An Analytical Modelling Technique for Computing Transfer Time Distributions in Peer-to-Peer Networks

R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno
Dipartimento di Informatica, Università di Torino,
Corso Svizzera, 185, 1049 Torino, Italia
{rossano,marcog,manini,matteo}@di.unito.it

Abstract— **This paper presents an analytical modeling technique for the evaluation of the distribution of transfer times in file sharing peer-to-peer (p2p) applications. Transfer times in such applications follow the resource search phase and depend on parameters such as the size of the resource to be transferred, the number of peers holding a copy of the requested resource, the selection criteria employed by the requesting peer when multiple peers hold the requested resource, the constraints posed by each peer (i.e., maximum number of concurrent downloads and uploads, maximum bandwidth dedicated to download and upload operations, variability of the available bandwidth due other peers that are simultaneous downloading from the selected peer). Neglecting the state of the underlying communication network, the proposed analytical modeling technique accounts all these aspects and provides an estimation of the distribution of the file transfer time after the search for a given resource has been performed. The technique is based on the combined use of first-order Fluid Stochastic Petri Nets (FSPN), queuing systems, and combinatorial manipulations allowing the derivation of the distribution of transfer times in p2p file sharing applications. Parameters of the models are derived from published measured data on the characteristics of p2p users and traffic. Numerical results are presented to prove the flexibility and the potentialities of the proposed technique.**

I. Introduction

File sharing applications are generating an increasing fraction of the traffic on Internet. These applications are based on the peer-to-peer (p2p) paradigm where components of the application can act both as clients and as server in requesting and providing a service. Several p2p file sharing applications have been developed so far, e.g., Napster [1], Gnutella [6], Freenet [5], Kazaa [3], WinMX [2].

The peer acting as a server (the *server-like peer*), accepts and forwards incoming queries for the search of resources (files), provides response to search queries, and serves requests when selected by clients. The peer acting as a client (the *client-like peer*), alternates between the search of a resource and the transfer of the resource (downloading) from a server-like peer. Following a successful search, p2p applications provide the client-like peer a list of peers holding a copy of the requested resource: depending on the particular p2p application, additional information describing the peers are included such as bandwidth between the server-like peer and its Internet Service Provider (ISP), number of client-like peers that are using this server-like peer, and other information the client-like peer may use for its server-like peer selection policy.

Both the search and the transfer phase may be time consuming: the search time is mainly influenced by architectural characteristics of the particular p2p application such as signaling, routing, searching protocols. On the contrary, the transfer time of a resource is mainly dominated by client-like peer bandwidth (last mile), server-like peer outgoing bandwidth (first-mile), maximum number of allowed concurrent downloads set by server-like peers, the number of server-like peers holding a copy of a resource as a function of its popularity, the size of the resource to be downloaded, the competing load on the server-like peer that is experienced by a client-like peer during download, the server-like peer selection policy when multiple peers hold a given resource, and the latency along the path connecting the two peers.

In this paper we develop an analytic modeling technique for the analysis of p2p file sharing systems with the aim to provide Quality of Service (QoS) user-perceived measures that are related with the transfer phase for a given resource; in particular, we provide a method for the estimation of the distribution of the file transfer time. This analysis is both general (since it might be applied to different p2p file sharing architectures) and flexible enough to be adapted to the analysis of other p2p applications, e.g., streaming content distribution, information management for vehicular traffic.

We develop a hybrid modeling technique based on the combined use of Fluid Stochastic Petri Nets (FSPN) [10], queueing systems, and combinatorial analysis. The joint use of different modeling paradigms allows to capture several features that dominate the resource transfer time as well as to obtain an efficient model solution. Parameters of the models we develop are obtained from measured data on p2p applications presented in the literature.

The outline of the paper is the following: in Section II we briefly summarize the related work on the subject of analytical models of p2p systems including a description of measure based work we exploit to match parameters of the models we develop. Section III discusses the main issues dominating the transfer time for a resource that are captured by our analytical modeling technique that is illustrated in details in Section

IV. Preliminary results of the model analysis are presented in Section V while Section VI draws conclusions and outlines several development of the current work.

