additional chunk with probability $\lambda - \lfloor \lambda \rfloor$, corresponding to an arrival rate λ. Note that for $\lambda < 1$, the source sends chunks according to a Bernoulli process. Similarly, peer u sends $\lfloor s(u) \rfloor$ chunks per time slot, plus one additional chunk with probability $s(u) - \lfloor s(u) \rfloor$, corresponding to an average upload speed of $s(u)$.

We assume that at each slot, each peer has a perfect knowledge of the state of its target peer, including the intended transmissions of other peers to the same target peer. In particular, all conflicts are solved at the beginning of each slot, prior to the chunk transmission. The impact of imperfect knowledge resulting in transmissions of the same chunk to the same target peer will be analyzed for the example of the *lb/ru* scheme in Section 4.

2.3 Diffusion rate, diffusion delay

The rate/delay performance trade-off achieved by each scheme is evaluated through the diffusion function r, where $r(t)$ is the probability that it takes no more than t time slots for an arbitrary chunk created by the source to reach an arbitrary peer. Equivalently, $r(t)$ is the fraction of peers that receive any given chunk no later than t time slots after its creation, averaged over all chunk transmissions.

The diffusion function has the typical S-curve illustrated by Figure 2. We refer to the asymptotic value of $r(t)$ as t tends to infinity as the *diffusion rate*. This corresponds to the average fraction of chunks received by an arbitrary peer; equivalently, this is the average fraction of peers that eventually receive any given chunk. The maximum diffusion rate is equal to 1 if $\lambda \leq 1$ and to $1/\lambda$ in the overload regime $\lambda > 1$.

Another key performance metric is the *diffusion delay*, which is defined as the delay it takes for an arbitrary chunk to reach a fraction $1 - \varepsilon$ of the peers that will eventually receive that chunk, where ϵ is an arbitrary, small constant. We take $\varepsilon = 5\%$ in the simulation results of Section 5. In the homogeneous case where $s(u) = 1$ for all peers u, the population of peers that have received any given chunk at most doubles every time unit, so that the minimum diffusion delay is of order $\log_2(N)$ (cf. Theorem 1 below for a more precise formulation); it may be less in heterogeneous cases (in particular, if the upload capacity is concentrated on a few peers only).

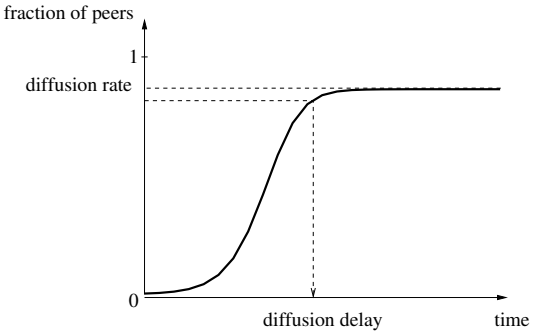


Figure 2: Performance metrics associated with the diffusion function: diffusion rate and diffusion delay.

There is a natural trade-off between rate and delay. The diffusion rate is typically maximized by a homogeneous dissemination of chunks among peers, irrespective of the age of these chunks. This is achieved for instance by the *dp/ru* scheme that has indeed been proved rate-optimal for a complete graph in the underload regime, assuming the source selects the most deprived peer [14]. We shall see that the corresponding diffusion delay is actually far from optimal.

To minimize the diffusion delay, priority should be given to the transmission of the latest chunk rather than to the homogeneous dissemination of chunks among peers. The *rp/lb* scheme for instance has been proved delay-optimal in the reference scenario of a complete graph with homogeneous distribution of upload capacity, in both the critical and overload regime [17]. The price to pay is a sub-optimal diffusion rate, growing to $1 - e^{-1}$ in the critical regime as the number of peers N tends to infinity, due to the reception of multiple copies of the same chunk by some peers.

It is in fact unclear whether there exist push-based diffusion schemes that are *both* rate- and delay-optimal. A key result of the paper, proved in Section 3, is that the *rp/lu* scheme can achieve dissemination at an optimal rate and within an optimal delay, up to an additive constant term, for the reference scenario in the underload regime. Note that the question remains open in the general case. We use simulations in Section 5 to compare the rate/delay tradeoffs achieved by the schemes of Table 1 in various scenarios, including the three load regimes described above, heterogeneous upload capacity distributions and uncomplete graphs.

2.4 Implementation issues

There are of course a number of other key parameters that should be considered in the design of efficient P2P live streaming systems. One of them is the *overhead* that is generated by the exchange of control messages between peers to maintain at each peer a fresh view of the collections of chunks owned by its neighbors. The considered push-based schemes differ significantly in this respect, ranging from the *rp/lb* scheme that generates virtually no overhead (it is sufficient for each peer to know its neighbors) to the *dp/lu* scheme that generates a lot of control messages (it is necessary for each peer to know the current collections of chunks owned by all its neighbors).

The impact of overhead on the rate/delay performance of a diffusion scheme could be quantified. This would require a complete model of the P2P system, however, including the transmission of control messages, whose frequency and content play a critical role. We prefer to model transmissions of chunks only, assuming peers have a perfect knowledge of the collections of chunks owned by their neighbors. We shall simply compare qualitatively the overheads of the considered schemes and analyze the impact of some approaches to reducing that overhead, by restricting the neighborhood of each peer, on the efficiency of the chunk dissemination.

Another key parameter is related to *source coding*. In practical P2P live streaming systems, the source contains some redundancy in the original signal to recover from chunk losses or delivery beyond the playback delay. The design of the source coding scheme should then depend on the rate/delay trade-off achieved by the diffusion scheme. Thus, since $r(t)$ corresponds to the fraction of chunks received by an arbitrary peer after t time slots, the source speed must be at least $1/r(t)$ that of the original signal, where t is the playback delay, to allow the peers to successfully decode the signal. Of course, many other parameters should be taken into account. These include the fairness of the chunk dissemination among peers, the statistical characteristics of successive chunks received by any given peer, and the additional delay introduced by the source coding/decoding algorithms. An example of joint design of diffusion scheme and source coding scheme will be given in Section 3.

Finally, practical diffusion schemes should be robust to *cheating* and to *selfish behavior*. Those schemes for which a peer can improve its reception rate or delay by sending false information about its state may rapidly collapse. Peers should also be encouraged to upload chunks at the highest possible speed, using for instance some form of *tit-for-tat* mechanism similar to that of Bit-Torrent [6]. These issues are not addressed in this paper but offer interesting perspectives for future work.

3. DELAY OPTIMALITY

We shall now state the theoretical results on the delay optimality of the *random peer, latest blind chunk* scheme and the *random peer, latest useful chunk* scheme. Under the former, each peer simply sends the latest chunk it has, irrespective of whether the target peer has it or not; under the latter, it sends the latest among the chunks that it has and that the target peer has not yet received.

