P2P Live Streaming in a Dynamic Setting

Riku Lääkkölä

August 26, 2010

Abstract

Peer-to-peer streaming has been researched extensively in the past years, but mostly in models considering a static number of peers. In this paper we study the performance of certain previously presented push-based sharing mechanisms in a dynamic network of peers. The results show that the schemes that were simulated perform remarkably well in dynamic settings, even when the peers spend abnormally short times in the system.

1 Introduction

Streaming high definition video or other bandwidth consuming material in a large scale service creates enormous server requirements. Utilizing Peer-to-peer (P2P) technologies might enable us to circumvent these requirements at the cost of the upload bandwidth of the users. This makes P2P streaming technologies an interesting area of research.

Several different aspects of P2P streaming have been studied extensively in the past years, but most of the research concentrates on systems that expect a "flash crowd" type user behaviour. A system, where users constantly come and go hasn't received much attention. In this paper, we implement and apply a set of push-based P2P streaming mechanisms to a simulated dynamic network of peers.

1.1 Previous work

Different types of P2P live streaming mechanisms have been lately studied in a completely static setting [1] and in settings involving flash crowd behaviour [2], where the peers all arrive during a short starting period. A system, where some of the peers disconnect, or "churn", in the middle of streaming, have also been simulated [3, 4], but previous work on systems, where peers also *arrive* at random is rather rare.

1.2 Outline

The structure of the paper is the following. In section 2 we supplement the model presented by Bonald et al. [1] and in Section 3 we present the results gathered by simulation. In Section 4 we present the conclusions.

2 Model

2.1 Static model

The foundation for our work lies in the static model defined by Bonald et al. [1] and this subsection depicts their definition. The model consists of a source that creates numbered chunks at a rate λ_s, which means, that $\lfloor \lambda_s \rfloor$ chunks are created per time slot, and one additional chunk is created with probability $\lambda_s - \lfloor \lambda_s \rfloor$. The source sends the chunks one at a time into a network of N peers, to peers chosen at random. The goal is that, after the source has sent a chunk to a peer, the chunk gets disseminated to all the peers in the network as fast as possible by the peers themselves.

Each peer u in the network V has a limited upload capacity $s(u)$, which is the maximum number of chunks u can send per time unit (or time *slot*). The download capacity of the peers is left unconstrained. The average upload capacity of the peers in the network is 1, i.e.,

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

Also, unless specified otherwise, we consider the homogeneous case, where $\forall\ u \in V : \ s(u) = 1$. When the source rate is below the average upload capacity, i.e., $\lambda_s < 1$, the system is in underload regime. Respectively, when $\lambda_s > 1$, the system is in overload regime. When $\lambda_s = 1$ we say, that the system is in critical regime.

A number of push-based diffusion schemes by Bonald et al. [1] were implemented, and these schemes are defined as follows. We consider the neighbourhood of each peer in the network to be the entire network. Thus, all the peers are connected to each other via a single hop. If we denote the collection of chunks of peer $u \in V$ by $C(u)$ and the all the possible collections of chunks of a peer by $\mathcal{C}$, we can formally define a push-based scheme to be a mapping $V \times \mathcal{C}^N \mapsto V \times \mathcal{C}$ that gives a peer u the destination peer and the chunk $c \in C(u)$ to be sent. We consider the following schemes (unaltered from [1]):

Random peer (*rp*): The destination peer is chosen uniformly at random among the neighbours of u.

Random useful peer (*up*): The destination peer is chosen uniformly at random among those neighbours $v \in V$ 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 neighbours v such that $c \notin C(v)$.

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

Latest useful chunk (*lu*): 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 neighbours v. When the destination peer v is selected first, c is the most recent chunk in the set $C(u) \setminus C(v)$.

A push-based scheme comprises of the combination of a peer choice and a chunk choice. The schemes that were simulated are referred to, according to the aforementioned abbreviations, as *rp/lb*, *rp/lu*, *lb/up* and *lu/up*. Whichever is chosen first, the receiver or the chunk, is denoted by the order accordingly. For example, as *rp* is a peer selection scheme, in *rp/lu* the peer is chosen first, then the chunk.

Time in the system is discretized into slots which correspond to a single time unit. All events taking place in a single time slot are considered simultaneous. We consider each peer to be aware of the sending plans of all other peers in the current slot. This means that in the more intelligent (i.e., less random) schemes, there should be no wasted sending attempts, as the peers may adjust their choices according to the situation of the current slot.

