Modeling File Popularity in Peer-to-Peer File Sharing Systems

‡Raffaele Bolla, †Mirko Eickhoff, †Krys Pawlikowski, ‡Michele Sciuto

‡Dept of Communication, Computer and System Sciences, University of Genoa, Italy
{*raffaele.bolla, michele.sciuto*}*@unige.it*

†Dept of Computer Science and Software Engineering, University of Canterbury, New Zealand
m.eickhoff@cosc.canterbury.ac.nz, krys.pawlikowski@canterbury.ac.nz

Abstract— **From the birth of Peer-to-Peer (P2P) protocols, Internet traffic has changed its characteristics; the amount of traffic carried by the web has grown exponentially with strong effects on traffic matrixes, both for the distributed nature of P2P protocols and for their intense bandwitdh request. Many research works studied the features and the behaviors of such P2P protocols, simulating the overlay network topologies at the application layer. We present a scalable P2P model, particularly focused on file popularity, considered as the level of interest that a file arouses in the system. The work has the aim of describing the effects that such a quantity generates in the query behavior, assuming that it is strongly linked to marketing procedures. For this reason we make use of the product life cycle in marketing environment to model such level of interest. The model can represent huge overlay networks, up to 500000 peers. The properties of such a model are studied considering the resulting file query distribution, by running multiple independent replications for every simulation. The number of queries performed by every peer and the peer' uptime are also considered, in order to validate the model, comparing simulation output with the measurement results in Gnutella network.**

Keywords— **Distributed systems; File-sharing applications; Peer-to-Peer modeling; Traffic simulation; Internet traffic; Evolution of quantiles.**

I. Introduction

In the last years, Internet technologies and infrastructures have experienced deep advancements and evolutions to meet the increasing user requirements and to support new application requests. It is well known that the major part of traffic carried by Internet is related with a relatively small percentage of classical Internet data applications (i.e., Web, e-mail, FTP, etc.), and that it is basically generated by file-sharing applications. Reference [1] outlines that this kind of applications, based on the P2P technology, is generally characterized by a fast and epidemic spreading and a bandwidth-intensive nature, and generates about 60-80% of the overall Internet traffic. In this scenario it is really important modeling/simulating this kind of applications. One of the difficulties in developing effective P2P simulators/models is the large size of real overlay networks, so the scalability is a fundamental issue. With this respect, we have already presented in [2] a basic framework of a P2P simulator, mainly focused on scalability, which is able to represent huge overlay networks (we have successfully tested it with up to 500000 peers). In this work we are mainly interested in reporting an important improvement of that first proposal obtained by introducing a new and more detailed model for the query generation process , which can better represent the file popularity dynamic, without decreasing system scalability. The query generation process is one of the most significant elements of a P2P model. Its correct representation strongly depends on the precise description of file popularity dynamic, i.e. basically the evolution of the user interest with respect to different files. Many modeling and simulation works, such as [3], [4], [5], [6], [7], [8], [9], can be found about the Gnutella network and, in general, about P2P overlay networks, but none of them describes the popularity evolution with sufficiently accurate results from simulations or measurements. An interesting way of modeling file popularity can be found in [10], where two different classes of files are taken into account, for generating different file request arrival rates. In a P2P system, such as Gnutella, most of files belong to music or video applications, as reported in [11]. These files arouse a lively interest mainly because they can be obtained for free through the P2P system, even if they have a value on the conventional market. So, the success of P2P systems has to be considered not just in reliability and scalability [12], which improve the sharing of contents, but it has also to be found in economic reasons. Interesting goods are freely available in the network, besides the download processes are faster than in client/server systems and they are based on a reliable service. This motivated us to take marketing issues into account when modeling file popularity in a P2P system. Measurements performed by Reza et al. (see [11]) report a particular and interesting behaviour of the file popularity, which has many similarities with the product life cycle (PLC) behaviour reported in marketing literature [13]. For this reason, we have decided to use the product life cycle adopted in marketing environment to model the level of interest for the files. In this work we describe this new approach and we report some numerical results that show the effectiveness of our proposal.

The paper is organized as follows. Section II describes the proposed P2P simulation model. Section III presents the model results, with a focus on the validity of the model. Section IV explains the statistical analysis we are using for studying results, running multiple parallel replications and considering the time evolution of quantiles. In Section V we show the computation time and the utilization of memory required by the simulator. Finally, in Section VI the conclusions are reported.