At each slot, a new chunk is created at the source with probability $\lambda \in (0, 1]$ and sent to one of the N peers chosen uniformly at random. We assume a complete overlay graph and a homogeneous upload capacity distribution, so that the transmission of any chunk takes one slot. We denote by D the delay it takes for any particular chunk to reach a randomly selected peer after its creation by the source.

3.1 General lower bound

We first give a lower bound on D, valid for any diffusion scheme. Specifically, we prove that $D \geq \log_2(N) + O(1)$, where $O(1)$ is a random variable uniformly bounded in N:

THEOREM 1. *For any diffusion scheme, we have:*

$$\Pr(D \leq t) \leq \frac{2^t}{N}, \quad \forall t \geq 0. \tag{1}$$

In particular, we have for all $m > 0$:

$$\Pr(D \leq \log_2(N) - m) \leq 2^{-m}. \tag{2}$$

Proof. Let Y_t denote the number of peers that have a particular chunk t time slots after its creation at the source, assuming some diffusion scheme. Necessarily, $Y_{t+1} \leq 2Y_t$ irrespective of the diffusion scheme, since each peer has unit upload capacity. Since $Y_0 = 1$, we get $Y_t \leq 2^t$ for all $t \geq 0$. Now the probability that a randomly selected peer has the chunk at time t is equal to $\mathrm{E}(Y_t/N)$, from which (1) follows. We deduce equation (2) by taking $t = \lfloor \log_2(N) \rfloor - m$ in (1). □

3.2 Random peer, latest blind chunk

We now consider the *rp/lb* scheme. This scheme has been shown to achieve the optimal delay of $\log_2(N)$ in the critical regime $\lambda = 1$, at the cost of a sub-optimal rate growing to $1 - e^{-1}$ as N tends to infinity [17]. In particular, some redundancy must be added to the original stream so as to recover from the loss of chunks. The efficiency of the decoding scheme then critically depends on the distribution of those chunks received by any given peer. We prove that under the *rp/lb* scheme, each chunk is actually received by any peer with probability at least:

$$q = 1 - e^{-1/10}$$

by time $\log_2(N)$, independently of other chunks and peers, for sufficiently large N. We shall discuss the consequence of this result on the delay performance of a simple system that combines source coding and the *rp/lb* scheme. The result turns out to be also a key step in the proof of the joint rate-delay optimality of the *rp/lu* scheme considered below.

In the sequel, we assume that the system starts at time $t = 0$ and denote by A_t the arrival process of chunks at the source: $A_t = 1$ if a chunk is created by the source at time t and $A_t = 0$ otherwise. For all $u \in V$, we define $X_{t,t'}(u) = 1$ if the chunk created at time t has already been received by peer u at time t' and $X_{t,t'}(u) = 0$ otherwise. By convention,

we let $X_{t,t'}(u) = 1$ for all $t' \geq t$ if $A_t = 0$, i.e., no chunk was created by the source at time t. Finally, let:

$$T = \lfloor \log_2(N) \rfloor - 1. \quad (3)$$

The following result is proved in Appendix A:

THEOREM 2. *For any time interval $[a, b)$, consider the following event:*

$$\mathcal{E} = \{\forall u \in V,\ \forall t \in [a, b),\ X_{t,t+T}(u) \geq Z_t(u)\},$$

where $Z_t(u)$ are i.i.d. Bernoulli random variables of mean q. Then the probability of event $\mathcal{E}$ is at least $1 - N^{-\alpha}$, for any $\alpha \in (0, 1)$ and sufficiently large N.

Now consider a P2P system that combines source coding and the *rp/lb* push scheme. Assume that the source receives original chunks at a rate of $q - \epsilon$ per time unit, where ϵ is some small positive parameter. Select then some window size W and assume that at time nW, the source creates W encoded chunks from the $(q - \epsilon)W$ original data chunks it has received over time interval $[(n-1)W, nW)$, using erasure codes, so that this original data can be recovered from any $(q - \epsilon)W$ chunks out of the collection of W encoded chunks. Finally, let the source inject these W encoded chunks one by one over the time interval $[nW, (n + 1)W)$.

Then the injection rate of encoded chunks is $\lambda = 1$. Thus the theorem entails that for sufficiently large N, with high probability, the number of chunks received by any given peer over time interval $[nW + T, (n + 1)W + T)$ is at least a binomial random variable with parameters (W, q). The probability p that this exceeds $W(q - \epsilon)$, and hence that decoding can be performed, tends to 1 exponentially fast as W increases. We have thus established that a simple system combining source coding and the *rp/lb* scheme can achieve diffusion at rate $q - \epsilon$ with arbitrarily small loss probability in $T + W = \log_2(N) + O(1)$ time slots.

It is worth noting that this system is delay-optimal but far from rate-optimal, since the original streaming rate of the source cannot exceed $1 - e^{-1/10} \approx 0.1$. By an adaptation of the proof given in Appendix A, it may actually be proved that Theorem 2 is still valid for a delay of $\lfloor (1 + \epsilon) \log_2(N) \rfloor$ and a probability $q = 1 - e^{-1} - \epsilon$, where ϵ is an arbitrary positive number. Thus a simple P2P system that combines source coding and the *rp/lb* scheme can in fact achieve diffusion at rate up to $1 - e^{-1} \approx 0.63$ in $(1 + \epsilon) \log_2(N) + O(1)$ time slots. The improvement in diffusion rate, at the expense of higher delays, illustrates the various performance trade-offs that may be achieved by different combinations of source coding and diffusion schemes.

3.3 Random peer, latest useful chunk

Finally, we state the main result of the paper, showing the joint rate-delay optimality of the *rp/lu* scheme:

THEOREM 3. *Assume that $\lambda < 1$. There exists a constant $\gamma > 0$ such that for all $m \geq 1$:*

$$\Pr(D \geq \log_2(N) + m) \leq \frac{\gamma}{m}, \quad (4)$$

for sufficiently large N.

Equivalently, this result states that the transmission delay D of any chunk to any peer is equal to $\log_2(N) + O(1)$ time slots, where the additive $O(1)$ term is a random variable, bounded in probability uniformly in N. This delay is optimal in view of Theorem 1. The proof of Theorem 3 is given in Appendix B.

4. RECURSIVE FORMULAS

In this section, we derive recursive formulas for the epidemic diffusion function of the *latest blind chunk, random peer* and the *latest blind chunk, random useful peer* schemes through mean-field approximations. Under the former, each peer simply sends the latest chunk it has to a randomly chosen peer; under the latter, it sends the latest chunk it has to a randomly chosen peer among those peers that have not yet received this chunk, if any.

We consider the reference scenario with complete graph and homogeneous upload capacity distribution. We assume that $\lambda \leq 1$; the overload regime $\lambda > 1$ is considered in §4.4. The number of peers N is assumed to be sufficiently large so that the system may be considered in the mean-field regime where peers are mutually independent. We further assume that, for any given peer u, the event that a chunk belongs to the collection $C(u)$ of chunks owned by u is independent of the event that any other chunk belongs to $C(u)$. The validity of the derived formulas will be assessed by comparison with simulations in Section 5.

