Modeling of Epidemic Diffusion in Peer-to-Peer File-Sharing Networks

Kenji Leibnitz[1], Tobias Hoßfeld[2], Naoki Wakamiya[1], and Masayuki Murata[1]

[1] Graduate School of Information Science and Technology
Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
[leibnitz,wakamiya,murata]@ist.osaka-u.ac.jp
[2] Deptartment of Distributed Systems, University of Würzburg
Am Hubland, 97074 Würzburg, Germany
hossfeld@informatik.uni-wuerzburg.de

Abstract. In this paper we propose an analytical model for file diffusion in a peer-to-peer (P2P) file-sharing network based on biological epidemics. During the downloading process, the peer shares the downloaded parts of the file and, thus, contributes to distributing it in the network. This behavior is similar to the spreading of epidemic diseases which is a well researched subject in mathematical biology. Unlike other P2P models based on epidemics, we show that steady state assumptions are not sufficient and that the granularity of the diffusion model may be appropriately selected.

1 Introduction

The volume of traffic transmitted over the Internet has enormously increased recently due to the upcoming of peer-to-peer (P2P) file sharing applications. The most popular applications, such as Gnutella [1], eDonkey [2], or BitTorrent [3], are often abused for illegally sharing copyrighted content over the Internet. In P2P technology, each participant (*peer*) serves simultaneously as client and server which makes the system more scalable and robust and distinguishes it from conventional client-server architectures. However, this also comes at a slight drawback when considering content distribution. Since now, no longer a single trusted server distributes the file, malicious peers (*pollution/poisoning*) [4] can offer fake or corrupted files and disrupt the file dissemination process. On the other hand, this can be also used as a method for the rightful owners of the files to protect their copyrighted property from being illegally distributed.

P2P networks can be briefly classified into *pure* and *hybrid* types [5]. Unlike pure P2P networks, e.g. Gnutella, hybrid networks have additional entities which have special functions. In the eDonkey network, each peer connects to an index server which indexes all shared files and over which the search for a certain file is performed. In a similar manner, BitTorrent uses trackers accessed over WWW pages to provide the information about other peers sharing the file. *Seeders* are peers that offer the complete file for other peers to download. After a file has

been downloaded, the peer may itself become a seeder or a *leecher* who does not participate in the file sharing after downloading it.

The file diffusion process itself is comparable to the spreading of a disease in a limited population. There exist many models for population dynamics in mathematical biology [6] dealing with predicting if a disease will become an epidemic outbreak or what vaccination strategy [7] is most appropriate. Epidemic models are also well suited to model the diffusion behavior of specific information in a network, see [8]. In this paper we will use modeling techniques from biological epidemics to predict the diffusion characteristics of single files shared in a P2P network. While in most papers, e.g. [9, 10], the steady-state network performance is investigated, we emphasize on the time-dynamics of the system which requires us to consider a non-stationary process, e.g. caused by flash crowd arrivals of file requests. Additionally, our model takes the distinction between leechers and seeders into account and we show the influence of selfish peers on the file dissemination process.

2 The eDonkey P2P File-Sharing Application

In the following we will consider a file sharing application similar to eDonkey which belongs to the class of hybrid P2P architectures and comprises two separate applications: the *eDonkey client* (or *peer*) and the *eDonkey server*, see [11]. The eDonkey client shares and downloads files. The eDonkey server operates as an index server for file locations and distributes addresses of other servers.

A main feature of P2P file sharing applications like BitTorrent, Kazaa, and eDonkey is the ability to perform *multiple source downloads*, i.e., peers can issue two or more download requests for the same file to multiple providing peers in parallel and the providing peers can serve the requesting peer simultaneously. Before an eDonkey client can download a file, it first gathers a list of all potential file providers. To accomplish this, the client connects to one of the eDonkey servers. Each server keeps a list of all files shared by the clients connected to it. When a client searches for a file, it sends the query to its main server which may return a list of matching files and their locations. In [12], we showed from measurements that about 50% of the total number of eDonkey users are connected to the seven largest index servers with population sizes N of up to 500,000 peers. This large number allows us to assume a Poisson process for the arrival of file requests. More details on the file sharing process itself can be found in [12].

