Data-driven traffic and diffusion modeling in peer-to-peer networks: A real case study

Romain Hollanders[*], Daniel F. Bernardes[†], Bivas Mitra[*],
Raphaël M. Jungers[*‡], Jean-Charles Delvenne[*] and Fabien Tarissan[†]

Abstract

Peer-to-peer (p2p) systems have driven a lot of attention in the past decade as they have become a major source of Internet traffic. The amount of data flowing through the p2p network is huge and hence challenging both to comprehend and to control. In this work, we take advantage of a new and rich dataset recording p2p activity at a remarkable scale to address these difficult problems. After extracting the relevant and measurable properties of the network from the data, we develop two models that aim to make the link between the low-level properties of the network, such as the proportion of peers that do not share content (i.e., free riders) or the distribution of the files among the peers, and its high-level properties, such as the Quality of Service or the diffusion of content, which are of interest for supervision and control purposes. We observe a significant agreement between the high-level properties measured on the real data and on the synthetic data generated by our models, which is encouraging for our models to be used in practice as large-scale prediction tools. Relying on them, we demonstrate that spending efforts to reduce the amount of free-riders indeed helps to improve the availability of files on the network. We observe however a saturation of this phenomenon after 65% of free-riders.

Contents

[*] UCLouvain – Université catholique de Louvain / INMA, Avenue G. Lemaître 4, 1348 Louvain-la-Neuve – Belgium, Email: firstname.lastname@uclouvain.be
[†] LIP6 – CNRS and Université Pierre et Marie Curie / Paris 6, Place Jussieu, 75252 Paris cedex 05 – France, Email: firstname.lastname@lip6.fr
[‡] F.R.S./FNRS Research Associate.

1 Introduction

Peer-to-peer (p2p) file sharing systems have evolved into a large traffic source in the Internet [Ban et al., 2011, TorrentFreak, 2010, Azzouna & Guillemin, 2003, Karagiannis et al., 2004, Sen & Wang, 2004]. This development has crucial implications for traffic engineering and information diffusion at the same time, since p2p networks constitute a remarkable case of interaction between a *technological* layer (network of computers) where the traffic occurs, and a *social* layer (overlay network of peers, structured by related interests) where the content spreading occurs. Moreover, information exchanges in p2p networks have the special characteristic that, unlike e.g. in mobile phone networks, the content that is actually shared by the peers is public and traceable. These features motivate the need for the development of specific tools and models to capture how these networks behave.

In this work, we study a new dataset that records p2p activity at a particularly fine scale, and we seize this opportunity to try enhancing both the user experience and the administrator's control over the network.

The main concerns of a peer when initiating a search for a file are usually (1) to find the desired file and (2) to acquire this file as quickly as possible. The ability of a p2p system to guarantee a certain level of performance when providing a file is defined as its Quality of Service (QoS). While our data do not enable us to assess the quality of the download speed provided by the network, which is mostly determined by the architecture of the network, they do enable us to evaluate the file availability. It is indeed striking that files sometimes become unavailable for some time, mainly because no provider is available. We call such unavailability periods "*silent periods*".

From the network administrator's point of view, one of the main concerns is to be able to observe and eventually control the data flow on the network, or more specifically, the way files diffuse on the network of peers. The diffusion of files is also a feature that can be inferred from our data and represented through "*spreading cascades*".

Both the QoS and the spreading cascades cannot be controlled directly and they depend on numerous characteristics of the p2p network, including its size and its architecture, but also on the way files are distributed on the network of peers, on the proportion of free riders (i.e., peers that do not share content), etc. Hence, it is important to understand how low-level properties of the network, such as file popularity, peer activity or their sharing behavior are related with its high-level properties, like spreading cascades or silent periods. Ideally, one would like to be able to influence the high-level properties of the network for supervision and control purposes, and this through the manipulation of its low-level properties. This of course requires understanding the relationship between these low- and high-level properties. This paper addresses precisely this question.

Two models. We here propose two different models to reproduce different high-level properties of the network from its low-level properties. The first model, based on Markov chains, is designed to reproduce synthetic but realistic traffic data and is particularly well suited for reproducing convincing silent periods. The second model is designed to capture diffusion features on the social network of peers and it performs well at reproducing spreading cascades. The global procedure used to calibrate these models and validate their ability to reproduce the high-level properties can be summarized as follows:

1. identify the meaningful and measurable low-level properties of the network from our dataset and use them as model parameters;
2. run the models and extract the relevant statistics about silent periods and spreading cascades from the simulated data;
3. compare these statistics with the real ones, extracted from the dataset.

The traffic model. Our first model aims at reproducing realistic synthetic traffic data. It represents the activity of peers with a Markov chain that translates the dynamics of the system, while keeping track of available files on the network. The main idea of this model is to assume that the network dynamics comes down to an entanglement of a number of simple and independent renewal processes (e.g., for each peer, a request, a login and a logout process). Each process is then assimilated to a Poisson process.

Improving upon a previous model from [Ge et al., 2003], the main features of this model are its simplicity, its flexibility and its accuracy. Indeed, the intuition of the model is easily explained and extensions of the model towards more complex behaviors can be added in a natural way. But above all, the model is efficient for reproducing realistic silent periods, even for a small generated network, as evidenced by our results. On the other hand, the computational and space complexity of the procedure are quite high.

The diffusion model. Our second model is designed to describe diffusion of content on the network. Instead of modeling diffusion with an agent-based model as previously, this model assumes diffusion of content occurs similarly to epidemic outbursts. Indeed, we present a model based on the classical SI model, which can incorporate peer behavior heterogeneity, and show how this model and its extensions can capture key properties of diffusion cascades.

In addition to the model describing file spreading dynamics we also present a method to reconstruct the social network of peers – connected by common interest – from the data. This reconstructed network is necessary to calibrate and simulate the model described previously. The interplay between network and spreading dynamics is interesting in and of itself and important to yield realistic results. In particular we demonstrate the importance to consider a dynamic network, integrating the connection patterns in the data, to reconstruct spreading cascade properties.

Contributions at a glance. Our two models are based both on the knowledge that we have of p2p systems and on our dataset; our knowledge helps us identify the key features of the network dynamics, give structure to the system under consideration and get rid of irrelevant details, whereas data is used to calibrate the established structure to a specific network. As developed in the next sections, these models succeed in reproducing some of the high-level features of the network, such as characteristics of silent periods and the size and number of links of cascades.

Moreover, the models provide predictive power. For example, it is interesting to measure how the fraction of free riders affects the frequency and length of silent periods for a file, and therefore the availability of the file. Relying on the model, we observe that file availability is improved as we reduce the fraction of free riders down to 65%, but the gain is limited below this threshold. Although it is not possible to directly control the fraction of free riders, this gives us insight on the intrinsic dynamics of p2p networks and shows the potential use of our models for predicting the effect of new p2p policies on diffusion or availability of files. Comparison between the two models also gives us insight on the features best captured by Markov chains on the one hand, or SI models on the other hand. As regard the diffusion model, we showed how to integrate temporal patterns into standard epidemiology models in order to reproduce qualitative properties of real spreading cascades.

More broadly, we believe that the constitutive principles behind our models are in no way limited to p2p networks, but are ultimately applicable to other situations in networks of interactive agents such as, e.g., in mobile phone networks or to model the diffusion of rumors in social networks.

Outline of the paper. The paper is organized as follows. First in Section 2, we review the existing studies and models of p2p traffic and diffusion. In Section 3, we present the dataset as well as the low-level properties of the network, used as model parameters, and its high-level properties, used as validating metrics. Section 4 is dedicated to the traffic model that reproduces realistic traffic data and silent periods. Then, Section 5 presents the diffusion model, showing how it is able to reproduce key properties of spreading cascades. We finally conclude the paper in Section 6 by discussing the results and laying some foundations for future works.

2 Related work

In the literature, several measurement-based studies have been done to investigate the properties of real p2p traffic. In [Gummadi et al., 2003], Gummadi et al. crawled the KaZaA traffic for 200 days to explore the client behavior as well as rise and fall in the file popularity over time. The studies in [Gummadi et al., 2003] as well as in [Hoßfeld et al., 2004] showed that the file popularity distribution deviates substantially from the Web traffic distribution and does not follow Zipf's law. In [Handurukande et al., 2006], a similar kind of measurement study has been done in eDonkey file sharing system which revealed a strong discrimination between download traffic flow and non-download streams [Tutschku, 2004]. Similar studies for Gnutella [Tutschku & de Meer, 2003] and BitTorrent [Izal et al., 2004] exist as well. Measurements performed by Zhao et al. [Zhao et al., 2006] report a particular and interesting behavior of the file popularity, which has many similarities with the product life cycle behavior reported in marketing literature.