4.1 Latest blind chunk, random peer

We first consider the *lb/rp* scheme. Recall that $r(t)$ corresponds to the average fraction of peers that receive any given chunk no later than t time slots after its creation. Without any loss of generality, we assume that some tagged chunk is created at time $t = 0$ and that the system is in steady state at that time. Since the source sends each new chunk to a randomly chosen peer, we have $r(1) = 1/N$. Now at any time $t \geq 1$, the tagged chunk is the latest of the collection owned by an arbitrary peer u with probability:

$$p(t) = r(t) \prod_{k=1}^{t-1} (1 - \lambda r(k)). \quad (5)$$

This follows from the independence assumption, noting that for all $k = 1, 2, \ldots, t$, $r(k)$ is the probability that a chunk created at time $t - k$ is in the collection $C(u)$ of chunks owned by peer u at time t.

Due to the random peer selection strategy, the number of copies of the tagged chunk that are received by an arbitrary peer at time $t + 1$ is a binomial random variable with parameters $(N - 1, p(t)/(N - 1))$. For large N, this can be approximated by a Poisson random variable with mean $p(t)$. Thus the probability that an arbitrary peer receives at least one copy of the tagged chunk at time $t + 1$ is approximately equal to $1 - e^{-p(t)}$. A fraction $1 - r(t)$ of the peers that receive the chunk at time $t + 1$ actually need it. We deduce the recursive formula:

$$r(t + 1) = r(t) + (1 - e^{-p(t)})(1 - r(t)), \quad t \geq 1, \quad (6)$$

where $p(t)$ is given by (5).

4.2 Latest blind chunk, random useful peer

We now consider the *lb/ru* scheme. The only difference with the *lb/rp* scheme is that all transfers are useful as long as some peers need the considered chunk. This gives the recursion:

$$r(t + 1) = r(t) + \min(p(t), 1 - r(t)), \quad t \geq 1, \quad (7)$$

where $p(t)$ is given by (5).

4.3 Delayed updates

As explained in 2.4, some control messages are needed to maintain a fresh view of the collection of chunks owned by each peer. Delaying some control messages reduce the overhead but may impact the performance of the system. We model such delayed updates by assuming that peers know the state of system in the previous slot, but are not aware of the ongoing transfers of the current slot. Therefore, collisions can occur even under the *lb/ru* scheme when several peers send the same chunk to the same target peer.

Consider as in §4.1 the diffusion of the chunk created at time $t = 0$. A fraction $1 - r(t)$ of the N peers has not yet received this chunk at time t. Thus the number of copies of this chunk that are received by one of these $N(1-r(t))$ peers at time $t+1$ is a binomial random variable with parameters $(N-1, p(t)/N(1-r(t)))$, where $p(t)$ is given by (5). For large N, this can be approximated by a Poisson random variable with mean $p(t)/(1-r(t))$. Thus the probability that a peer that has not yet received the considered chunk at time t receives at least one copy of this chunk at time $t+1$ is approximately equal to $1 - e^{\frac{-p(t)}{1-r(t)}}$. We deduce the recursive formula:

$$r(t+1) = r(t) + (1-r(t))(1 - e^{\frac{-p(t)}{1-r(t)}}), \quad t \geq 1. \qquad (8)$$

4.4 Overload regime

In the overload regime, $\lfloor\lambda\rfloor$ new chunks are created by the source at each slot, plus one additional chunk with probability $\lambda - \lfloor\lambda\rfloor$. The diffusion processes of these $\lfloor\lambda\rfloor$ or $\lfloor\lambda\rfloor + 1$ chunks will interfere in the diffusion process. We number these chunks as $1, 2, \ldots, \lfloor\lambda\rfloor$ (or $\lfloor\lambda\rfloor + 1$), where chunk 1 corresponds to the last created chunk. Thus chunk 1 has priority over chunk 2, chunk 2 over chunk 3, and so on.

Now let r_i be the diffusion function associated with a chunk of index i. Again, we assume that some tagged chunk of index i is created at time $t = 0$ and that the system is in steady state at that time. At any time $t \geq 1$, this chunk is the latest of the collection owned by an arbitrary peer u if u has got it and hasn't got any fresher chunk. This happens with probability:

$$p_i(t) = r_i(t) \prod_{j=1}^{i-1} (1 - r_j(t)) \times \prod_{k=1}^{t-1} \left((1 - (\lambda - \lfloor\lambda\rfloor) r_{\lceil\lambda\rceil}(k)) \prod_{j=1}^{\lfloor\lambda\rfloor} (1 - r_j(k)) \right). \qquad (9)$$

There are now $\lceil\lambda\rceil$ recursive formulas, one per diffusion function r_i. These can be deduced from (6), (7), (8) by replacing the functions r and p by r_i and p_i, respectively, for each considered diffusion scheme.

The global diffusion function follows by averaging:

$$r(t) = \frac{1}{\lambda} \left((\lambda - \lfloor\lambda\rfloor) r_{\lceil\lambda\rceil}(t) + \sum_{i=1}^{\lfloor\lambda\rfloor} r_i(t) \right). \qquad (10)$$

5. SIMULATION RESULTS

In this section, we validate the above theoretical results and evaluate the rate/delay performance trade-offs achieved by the push-based diffusion schemes of Table 1 by means of simulations.

Unless otherwise specified, results are derived for $N = 600$ homogeneous peers with a complete graph, which corresponds to an optimal diffusion delay of $\log_2(N) \approx 9$ slots. Chunks that arrive more than 50 slots after their creation are not taken into account, which is representative of a real live streaming system with limited playback delay. In particular, the diffusion rate is approximated by the value of the diffusion function $r(t)$ at time $t = 50$.

5.1 Reference scenario

We first consider a reference scenario that consists of a complete graph with a common upload speed $s(u) = 1$ for all peers u. Figure 3 shows the diffusion functions in the critical regime $\lambda = 1$.

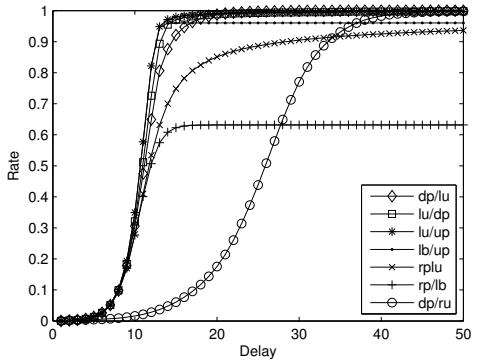
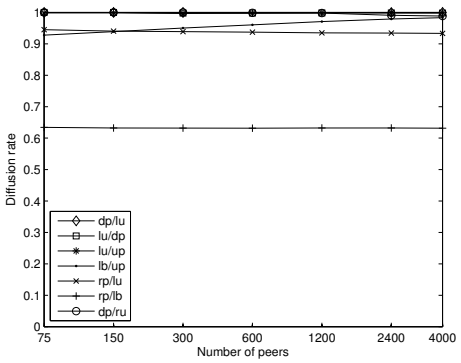


Figure 3: Diffusion in the reference scenario.