The general structure of an arbitrary file f that is shared in the eDonkey network is depicted in Fig. 1. The file with a size of f_s kB comprises a number of $c_{max} = \lceil \frac{f_s}{c_s} \rceil$ chunks, each with a constant size of $c_s = 9500$ kB with exception of the final chunk c_{max} which may be smaller in size. A full chunk is not exchanged between the peers in whole, but is transmitted in blocks of size $b_s = 180$ kB.

A block is requested from a peer who shares the whole chunk containing this block. After all blocks of a chunk have been downloaded by a requesting peer, an error detection mechanism is performed. In eDonkey, this is done via comparing the hash value of the received chunk with the sender's hash value of the chunk.

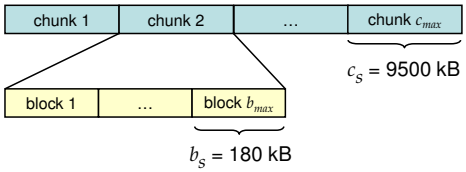


Fig. 1. Structure of a file on eDonkey application layer

In case of an error, i.e., at least one block is corrupted, the complete chunk is discarded and has to be requested and downloaded again.

After a peer has successfully downloaded all blocks of chunk i, he immediately acts as a sharing peer for this chunk and the number of sharing peers is incremented by one. Thus, all users in an eDonkey network may act simultaneously as sharing peers and downloading peers. Although, the user cannot influence that each chunk is shared during downloading, he can show a different behavior after the file has been entirely downloaded. We take this into account in our model by introducing p as the probability that a user shares the file for an exponentially distributed period B. All users in the system use the identical values of p and B. Hence, $p = 0$ indicates a system consisting entirely of *leechers*, i.e., users who only share the file during the download and immediately stop sharing it once the download has been completed.

3 Epidemic Model of File Diffusion

In the following, let us consider a basic epidemic model for P2P file sharing. In general, epidemic models categorize the population into groups depending on their state. A commonly used approach is the SIR model [6]. SIR is an abbreviation for the states that are taken during the course of the spread of the disease. At first, there are *susceptibles*, which are users that can be possibly infected with a certain rate. When they are contacted with the disease, they move to the state of *infectives* and can pass the disease on to other members of the susceptible population. Finally, there is the *removed* population, consisting of users who have either fatally suffered from the disease or have recovered and become immune to it. In either case, they can not get infected by the disease again. An important issue is that the total population N remains constant.

3.1 Analogy of P2P to Biological SIR Model

In this section we will describe the basic underlying biological model and show the commonalities with P2P file sharing. Although there are various analogies between both models, we will see that simply applying an SIR model is insufficient due to the complexity of the P2P applications. However, the principle time-dynamic modeling technique from biology will be maintained and unlike [9] we are able to consider cases that are not in steady state.

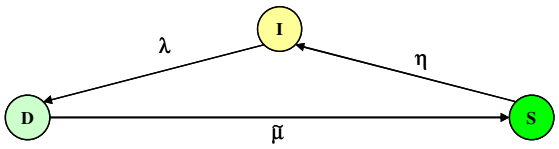


Fig. 2. Simple IDS state space

We denote the number of susceptibles as *idle peers* I at a certain time t. From this set, the file requests are generated with a rate of λ, which can be a time dependent function or a constant reflecting the popularity of the file over time, see [12]. Once the peer starts to download the file, he is attributed to the set of *downloading* peers D. The download rate $\tilde{\mu}$ depends on the number of peers sharing the file and the other downloading peers, which all compete for the download bandwidth. Once downloading of the complete file with size f_s is finished, the peer joins the *sharing* peers S, that offer the file to the other users. The peer shares the file only for a limited time after which he returns with rate η to the idle peers, see Fig. 2. This is a rather simplified view for a generic file sharing application, as the detailed mechanism in eDonkey involves downloading and sharing chunks of the file. Note that all of the above quantities are functions of time, but we will drop the time index in the notation for simplification.