The peers have a buffer of 50 time slots, and thus chunks older than 50 slots are considered old and are discarded. The main characteristics of each scheme are the *diffusion rate*, which designates the average number of peers a chunk has reached after 50 slots of diffusion, and *diffusion delay*, which is the average number of slots a chunk has to spend in the system until it has reached 95% of its diffusion rate.

2.2 Dynamic model

We wanted to study a streaming service where users arrive and leave at a constant rate instead of a flash crowd behaviour, where all the users arrive when the program starts almost simultaneously. In our model, the peers arrive and leave according to an $M/G/\infty$-model. In simulation terms this means that in each time slot n peers arrive according to a Poisson process with rate λ_p, and the peers have independent and identically distributed service times (rounded to the nearest integer). In the simulations we used the exponential distribution with mean $\frac{1}{\mu_p}$. Thus, after the system has reached steady state, the mean number of peers in the system is $\bar{N}_p = \frac{\lambda_p}{\mu_p}$. Consequently, it also holds that

$$\bar{N}_p \sim \text{Poisson}\left(\frac{\lambda_p}{\mu_p}\right),$$

at least in a continuous-time model. In the discrete model it is a good approximation.

3 Results

The simulations were carried out with a custom C program created specifically for this purpose. This section discusses the results of the simulations and the figures presented in the appendices. The main attention is on the attributes diffusion rate and diffusion delay. Unless otherwise noted, the mean number of peers in the system is 600. For the most part, different scenarios were simulated for a time interval during which the source sends 1000 chunks, as no significant difference in results were found in longer test simulations. The computing times of different schemes varied from a few seconds (mostly for *rp/lb* and *rp/lu*) to a half an hour (for the more computing intensive schemes).

3.1 Critical regime

With a source rate of 1 and homogeneous upload capacities among peers, we simulated systems of mean sizes of $\bar{N}_p = 600$ and $\bar{N}_p = 60$ peers. The dynamics of the systems were varied by choosing 5 different mean service times $1/\mu_p = \{6000, 600, 60, 6, 3\}$ and adjusting the arrival rate to get a system with the correct mean number of peers. For comparison, the diffusion graphs in a static setting are presented in figure A1, and the results seem entirely identical to previous research [1]. The results for dynamic networks are visualized in figures A2–A6.

Instead of visualizing the diffusion rate only at the 50 slot mark the figures present the diffusion of the chunk at each point of its life. Also the fraction of chunks that have been able to spread (in the available 50 slots) to a certain number of peers is visualized.

The most interesting point visible in the results is that at reasonable levels of dynamics, the schemes do not seem to degrade almost at all compared to the static setting. Even at extreme levels of dynamics ($1/\mu_p = 3$ slots), the schemes seem to function at a reasonable rate.

3.2 Effects of source rate

A system of 600 peers was simulated with a varying source rate λ_s from 0.1 to 1.9. The results are shown in figure B1. In an almost static $\lambda_p = 0.1$, and thus $1/\mu_p = 6000$ system, the results seem very similar to earlier results for static systems [1].

An interesting effect seems to arise in more dynamic settings in underload regime ($\lambda_s < 1$): The diffusion rate becomes larger than 1 with sufficiently small λ_s. In the static setting this would never happen, as we scale the rate with the number of peers. On the other hand, in the dynamic setting the diffusion rate is calculated by dividing by the **mean** number of peers, when actually an individual chunk "sees" more than the mean amount of individual peers during its lifetime. In underload regime the chunks have more time to spread, as there are fewer new chunks to take precedence over them. This taken into account, the results seem again differ very little from the results for a static system.

3.3 Effects of heterogeneity

We define the *heterogeneity* of the system to be the variance in the upload capacities of the peers in the network. Similarly to [1], in our system, the heterogeneity is portrayed by the property $h \in [0, 1]$, which is tied to the peers' upload capacities in the following manner: a fraction $\frac{1}{3}h$ of the peers have an upload capacity of 2, a fraction $\frac{2}{3}h$ a capacity of 0.5 and the rest $(1 - h)$ a capacity of 1. Thus, increasing h also increases the variance in the peers' upload capacities.

Figure C1 visualizes the effects of increasing heterogeneity in three different dynamic settings. In the least dynamic setting the results are again almost similar to the static setting, and in the more dynamic settings the differences between the schemes under observation seem to diminish. The major effect seems to be that as the dynamics increase, the performance of the rp/lu scheme drops below that of the lb/up.