II. Model Description

In this section we present the features of the proposed model; the general framework has been already presented in [2]. The application layer has been enriched with some new features, in order to represent P2P traffic in a complete way. The most important improvements regard the birth of new files in the system and the introduction of a new model for file popularity and query mechanism. We took Gnutella 0.6 as a reference system, which relies on a hierarchical unstructured overlay network. We decided to model the Gnutella protocol because it is one of the most used P2P protocols and also because it is well described in literature; moreover there are many measurement results in Gnutella network [11], [14] and [15] we found useful both as input data and for validation. In this Section we provide a general description of the model; more detail can be found in [2].

A. Simulator General Description

The most significant overlay network elements of a file-sharing application are peers, ultra-peers and files. Peers represent the leaves of the overlay network; ultra-peers have the same properties of peers, but they also have to satisfy the queries; files represent the content, which interests peers; see [2]. The developed simulator tries to represent two dynamic processes of a Gnutella P2P network: the overlay network evolution process and the query process. The first one models the ingresses and egresses of peers in and out of the overlay network. The arrival of a new peer causes the creation of new relationships in the overlay network and the potential associations of newly shared file copies to the peer. On the contrary, when a peer exits the overlay network, all its relationships and its shared files have to be erased from the system. The query process is reported in one of the next sections. A new dynamic and more precise model for file popularity than in [2] has been introduced. The structure of the simulation tool can be divided roughly into: the initialization procedure, the overlay network and its dynamics, the birth of new files in the system, the file popularity model, and the query process model.

B. Initialization Procedure

In this Subsection we describe the initialization of the main quantities of the simulator:

- total number of peers and ultra-peers,
- total number of different files in the networks,
- number of file replicas,
- peer and ultra-peer relationships.

The initial numbers of peers and ultra-peers are input parameters. The initial number of replicas is a specific parameter associated with each different file. We assign this parameter to each file, according to the Zipf distribution, as reported in [6]. For instance, starting with 5000 files, the resulting number of total files, considering also replicas is 103387. It is assumed that each peer can maintain no more than one copy of each file. In this way it is clear that the maximum number of replicas for every file is bounded by the total number of peers. We consider an average number of 15 neighbors for every ultra-peer. Each peer has one relationship with an ultra-peer. This ultra-peer is selected randomly with uniform probability among the ones that have less than 150 connected leaves. The overlay network is initialized considering the configuration parameters, and with the strong boundary that every ultra-peer needs at least one neighbor, in order to provide connection for every peer in the network. At this step we initialize the simulator using a specific developed software; the measures performed by Reza et al. in [11], and their conclusions, gave us the input to use real parts of the Gnutella network for initializing the simulator. We actually can acquire real portion of the Gnutella network with a crawler, which we developed for teletraffic classification (see [16]).

C. Overlay Network and its Dynamics

In this sub-section we describe the events provided by the simulation tool to represent the overlay network dynamics, i.e. the login and logout processes related with ultra-peers and leaves. An ultra-peer can enter the system in case of a new ultra-peer's birth or in case of a peer's election to an ultra-peer status. The model adds a new ultra-peer element in the overlay network by performing the creation of the neighbour list for the ultra-peer. The new ultra-peer joins the network by contacting a randomly chosen ultra-peer. This ultra-peer forwards the list of its neighbors to the joining ultra-peer, which contacts immediately the ultra-peers from the list. Every neighbor's list gets an update, derived from the ultra-peer's presence. An ultra-peer logging out causes the assignment of its leaves to a randomly chosen ultra-peer. Also its neighbor list is assigned to another ultra-peer. A leaf logging in provides the assignment of a number of files to share. The peer joins the overlay network by establishing a relationship with an ultra-peer. It is selected randomly with uniform probability from the ones that have less than 150 connected leaves. The number of files shared by such a peer, is determined by following the distribution introduced in [2], which we report here: 25% of the total number of peers share zero files (these peers are called *free-riders*), 70% of peers share up to 100 files and 5% of users share more than 1000 files. A peer logging out causes a decrement in the number of replicas, for all shared files. For this reason the ultra-peer, which was connected to the peer, needs to update its list of leaves. The churning of overlay network is simulated considering the birth of peers and ultra-peers in the system. As in [2], exponential distributions are used to generate the birth and the death of peers and ultra-peers in the system.