There are several studies done on the *modeling of p2p network traffic* which primarily focused on the behavior of the peers in the system. In this context, Ge et al. [Ge et al., 2003] proposed an agent based traffic model and used it to explore the impact of free riders on p2p system performance; however, their model only focused on peer query characteristics and was not really data driven. In [Qiu & Srikant, 2004], Qiu et al. presented a simple fluid model for BitTorrent-like networks and studied the steady-state network performance. Experimental results showed that the model can capture the behavior of the system even when the arrival rate is small. Schlosser et al. [Schlosser et al., 2002] proposed a query-cycle simulator concentrating on p2p traffic and network behaviors. In [Xiangying & de Veciana, 2004], Yang et al. modeled the service capacity of a p2p system in two regimes. One is the transient phase in which the system tries to catch up bursty demands (flash crowd) and the second one is the steady state where the service capacity of a p2p system will scale with and track the offered loads. In [Menasche et al., 2009], Menasche et al. proposed a framework to model p2p systems where files may become unavailable. They show the applicability of the model to decide the optimal bundling of files to improve the file availability in BitTorrent. In [Feng et al., 2009], Feng et al. build user behavior models which incorporates several important characteristics including retry behavior, free-riding, file checking, and file removal. The model parameters are empirically derived from real user logs.

When we explore the aforementioned papers in the light of our large scale measured dataset, some limitations of the existing literature appear. In particular, although some works mention ways of measuring the unavailability of files similarly to the “silent periods” that we use, a more comprehensive understanding is required about their characteristics. Indeed, the related literature fails to provide satisfactory insight regarding the connection of the silent periods with the other network parameters, which is important considering the close link between the silent periods and the Quality of Service. Moreover, little has been done to study in a unified way both the traffic engineering on the physical network and the social aspects on the overlay network. Our work sheds some light on those issues.

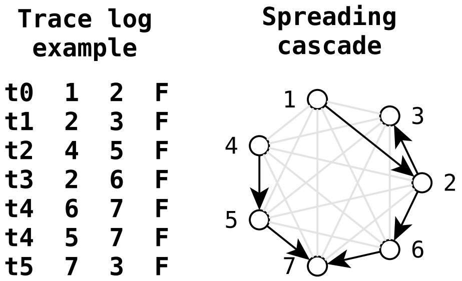
As for *content diffusion*, in the literature, this concept can allude broadly to the dissemination of a piece of information among individuals. In this case one is typically interested in the evolution of the number of peers which possess the piece of information in question. This notion of diffusion has been primarily investigated in biology (in connection with epidemic/contagion outbursts [Andersson & Britton, 2000]), but has also proven relevant in the context of p2p networks [Leibnitz et al., 2006, Hosanagar et al., 2010].

The interest in exploring embedded social networks upon technological ones and the increasing availability of real world data [Kleinberg, 2008] have pushed for a more detailed notion of diffusion on networks, characterized by the spreading of information *among neighbors* in this network. More realistic spreading models compatible with this notion, which propose microscopic evolution mechanisms for the diffusion phenomenon, have been developed building upon traditional models from epidemiology [Barrat et al., 2008]. These models (particularly the classical SIR model and derivatives) have been extensively used in recent works to analyze information diffusion on overlay networks, such as emails on corporate networks [Iribarren & Moro, 2009], SMS on mobile networks [Onnela et al., 2007], hypertext on web blogs [Leskovec et al., 2007], and files on p2p networks [Bernardes et al., 2012].

In parallel with the theoretical evolution of diffusion models on networks, new approaches emerged in empirical studies of information spreading as well. In particular recent works have focused on the study of *spreading cascades* (also known as diffusion/information cascades) [Leskovec et al., 2007, Liben-Nowell & Kleinberg, 2008, Lerman & Ghosh, 2010, Bernardes et al., 2012]. These graphs are more complex objects than macroscopic quantities such as number of infected individuals and reveal more information about the spreading trail.

3 Dataset and Statistics

The data used in this study comes from a 48 hour record of the file sharing activity in an eDonkey server (akin to [Aidouni et al., 2009]) located in France, suitably anonymized for privacy protection purposes. In this setting, peers query the server and for each requested file they get a list of available peers in the network possessing it. Next, the interested peer contacts the potential providers directly and the transmission between them ensues. The dataset is a collection of these satisfied queries, encoded as tuples of integers in the following format: $(t, \{P_k\}_{k \leq n}, C, F)$ where the capital letters represent unique ids. Each tuple accounts for a request made at time t of the file F by the client peer C, satisfied by the provider peers P_k.



Trace log example

t0	1	2	F
t1	2	3	F
t2	4	5	F
t3	2	6	F
t4	6	7	F
t4	5	7	F
t5	7	3	F

Fig. 1. A trace log example (left) with the corresponding spreading cascade in black (right).

Each request in the latter format can be decomposed into individual transfer interactions, represented by the tuples $(t, P_0, C, F) \ldots (t, P_n, C, F)$. An example of the trace log given in terms of individual interactions is presented in Fig. 1 (left).

Let $\mathscr{R}$ be the set of all requests, $\mathscr{D}$ be the set of individual transfer interactions, $\mathscr{P}$ the set of all peers appearing in these tuples and $\mathscr{F}$ the set of all files exchanged. In this dataset we have registered $|\mathscr{P}| = 5\ 380\ 616$ peers, $|\mathscr{F}| = 1\ 986\ 588$ files, $|\mathscr{R}| = 212\ 086\ 691$ requests, $|\mathscr{D}| = 471\ 134\ 409$ transfer interactions, all happening during $T = 170353$ seconds. This massive amount of data offers the possibility for a more in-depth statistical analysis than previous studies such as [Gummadi et al., 2003], who mostly focused on the characterization of the p2p clients and objects.

Let us now analyze the properties of the network that can be measured from the dataset. We distinguish two types of properties: the low-level ones that will serve as model parameters and the high-level ones that will serve as validating metrics since they represent the properties that we want to control.

3.1 Low-level properties of the network

We now describe the properties that can be measured from the dataset and that we will use as parameters for our models.

3.1.1 Peer activity

Let us define the *activity* $\alpha_P(t)$ of a peer $P \in \mathscr{P}$ as the number of requests made by that peer on $[0,t]$. By convention, if we write α_P without the reference to the time t, we refer to the average request frequency rather than the actual number of requests, assuming it is not changing with time.

3.1.2 File popularity

Similarly to the activity, we define the *popularity* $\pi_F(t)$ of a file $F \in \mathscr{F}$ as the number of requests for file F made on $[0,t]$. Again, π_F (without the reference to t) refers to the average request frequency for file F.

Fig. 2 shows the complementary cumulative distributions for peer activity and file popularity in our dataset. It can be observed that these distributions are heavy tailed, ranging over several orders of magnitude, though not properly scale-free.

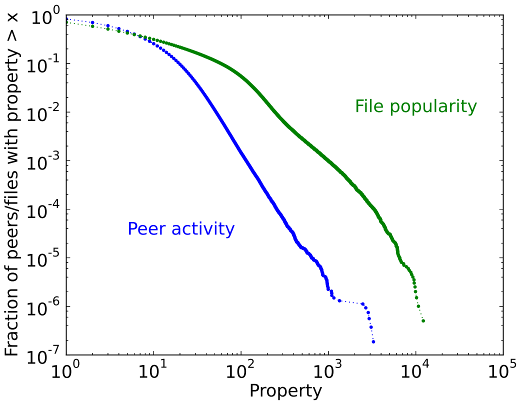


Fig. 2. The peer activity and the file popularity complementary cumulative distributions are heavy tailed, even though not properly scale-free.