3.4 Effects of the number of peers

The effect of the mean number of peers in the system was studied by fixing the peer arrival rate ($\lambda_p = 1$) and varying the service time ($1/\mu_p$) between simulations. The results illustrated in figure D1 show us that the dynamics have again very little or no effect in the overall performance of the diffusion schemes in comparison with the static setting.

4 Conclusion

We analyzed the effect of $M/G/\infty$-type peer dynamics in four push-based diffusion schemes in contrast with the performance in a static setting. The main finding is that adding peer arrivals and departures does not have a significant effect on the performance of the schemes in question, at least at reasonable rates of dynamics.

Various points of a dynamic setting are however still left open, like the performance from the point of view of the peer. In these results the analysis is based on the chunk point of view, and it would be interesting to simulate a setting that enables the analysis of the user's perceived service quality.

Acknowledgement

This work was supported by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT)

References

[1] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: optimal performance trade-offs," in *SIGMETRICS '08: Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, (New York, NY, USA), pp. 325–336, ACM, 2008.

[2] Z. Liu, Y. Shen, K. W. Ross, S. S. Panwar, and Y. Wang, "Layerp2p: Using layered video chunks in p2p live streaming," *Multimedia, IEEE Transactions on*, vol. 11, pp. 1340–1352, August 2009.

[3] R. J. Lobb, A. P. Couto da Silva, E. Leonardi, M. Mellia, and M. Meo, "Adaptive overlay topology for mesh-based p2p-tv systems," in *NOSSDAV '09: Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video*, (New York, NY, USA), pp. 31–36, ACM, 2009.

[4] C. Feng and B. Li, "On large-scale peer-to-peer streaming systems with network coding," in *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, (New York, NY, USA), pp. 269–278, ACM, 2008.

Appendix

A Figures: Critical regime

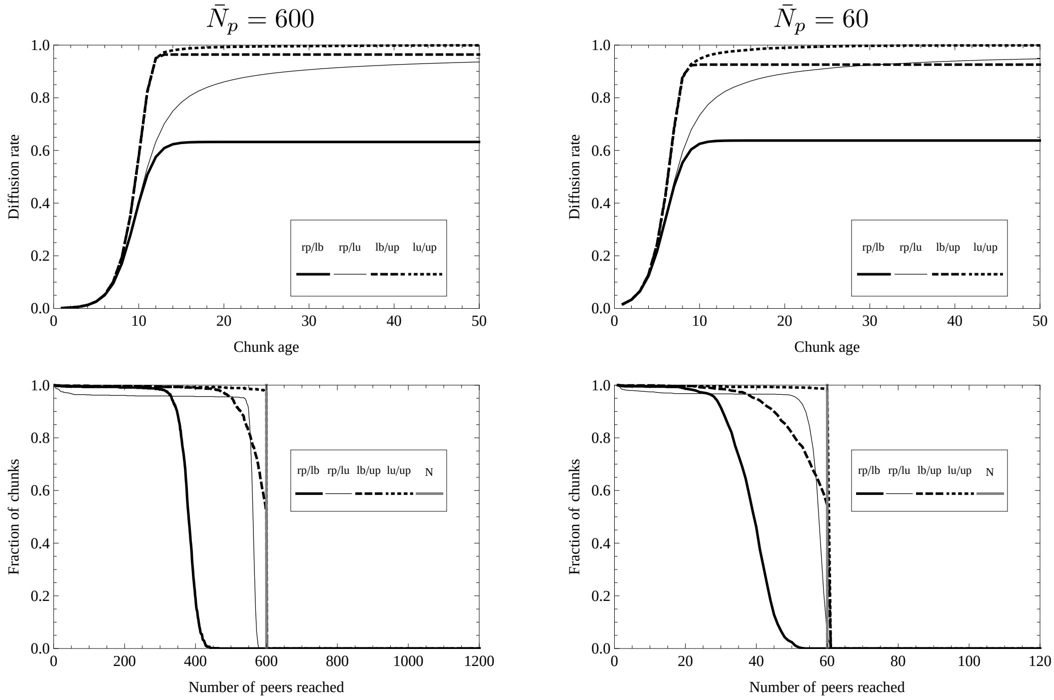

Figure A1: Diffusion graphs of the policies in a static setting.