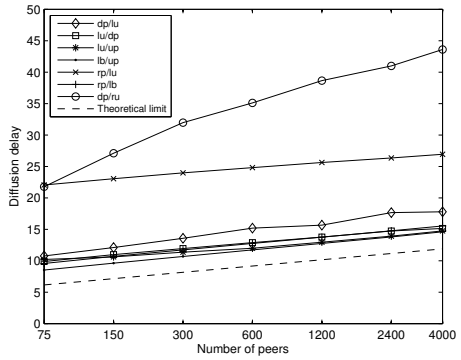
The first four schemes clearly outperform the three others. Among these four top schemes, the *dp/lu* tends to be slower than the three others, which is a consequence of the priority given to the peer selection over the chunk selection. The performance of the last two schemes is good regarding either rate or delay but not both, as announced in §2.3. Finally, the high delay suffered by the *rp/lu* scheme may be surprising in view of the delay optimality of this scheme stated in Theorem 3. This is because the theoretical result is not valid in the considered critical regime. Moreover, we shall see in §5.3 that the additional constant delay predicted by Theorem 3 is significant even in the underload and overload regimes, as soon as the source speed λ is close to 1.

5.2 Impact of the number of peers

We now let the number of peers N vary from 75 to 4000. The results are shown in Figure 4. The diffusion rate is constant for all schemes but the *lb/up* and *dp/ru* schemes. For the *lb/up* scheme it increases with N, which suggests the asymptotic rate optimality of this scheme. As expected, the diffusion rate of *rp/lb* is equal to $1 - e^{-1}$. The *rp/lu* scheme, where the last useful chunk is selected, achieves a rate close to 0.93. All schemes but the *dp/ru* scheme have an optimal diffusion delay of $\log_2(N) + O(1)$, which shows the good scalability of these schemes. The additional constant is significant for the *rp/lu* scheme (around 25 slots), moderate for the *dp/lu* scheme (between 5 and 10 slots), slight for the other schemes (less than 5 slots). Finally, the *dp/ru* scheme has poor delay performance, which is a consequence of the random chunk selection and induces a decrease of its rate for large N (some chunks are received after the maximum delay of 50 slots).

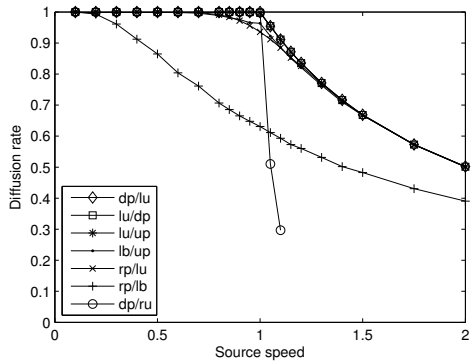


(a) Rate

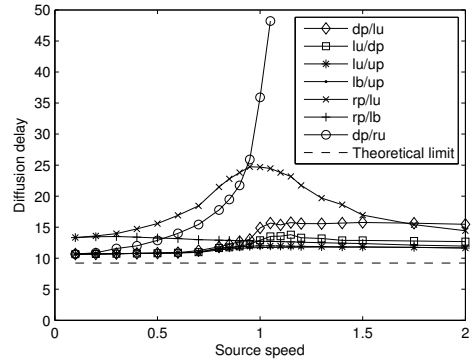


(b) Delay

Figure 4: Impact of the number of peers.



(a) Rate



(b) Delay

Figure 5: Impact of source speed.

5.3 Impact of source speed

We now analyze the impact of the upload speed λ. Results are shown in Figures 5(a)-5(b) when λ varies from 0 to 2, for $N = 600$ peers. Observe that the *rp/lu* scheme has poor delay performance not only in the critical regime $\lambda = 1$ but in all regimes close to critical, as announced. This means that the additional constant delay given in Theorem 3 is far from negligible. The *rp/lb* scheme achieves a diffusion rate close to $1-e^{-1/\lambda}$ in low delay, as expected [17]. The performance of the other schemes is nearly optimal for both rate and delay, except for the *dp/ru* and *dp/lu* schemes that behave poorly in overload regime. Note that the *dp/ru* scheme doesn't reach any steady state, which is a consequence of the random chunk selection coupled with the fact that each peer receives at most a fraction $1/\lambda$ of the chunks.

5.4 Validation of the recursive formulas

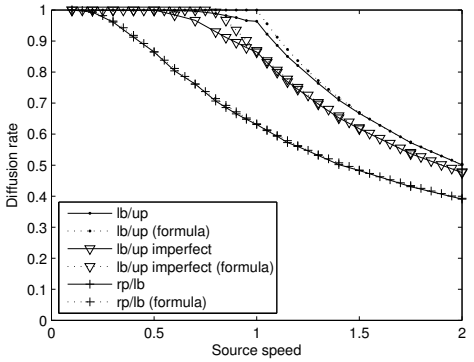
We now validate the mean-field approximation used to derive the recursive formulas of Section 4. Figures 6(a)-6(b) compare the diffusion rate and diffusion delay obtained by analysis and by simulation, in the scenario of §5.3. The formulas are quite accurate for both rate and delay. The most significant difference concerns the *rp/lb* scheme, where the formula overestimates the delay for $\lambda \approx 0.3$ by 1.5 slots (corresponding to an error of 10%). Regarding the *lb/up* scheme, the delay estimation is very good but the formula slightly overestimates the rate for λ close to 1 (the error is less than 4%). Finally, the formula of the *lb/up* scheme with imperfect knowledge slightly overestimates both delay and rate for $\lambda \approx 0.8$ (error less than 6% for both metrics). Interestingly, these anomalies occur at the maximum source speed λ for which the diffusion rate is very close to 1; this is due to the fact that the fraction of peers that need any given chunk at time t, approximated by $1 - r(t)$, becomes hard to estimate in this specific regime.

5.5 Heterogeneous upload capacities

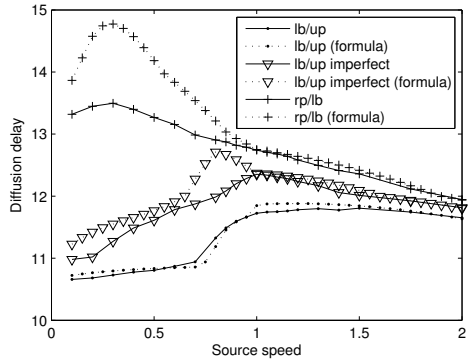
So far we have considered the case of homogeneous upload capacities only. The practically interesting case of heterogeneous upload capacities is much less well understood. Recall that, of all considered strategies, the *dp/ru* scheme is the only one for which optimality results exist for heterogeneous upload capacities (it is known to be rate-optimal).