3.1.3 Peer connection patterns

We have inferred the connection duration of peers by looking at their activity profile. Fig. 3 (top) illustrates such profiles for a few selected peers. Active periods can be assimilated to time spent online, as opposed to the time spent offline. We estimate the active and

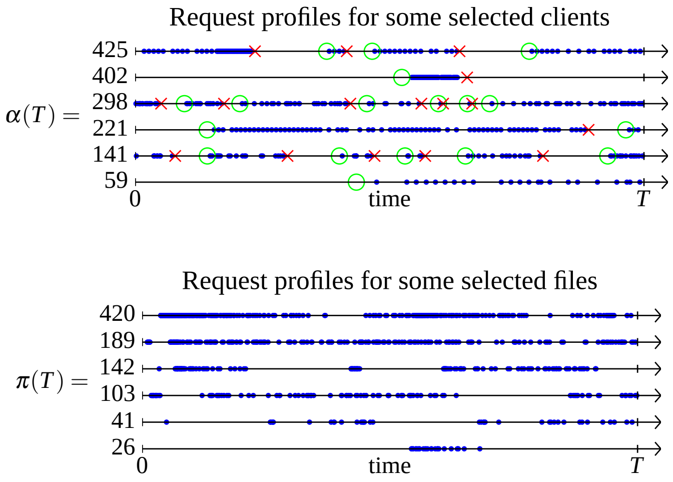


Fig. 3. The request profiles of six selected clients (top) and files (bottom) with variable activity/popularity. Here, each time line corresponds to a peer or a file and we draw a blue dot at time t on the time line of some peer or some file if it has been requested at that time. (Top) We also add green circle and red crosses for the estimated login and logout times of peers respectively. (Bottom) One can clearly distinguish silent periods on most of the illustrated profiles.

inactive time periods using a maximum likelihood approach based on an algorithm from [Jewell, 1982]. Roughly, the idea is to decide whether an interval between two requests corresponds to a period during which the peer was online or not. The procedure is explained in more details in Appendix A. On Fig. 3 (top), the estimated login and logout dates appear respectively as green circles and red crosses. From the estimated active periods we extract the distributions for the rate at which peers login to the network when they were offline and the rate at which they logout when they were online, which we use in our models.

Fig. 4 shows the complementary cumulative distributions for the estimated peer login and logout rates. As for the peer activity and the file popularity distributions, these distributions are also heavy tailed.

3.1.4 Sharing behavior

It has been observed that the peers are divided into two classes: those that provide file sharing facility (*sharing peers* or *providers*) and those that do not (*free riders*) [Ge et al., 2003]. Free riders bring in capacity only to the common service component of the system (e.g., routing queries), and do not contribute to the capacity of serving files. More precisely, we define a sharing peer as a peer that shares at least one of the files known to be in its possession to other peers. A peer who is not a sharing peer is a *free rider*. In our dataset, the proportion of sharing peers was around 4%.

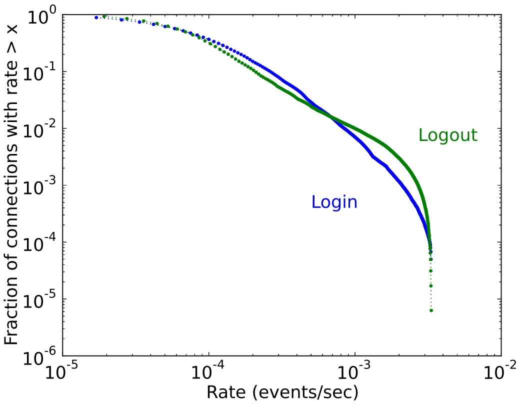


Fig. 4. The peer login and logout rate complementary cumulative distributions are also heavy tailed.

3.2 High-level properties of the network

We now describe silent periods and spreading cascades which are the properties of the network that we would like to study. They will serve as validating metrics for our models.

3.2.1 Silent periods

One critical feature of p2p requests is the fact that a file is not always available; availability of the file depends on the presence of providers for that file. For example, when a popular file is not available for some time, one can observe a "*silent period*" when looking to its request profile, i.e., a sudden stop of the requests for that file followed by an unexpectedly long period of inactivity. Fig. 3 (bottom) illustrates the request profiles of a few typical files on which one can neatly observe some silent periods. The presence of long silent periods is undesirable in practice because they are frustrating for the peers interested in the unavailable files. Additionally, silent periods indicate a temporary stop in the traffic flow in the underlying network for that specific file (these issues have already been mentioned in [Menasche et al., 2009]).

To identify silent periods, we used a similar procedure as the one used to determine the connection patterns of the peers. The idea is again to decide, using the maximum likelihood approach proposed in [Jewell, 1982], whether or not a time-interval between two requests of a file corresponds to a period during which the file was available. Again, the procedure is described with more details in Appendix A. Once the silent periods have been identified, we computed three of their characteristics, namely (1) the distribution of the number of silent periods in request profiles, (2) the distribution of the total unavailability time of files and (3) the distribution of the average length of silent periods. Although all the aforementioned distributions are based on the silent period profiles, they nevertheless complement one another and reveal different trends.

3.2.2 Spreading cascades

We also analyze the *spreading cascade*, which represents the diffusion of each file in the p2p network. For a file F, the spreading cascade is a directed graph featuring the set $\mathscr{P}_F$ of peers who have participated in the spread of F (as clients and/or providers) and the set $\mathscr{L}_F$ of links connecting each client C with the first peer(s) who provided F to it. More formally, let $\tau_F(C) = \inf\{t : (t, \cdot, C, F) \in \mathscr{D}\}$ be the first instant at which C requested F and let the directed graph $\mathscr{K}_F = (\mathscr{P}_F, \mathscr{L}_F)$ be the spreading cascade of F, with

$$\mathscr{P}_F = \{P \in \mathscr{P} : (\cdot, P, \cdot, F) \in \mathscr{D} \text{ or } (\cdot, \cdot, P, F) \in \mathscr{D}\},$$

$$\mathscr{L}_F = \cup_{C \in \mathscr{P}_F} \{(P, C) \in \mathscr{P}_F \times \mathscr{P}_F : (\tau_F(C), P, C, F) \in \mathscr{D}\}.$$

A client requesting a file may receive a response from potentially several providers simultaneously, which implies that nodes in the cascade graph not only have multiple outgoing links, but also multiple incoming links in general.

The first key property encoded in the spreading cascade of a given file F is the number of nodes who possess it at the end of the observed period, which is given by the *size* of the cascade $|\mathscr{P}_F|$. We also explore two other key topological properties of the cascade, namely its *depth* and *number of links*. The former is defined as the length of the longest path on the cascade and captures the maximum number of hops from peer to peer that the file has undergone before it was relayed from a provider to a client. The number of links, given by $|\mathscr{L}_F|$, combined with the size of the cascade gives information on the sharing pattern of the network. An example of observed trace and constructed spreading cascade is given in Fig. 1: the spreading cascade has size 7, depth 3 and 6 links.

4 A Traffic Model Based on Markov Chains

Our goal in this section is to propose a simple traffic model that can reproduce the basic characteristics of the real p2p traffic. Instead of proposing a complete and rigid model which is perhaps hard to customize, we aim at developing a simple[1], intuition-driven flexible model which is easily extendable depending on the specific requirement. With some directions to the possible extensions, we show how our model is able to reproduce the key features of the silent periods that can be observed in real data.

The model that we propose relies on the assumption that clients make new requests independently of previous ones and that the time between two requests follows an exponential distribution. In general the Poisson process parameter may be time-dependent, for instance varying according to the circadian rhythm of peers, which has been observed empirically [Locher et al., 2009]. In the following, we neglect the time-dependency as a first approximation. Thus, the requests of a client P follow a Poisson process with the activity α_P as parameter (i.e. request rate). According to our data, this is a reasonable assumption, as illustrated in Fig. 5. In this figure, we compared the average inter-request times with its standard deviation: the agreement of the two curves is indeed a property of exponential distributions. In a similar way, we assume that the time between a login and a logout (or vice versa) follows an exponential distribution. Such assumptions are frequently

[1] Our model features only 5 parameters, assuming that file popularity, peer activity, login and logout rates can be characterized by power law distributions.

used in the literature [Menasche et al., 2009,Clevenot & Nain, 2004,Gummadi et al., 2003]. The intermittent availability of files seems to be an important feature when studying their diffusion in p2p networks. Our model takes that into account as much as possible.

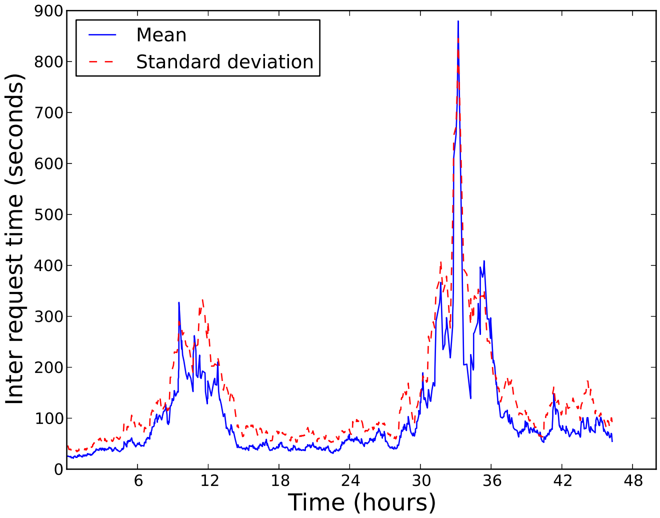


Fig. 5. (Blue) The average inter-request time for a typical active client ($\alpha(T) = 2674$) and (Red) its standard deviation, evolving with time. Both the average and the standard deviation have been estimated with a one-hour sliding time window over the two days' time range of our data. The two curves are close from each other, in agreement with the assumption that inter-event times follow an exponential distribution. The fact that the red curve lies a little bit above the blue curve reveals however some burstiness. Furthermore, note that the parameter of the Poisson distribution depends on time through the circadian effect, which we choose to neglect as a first approximation.