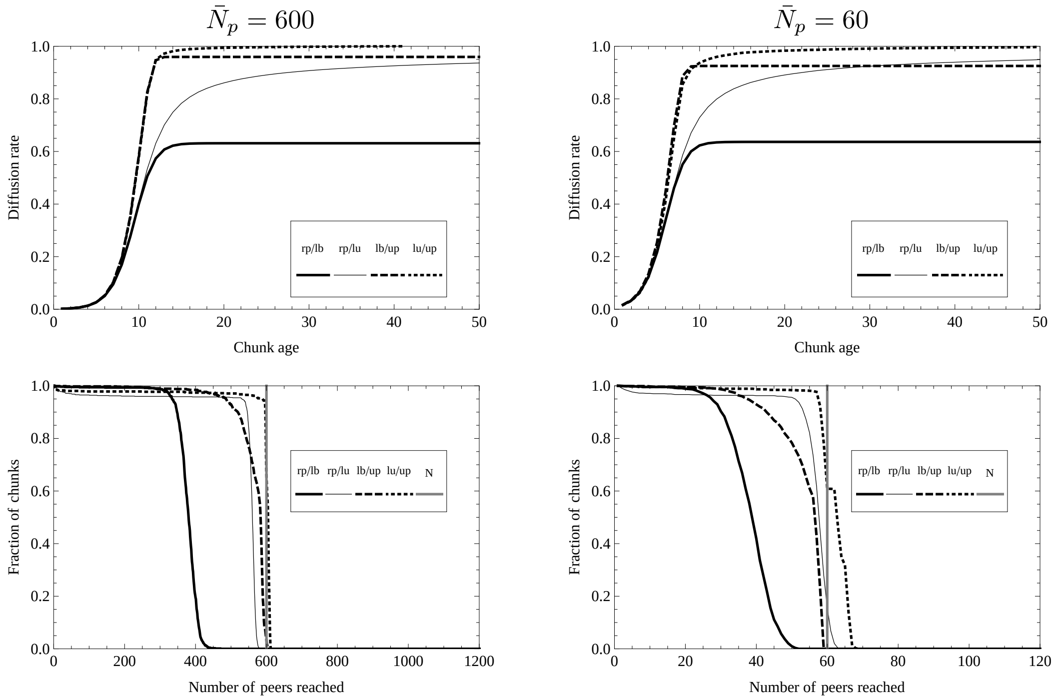

Figure A2: Diffusion graphs of the policies in a dynamic ($\frac{1}{\mu_p} = 6000$) setting.

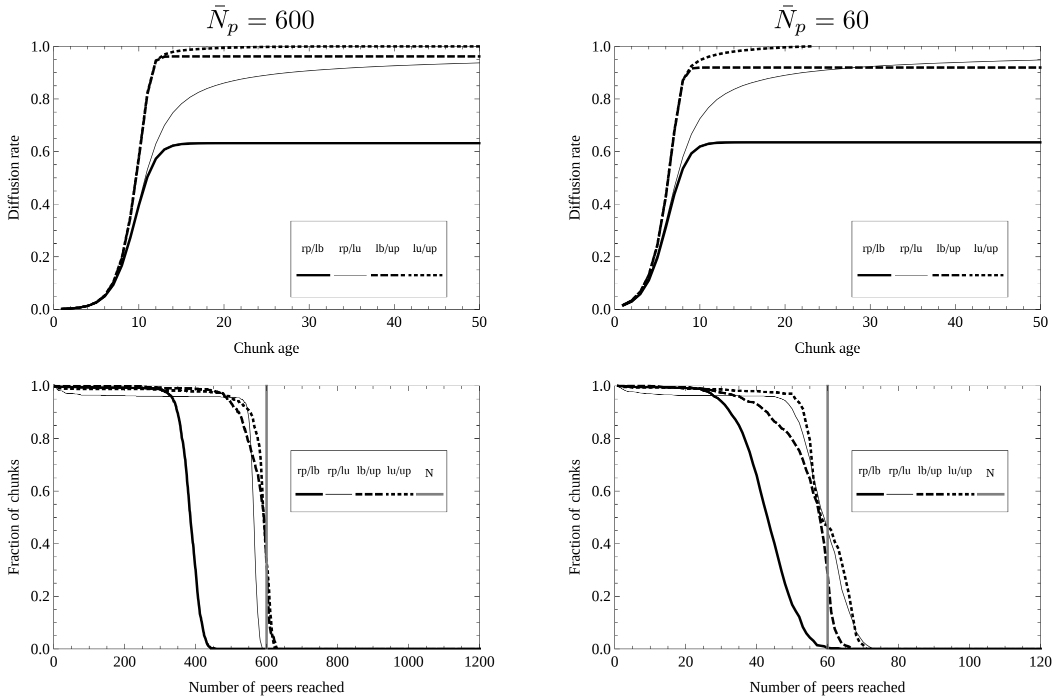

Figure A3: Diffusion graphs of the policies in a dynamic ($\frac{1}{\mu_p} = 600$) setting.