D. File Popularity and Product Life Cycle in Marketing Envvironment

File popularity in Gnutella network is measured in [11]. As reported there, file popularity can be defined as *as the percentage of successfully contacted peers with the file. Given the random distribution of files among peers, the popularity can be interpreted as the probability of having that file at a random peer.* They define *the change in popularity of a given file over interval τ as the difference between its popularity at the beginning and the end of such an interval (in percentage points).* It is clear that this definition involves the fact that popularity can be produced by the system. Considering that file popularity cannot be measured like a physical parameter in the system, the assumption made by Reza et al. is a good way to estimate this parameter in the Gnutella network. Analyzing the measurement results in [11] it is possible to find an interesting similarity between these graphs and the PLC behavior in marketing literature. We could link these apparently disconnetted quantitites taking into consideration that the success of P2P systems has also got economic reasons, as already mentioned before. In [13] a visualization of a product life cycle for an industry is shown. The curve expresses the total market sale that is the *industry product life cycle*. All products have a certain lifetime, during which they pass through certain stages. Their lifes begin with their market introduction. Then they go through a period during which their market grows rapidly. Eventually, they reach market maturity after which their market declines until the end of their life. The exact path traced by the product life cycle varies from case to case. Marketing theory cannot provide a general analytical function for this curve, because of the big differences between different goods. Each product in the market follows its own behavior. Taking account of these properties, and the measurement results reported in [11], we decided to model the popularity of files in the P2P system using the behaviour of product life cycle (PLC) in marketing environment. We are assuming the presence of two different types of files, as suggested by [10]. The files already present in the system are considered as having a constant popularity. They are modeled as old files, belonging to the system for a long time. Every time a peer decides to share a new content, a new file joins the network during the simulation. These files are considered as *new* and they are characterized by a popularity depending on time. The interest they generate in the system is heavily time dependent. As reported in [11] file popularity is assumed to have a life cycle of one year, and it can change with a daily granularity. We assumed that a file could join the network at every instant of its PLC. This means it can appear in the system in the first or in the final stage of its popularity cycle. This temporal instant, describing the position of the file in its PLC at the time of its birth, is defined as T_{pop_0}. Every time a birth of file occurs T_{pop_0} is randomly chosen in a set of 365 integer numbers. These integer numbers represent the number of days in a time period of one year. Another important parameter we consider for every file is its age in the P2P system, which clearly depends on the instant such a file is born in the system, denoted as T_{birth}. If we define T_{sim} as the current simulation time, T_{age} can be expressed like $T_{age} = T_{sim} - T_{birth}$. The analytical function

$$\Psi(\theta) = \begin{cases} (e^{\theta/183} - 1)/(e - 1) & \text{if } \theta \in [0, 183) \\ 182/\theta & \text{if } \theta \in [183, 365) \\ 0.5 & \text{if } \theta \in [365, \infty) \end{cases} \tag{1}$$

is a good approximation both for the PLC (shown in [13]), and for the measurements results [11]. They report a growing exponential behavior and, after an average period of 183 days (six months), a smoother decrease (see Fig. 1). It is important to underline that each file follows its own model of popularity behavior. The best way to model popularity should be to consider the different types of files present in a P2P system, measuring the file request arrival rates for each category (video, music, software, documents...) and choosing a correct popularity function. We decided to model the query event with reference to each file, and not to the peers, as it is usually done in P2P simulators [17]. At the beginning of the simulation the first query is scheduled for every file. After the query has been performed, a new query is scheduled, following a cyclic procedure. We chose to model the query process with such a technique in order to manage its behavior with respect to file popularity, as already suggested in [10]. The query interarrival time for every old file is modeled with the exponential distribution. If we define the random variable A_i as the query interarrival time for the old files, its expected value can be expressed like $E[A_i] = 1/\lambda$. The cumulative distribution function is given by $F_{A_i}(x; \lambda) = \mathrm{Exp}(x; \lambda) = 1 - e^{-\lambda x}$. We can define the random variable A'_i as the query interarrival time for the files born during the simulation. We model A'_i with the same exponential distribution, modulated by the popularity function. The cumulative distribution function, for the file request interarrival times is given by