Model parameters. To simulate a p2p system with m clients and n files, we need to fix the following parameters:

1. The activity of the clients;
2. The popularity of the files;
3. The login and logout rates of the peers;
4. The fraction of free riders.

These ingredients can be obtained from our data through the distributions of peer activity, file popularity, login rates and logout rates, and the observed proportion of free riders. See Section 3.1 for details about these statistics. We then use these statistics to generate a set of peers $\mathscr{P}$ and a set of files $\mathscr{F}$ of appropriate size. To each peer P, we assign an activity α_P according to the observed activity distribution as well as a login rate λ_{IN_P}, a logout rate λ_{OUT_P} and a sharing peer/free rider flag. We also assign a popularity π_F (i.e. the rate at which file F is requested) to every file F according to the popularity distribution. Finally, we choose the number of time steps T to be simulated.

The core of the model can be split in two parts: (1) the state of each peers in the system, represented by small Markov chains and (2) the state of the files, represented by an availability matrix. This matrix is meant to make the link between the states of all peers by remembering which files are possessed by which peers.

4.1 The behavior of the peers represented by Markov chains

The state of a peer P in the system is represented by a continuous-time Markov chain (see Fig. 6) with three states: an offline state (“OFF$_P$”), an online state (“ON$_P$”) and a download state (“DL$_P$”). The transition rates between the states are chosen according to the chosen activity ($\lambda_{\mathrm{REQ}_P} = \alpha_P$), login rate ($\lambda_{\mathrm{IN}_P}$) and logout rate ($\lambda_{\mathrm{OUT}_P}$) of peer P (which are initially attributed as described above). In this simple model, we assume that files are downloaded instantaneously, hence the download rate $\lambda_{\mathrm{DL}_P} = \infty$.

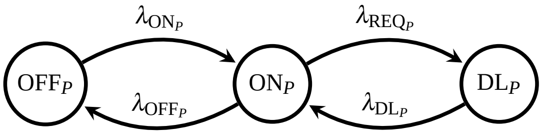

λ_{ON_P}
λ_{REQ_P}
OFF$_P$
ON$_P$
DL$_P$
λ_{OFF_P}
λ_{DL_P}

Fig. 6. A three-state continuous-time Markov chain provides a natural model of a peer’s dynamics.

Of course, the initial state of each peer should be initialized at time $t = 0$ to either “ON$_P$” or “OFF$_P$”, proportionaly to “λ_{ON_P}” and “λ_{OFF_P}” respectively.

The main strength of such a simple modeling of the peers rests in its modularity. Indeed, it lays the foundations from which almost any feature that may represent the behavior of a peer can be added in a fairly straightforward way. We will mention a number of natural possible extensions of these Markov chains in Section 4.4.

4.2 The availability matrix

The presence of silent periods in the p2p system highly depends on the availability of the files. For a file to be available, it must be possessed by a peer which is both online and ready to share the file. To capture this feature, we introduce the availability matrix $A(t) \in \{0, \pm 1\}^{m \times n}$ whose entries are defined as follows:

$$A_{P,F}(t) = \begin{cases} 0 & \text{if peer } P \text{ does not possess file } F \text{ at time } t \\ 1 & \text{if peer } P \text{ possesses file } F \text{ at time } t \\ & \text{and is both online and ready to share } F \\ -1 & \text{if peer } P \text{ possesses file } F \text{ at time } t \\ & \text{but is either offline or not ready to share } F. \end{cases}$$

We also define the availability vector $a(t)$ as:

$$a_F(t) \triangleq \sum_{P \in \mathscr{P}} \max\{A_{P,F}(t), 0\}$$

for all files $F \in \mathscr{F}$, which counts the number of available providers for F at time t, and its binary version $a_F^{(\mathrm{bin})}(t) \triangleq \min\{a_F(t), 1\}$ for all $F \in \mathscr{F}$. Hence, a file F is available at time t whenever $a_F(t) > 0$ (or $a_F^{(\mathrm{bin})}(t) = 1$).

Then, when simulating a p2p system using the Markov chains defined in Section 4.1 to represent the state of the peers, we update $A(t)$ as follows:

- When an event "$\mathrm{ON}_P \to \mathrm{DL}_P$" occurs at time t, peer P chooses a file F according to its preference vector $p^P(t)$ (which is a probability vector that depends on t since it depends on the files that are available at the time of the request). In the simplest version of our model, we choose $p^P(t)$ to be proportional to the popularity of the available files (and independent from the number of available providers) such that:
$$p_F^P(t) = \frac{\pi_F \, a_F^{(\mathrm{bin})}(t)}{\sum_{F\in\mathscr{F}} \pi_F \, a_F^{(\mathrm{bin})}(t)}$$
for all P. Once a file F has been chosen to be requested, we update $A(t)$:
$$A_{P,F}(t) = \begin{cases} 1 & \text{if peer } P \text{ is a sharing peer} \\ -1 & \text{otherwise.} \end{cases}$$
- When an event "$\mathrm{ON}_P \to \mathrm{OFF}_P$" occurs at time t, and if peer P is a sharing peer: $A_{P,F}(t) = -|A_{P,F}(t-1)|$, for all $F \in \mathscr{F}$.
- When an event "$\mathrm{OFF}_P \to \mathrm{ON}_P$" occurs at time t, and if peer P is a sharing peer: $A_{P,F}(t) = |A_{P,F}(t-1)|$, for all $F \in \mathscr{F}$.

To summarize, the availability matrix keeps track of how files diffuse among the peers while providing information about the availability of the files for other potential clients at the same time. Of course, it is necessary for the matrix A to be initialized at time $t = 0$ such that there is at least one non-zero entry in every column. The assignment of the non-zero entries for each line of the matrix is therefore randomly chosen in proportion to the activity of the corresponding peer.

Modeling procedure. Once all peers and files in the system have been created with their parameters (π_j for files and $\lambda_{\mathrm{REQ}_P}, \lambda_{\mathrm{ON}_P}, \lambda_{\mathrm{OFF}_P}$ and the sharing peer/free rider flag for peers), and once the availability matrix $A(0)$ and the initial states of the peers have been initialized, one can simulate the traffic generation process. We first simulate for every peer the moment of its next transition in the Markov chain. Then, we iteratively treat every transition event in the Markov chains of the peers in their order of appearance. After the treatment of the transition event of a peer, we first determine the time of its next transition before considering the next event to happen. We do this until the time limit has been reached.

We can determine the complexity of the modeling process which essentially corresponds to the product of the expected number of events to treat with the cost of the treatment for one event, which gives a number of flops of about $m\,n\,T\,R$, where m is the number of peers, n is the number of files, T is the total time to be simulated and R is the average number of requests of a client per unit of time. In our dataset, R is typically worth about 2.5 requests every day, hence in our case $T\,R \sim 5$.

4.3 Model validation and insights

We simulated the above described model to generate a synthetic dataset with 5000 peers and 2000 files over a period of time of 2 days. Here, the ratio between the number of peers and the number of files is approximately the same as in the data. Fig. 7 illustrates request profiles for some clients and files obtained from our simulations.

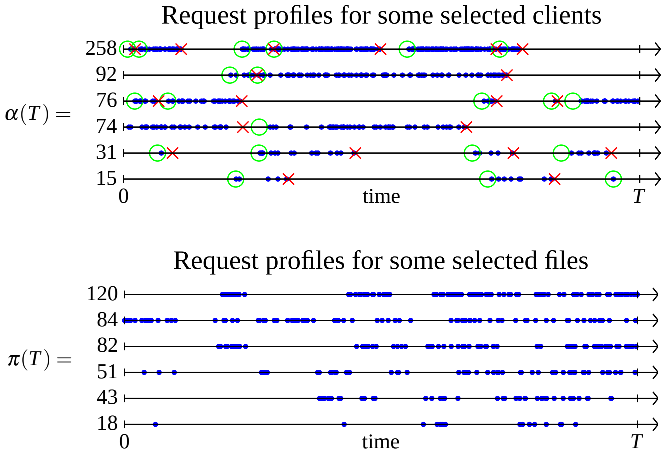


Fig. 7. Request profiles obtained from simulations for a few selected peers and files. They can be qualitatively compared with the real profiles shown earlier in Fig. 3.