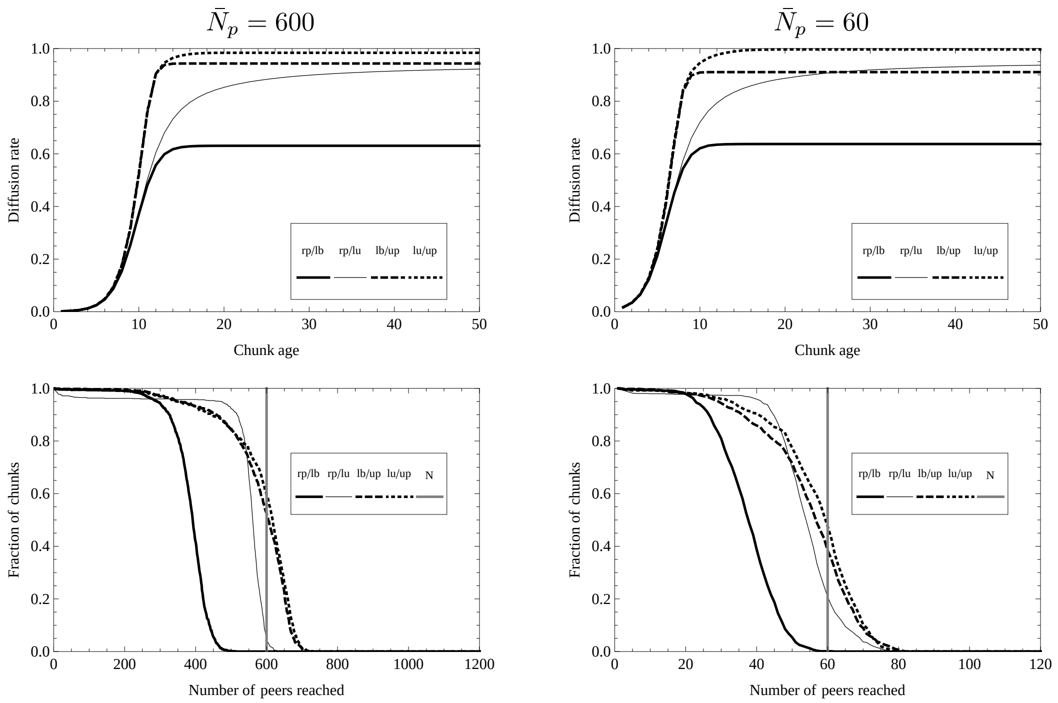

Figure A4: Diffusion graphs of the policies in a dynamic ($\frac{1}{\mu_p} = 60$) setting.