$$F_{A'_i}(x) = \mathrm{Exp}(x; \lambda/[1 - \Psi(\tau)]), \tag{2}$$

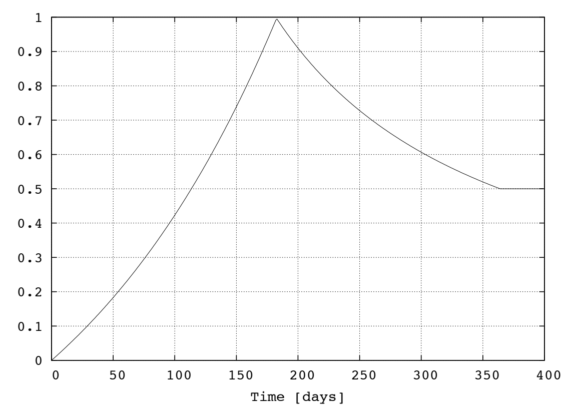


Fig. 1. Popularity function

the expected value of A'_i is

$$E[A'_i] = \frac{[1 - \Psi(\tau)]}{\lambda}, \tag{3}$$

where Ψ is the popularity function shown in (1) and

$$\tau = T_{pop_0} + T_{age}. \tag{4}$$

So, during the simulation the query inter-arrival time for each new file follows the behavior of the popularity function. τ expresses the popularity time dependency. At the birth of a new file, $\tau = T_{pop_0}$ which describes the position of the file in its PLC at the time of its birth (as reported in II-D). File popularity shifts on time depending on the age of the file, expressed by the term T_{age} in (4). At the maximum of the popularity function, a given file becomes very popular. Thus the inter-arrival time of queries gets minimal to allow a high frequency of requests. We set an average arrival rate λ for old file requests equal to 5 requests/day. The resulting maximum value of the file request arrival rate is 915 requests/day, which fits well with measures, taking into account also the results and the model presented in [10]. Values of popularity after 1 year (over the domain of the popularity function) are considered as constant. The constant value of popularity is set to 0.5 (457.5 requests/day on average), after the end of their PLC the new files will double the popularity of the old files.

During the file query process, the peer processing the query is chosen randomly among the ones joining the overlay network. When the file is queried, two conditions must be satisfied: the file must not be present in the list of shared files, nor in the list of non-shared but owned files, maintained by the peer. If these conditions are satisfied, the query process comes to a good end and the requesting peer can ask its ultra-peer to search the file. During search process the involved ultra-peer receives the requested file ID. It searches the file at one-hop distance, contacting its leaves and the directly connected ultra-peers. The neighbouring ultra-peers receive the file ID and forward it to their leaves. The process terminates when the information is completely received from the peer. The final result is a list of peers at one-hop distance who share the requested file. If no peer is found, the process is repeated also for two-hop distance ultra-peers, which search the file in their attached leaves. When the list of peers sharing the file is received, the best peer must be selected. This choice is based on two important parameters: the amount of bandwidth of the peer and the number of download processes the peer is satisfying. The selected peer is the one having the largest ratio between the available bandwidth and the number of downloads. When the file is received by the peer, the requesting user can choose whether to insert the file in the list of shared files or not to share it. The number of replicas of the file is incremented or not, according to this choice. The decision to share a file is determined with a probability of 50%. This probability is zero in the presence of free-ride peers, which share no files. The peer which decides not to share the file is added to the list of non-shared files. The file is not shared and the peer cannot request it anymore.

III. Simulation Results

Following we present the effects of such a model of popularity on the system. We considered a huge network with 250000 peers, 2500 ultrapeers and 5000 single files. The total number of files at the beginning of the simulation, considering replicas is 103387.

The results we show in this Section are not just limited to popularity. Initially we report the number of queries measured over whole simulated time received by two particular files: the first one ("file 1" in Fig. 2) at the time of its birth stands at the beginning of its PLC ($T_{pop_0} = 0$ days). For this reason the file will reach its peak of popularity in 182 days, after that its popularity will decrease. The second considered file has a T_{pop_0} almost at the end of its PLC; the effect is a linear behavior in the number of received queries during the simulation. In this case the simulation time is 1 year. The same (linear) behavior is followed by files already present in the system at the beginning of the simulation, the rate for these files is described by the exponential distribution.