In order to investigate the impact of heterogeneity, we consider, for any fixed parameter $h \in [0, 1]$, a scenario where $s(u) = 2$ for a fraction $\frac{1}{3}h$ of the peers, $s(u) = 0.5$ for another fraction $\frac{2}{3}h$ of the peers, and $s(u) = 1$ for all other peers. The average upload capacity is therefore still equal to 1. We refer to h as the factor of *heterogeneity*: $h = 0$ corresponds to the homogeneous case, and the upload variance grows linearly with h. Figures 7(a)-7(b) show the diffusion rate and the diffusion delay of the considered schemes as a function of the heterogeneity factor in the critical regime ($\lambda = 1$).

Observe that the performance of the top three schemes

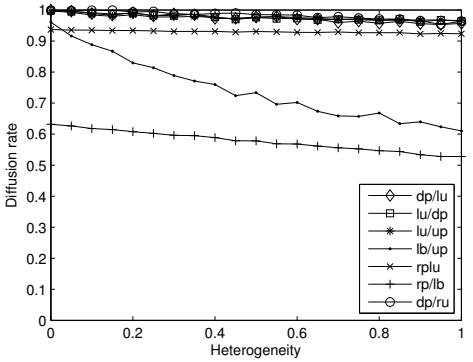


(a) Rate

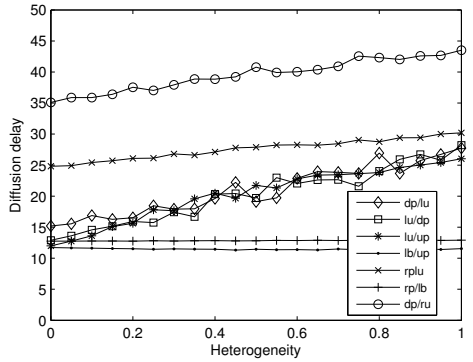


(b) Delay

Figure 6: Validation of the recursive formulas.



(a) Rate



(b) Delay

Figure 7: Diffusion as a function of heterogeneity.

worsens with h, for both rate and delay. In particular, the diffusion delay approximately doubles when h grows from 0 to 1. The impact is less significant for the *rp/lu* scheme: the diffusion rate remains approximately unchanged, while the diffusion delay increases by 25%. Regarding the *rp/lb* and *lb/up* schemes, the diffusion delay is almost insensitive to heterogenity, but the diffusion rate is strongly impacted, especially for the latter that looses about 35%.

5.6 Restricted neighborhoods

As mentioned in §2.4, a complete overlay graph is hard to implement in practice because of overhead issues. We propose to investigate three simple ways to bypass this issue:

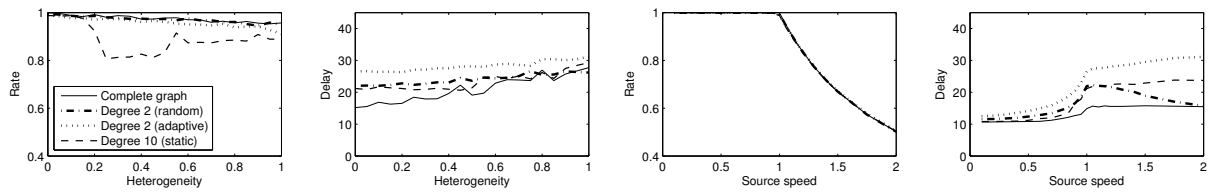
Static graph: The graph G is an Erdös-Rényi graph with an average degree of 10, that ensures connectivity with high probability for the considered set of $N = 600$ peers. The graph remains the same during the whole diffusion process.

Random graph: For each chunk transmission, the sender peer selects uniformly at random two peers among the $N-1$ other peers; the diffusion scheme then applies to these two potential target peers. Note that the graph is now dynamic.

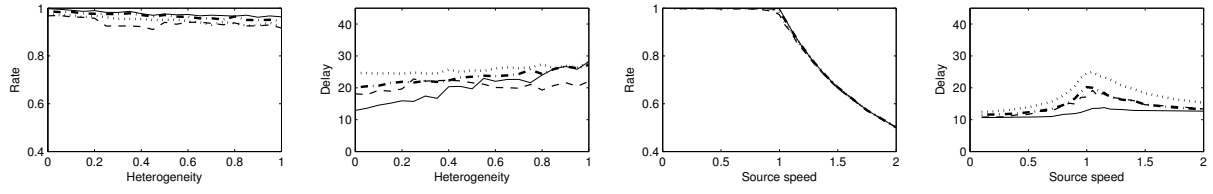
Adaptive graph: For each chunk transmission, the sender peer keeps track of the last target peer and select uniformly at random another peer among the $N-2$ other peers; again, the diffusion scheme then applies to these two potential target peers. Note that this technique is somewhat reminiscent of the "optimistic unchoking" used by BitTorrent [6].

The results are shown in Figure 8, in the scenarios of §5.3 for the source speed λ and §5.5 for the heterogeneity factor h. The same instance of the Erdös-Rényi graph is used for all plots. We observe that for most diffusion schemes, this static restriction of neighborhood strongly reduces the diffusion rate. This is particularly true for the *dp/lu* and *dp/ru* schemes in heterogeneous cases. The chaotic nature of the diffusion rate as a function of heterogeneity is due to the fact that results are derived for each value of h from a different distribution of upload capacities over the nodes of the Erdös-Rényi graph. This exemplifies the sensitivity of the most deprived peer selection scheme to the network structure.

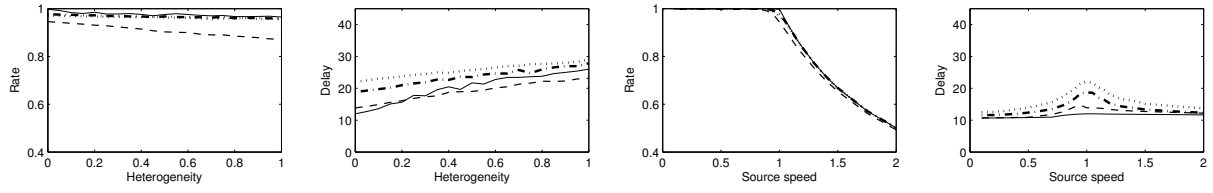
The adaptive neighborhood, on the other hand, increases the diffusion delay of most schemes. It turns out that the basic random graph approach, where the sender peer selects two potential target peers at random, achieves the best trade-off. The performance degradation is slight in most cases compared to the complete graph. In particular, the top three schemes have very good performance, even in the worst case of heterogeneous networks in the critical regime.