Thus, the dynamic system of the sharing process can be expressed by the equation system given in (1). In analogy to the SIR model, we will refer to it as the IDS model.

$$\frac{dI}{dt} = \eta S - \lambda I \qquad \frac{dD}{dt} = \lambda I - \tilde{\mu} D \qquad \frac{dS}{dt} = \tilde{\mu} D - \eta S \tag{1}$$

The initial values at time $t = 0$ are I_0, S_0, and $D_0 = N - I_0 - S_0$, respectively.

In Eqn. (1) we can at first assume a constant request arrival rate λ which is adapted to match a Poisson arrival process and the main problem lies in the determination of the download rate $\tilde{\mu}$. Let us define the upload and download rates as r_u and r_d, respectively. We assume homogeneous users with ADSL connections, resulting in rates of $r_u = 128$ kbps and $r_d = 768$ kbps. Since eDonkey employs a fair share mechanism for the upload rates, there are on average S/D peers sharing to a single downloading peer and we multiply this value with r_u which gives us the bandwidth on the uplink. However, since the download bandwidth could be the limiting factors, the effective transition rate μ consists of the minimum of both terms divided by the file size f_s, see Eqn. (2).

$$\tilde{\mu} = \frac{1}{f_s} \min \left\{ \frac{r_u S}{D}, r_d \right\} \tag{2}$$

The dynamics of the populations of D and S are shown in Fig. 3 and compared to the mean population sizes, i.e., mean number of peers, obtained from the average over 5000 simulation runs. We selected $S_0 = 5000$, $I_0 = 100$ and a constant λ of 1300 requests per hour. For the sake of simplicity we consider at

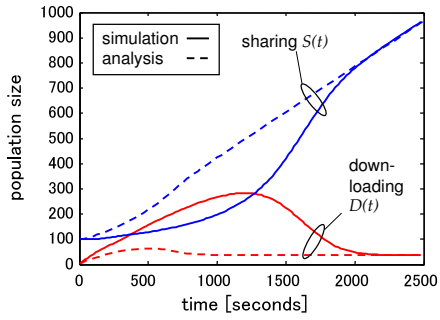


Fig. 3. Comparison of simulation results with basic IDS model

this point $\eta = 0$, i.e., all peers remain sharing peers after a completed download and do not leave the system. The shape of the I curves is not very interesting to us in this scenario, since it just linearly decreases due to the Poisson assumption.

When comparing the simulation with the analytical model, we can see that the same general shape matches for $t > 2000$, whereas a problem arises w.r.t. the accuracy of the model for smaller values of time t. This can be explained as follows. The transition from D to S is performed only after the complete file with fixed size f_s has been downloaded. The current model using the states I, D, and S, however, is memoryless and does not take into account the number of bits that have already been downloaded. The transitions between these states are given here as rates indicating the "average" number of transitions per time unit. In reality, the average download rate changes during the downloading process of an individual peer and it is insufficient to consider it a priori as constant for the complete file. While this assumption is generally applied in epidemic modeling of diseases, we wish to provide an enhanced mathematical model by considering a finer granularity. In the following we will, therefore, minimize the error by splitting the macro state D into M smaller states corresponding to the number of bits downloaded. We expect that when M approaches infinity, the error will be reduced to zero.

3.2 Detailed File Sharing Model

We consider in the following the last downloaded chunk of a file which is the most interesting case, as its completion results in the completion of the entire file. The user can then decide whether the whole file is shared or not, i.e., whether the peer becomes a leecher or a seeder. In the following the terms file and last downloaded chunk will be used interchangeably.

Let us split the file with size f_s into M logical units which we will consider individually. Our model thus increases by the states $D_0, \ldots, D_M$. We can interpret the states D_i as the state where i logical units have been successfully downloaded, i.e., D_0 means that the download is initiated and D_M indicates a complete download. After reception of each block, the queue mechanism of eDon-