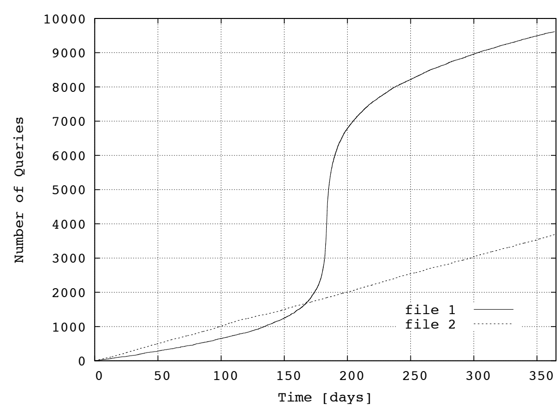


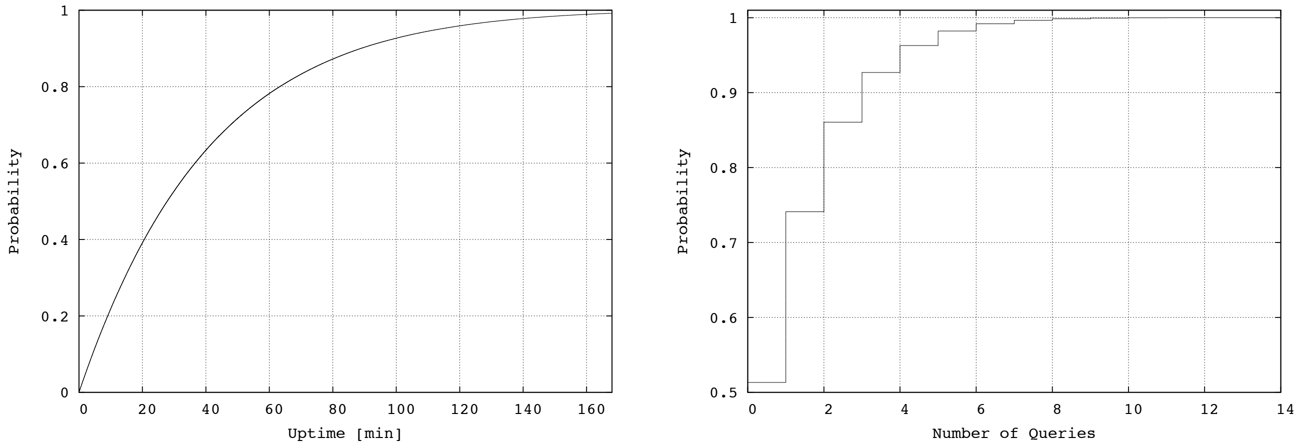
Fig. 2. Number of Queries received by files with different popularities

In Figures 3(a) and 3(b) the behavior of queries is considered from the peer's viewpoint. We report the cumulative distribution function for the peer' uptime and for the number of queries done by every peer. The details about the simulation are reported in TABLE I.

Using these rates for the events, we can study the system in a realistic way, respecting the high dynamism of a P2P system. In one week of simulation time, the number of peers joining the system is bigger than 10^6. Analyzing the figures, it is possible to see the validity of the model. The number of queries performed by

Query for the Old Files	5 every day
Birth of Files	1 every 40 sec
Birth/Death of Peers	5000 every hour
Birth/Death of Ultrapeers	1 every 150 min
Simulation Time	7 days

TABLE I: Interarrival rates of the events and simulation time.



(a)Uptime of the peer

(b)Number of queries performed by peers over whole simulated time

Fig. 3. Empirical CDF of observed measures

every peer respects the results of measurements published in [15]. The results in Fig. 3(a) fits well with the distribution reported in [14] for the peer' uptime. It is important to underline that the first aim of such a model is to maintain a good level of scalability. We can also claim that these results confirm the measurements done in the Gnutella network. In particular we started from modeling file popularity, and scheduling the requests with a behavior depending on time. It can be validated by looking at the query behavior from the peer' viewpoint.