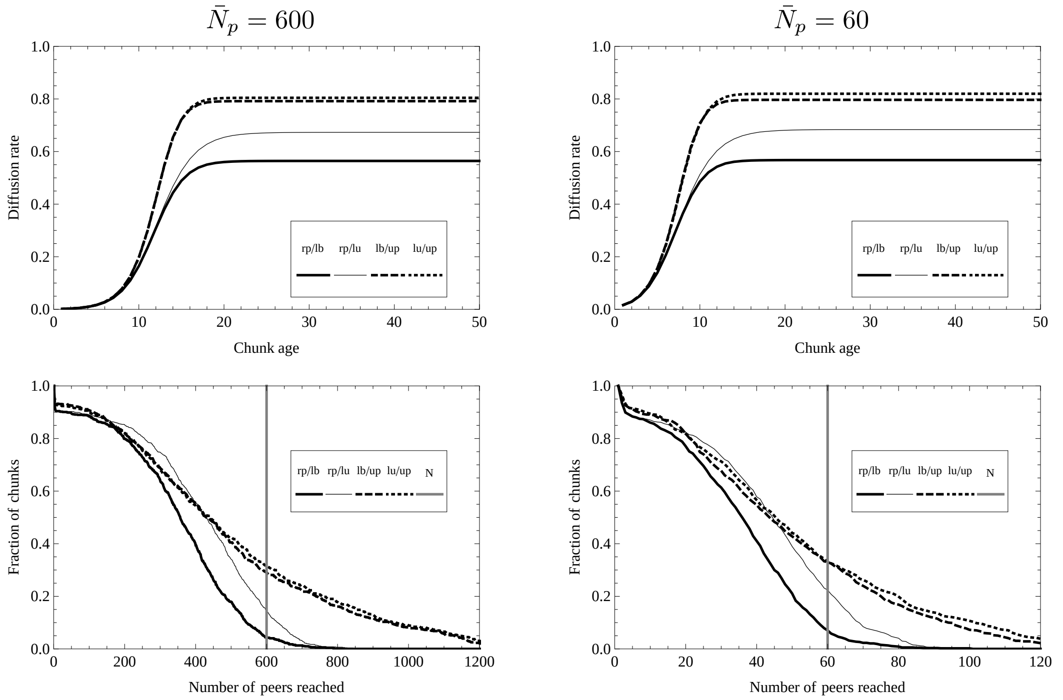

Figure A5: Diffusion graphs of the policies in a dynamic ($\frac{1}{\mu_p} = 6$) setting.

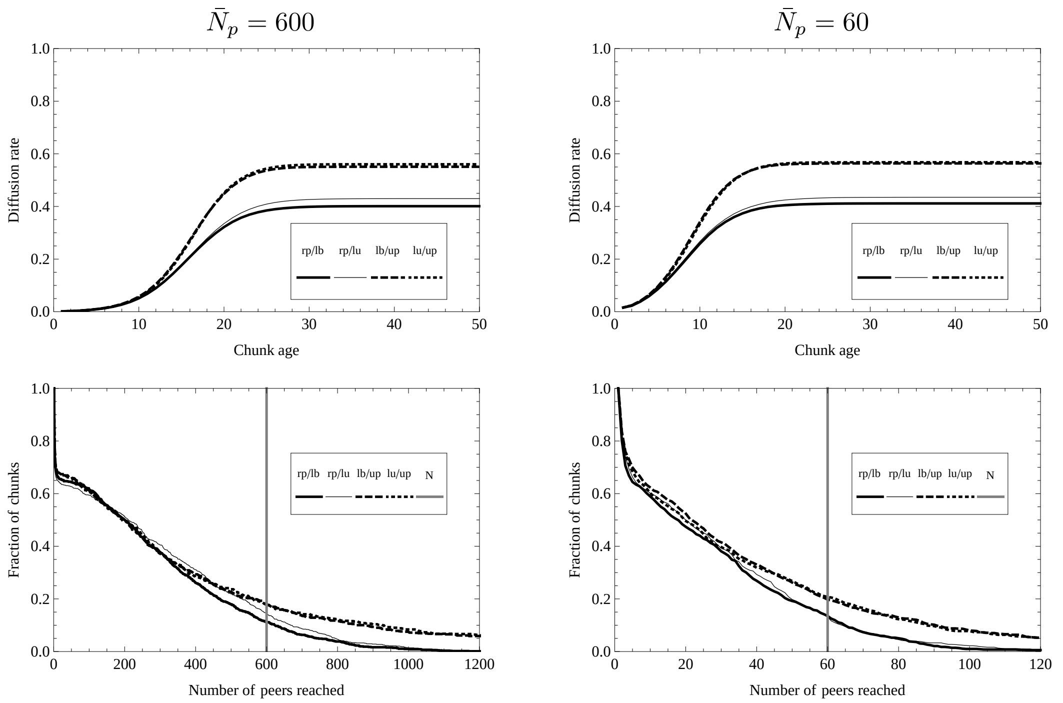

Figure A6: Diffusion graphs of the policies in a dynamic ($\frac{1}{\mu_p} = 3$) setting.