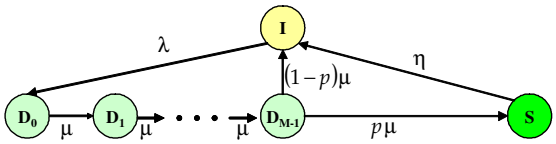


Fig. 4. Detailed IDS state space

key determines the sharing peers from which the next block will be downloaded. This involves an update of the download rate μ after each logical unit. If we choose the logical unit as blocks, our model is exact and the obtained numerical error is acceptably small, cf. Fig. 5(a). The transitions from the states D_i use a rate μ similar to the one described in Eqn. (2).

$$\mu = \frac{M}{f_s} \min \left\{ \frac{r_u \, S}{\sum_{i=0}^{M-1} D_i}, r_d \right\} \tag{3}$$

A further enhancement of the simple model is the introduction of p as the probability of sharing a file. The updated state space with transitions is illustrated in Fig. 4. After the M-th logical unit has been downloaded, the peer enters the sharing peers with probability p and returns to the idle state with $1 - p$. This corresponds to the user leaving the system after downloading (leecher) or downloading it another time again at a later time.

The new equation system is summarized below. The original model given in Section 3.1 corresponds to a value of $M = 1$. Obviously, the larger M is, the more accurate is the model, but the computational requirements for solving the equation system increase as well. Finding a good value of M involves a tradeoff between accuracy and computation speed.

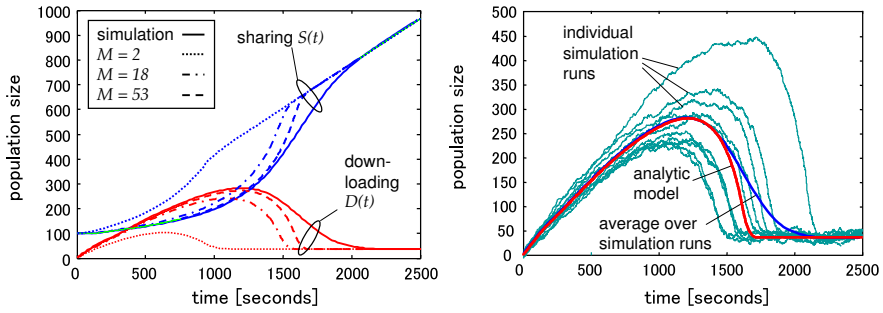
$$\frac{dI}{dt} = (1-p)\mu D_{M-1} - \lambda I + \eta S \qquad \frac{dS}{dt} = p\,\mu\, D_{M-1} - \eta S \tag{4}$$

$$\frac{dD_0}{dt} = \lambda I - \mu\, D_0 \qquad \frac{dD_i}{dt} = \mu\,(D_{i-1} - D_i) \quad \forall_{1 \le i < M} \tag{5}$$

Again, we must include the condition to keep the total population at the index server constant at $N = I + \sum_{i=1}^{M} D_i + S$.

However, since the equation system is a closed system, it is sufficient to ensure that the initial values obey this constraint. Hence, we assume that $N = I_0 + S_0$ and $D_i = 0$ for all i. The considered values for M are 1, 2, 18, and 53, corresponding to the download units of a chunk. Thus, the largest number of equations is when $M = 53$ and the units are blocks as described in Section 2.

The extended model is compared to simulation results in Fig. 5(a). We can recognize that using a large value of M greatly improves the accuracy of the model. Note that the task of comparing results averaged from simulation runs to the mathematical model is not fully appropriate. The differential equations describe the general behavior of a single evolution over time, depending on the initial values and boundary values. We can easily match the initial values, but the