IV. Time Evolution of Quantiles

In this Section we present the analysis performed using a proprietary software, developed at the Computer Science Department of the University of Canterbury, New Zealand. This software estimates the evolution of quantiles over time with controlled error. We analize the correctness of our model, by considering the time evolution of quantiles of query interarrival times. What we want to verify is the behavior for the file born during the simulation. The software we used can study the time evolution of quantiles, running a number of parallel simulations at the same time. From the final results in Fig. 4(a), 4(b) and 5, we want to verify that the query interarrival time still follows the exponential distribution which we use at the beginning for scheduling processes. Most simulation output analysis is confined to the estimation of mean values. However, the estimation of quantiles provides a deeper insight into the simulated model. Quantile estimation can answer questions like: What is the probability of a query interarrival time longer than t? Questions of this kind are often of more interest than the average system behaviour. A set of several quantiles can be used to approximate a probability distribution function. The analysis of the time evolution of these quantiles as the simulation progresses provides deeper insight into the transient behaviour of the system of interest.

A. Method of Estimation

We use the method of quantile estimation that was proposed in [18] and [19]. This method can show the evolution of a set of quantiles of $F_{A'_i}(x)$ over time, i.e. for increasing i. This enables us to compare the results of our simulator with the expected behaviour given by Equation (2), especially the influence of $\Psi(\tau)$ will be evident. The q-quantile $x_q = F^{-1}_{A'_i}(q)$ can be estimated on the basis of an independent and identically distributed random sample of A'_i, which is provided by multiple independent replications of the same simulation run. The half width of a confidence interval of the estimate $\hat{x}_q$ can be described by:

$$\hat{x}_q \in x_{q \pm \epsilon_q}. \tag{5}$$

ϵ_q is an interval in the range of the probability (see [20]). Note, $q \pm \epsilon_q$ should not exceed the bounds 0 and 1.

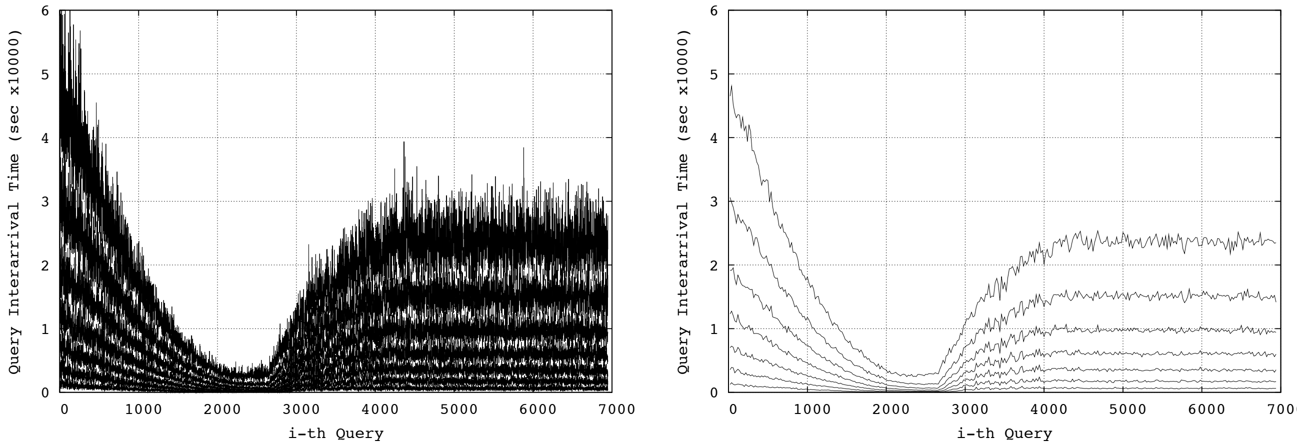
Consecutive quantiles q_k and q_{k+1} are selected automatically so that their confidence intervals do not overlap:

$$\begin{aligned} q_k < 0.5: &\qquad q_k - \epsilon_{q_k} = q_{k+1} + \epsilon_{q_{k+1}}, \\ q_k > 0.5: &\qquad q_k + \epsilon_{q_k} = q_{k+1} - \epsilon_{q_{k+1}}. \end{aligned}$$

This selection starts with the median and searches in both directions, $q_k < 0.5$ and $q_k > 0.5$, for more possible quantiles. The number of selected quantiles depends mainly on the size of the random sample. Non-overlapping confidence intervals avoid high correlation between the estimated quantiles.