To compare both the real and the synthetic datasets, we used the metrics that we defined in Section 3 for silent periods, i.e., the distributions of the number of silent periods as well as their total and average length. These three metrics are complementary to each other and they reveal different trends. Fig. 8 illustrates the results obtained on these metrics for both datasets. We observe a remarkable correspondence between simulated and real data, especially for the first two distributions, which is a non-trivial and encouraging result. Furthermore, it is striking to see that this correspondence is observed on datasets with a huge difference of scale. This is a good sign that our model is able to reproduce some essential intrinsic features of a network even at a fairly small scale, which is an interesting achievement, especially given the simplicity of the model.

Next we turn our attention to understand the impact of free riders on the QoS and user satisfaction. Taking advantage of the above described results, we used our model to simulate traffic with an increasing proportion of free riders. We recorded the average number of silent periods and the average total length of silent periods and plotted the results in Fig. 9. As expected, the number of silent periods and their total length decrease when the proportion of free riders decreases. Even though the fraction of free riders is not an exogenous parameter that can usually be controlled, it is interesting to note that the characteristics of silent periods—thus the availability of files—seem to stabilize when this fraction falls below 65%. Hence, reducing the number of free riders seems to be a way to improve the QoS, yet only until some point, whereas other effects that are more difficult to deal with also affect the QoS, e.g., the disconnection of the few peers that possess a rare file. In that respect, the contribution of a file on the QoS should probably be defined as

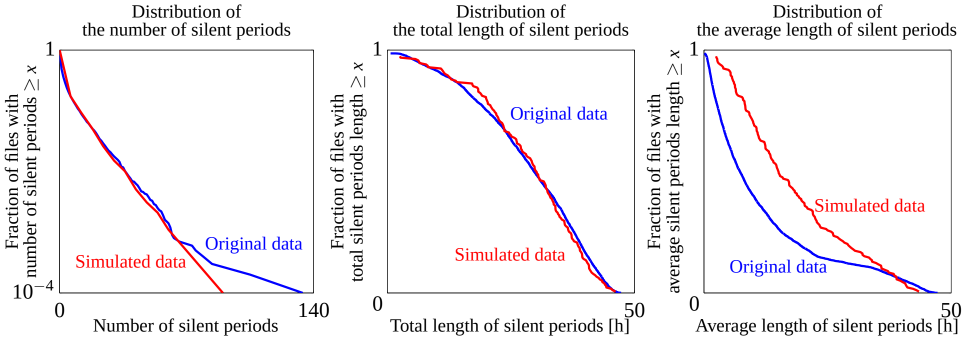


Fig. 8. The scaled[2] simulated data exhibit a remarkable correspondence with the real data on the three chosen metrics that feature silent periods, especially for the number of silent periods (left) and their total length (middle). The mismatch for the average length of silent periods (right) reveals that, for a given file, the lengths of its different silent periods have not enough variation in the simulated data. There is also a minor mismatch for the distribution of the number of silent periods (left) that shows a higher tail in the real data than in the simulated data. This higher tail probably comes from a bias in the way we identify periods at all for which we detect (lots of short) silent periods. As supported by both the graph for the number of silent periods and the one for their average length, this effect is more likely to appear in the real data for which inter-event times are more variable.

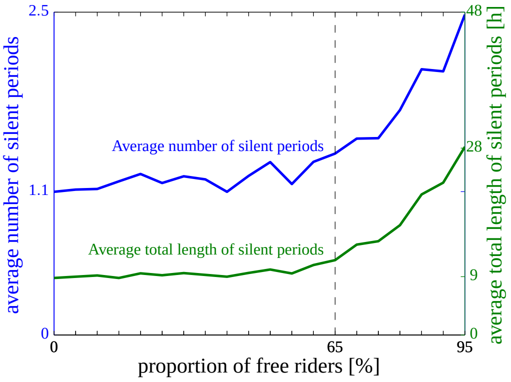


Fig. 9. The average number of silent periods (blue) and their average total length (green) increase when the proportion of free riders increases. Below around 65% of free riders, both the average number and the average length seem to stabilize. Note that the curves become ill-defined at 100% of free-riders, as no sharing occurs at all in this case.

the total time of its silent periods weighted by its popularity since the unavailability of an unpopular file should not be too much of a problem.

The model in its present simplest form creates a complete dataset that is similar to the real dataset we study. Hence, diffusion cascades can be extracted and their characteristics

[2] The simulated networks were obtained with fewer files and peers and hence with a smaller request density than the original network. Therefore, the time line in the simulated data was scaled by a factor that best highlights the comparison between original and simulated data. The same scaling was used in the three graphs, also affecting the computation of the silent periods. Because of this scaling, the comparison between original and simulated data should remain qualitative and focus on the shape of the curves.

can be studied (see Section 3). However, the obtained cascades are quite different from real cascades. In particular, their depth is in general significantly smaller. To obtain realistic cascades, one should improve the model, for instance by adding community effects to it as we mention in Section 4.4. We expect the model with communities to be able to reproduce realistic cascades and we plan to test its ability to do so in future work. However, even if it does reproduce realistic cascades, it will nevertheless remain a costly method if the only feature one is interested in is the diffusion of files. Therefore, in that case, a model which is specialized for diffusion should definitely be preferred. Proposing such a model is the goal of Section 5.

4.4 Natural extensions of the core model

The model presented above is meant to be as simple as possible, while capturing the main features of diffusion and silent periods in p2p networks. Even though simplicity is one of its main strengths, the model remains highly modular and refinable in many natural ways. To illustrate that, we here mention a number of natural extensions that may be of interest while generating the synthetic p2p traffic. On the one hand, these extensions increase the complexity of the model, but on the other hand, the latter can eventually become more sophisticated and realistic.

- *Add community effects*: this can be done by modifying the definition of the preference vector p^P of the peers such that they only are interested in some of the specific files. Such a modification can be done without modifying the initially chosen activity and popularity distributions. Note that community effects may play a crucial role in the way files are diffused in the network.
- *Add a circadian effect or other time-dependent effects*: it is always possible to add a time dependency to the parameters of the model. For instance, a circadian effect can be taken into account by modulating the request rate of the peers by a constant that oscillates around 1, depending on the time of the day. Note that time effects can be used to take burstiness into account.
- *Add a sharing obligation for free riders to share the files in their possession while downloading a file*: while spending time in the “DL$_P$” state, the client P is forced to share its files (which can be imposed by adding the new update rule: $A_{P,F}(t) = |A_{P,F}(t-1)|$, for all $F \in \mathscr{F}$ whenever entering that state). For that, a finite download rate λ_{DL_P} can be chosen so that a non-zero time is spent in that state. Special care must be taken however to take into account the fact that other requests can be made while spending time in the “DL$_P$” state: this can be settled by extending the Markov chains from Section 4.1 into a download queue.
- *Include correlation between the parameters*: if correlation data is available, it can be used to assign activities, login rates, logout rates and sharing peer/free rider flags to the peers in a more realistic way.
- *Include the fact that peers may clear their sharing directory*: this can be done by adding new “clearing” states to the original Markov chains and by adding a new update case for the availability matrix which erases all entries related to a peer that visits the new state. Adding such a clearing state can make files disappear from the network after some time or prevent some files from acquiring too many providers.

- *Add a dependency between the popularity of a file and the number of providers for this file*: this can be obtained again by modifying the definition of the preference vector p^P of the peers, making it dependent in $a(t)$ and not only in its binary version.

The above extensions illustrate the flexibility of our model. It is likely that most refinements that one could be willing to add to the model could be added in a fairly straightforward way.

4.5 A brief comparison with an existing model

We here compare our model with the existing agent based model described in [Ge et al., 2003]. The differences that one can expect from our model are the following:

- Our model aims to be as simple and natural as possible, where the model from [Ge et al., 2003] introduces some sophistications like for instance the classification of peers into several classes that behave all differently. At the same time, our model leaves room for extensions that can be added in a straightforward way.
- Our model generates synthetic datasets that exhibit silent periods even for the popular files, which we often observe in real traces, whereas datasets generated from the model described in [Ge et al., 2003] only generates such silent periods for unpopular or moderately popular files.
- Finally, our model can be implemented in order to perform computations in parallel by treating several transition events close in time at the same time.