B Figures: Source rate

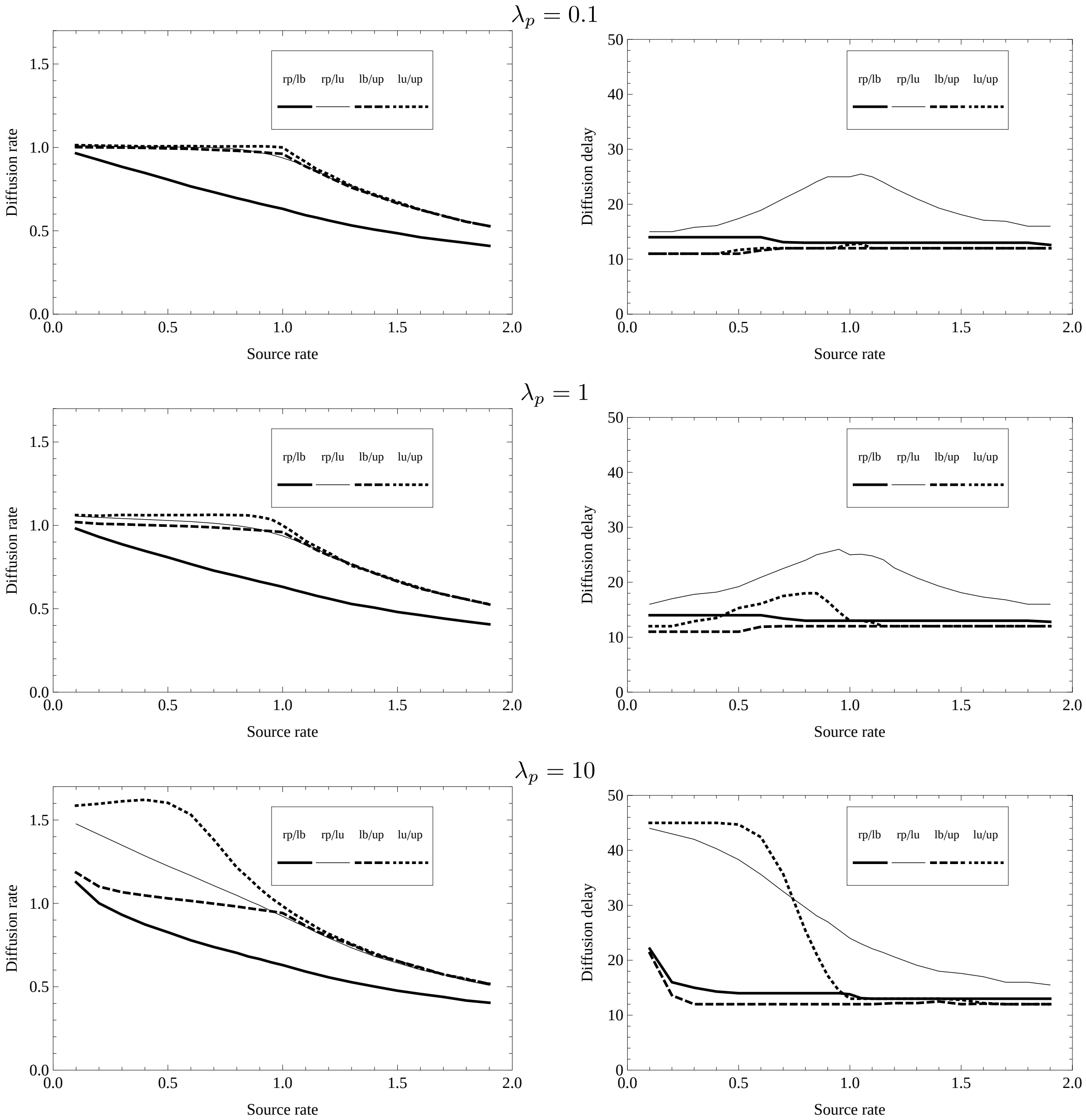

Figure B1: Diffusion rate and delay as a function of source rate.