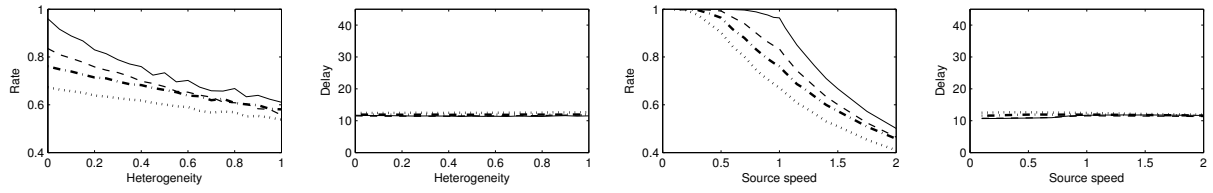
(a) dp/lu



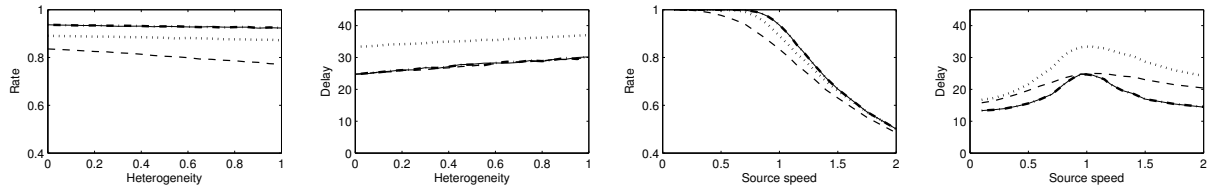
(b) lu/dp



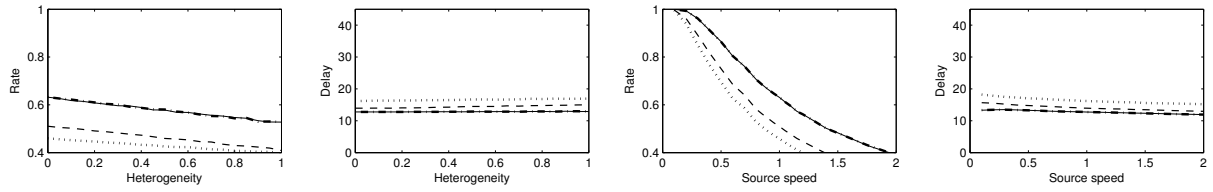
(c) lu/up



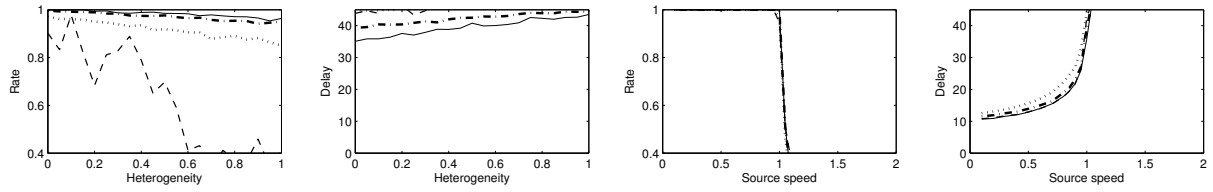
(d) lb/up



(e) rp/lu



(f) rp/lb



(g) dp/ru

Figure 8: Impact of restricted neighborhoods on performance.

6. CONCLUSION

We have analyzed the rate/delay performance trade-offs of various push-based diffusion schemes. A key result of the paper is that joint rate and delay optimality can be achieved by these epidemic-style algorithms, whereas such optimality results were known for structured systems only.

By simulation, we have identified some good diffusion schemes like *dp/lu*, *lu/dp*, *lu/up* and *lb/up*, and provided an explicit formula for the diffusion function of the latter. These are strong practical contenders for a live streaming system. Some key implementation issues remain open, however, like the joint design of the source coding and diffusion schemes, the building and evolution of the overlay graph, the frequency and size of control messages and the robustness to cheating and selfish behavior.

7. REFERENCES

[1] TVants, http://tvants.en.softonic.com/.

[2] Sopcast, http://www.sopcast.com/.

[3] UUsee inc., http://www.uusee.com/.

[4] S. Ali, A. Mathur, and H. Zhang. Measurement of commercial Peer-to-Peer live video streaming. In *Workshop in recent advances in Peer-to-Peer streaming*, 2006.

[5] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in cooperative environments. In *Symposium on Operating System principles (SOSP 2003)*, Bolton Landing, NY, October 2003.

[6] B. Cohen. Incentives build robustness in BitTorrent. Technical report, bittorrent.org, 2003.

[7] G. Dan, V. Fodor, and I. Chatzidrossos. On the performance of multiple-tree-based Peer-to-Peer live streaming. In *INFOCOM*, 2007.

[8] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross. A measurement study of a large-scale P2P IPTV system. In *IEEE Transactions on Multimedia*, 2007.

[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 *Proc. of IPTV Workshop, International World Wide Web Conference*, 2006.

[10] R. M. Karp, A. Sahay, E. E. Santos, and K. E. Schauser. Optimal broadcast and summation in the logp model. In *ACM Symposium on Parallel Algorithms and Architectures*, 1993.

[11] X. Liao, H. Jin, Y. Liu, L. Ni, and D. Deng. AnySee: Peer-to-Peer live streaming. In *INFOCOM*, 2006.

[12] T. Locher, R. Meier, S. Schmidt, and R. Wattenhofer. Push-to-pull Peer-to-Peer live streaming. In *DISC*, 2007.

[13] N. Magharei, R. Rejaie, and Y. Guo. Mesh or multiple-tree: A comparative study of live p2p streaming approaches. In *INFOCOM*, 2007.

[14] L. Massoulié, A. Twigg, C. Gkantsidis, and P. Rodriguez. Randomized decentralized broadcasting algorithms. In *INFOCOM*, 2007.

[15] J. Mundinger, R. Weber, and G. Weiss. Optimal scheduling of Peer-to-Peer file dissemination. In *Journal of Scheduling (to appear)*, 2007.

[16] F. Pianese, D. Perino, J. Keller, and E. Biersack. Pulse: an adaptative, incentive-based, unstructured p2p live streaming system. In *IEEE Transaction on Multimedia*, 2007.

[17] S. Sanghavi, B. Hajek, and L. Massoulié. Gossiping with multiple messages. In *INFOCOM*, 2007.

[18] T. Small, B. Liang, and B. Li. Scaling laws and tradeoffs in Peer-to-Peer live multimedia streaming. In *ACM Multimedia*, 2006.

[19] A. Vlavianos, M. Iliofotou, and Faloutsos. BiTos: Enhancing BitTorrent for supporting streaming applications. In *9th IEEE Global Internet Symposium 2006*, April 2006.

[20] M. Zhang, Y. Xiong, Q. Zhang, and Q. Yang. Optimizing the throughput of data-driven peer-to-peer streaming. In *Lecture Notes in Computer Science*, volume 4351, 2007.

[21] M. Zhang, Q. Zhang, L. Sun, and S. Yang. Understanding the power of pull-based streaming protocol: Can we do better? In *IEEE JSAC, special issue on Advances in Peer-to-Peer Streaming Systems*, 2007.

[22] M. Zhang, L. Zhao, Y. Tang, J. Luo, and S. Yang. Large-scale live media streaming over Peer-to-Peer networks through global Internet. In *Workshop on advances in Peer-to-Peer multimedia streaming*, 2005.