5 Social Network and Diffusion Modeling

In this section, we examine the observed file spreading cascades on the social network of peers participating in the p2p system. To this end, we model the spreading cascade of files using a standard contagion model adapted to networks: the SI contagion model [Barrat et al., 2008]. As discussed in the introduction, this is a key reference model in the study of diffusion in a wide range of fields. This model treats each file spreading as an independent epidemic on the underlying social network of peers, where peers infect their neighbors in the network according to local rules of transmission.

Given this setting, in order to analyze the empirical spread of files among peers we need not only the detailed chronological data of who transmitted the information to whom (observable in the trace) but also data on the underlying social network on which the diffusion takes place. As pointed out in [Gomez-Rodriguez et al., 2012] it is challenging to reconstruct the network on which the diffusion takes place. One strategy to unfold this network is to explore relations among peers and their common shared files. Such a strategy was hinted at in [Handurukande et al., 2006] and developed more substantially in [Latapy et al., 2008, Iamnitchi et al., 2011, Bernardes et al., 2012]. We follow this approach to reconstruct the underlying social network and we build upon this model, integrating the temporal information in our trace to reconstruct a dynamic social network of peers. Finally, we calibrate the diffusion model using available trace data and evaluate it using numerical simulations.

5.1 Social network reconstruction

It is reasonable to assume that peers store and share content related to their interests and, likewise, peers will search for content matching their interests. In this sense it is natural to study the diffusion of files in the network of peers, related by common interests. More precisely, let the *interest graph* be the graph in which each node represents a peer and each edge joining two peers stand for common interest. Hence the spread of files among peers takes place on the interest graph and occurs from neighbor to neighbor, in agreement with the notion of diffusion on networks. It is important to stress that one cannot directly observe this graph in general – especially at large scale – but it is possible to approximate it using the spreading trace presented in section 3[2]. Using the framework in [Bernardes et al., 2012] we construct the bipartite graph $\mathscr{B} = (\mathscr{P}, \mathscr{F}, \mathscr{A})$ where $\mathscr{A}$ is the set of edges connecting the disjoint sets $\mathscr{P}$ and $\mathscr{F}$, respectively of peers and files, connecting each peer to the files it has shared, that is:

$$\mathscr{A} = \{(P,F) \in \mathscr{P} \times \mathscr{F} : (\cdot, P, \cdot, F) \in \mathscr{D} \text{ or } (\cdot, \cdot, P, F) \in \mathscr{D}\}.$$

Next, we construct the inferred interest graph of peers, connecting any two peers, which have demonstrated a common interest in the trace log – by requesting or providing a common file. More precisely, the interest graph $\mathscr{G} = (\mathscr{P}, \mathscr{E})$, is given by the projection of $\mathscr{B}$ on $\mathscr{P}$ such that

$$\mathscr{E} = \{(P,P') \in \mathscr{P} \times \mathscr{P} : \exists F \in \mathscr{F}, (P,F) \in \mathscr{A} \text{ and } (P',F) \in \mathscr{A}\}.$$

In other words, peers belonging to the neighborhood of a common file in $\mathscr{B}$ are connected in $\mathscr{G}$ – cf. example in Fig. 10. If a peer P provides a file F (corresponding to a music album for example) to another peer P', then there is link between them in the interest graph since both are interested in the same content, namely F.

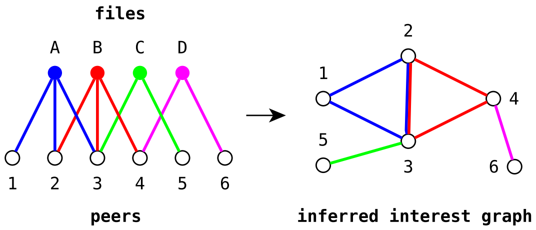


Fig. 10. The interest graph is the projection of the bipartite graph of peers and files, on the set of peers.

The interest graph is a comprehensive synthesis of peers' interest relations revealed in the observed time window. These relations are key to diffusion, since the spread of files occurs on the interest graph, as pointed out previously. However, even if the spread of

[2] In this section we use a reduced portion of the dataset, consisting of the first 8hrs of measure.

files between neighbors in the interest graph is likely, the actual transfer of files may not occur concretely because they may never be simultaneously connected to the p2p system or have a small *co-presence time* – i.e., the amount of time online in the presence of each other in the system. Hence, in order to make simulations more realistic, in the sense of reproducing observed file spreading cascades, we used temporal information to enhance the social network reconstruction.

A strategy to use temporal information, integrating the connection data estimated in Section 3.1.3, is to reconstruct a *dynamic interest graph*. In this graph, two peers will be connected at time $t > 0$ if they share a common interest (as in the interest graph) and if they are both online at time t. More formally, let $\mathscr{P}_t$ be the set of nodes online at time $t > 0$ and let the dynamic interest graph be defined as $\mathscr{G}_t = (\mathscr{P}_t, \mathscr{E}_t)$, with

$$\mathscr{E}_t = \{(P,P') \in \mathscr{P}_t \times \mathscr{P}_t : \exists F \in \mathscr{F}, (P,F) \in \mathscr{A} \text{ and } (P',F) \in \mathscr{A}\}.$$

Intuitively, the dynamic interest graph is built similarly to the original interest graph, but evolves with the addition/suppression of links between connecting/disconnecting nodes and their neighbors. The dynamic interest graph is a subgraph of the interest graph $\mathscr{G} = (\mathscr{P}, \mathscr{E})$ defined previously, in the sense that for all $t > 0$, $\mathscr{P}_t \subset \mathscr{P}$ and $\mathscr{E}_t \subset \mathscr{E}$. In the following, we examine both the original (static) interest graph and the dynamic interest graph as the underlying social network on which we perform file spreading simulations. That is, we consider a simple, baseline setting in which we suppose all the users are continuously online during the whole observation period and a second setting where we integrate peers connection patterns. Let us refer to these settings as *static* and *dynamic* respectively.

5.2 Diffusion model and calibration

As pointed out in the introduction, we model the spreading of files using the SI model for networks [Barrat et al., 2008]. In this model, each individual is either *susceptible* or *infected* (hence the acronym). Susceptible nodes do not possess the file and may receive it from an infected node, thus becoming infected. Infected nodes, in turn, try to spread the file to each of their neighbors in the network, one at a time in a uniform random way. The time between two infections is also random and follows an exponential distribution, which we refer to as the *inter-contagion time (ICT)*. Thus, if P possesses the file F, the number of peers who received the file F from P (after P obtained it) is a Poisson process characterized by the inter-contagion time rate. Alternatively, this process can be characterized by the average ICT, since it is the inverse of the ICT rate. We will examine SI models with *homogeneous* and *heterogeneous* inter-contagion time. In other words, in the first case we suppose all nodes have the same spreading behavior (global ICT rate) and in the second, an individual one (a different ICT rate for each node).

In order to calibrate these models, we use the temporal data in our trace: the estimation process takes into account the number of files provided by each node and how long the node was online. Therefore, it yields different estimates for the average inter-contagion time in the static and dynamic settings – i.e., if we suppose nodes were continuously online the whole period or not. Considering the homogeneous SI model first, we estimate average inter-contagion times of 10 064 seconds (2h48min) in the static setting and 4 926 seconds (1h22min) in the dynamic setting.

Next, considering the heterogeneous SI model, we also have different average inter-contagion time estimates for different settings: similarly to the homogeneous model, individual estimates are also generally greater in the static setting. Indeed, nodes seem less active if we suppose they were continuously online in the whole observation period (since the number of transfers remains the same). An important difference in this model, compared to the homogeneous one is the following: individual average inter-contagion times imply that observed free riders (clients who do not provide files) have null ICT rate estimates. Hence they will also behave as free riders in simulations of this model. The estimated complementary cumulative distributions in both settings (static and dynamic) are plotted in Fig. 11. As noted in section 3, more than 95% of the peers in the system are free riders, and thus, are not represented in the graph.

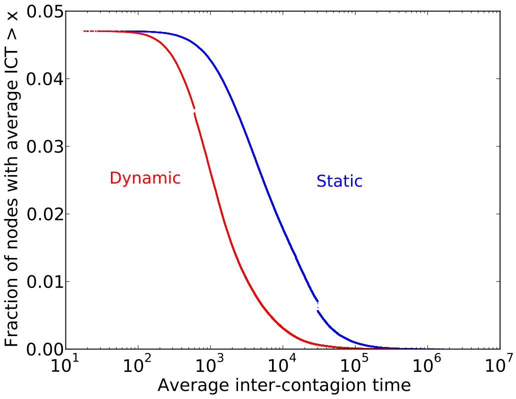


Fig. 11. Complementary cumulative distributions of individual average inter-contagion time estimates for nodes in the static and dynamic interest graphs. Free riders ($>$ 95%) have null inter-contagion time rate and are not shown.