II. Background and Related Works

The literature on performance issues related with p2p applications is quite vast and therefore it is difficult to provide here a comprehensive summary of previous contributions. In this section we only summarize some of the results appeared so far in this field and that are most closely related or that influence our work.
Several works has been performed for characterizing the properties of p2p file sharing and a large number of these works are based on measurement.

In [23] a study based on measurement supplies a precise characterization of end-user hosts that participate in two popular p2p file sharing system like Napster and Gnutella. This characterization accounts the bottleneck bandwidth between these hosts, Internet at large, IP-level latencies to send packets to these hosts, how often hosts connect and disconnect from the system, how many files hosts share and download, and correlation between these aspects. The results of this study show that there is significant heterogeneity in peers' bandwidth, availability, transfer rates, and peer behaviors.

The work in [11] presents an analysis of a modern p2p multimedia file-sharing workload: a trace-based analysis of Kazaa is presented and then a model is derived based on this analysis. The results show that p2p file-sharing workloads are driven by considerably different processes than the Web. It is also demonstrated that there is significant locality in Kazaa workload, and therefore substantial opportunity for caching to reduce bandwidth consumption.

In [24] a systematic characterization of p2p traffic and its impact on the underlying network is performed; a novel approach for conducting large scale measure of p2p traffic for collecting data from multiple routers in a large ISP is presented. Three popular p2p system are analyzed: FastTrack, Gnutella, and DirectConnect. The results reveal significant skew in the distribution of traffic across IP addresses, subnets, and autonomous systems.

The work presented in [21] analyzes the topology graph of Gnutella and evaluate the generated network traffic. This analysis shows that Gnutella connectivity follows a structure that keeps the network as reliable as a pure power-law network when assuming random node failures.

A measurement study of FastTrack-based p2p systems is performed in [17]. In [4] a set of experiments has been conducted to analyze the behavior of free riders present in Gnutella system and its impact on performance.

A mathematical model is deployed in [9] to explore and illustrate fundamental performances issues of p2p file-sharing system. This model is applied in three different type of architecture (centralized indexing, distributed indexing with flooded queries, and distributed indexing with hashing directed queries), and it is used for analyzing important aspects regarding performance like system scaling, freeloaders, file popularity and availability.

In [14] a random-graph based model is introduced for studying the evolution of ad hoc p2p communities such as Gnutella or Freenet; this model is used for analyzing basic properties such as reachability from a given node in the network. The same model is used in conjunction with a simulation approach in [15], [16] for studying complex properties such as queueing behavior.

An other simulation approach is presented in [18] where is developed a tool that can simulate p2p networks on top of representative Internet topologies.

In [13] a framework for an extensible and scalable p2p simulation environment that can be built on top of existing packet-level network simulators is designed and developed.

In [8] an investigation that shares some of the assumptions and simplifications that we use in our paper is presented. In particular, the paper presents a study to quantify a user's performance as a function of the percentage of users that share their resources in a p2p file sharing system. The approach presented in [8] is based on a use of simulative results combined with some simple analytical considerations.

III. Modeling p2p Networks

In this paper we propose an analytic modeling technique for the analysis of p2p file sharing systems. The aim is to provide Quality of Service (QoS) user-perceived measures that are related to the transfer phase for a given resource (file); in particular, we provide a method for the estimation of the distribution of the transfer time, i.e., the phase starting after a successful search phase. The investigations presented in this paper can be considered complementary to those presented in [9], [14], i.e., these papers present two different modeling-based analysis of p2p networks that can be considered as examples of system-oriented performance analysis of p2p networks.

In general, for the resource transfer phase all p2p applications provide a list of peers holding a copy of the requested resource. In the following we denote the peer that requests the resource as the *client-like peer* and the peers holding a copy of the requested resource as the *server-like peers*. For each server-like peer the p2p applications also provide additional information such as bandwidths between the server-like peer and its ISP, number of client-like peers that are using this server-like peer, and other information that help the client-like peer to choose the server-like peer to download the resource.

In the following we discuss those parameters that influence the transfer phase duration that are taken into account by our modeling technique.