[23] X. Zhang, J. Liu, B. Li, and T. Yum. Coolstreaming/donet : A data-driven overlay network for Peer-to-Peer live media streaming. In *INFOCOM*, 2005.

[24] Y. Zhou, D. Chiu, and J. Lui. A simple model for analysis and design of P2P streaming protocols. In *IEEE ICNP*, October 2007.

APPENDIX

A. PROOF OF THEOREM 2

Let us first establish the following intermediate result.

LEMMA 1. *Let $\beta \in (0,1)$ be a fixed constant. The number of attempts η to push any given chunk during the T time slots following its creation verifies:*

$$\Pr(\eta \geq N/9) \geq 1 - 2N^{-\beta}, \tag{11}$$

for sufficiently large N.

Proof. Consider the classical gossip process, described as follows. There is a population of N users. In each round, each infected user selects uniformly at random another user, which becomes infected after its being selected.

Denoting by Y_t the number of infected individuals after t rounds, starting with $Y_0 = 1$, we now establish bounds on the values of Y_t that strengthen the bounds provided in [17]. Let $\bar{Y}_t$ denote the expected value of Y_t.

To this end, we first recall the following result from [17]:

LEMMA 2. *For any $t \geq 0$, let:*

$$G(y) = \mathrm{E}(Y_{t+1}|Y_t = y).$$

We have:

$$G(y) = y + (N - y)[1 - (1 - 1/N)^y].$$

Moreover, the function G is non-decreasing, and verifies:

$$2y(1 - y/N) \leq G(y) \leq 2y. \tag{12}$$

This yields the following corollary:

COROLLARY 1. *For any $t \geq 0$, we have the following bounds:*

$$2^t \left[1 - \frac{2^t}{N}\right] \leq \bar{Y}_t \leq 2^t. \tag{13}$$

Proof. Since $Y_0 = 1$ and $Y_t \leq 2Y_{t-1}$ for all $t \geq 1$, we have $Y_t \leq 2^t$ for all $t \geq 0$, from which the second inequality in (13) follows.

The first inequality in (13) is obvious when $2^t > N$. Let us thus assume that $2^t \leq N$. Write then

$$\bar{Y}_t = \mathrm{E}[G(Y_{t-1})].$$

Thus, in view of (12),

$$\mathrm{E}[2Y_{t-1}(1 - Y_{t-1}/N)] \leq \bar{Y}_t.$$

Using the fact that $Y_{t-1} \leq 2^{t-1}$ almost surely, this entails

$$2\bar{Y}_{t-1}(1 - 2^{t-1}/N) \leq \bar{Y}_t.$$

By induction, this further implies:

$$2^t \prod_{n=0}^{t-1} (1 - 2^n/N) \leq \bar{Y}_t.$$

By the inequality $(1-x)(1-y) \geq 1-x-y$, valid for any two numbers $x, y \in [0,1]$, the previous inequality implies that:

$$\bar{Y}_t \geq 2^t \left[1 - N^{-1} \sum_{n=0}^{t-1} 2^n\right].$$

The desired inequality follows. □

The argument detailed in [17, Proof of Theorem 5] implies the following result:

LEMMA 3. *Let $\beta \in (0,1)$ be a fixed constant. There exists some $\epsilon > 0$ such that the number of attempts η to push any given chunk in the T time slots following its creation verifies:*

$$\Pr(\eta \geq \bar{Y}_T - 2^T N^{-\epsilon}) \geq 1 - 2N^{-\beta}. \tag{14}$$

We are now ready to complete the proof of Lemma 1. In view of (3) and (13), one has:

$$\begin{aligned} \bar{Y}_T - 2^T N^{-\epsilon} &\geq 2^T[1 - (1/N)2^T - N^{-\epsilon}] \\ &\geq (N/4)[1 - 1/2 - N^{-\epsilon}] \\ &\geq N/9, \end{aligned}$$

for sufficiently large N. Combined with (14), this yields the desired result (11). □

The proof of Theorem 2 proceeds as follows. For each time slot $t \in [a,b)$, we consider the event $\mathcal{B}_t$ that the chunk created at time t has been pushed at least M_t times by time $t+T$, where M_t is a Poisson random variable M_t with mean $N/10$. By Chernoff's inequality, the probability that $M_t > N/9$ is exponentially small in N. It then follows from Lemma 1 that:

$$\Pr(\mathcal{B}_t) \geq 1 - 3N^{-\beta}, \tag{15}$$

for sufficiently large N. Since each target peer is chosen uniformly at random, the total number of selections of a given peer u among the first M_t attempts to push a chunk generated at time t is a Poisson random variable with mean $1/10$. Furthermore, these Poisson random variables are mutually independent across peers and across chunks. Let $Z_t(u) = 1$ if the corresponding Poisson random variable is positive, and $Z_t(u) = 0$ otherwise. Since

$$\Pr(Z_t(u) = 1) = 1 - e^{-1/10},$$

these variables are distributed as specified in Theorem 2. Finally, note that on the event $\cap_{t\in[a,b)}\mathcal{B}_t$, each peer u has received each chunk generated at time t by time $t+T$ provided $Z_t(u) = 1$. The probability of this event is, by the union bound, at least:

$$1 - 3(b-a)N^{-\beta} = 1 - \frac{3(b-a)}{N^{\beta-\alpha}} N^{-\alpha}.$$

The proof follows by choosing $\beta > \alpha$. □

B. PROOF OF THEOREM 3

We first give an informal description of the argument before providing the details. Our aim is to find a constant $\gamma > 0$ such that for any $m \geq 1$, the probability of a given peer not receiving a given chunk after $T + m$ time slots is bounded by γ/m.

To this end, we shall consider, for some arbitrary time slot a and some constant $\delta \in (0,1)$, the number of chunks created over the time interval $[a,b)$, with $b - a = \lceil \delta m \rceil$, and not yet received by some arbitrary peer at time $a' = b+T$. We shall show that over the time interval $[a',b')$, with $b' - a' = m - \lceil \delta m \rceil$, with probability $1 - O(1/m)$, all such originally missing chunks must have been received at the exception of at most ℓ missing chunks, for some fixed ℓ. By a monotonicity argument, the probability that a chunk created at time a is not received by time b' is at most $\ell/(b-a) + O(1/m) = O(1/m)$. Thus the probability that a chunk is not yet received $b' - a = T + m$ slots after its creation is $O(1/m)$, which is the announced result.

To establish the intermediate result, we consider for any given set $\mathcal{L}$ of ℓ chunks created over $[a,b)$ the number $M(\mathcal{L})$ of peers who made push attempts during $[a',b')$ towards the considered peer, say peer v, while these pushers did hold some chunk with time stamp in $\mathcal{L}$. We show that with probability $1 - O(1/m)$, one such opportunity must have been used to provide the target peer v with one of these chunks, by showing that $M(\mathcal{L})$ is larger than the number of later chunks that can be provided to v during $[a',b')$.