B. Query Interarrival Time

The query interarrival times A'_i of the ith query form a sequence of random variables, where $1 \leq i < \infty$. All A'_i are unequally distributed, i.e. $F_{A'_i}(x) \neq F_{A'_{i+\Delta}}(x)$, because they are influenced by $\Psi(\tau)$. To assure a correct behaviour of our simulator we analyse the evolution of $F_{A'_i}(x)$ for increasing i. For small i a long interarrival time is expected, i.e. A'_i should show relatively large values. With increasing i up to the maximum of the file's



(a)Quantile estimates. (b)Smoothed quantile estimates.

Fig. 4. Time evolution of quantiles of $F_{A'_i}(x)$ for a file with $T_{\mathrm{pop}_0} = 0$ days. The query interarrival time depends on file popularity. The end of the PLC is reached in one year of simulation time, roughly around the 4500th query

popularity the interarrival time A'_i should decrease to a low level, implying that a relatively large number of queries take place. After this period A'_i should grow again with increasing values of i because the popularity is shrinking.

In Figure 4 we depicted quantiles of $F_{A'_i}(x)$ for increasing i. In this case we considered a small network with 2500 peers, 25 ultrapeers and an initial number of files equal to 50 (594 considering also replicas); the interarrival query time concerns a file with $T_{pop_0} = 0\ days$. Using $\alpha = 0.1$ and 99 independent replications 7 quantiles could be selected with non-overlapping confidence intervals. All results were obtained with a controlled statistical error given by the halfwidth of the confidence interval of the final estimates not greater than 10% in the probability domain at the 0.9 confidence level. Figure 4(a) shows the original quantile estimates, whereas a smoothed curve for every quantile is shown in Figure 4(b). The original estimates are smoothed by averaging 20 consecutive values. As we can see, the behaviour is as expected. The curve of the quantile estimates of A_i start and finish on a relatively high level indicating a low popularity. In between they are on a low level caused by higher popularity. The big variance in the behavior of each single quantile is because interarrival times are independent of each other.

We verifed the exponential distribution of A'_i at the first and the 364th day of the simulation, with $\Psi(1) = 0$ and $\Psi(364) = 0.5$ (see Fig. 5). The smoothed quantiles and their confidence intervals are checked against $F_{A'_i}$. As one can see, the estimates are as expected.

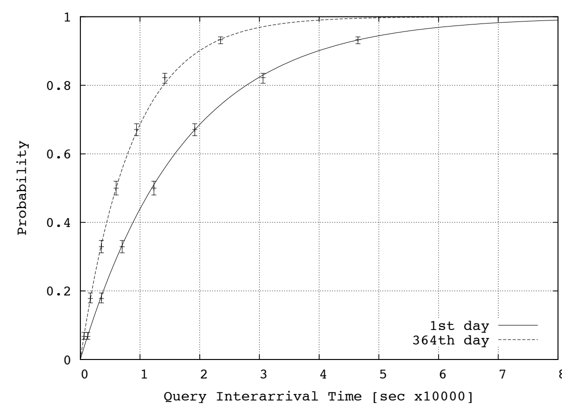


Fig. 5. Interval estimates of quantiles with $\Psi(1)$ and $\Psi(364)$ checked against the expected cumulative distribution function.

V. Some Comments about Scalability

Finally, we report the computation time and the memory utilization values needed by the simulator. The simulated overlay network is the one we used for the previous results (overlay network with an initial number of 250000 peers, 2500 ultrapeers and 5000 files). The considered simulation time has been fixed to 1 year. We have performed all the simulations using a server equipped with a CPU Dual Intel(R) Xeon(TM) 3.00 GHz and with 4 GBytes of memory. The required computation time is 72 hours, which reveals a good level of scalability also in case of huge overlay networks. The memory utilization is always less than 10%.

VI. Conclusions

We presented a scalable P2P model, particularly focused on file popularity. The work describes the effects that such a quantity generates in the query behavior, assuming that it is strongly linked to marketing procedures. The number of queries performed by every peer and the peer' uptime are considered in order to validate the model, comparing simulation output with the measurement results in Gnutella network. The properties of such a model are studied considering the resulting file query distribution, by running multiple independent replications for every simulation. The model can represent huge overlay networks, up to 500000 peers. Future works will include the use of such a simulative model for studying the impact of P2P traffic on the network, driving our work in the application of this model for network planning reasons.

References

[1] S. Sen, J. Wang. Analyzing Peer-to-Peer Traffic Across Large Networks. ACM Transactions on Networking, Vol. 12, No. 2, pp. 219-232, April 2004.