5.3 Results

We have simulated the SI model with homogeneous and heterogeneous spreading behavior as outlined above on the static and dynamic interest graphs for each file present in the trace. The profiles of real and simulated cascades are summarized in Fig. 12: we have plotted the complementary cumulative distributions of cascades' size, number of links and depth. For each cascade property, we plot the same distribution in lin-log and log-log (inset) scales, which highlight respectively smaller/short cascades (most cascades) and bigger/deeper cascades (rare cascades).

The first observation, comparing simulated cascade profiles on the static interest graph and the dynamic interest graph, is that cascades are generally smaller and feature a smaller number of links in the dynamic graph. This is due, in part, to the fact that in the dynamic graph, in contrast to the static interest graph, there are no links between nodes which were never simultaneously online in the trace. In our case, these missing links amount to 29%

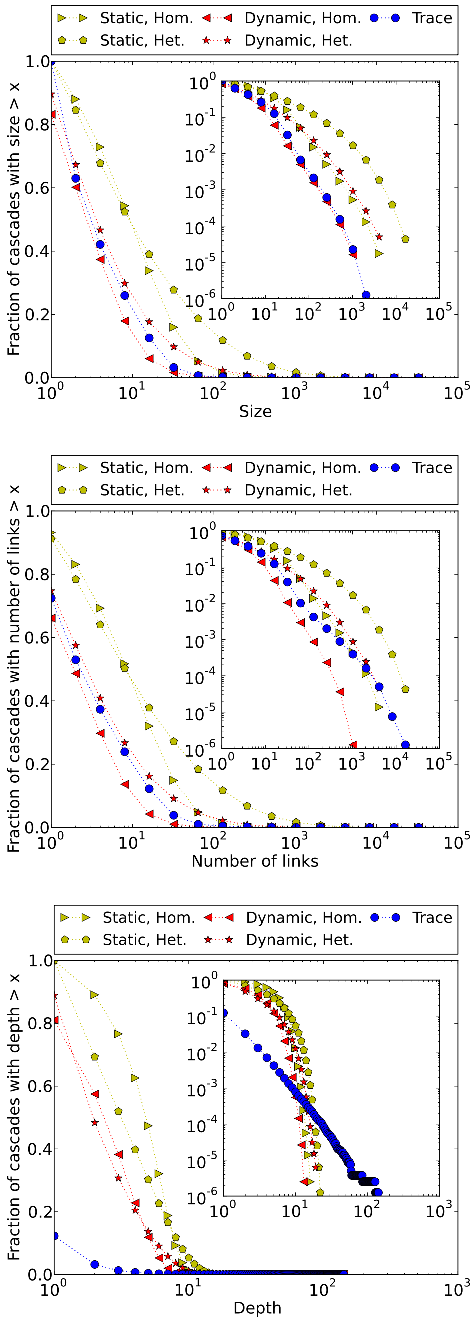


Fig. 12. Spreading cascades profile in terms of size, number of links and depth, respectively. Plots feature the complementary cumulative distribution of these properties in lin-log and log-log (inset) scales. Simulations on the dynamic graph remain closer to real cascades (trace), with the homogeneous model reproducing well real cascades' size and the heterogeneous one, their number of links; no model was able to reproduce the observed depth distribution.

of the links in the static interest graph. So, in order to evaluate their impact, we have also simulated our models on the static interest graph without these links (not shown) and found that the impact was minor: simulated cascades in this new static graph featured the same profile of simulated cascades on the original static interest graph. Thus, we conclude that the difference in cascade profiles simulated on static and dynamic graphs is primarily due to the reduction of the co-presence time among neighbors in the dynamic graph (in the static interest graph the co-presence time correspond to the whole observation period) and potential causality effects. This cascade profile difference is not trivial however since one could have thought that the co-presence time reduction could have been compensated (or overcompensated) by the fact that nodes in the dynamic graph are more active than nodes in the static graph, as discussed previously.

Focusing on the cascade properties, we note that none of the proposed models was able to reproduce the scale-free depth distribution featured by the real cascades; simulated cascades exhibit, in contrast to real ones, a sharp decrease in the proportion of cascades with depth greater than 10. In terms of size and number of links, we find encouraging results: both homogeneous and heterogeneous models perform relatively well in the dynamic setting, in the sense that simulations on the dynamic graph feature a proportion of small cascades similar to the real ones (most cascades). In terms of larger (and infrequent) cascades, the homogeneous model reproduces well the size distribution of real cascades; in terms of number of links, the heterogeneous model is superior.

6 Conclusion and Perspectives

During the last decade, substantial research has been done in the field of p2p systems. Most of the studies focused primarily on measurements or on theoretical modeling. In this work, we took benefit of a new rich dataset obtained by measuring p2p activity to bridge this gap, proposing a data-driven approach to model two important aspects surrounding p2p sharing.

Our measurement study of an eDonkey network has revealed nontrivial properties such as heterogeneous distributions of file popularity and peer activity, Poissonian query and download profiles, high presence of free riders, etc. The study of silent periods revealed that unavailable files are frequent, even for popular files and the profiles of diffusion cascades revealed elongated cascades, with a scale-free depth distribution.

The insights obtained from the dataset enabled us to propose a traffic model based on Markov chains and Poisson processes. The model is able to generate synthetic traffic data that reproduce several key properties of the p2p network, such as for instance the silent periods – that matter for the Quality of Service – which are convincingly reproduced. The model is simple but nevertheless easily extendable to also reproduce the other characteristics of the p2p network. Further exploration of the model has revealed that the presence of free riders above 65% significantly deteriorates the Quality of Service, whereas the network mostly remains unaffected before this fraction.

In the study of diffusion in p2p systems, epidemic models have been used in the literature to reproduce the evolution of the number of infected individuals [Leibnitz et al., 2006, Hosanagar et al., 2010]. In this work we have explored a rich empirical notion of diffusion in the context of p2p systems – namely, file spreading cascades – which not only contains the information on the number of infected individuals, as usual, but also encodes the file diffusion trail. To study those objects and assess the relevance of epidemic/contagion network models which generate diffusion cascades, we have reconstructed the social network of peers in the p2p system using the available temporal data in the trace and integrated it into the spreading model.

Spreading cascades feature a complex structure, which we summarize in terms of three key properties: size, number of links and depth. Previous studies pointed out that these properties are challenging to reproduce with simple spreading models [Bernardes et al., 2012]. Our results are coherent with these findings, in the sense that the SI models we examined were unable to reproduce all these key properties simultaneously. In particular, the depth distribution of real observed cascades is qualitatively very different from corresponding distribution of the simulated cascades. That said, our work demonstrates the benefit of incorporating available temporal data into the models to make simulations more realistic. In particular, we present a framework capable of reproducing the distribution of cascades size or number of links using a reconstructed dynamic social network of peers. We have also shown that assuming homogeneous or heterogeneous spreading behavior impacts the cascade profiles, albeit to a lesser extent than the difference between simulations on the static or dynamic interest graphs.

Although we have explored improvements to epidemic models in this work, they remain based on “push” dynamics whereas peers in p2p systems “pull” content from one another. Thus, we plan to analyze adoption/threshold models in the future, for their spreading dynamic might be more adapted to this context than standard diffusion models currently used.

Acknowledgements

This work is partly funded by the European Commission through the FP7 FIRE project EULER (Grant No.258307).

A How to detect offline and silent periods

Let $\{t_0, t_1, ..., t_n\}$ denote event dates and let $d_i = t_i - t_{i-1}$ for $0 < i \leq n$ be the inter-event times. Here, we think of the $t_i{}'s$ as the download dates from some given peer or for some given file. We assume that if the peer is connected (resp. the file is available) the time between two downloads follows an exponential distribution with parameter α. Similarly, if the peer goes offline (resp. the file becomes unavailable), we assume that the time before the next event also follows an exponential distribution with parameter β. Clearly, α is larger than β. Furthermore, each d_i either follows one or the other distribution. Our goal is, for each d_i, to be able to link it to the most likely distribution in order to detect offline and silent periods in real event sequences.

Given that two distinct exponential distributions alternate randomly, the algorithm from [Jewell, 1982], Section 4, tells us how to find the most likely parameters α and β of the two distributions. Then, the next step is to link every d_i with the right distribution. Let $X \sim \text{Expo}(\alpha)$, $Y \sim \text{Expo}(\beta)$ and let a given inter-event time d_i be equal to D. We can compute the following probabilities:

$$P(X > D) = e^{-\alpha D}$$
$$P(Y < D) = 1 - e^{-\beta D}.$$

Hence, it means that it is most likely that $d_i \sim \text{Expo}(\alpha)$ iff $e^{-\alpha D} > 1 - e^{-\beta D}$. This enables us to conclude.