(a) Influence of different values of M (b) Error in average over simulation runs

Fig. 5. Extended IDS model

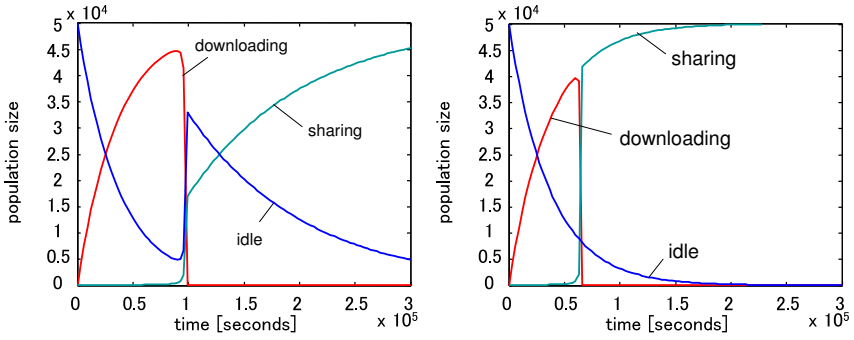
boundary conditions in the simulation depend for example also on the realization of each random variable. Each individual simulation run matches exactly the shape of the analytical model, however, depending on the random variables can be different in scale, see Fig. 5(b). When we average over the series of simulation runs, this leads to the different decreasing slope at about 1700 s in Fig. 5(a).

With our model, we can evaluate the influence of the parameters on the system behavior. In this paper, we focus on the sharing probability p. Two values of p are shown in Fig. 6. In Fig. 6(a), $p = 0.35$ and this percentage of peers becomes seeders right after downloading. The others return to the idle state and download the file again at a later time, only then there are more seeders available which makes the download time very short. The idle users decrease exponentially, since $\eta = 0$ and the sharing users increase accordingly. Finally, all peers will become seeders in spite of p being less than 1. The higher p is, the faster the file is distributed among all peers, see Fig. 6(b).

4 Conclusion and Outlook

We presented an analytical model for file diffusion in an eDonkey-like P2P file sharing network. It is based on an epidemic model like the well-known SIR model, but in our case corresponds to the populations of idle peers, peers currently downloading the file (or chunk), and those sharing it. We could see that using a simple SIR-like model is not very accurate, nor is the steady state assumption found in many publications. We, therefore, considered separate populations for peers having downloaded certain parts of the file and could improve the accuracy of the model when we compared the results to simulations.

The model provides the foundation to investigate many aspects of file diffusion properties. We are especially interested in the effects of pollution in P2P file sharing. Our main objective in the future will be to investigate the influence of peers sharing polluted data on the dissemination process.



(a) $p = 0.35$ (b) $p = 1.0$

Fig. 6. Influence of sharing probability p

Acknowledgement This research was supported by "The 21st Century COE Program: *New Information Technologies for Building a Networked Symbiosis Environment*" and a Grant-in-Aid for Scientific Research (A)(2) 16200003 of the Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

1. Gnutella Protocol Development Website. (http://rfc-gnutella.sourceforge.net/)
2. eDonkey2000 Home Page. (http://www.eDonkey2000.com/)
3. The Official BitTorrent Home Page. (http://www.bittorrent.com/)
4. Liang, J., Kumar, R., Xi, Y., Ross, K.: Pollution in P2P file sharing systems. In: IEEE INFOCOM, Miami, FL (2005)
5. Schollmeier, R.: A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In: IEEE 2001 International Conference on Peer-to-Peer Computing (P2P2001), Linköping, Sweden (2001)
6. Murray, J.: Mathematical Biology, I: An introduction. 3 edn. Springer (2002)
7. Verriest, E., Delmotte, F., Egerstedt, M.: Control of epidemics by vaccination. In: American Control Conference, Portland, OR (2005)
8. Khelil, A., Becker, C., Tian, J., Rothermel, K.: An epidemic model for information diffusion in MANETs. In: 5th ACM MSWiM, Atlanta, GA (2002) 54–60
9. Qiu, D., Srikant, R.: Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In: ACM SIGCOMM'04, Portland, OR (2004)
10. Lo Piccolo, F., Neglia, G., Bianchi, G.: The effect of heterogeneous link capacities in BitTorrent-like file sharing systems. In: Intern. Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P'04), Volendam, The Nederlands (2004) 40–47
11. Tutschku, K.: A measurement-based traffic profile of the eDonkey filesharing service. In: 5th Passive and Active Measurement Workshop (PAM2004), Antibes Juan-les-Pins, France (2004)
12. Hoßfeld, T., Leibnitz, K., Pries, R., Tutschku, K., Tran-Gia, P., Pawlikowski, K.: Information diffusion in eDonkey-like P2P networks. In: Australian Telecommun. Networks and Applications Conference (ATNAC), Bondi Beach, Australia (2004)