[2] Raffaele Bolla, Roberto Bruschi, Franco Davoli, Andrea Ranieri, Riccardo Rapuzzi, Michele Sciuto. A Simulative Framework to Estimate P2P Performance Indexes and Traffic Matrixes. In Proceedings of the 2nd European Modeling and Simulation Symposium (EMSS06), Barcelona, Spain, October 2006.

[3] J. Hess, B. Poon. Improving Performance in the Gnutella Protocol. Available online at http://www.cs.berkeley.edu.

[4] M. Karakaya, I. Korpeoglu, . Ulusoy. A General Purpose Simulator for Gnutella and Unstructured P2P Networks. Technical Report BU-CE-0505, Department of Computer Engineering, Bilkent University, 2005.

[5] Q. He, M. Ammar, G. Riley, H. Raj, R. Fujimoto. Mapping Peer Behavior to Packet-level Details: A Framework for Packet-level Simulation of Peer-to-Peer Systems. In Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MASCOTS 2003), Atlanta, GA, USA, October 2003, pp.71-78.

[6] T. Schlosser, T. E. Condie, S. D. Kamvar. Simulating A File-Sharing P2P Network. In Proceedings of the 1st Workshop on Semantics in P2P and Grid Computing (SemPGRID03), Budapest, Hungary, May 2003.

[7] PeerSim: A P2P Simulator, accessed 01-May-2006. [Online]. Available: http://peersim.sourceforge.net/

[8] PlanetSim: An Overlay Network Simulation Framework, accessed 01-May-2006. [Online]. Available: http://planet.urv.es/planetsim/

[9] NeuroGrid, accessed 01-May-2006. [Online]. Available: http://www.neurogrid.net/php/index.php

[10] T. Hofeld, K. Leibnitz, R. Pries, K. Tutschku, P. Tran-Gia, K. Pawlikowski. Information Diffusion in eDonkey Filesharing Networks. In Proceedings of the 2004 Australian Telecommunication Networks and application Conference (ATNAC 2004), Sydney, Australia, December 2004, pp. 390-397.

[11] Daniel Stutzbach, Shanyu Zhao, Reza Rejaie. Characterizing Files in the Modern Gnutella Network. To appear in the Proceedings of Multimedia Systems Journal, 2007. Technical Report CIS-TR-06-09, University of Oregon, July 2006.

[12] Dongyu Qiu and R. Srikant. Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. In Proceedings of the 2004 Conference on Applications, technologies, architectures, and protocols for computer communications, Portland, Oregon, USA, 2004

[13] E. W. Candiff, R. R. Still. Basic Marketing, Concepts, Decisions and Strategies. Second edition, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1971.

[14] Daniel Stutzbach, Reza Rejaie. Understanding Churn in Peer-to-Peer Networks. Proceedings of ACM SIGCOMM/USENIX Internet Measurement Conference, Rio de Janeiro, Brazil, October 2006.

[15] Klemm, A., Lindemann, C., Vernon, M.K., Waldhorst, O.P. Characterizing the query behavior in peer-to-peer file sharing systems. In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, Taormina, Sicily, Italy, October 2004, pp. 55 - 67.

[16] Raffaele Bolla, Riccardo Rapuzzi, Michele Sciuto. Monitoring and Classification of Teletraffic in P2P Environment. In Proceedings of the 2006 Australian Telecommunication Networks and application Conference (ATNAC 2006), Melbourne, Australia, December 2006.

[17] Stephen Naicken, Anirban Basu, Barnaby Livingston, Sethalat Rodhetbhai, Ian Wakeman. Towards Yet Another Peer-to-Peer Simulator. In Proceedings of the Forth International Working Conference in Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs '06), West Yorkshire, U.K., September 2006.

[18] M. Eickhoff, D. McNickle and K. Pawlikowski. Depiction of Transient Performance Measures using Quantile Estimation. Proceedings of the 19th European Conference on Modelling and Simulation, pp. 358-363, 2005

[19] M. Eickhoff, D. McNickle and K. Pawlikowski. Analysis of the Time Evolution of Quantiles in Simulation. International Journal of Simulation, Vol. 7, No. 6, pp. 44-55, 2006

[20] E. J. Chen and W. D. Kelton. Simulation-Based Estimation of Quantiles. Proceedings of the 1999 Winter Simulation Conference, pp. 428-434, 1999