Of course, in practice, some implementation details must be taken care of. For instance, if α and β are close to each other, it probably means that the peer never went offline or that the file was always available. Additionally, border effects may appear and must be treated appropriately.

References

Aidouni, Frederic, Latapy, Matthieu, & Magnien, Clémence. (2009). Ten weeks in the life of an edonkey server. Pages 1–5 of: 23rd ieee international symposium on parallel and distributed processing, ipdps 2009, rome, italy, may 23-29, 2009.

Andersson, Hakan, & Britton, Tom. (2000). Stochastic epidemic models and their statistical analysis (lecture notes in statistics) (v. 151). 1 edn. Springer.

Azzouna, N.B., & Guillemin, F. (2003). Analysis of adsl traffic on an ip backbone link. Pages 3742–3746 of: Ieee globecom 2003, vol. 1. IEEE.

Ban, Tao, Guo, Shanqing, Zhang, Zonghua, Ando, R, & Kadobayashi, Y. (2011). Practical network traffic analysis in p2p environment. Pages 1801–1807 of: Proceedings of the 7th international conference on wireless communications and mobile computing conference (iwcmc). IEEE.

Barrat, Alain, Barthlemy, Marc, & Vespignani, Alessandro. (2008). Dynamical processes on complex networks. New York, NY, USA: Cambridge U. Press.

Bernardes, Daniel F., Latapy, Matthieu, & Tarissan, Fabien. (2012). Relevance of SIR model for real-world spreading phenomena: Experiments on a large-scale p2p system. Proceedings of the international conference on advances in social networks analysis and mining (asonam). IEEE. Istanbul, Turkey; 2012-08-26 – 2012-08-29.

Clevenot, F., & Nain, P. 2004 (Mar.). A simple fluid model for the analysis of the squirrel peer-to-peer caching system. Ieee infocom 2004, vol. 4.

Feng, Qinyuan, Wu, Yu, Sun, Yan, Jiang, Jing, & Dai, Yafei. (2009). User behavior modeling in peer-to-peer file sharing networks: Dissecting download and removal actions. Pages 3477–3480 of: Proceedings of the 2009 ieee international conference on acoustics, speech and signal processing. ICASSP '09. Washington, DC, USA: IEEE Computer Society.

Ge, Z., Figueiredo, D. R., Jaiswal, S., Kurose, J., & Towsley, D. (2003). Modeling peer-peer file sharing systems. Ieee infocom 2003.

Gomez-Rodriguez, Manuel, Leskovec, Jure, & Krause, Andreas. (2012). Inferring networks of diffusion and influence. Acm trans. knowl. discov. data, **5**(4), 21:1–21:37.

Gummadi, K. P., Dunn, R. J., Saroiu, S., Gribble, S. D., Levy, H. M., & Zahorjan, J. 2003 (October). Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. Proceedings of the 19th acm symposium on operating systems principles (sosp-2003).

Handurukande, S. B., Kermarrec, A.-M., Le Fessant, F., Massoulié, L., & Patarin, S. (2006). Peer sharing behaviour in the edonkey network, and implications for the design of server-less file sharing systems. Pages 359–371 of: Proceedings of the 1st acm sigops/eurosys european conference on computer systems 2006. EuroSys '06. New York, NY, USA: ACM.

Hosanagar, Kartik, Han, Peng, & Tan, Yong. (2010). Diffusion models for peer-to-peer (p2p) media distribution: On the impact of decentralized, constrained supply. Info. sys. research, **21**(2), 271–287.

Hoßfeld, Tobias, Leibnitz, Kenji, Pries, Rastin, Tutschku, Kurt, Tran-Gia, Phuoc, & Pawlikowski, Krzysztof. 2004 (12). Information diffusion in edonkey filesharing networks. Page 8 of: Atnac 2004.

Iamnitchi, Adriana, Ripeanu, Matei, Santos-Neto, Elizeu, & Foster, Ian. (2011). The small world of file sharing. Ieee trans. parallel distrib. syst., **22**(7), 1120–1134.

Iribarren, J. L., & Moro, E. (2009). Impact of Human Activity Patterns on the Dynamics of Information Diffusion. Physical review letters, **103**(3), 038702–+.

Izal, M., Urvoy-Keller, G., Biersack, E., Felber, P., Hamra, A. Al, & Garces-Erice, L. 2004 (April). Dissecting BitTorrent: Five months in a torrents lifetime. Passive and active measurements.

Jewell, N.P. (1982). Mixtures of exponential distributions. The annals of statistics, 479–484.

Karagiannis, Thomas, Broido, Andre, Faloutsos, Michalis, & claffy, Kc. (2004). Transport layer identification of p2p traffic. Pages 121–134 of: Proceedings of the 4th acm sigcomm conference on internet measurement. IMC '04. New York, NY, USA: ACM.

Kleinberg, Jon. (2008). The convergence of social and technological networks. Commun. acm, **51**(11), 66–72.

Latapy, Matthieu, Magnien, Clémence, & Vecchio, Nathalie Del. (2008). Basic notions for the analysis of large two-mode networks. Social networks, **30**(1), 31 – 48.

Leibnitz, Kenji, Hossfeld, Tobias, Wakamiya, Naoki, & Murata, Masayuki. (2006). Modeling of epidemic diffusion in peer-to-peer file-sharing networks. Pages 322–329 of: Proceedings of the second international conference on biologically inspired approaches to advanced information technology. BioADIT'06. Berlin, Heidelberg: Springer-Verlag.

Lerman, K., & Ghosh, R. (2010). Information contagion: An empirical study of the spread of news on digg and twitter social networks. Proceedings of 4th international conference on weblogs and social media.

Leskovec, Jure, McGlohon, Mary, Faloutsos, Christos, Glance, Natalie, & Hurst, Matthew. 2007 (Apr.). Cascading Behavior in Large Blog Graphs. Proceedings of 7th siam international conference on data mining (sdm).

Liben-Nowell, David, & Kleinberg, Jon. (2008). Tracing information flow on a global scale using Internet chain-letter data. Proceedings of the national academy of sciences, **105**(12), 4633–4638.

Locher, Thomas, Mysicka, David, Schmid, Stefan, & Wattenhofer, Roger. (2009). A peer activity study in edonkey & kad. International workshop on dynamic networks: Algorithms and security (dynas).

Menasche, Daniel Sadoc, de Aragao Rocha, Antonio A., Li, Bin, Towsley, Don, & Venkataramani, Arun. (2009). Modeling unavailability in peer-to-peer systems. Pages 375–376 of: Proceedings of the 28th ieee international conference on computer communications workshops. INFOCOM'09. Piscataway, NJ, USA: IEEE Press.

Onnela, J.-P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., Kertész, J., & Barabási, A.-L. (2007). Structure and tie strengths in mobile communication networks. Proceedings of the national academy of sciences, **104**(18), 7332–7336.

Qiu, Dongyu, & Srikant, R. (2004). Modeling and performance analysis of bittorrent-like peer-to-peer networks. Pages 367–378 of: Proceedings of the 2004 conference on applications, technologies, architectures, and protocols for computer communications. SIGCOMM '04. New York, NY, USA: ACM.

Schlosser, Mario T., Condie, Tyson E., Kamvar, Sepandar D., & Kamvar, Ar D. (2002). Simulating a p2p file-sharing network. First workshop on semantics in p2p and grid computing.

Sen, Subhabrata, & Wang, Jia. (2004). Analyzing peer-to-peer traffic across large networks. Ieee/acm trans. netw., **12**(2).

TorrentFreak. (2010). Cisco expects p2p traffic to double by 2014. http://torrentfreak.com/cisco-expects-p2p-traffic-to-double-by-2014-100611/.

Tutschku, K., & de Meer, H. (2003). A Measurement Study on Signaling on Gnutella Overlay Networks. Proceedings of the fachtagung - kommunikation in verteilten systemen (kiVS).

Tutschku, Kurt. (2004). A measurement-based traffic profile of the edonkey filesharing service. Proceedings of the 5th international workshop on passive and active network measurement, pam 2004, antibes juan-les-pins, france. Lecture Notes in Computer Science, vol. 3015. Springer.

Xiangying, Yang, & de Veciana, G. 2004 (Mar.). Service capacity of peer to peer networks. Ieee infocom 2004, vol. 4.

Zhao, Shanyu, Stutzbach, Daniel, & Rejaie, Reza. (2006). Characterizing Files in the Modern Gnutella Network: A Measurement Study. 13th annual multimedia computing and networking (mmcn'06).