This number of later chunks is bounded from above by the quantity $I + J$, where I and J are defined as follows: I denotes the number of chunks created over $[a, b' - T)$; J denotes the number of chunks created over $[b' - T, b')$ and received by peer v. Thus the main step consists in showing that with probability $1 - O(1/m)$, for all $\mathcal{L} \subset [a,b)$ such that $|\mathcal{L}| = \ell$, it holds that

$$M(\mathcal{L}) > I + J. \tag{16}$$

We now provide the detailed arguments. For any given $\lambda < 1$, define:

$$\delta = \frac{1-\lambda}{7}. \tag{17}$$

Select $\ell > 0$ such that

$$\Pr(B \geq 1) \geq 1 - \delta, \tag{18}$$

where B is a binomial random variable with parameters (ℓ, q), with q as defined in Theorem 2. It can be readily

checked that the following choice will do:

$$\ell = \left\lceil \frac{\log(1/\delta)}{q} \right\rceil.$$

Consider the event $\mathcal{E}$ defined in Theorem 2, and the corresponding random variables $Z_t(u)$. Recall that on $\mathcal{E}$, at time $a' = b + T$, peer u holds the chunk created at time $t \in [a, b)$ whenever $Z_t(u) = 1$. For a given subset $\mathcal{L} \subset [a, b)$, let:

$$K(\mathcal{L}) = \sum_{u \in U} \max_{t \in \mathcal{L}} Z_t(u) \times \mathbf{1}_{\{u \text{ contacts } v \text{ during } [a', b')\}}.$$

Clearly, on the event $\mathcal{E}$, it holds that $K(\mathcal{L}) \leq M(\mathcal{L})$.

We now introduce the following event:

$$\mathcal{F} = \mathcal{E} \cap \mathcal{I} \cap \mathcal{J} \cap \mathcal{K},$$

where

$$\mathcal{I} = \{I \leq (\lambda + \delta)m\}, \quad \mathcal{J} = \{J \leq \delta m\},$$

and

$$\mathcal{K} = \{\forall \mathcal{L} \subset [a, b),\ |\mathcal{L}| = \ell, \quad K(\mathcal{L}) \geq (1 - 4\delta)m\}.$$

On the event $\mathcal{F}$, inequality (16) must hold for all size-ℓ subsets $\mathcal{L}$ of $[a, b)$. Indeed, necessarily

$$M(\mathcal{L}) \geq K(\mathcal{L}) \geq (1 - 4\delta)m,$$

and, in view of (17),

$$I + J \leq (\lambda + 2\delta)m = (1 - 5\delta)m.$$

Let us show that event $\mathcal{F}$ has probability $1 - O(1/m)$. To this end, note first that I follows a binomial distribution with parameters (m, λ); hence, by Chernoff's inequality, the probability of event $\mathcal{I}$ verifies:

$$\Pr(\mathcal{I}) = \Pr(I \leq (\lambda + \delta)m) \geq 1 - e^{-\theta m}, \tag{19}$$

for some positive constant θ which depends on λ and δ only. To bound the probability of event $\mathcal{J}$, note that, using an argument similar to that in the proof of Theorem 1, the random variable J verifies:

$$\mathrm{E}[J] \leq \frac{1}{N} \sum_{n=0}^{T} 2^n \leq 1.$$

Hence, Chebitchev's inequality entails that:

$$\Pr(\mathcal{J}) = \Pr(J \leq \delta m) \geq 1 - \frac{1}{\delta m}. \tag{20}$$

Finally, the random variable $K(\mathcal{L})$ is distributed as a Binomial random variable with parameters (N, p), with

$$p = \Pr(B \geq 1) \times (1 - (1 - 1/N)^{m - \lceil \delta m \rceil}).$$

The first term is the probability that $Z_t(u) = 1$ for some $t \in [a, b)$, for some arbitrary peer u; the second term is the probability that an arbitrary peer contacts v during $[a', b')$. In view of (18), the probability p satisfies $p \geq p'$ for sufficiently large N, with:

$$p' = (1 - 3\delta)\frac{m}{N}.$$

Denoting by K' a Binomial random variable with parameters (N, p'), we obtain for sufficiently large N,

$$\begin{aligned} \Pr(\mathcal{K}) &= \Pr(\forall \mathcal{L} \subset [a, b),\ |\mathcal{L}| = \ell,\ K(\mathcal{L}) \geq (1 - 4\delta)m\}), \\ &\geq \Pr(K' \geq (1 - 4\delta)m)^{\binom{b-a}{\ell}}, \\ &\geq 1 - \binom{b-a}{\ell} e^{-\kappa m}, \end{aligned} \tag{21}$$

where the last inequality follows from Chernoff's inequality, for some positive constant κ which depends on δ only.

Put together, inequalities (19)-(21) and Theorem 2 imply

$$\Pr(\mathcal{F}) \geq 1 - N^{-\alpha} - e^{-\theta m} - \frac{1}{\delta m} - \binom{b-a}{\ell} e^{-\kappa m}, \tag{22}$$

for sufficiently large N.

Now it is easily seen that the following monotonicity property holds, for an arbitrary peer u, any time slots t, t' such that $t < t'$:

$$X_{t,t'}(u) \geq_d X_{t+1,t'}(u).$$

In words, the older the chunk, the more likely it has already been received by peer u. This provides the first inequality in the following:

$$\begin{aligned} \Pr(X_{a,b'}(v) = 0) &\leq \frac{1}{b-a} \sum_{t \in [a,b)} \Pr(X_{t,b'}(v) = 0) \\ &\leq \frac{1}{b-a} \sum_{t \in [a,b)} \Pr(X_{t,b'}(v) = 0 | \mathcal{F}) + \Pr(\overline{\mathcal{F}}) \\ &\leq \frac{\ell}{b-a} + \Pr(\overline{\mathcal{F}}), \end{aligned}$$

where $\overline{\mathcal{F}}$ is the complement of event $\mathcal{F}$. The last inequality follows because on event $\mathcal{F}$, the number of chunks created over $[a, b)$ and not yet received by peer v is no larger than ℓ. In view of (22), we get:

$$\Pr(X_{a,b'}(v) = 0) \leq \frac{\ell}{b-a} + N^{-\alpha} + e^{-\theta m} + \frac{1}{\delta m} + \binom{b-a}{\ell} e^{-\kappa m}.$$

Thus, provided $m \leq N^{\alpha}$, and using $b - a = \lceil \delta m \rceil$, we obtain:

$$\Pr(X_{a,b'}(v) = 0) \leq \frac{\gamma}{m},$$

where γ is the constant given by:

$$\gamma = \left\lceil \frac{\ell + 1}{\delta} + 1 + \sup_{m \geq 1} \left(m e^{-\theta m} + m \lceil m\delta \rceil^{\ell} e^{-\kappa m} \right) \right\rceil.$$

This constant only depends on λ since so do δ, ℓ, θ and κ, cf. (17), (18), (19) and (21). □