C Figures: Heterogeneity

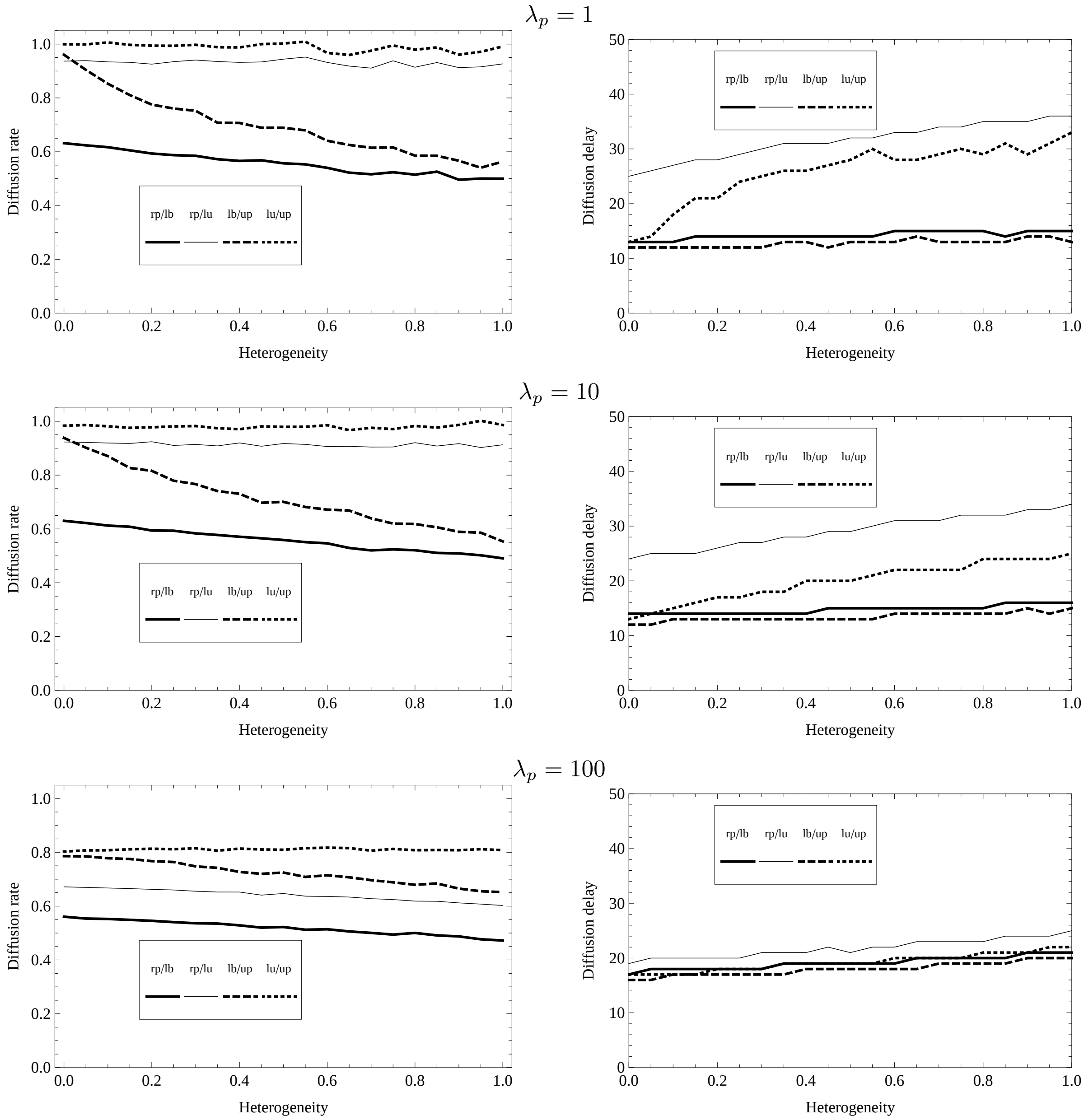

Figure C1: Diffusion rate and delay as a function of heterogeneity.

D Figures: Mean number of peers

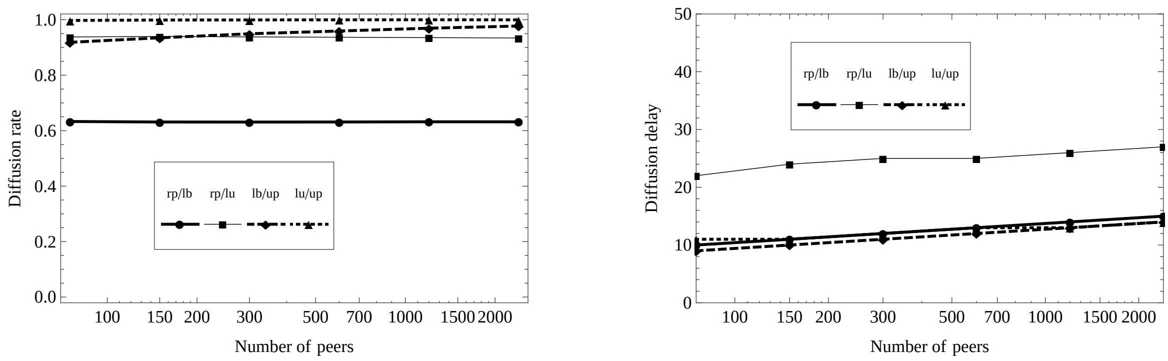

Figure D1: Diffusion rate and delay as a function of the mean number of peers $\bar{N}_p = \frac{\lambda_p}{\mu_p}$.