- *Resource characteristics*: the size of the resource to be downloaded has an obvious impact on the distribution of the transfer time. Measure-based analysis of p2p applications (see for instance [11], [22]) show that that there is a substantial difference in typical resource size between p2p and WWW traffic. The measures presented in these papers show three prominent regions: *small* resources,

Bandwidths	%
14.4 Kbps	4%
28.8 Kbps	1%
33.6 Kbps	1%
56 Kbps	23%
64 Kbps	3%
128 Kbps	3%
DSL	14%
Cable	44%
T1	5%
T3	2%

TABLE I

DISTRIBUTION OF THE USER BANDWIDTHS (FROM [23] VOLUNTARILY REPORTED BY NAPSTER USERS.

typically mp3 files, that are less than 10 MBytes, *medium-size* resources, 10 to 100 MBytes that correspond to small-medium video files, and *large-size* resources over 100 MBytes, that correspond to large video files.

It is interesting to observe that small p2p resources are three order of magnitude larger than an average Web resources, while large resources are five order of magnitude larger [22]. Another interesting observation that can be found in [11] is that the majority of requests (91%) are for resources smaller that 10 MBytes, while the majority of bytes transferred (65%) are due to the largest resources.

- *Bandwidth Characteristics of the Peers*: the rate at which files can be downloaded from a server-like peer depends on the bottleneck bandwidth between the client-like peer and the chosen server-like peer, the available bandwidth, and the latency along the path connecting the two peers. The connection bandwidth between peers and ISPs has a clear impact on the transfer phase. In [23] measured based results for the Napster and Gnutella p2p applications show that there is a significant amount of heterogeneity in bandwidth, latency, and other characteristics that vary several orders of magnitude across the peers of the system. Table I reports these measures for the distribution of downstream bottleneck bandwidth.
  Our investigations are not related with a specific p2p application but they can applied to different p2p file sharing system nevertheless we use results derived for Napster and Gnutella for setting some parameters in our models. This use of the results presented in [23] might be considered an improper extension but the bottleneck bandwidth results derived for Napster and Gnutella are quite similar even if the two p2p applications are based on different architectures. It is important to point out that the measures reported in Table I, that we use as basis for our experiments have been presented in 2002 and these measures represent a reasonable "picture" of the last-mile connections at that time. The trend is towards high-speed bandwidth connections and than the results presented in this paper should be considered as a sort of worst-case analysis.
- *Parallel/Partial Downloads*: the speed at which the client-like peer downloads the requested file also depends on the possibility that the file can be downloaded in pieces or "chunks" from several different server-like peer (for instance, Kazaa allows this possibility).
- *Current Bandwidth Peer State*: after the client-like peer chooses a server-like peer (or more server-like peers in case of parallel downloads) the resource bandwidth allocated to this (these) transfer(s) may change during the transfer phase. These bandwidth fluctuations are mainly due to the variation of the load on the chosen server-like peer. In general, p2p applications implement sharing bandwidth policies among the different client-like peers that download resource(s): in some case the server-like peer equally shares the available download bandwidth among the client-like peers, in other case the sharing policy depends on some parameters that account for the participation level the client-like peer. In particular, in Kazaa system the peer performance improves according the peer's behavior, i.e., this p2p application accounts the integrity level of the shared file, and the number of uploads. In other cases the server-like peer favors the client-like peers with longer on-line periods, etc. Furthermore, bandwidth fluctuations are also influenced by the maximum number of downloads operations allowed by the server-like peer.
- *Resource popularity*: the influence of this parameter on the transfer rate is quite clear. If the client-like peer is looking for a very popular resource, than the probability that a copy of this resource is held by a server-like peer with high speed connection bandwidth is higher than the the case of a search for a "rare" resource. On the other hand the probability that a server-like peer holding a very popular resource is overloaded (because there are many client-like peers that require its resource(s)) increases with the resource popularity.

IV. MODELING TECHNIQUE DESCRIPTION

The modeling technique we develop aims at the computation of the distribution of the file transfer for a client-like peer whose behavior is the following: after the search of a given resource (file), whose popularity dictates the number of server-like peers holding a copy, the client-like peer selects a server-like peer to transfer the resource and leaves the system as soon as the download terminates. The modeling technique does not take into account the search phase explicitly and assumes that the result of this search is the number of peers holding a copy of the resource. Therefore, our modeling technique neglects the quantitative aspects of the search of a resource that would require a separate analysis.

The modeling technique we develop takes into account several system parameters: we consider the client-like peer bandwidth (last mile), server-like peer outgoing bandwidth and maximum number of allowed concurrent downloads, the number of peers holding a given resource as a function of its popularity, the size of the resource to be downloaded, the competing load on the server-like peer that is experienced by a client-like peer during download, and the server-like peer selection policy when multiple peers hold a given resource.

Our modeling technique has been developed under several simplifying assumptions:

- The number of peers holding a copy of the requested resource is a function of its popularity p that we denote as $N(p)$. We view popularity as the fraction of peers holding a copy of the resource therefore, $0 < p \leq 1$. As a consequence, we assume that the higher the popularity of a resource the greater is the number of peers holding a copy. We assume that all the $N(p)$ server-like peers not only hold a copy of the resource but are available for download, i.e., a request for download is neither queued nor refused. Indeed, in our model we assume that the resource popularity p is an indirect input parameter, i.e., we give as input the number of server-like peers ($N(p)$) holding a copy of the resource requested by the client-like peer. In this manner we can perform experiments with different popularity parameter by varying the number of server-like peers holding the resource.
- We do not consider server-like peer availability issues, that is, we assume that the server-like peer is available for all the duration of the resource download time. This is a reasonable assumption in case of small resource downloads but becomes less reasonable in case large resource downloads.
- We model the activity of a client-like peer that is always on-line (we do not address availability issues for the client-like peer under investigation) and the session ends only when the resource has been completely downloaded.
- The time dynamic of a download is faster than the activity of the $N(p)$ server-like peers, i.e., the $N(p)$ server-like peers do not leave the system before the download terminates.
- The underlying IP network is never congested, i.e., network transfer times are dominated by first mile and last mile characteristics of the peers.
- The offered bandwidth of a server-like peer is equal to its first mile bandwidth, i.e., during the upload of the resource the server-like peer does not perform downloading. On the other hand, a client-like peer, during the transfer phase, dedicates all its bandwidth to the download of the resource, i.e., during the transfer phase the client-like peer does not allow uploads.
- The download of a resource is not split in parallel downloads of smaller chunks from different server-like peers.
- A server-like peer does not discriminate among clients, i.e., it equally shares its offered bandwidth.

We develop a hybrid modeling technique based on the combined use of Fluid Stochastic Petri Nets (FSPN), queueing systems, and combinatorial analysis. The joint use of different modeling paradigms allows to capture several different features of the system under investigation and their interactions, as well as to obtain an efficient model solution.

The basis of our approach is the development of a FSPN model representing a server-like peer serving the request of a particular client-like peer (the one whose file transfer distribution we are analyzing that we term as the *tagged client*); the FSPN model also represents the concurrent downloads interference by other client-like peers whose effect is to introduce fluctuations in the available bandwidth for the tagged client. The transient solution of the FSPN model will yield the distribution of the file transfer time for the tagged customer as a function of the following parameters: the client-like peer bandwidth (last mile), server-like peer outgoing bandwidth and maximum number of allowed concurrent downloads, the size of the resource to be downloaded, the competing load on the server-like peer that is experienced by a client-like peer during download.

In the following we denote as $F_t(t|B, i, s, b)$ the distribution of the transfer time for the tagged customer of a file of size s bytes, using a dedicated bandwidth of b Kbps, from a server-like peer whose offered bandwidth is equal to B Kbps where the initial number of competing clients when the tagged customer starts its download is equal to i. The FSPN model is described in Section IV-A.

In order to obtain the distribution of the transfer time for the tagged customer for a file of size s bytes, using a dedicated bandwidth of b Kbps regardless the initial number of competing clients when the tagged customer starts its download, we must correctly characterize the distribution $P(i|B)$, i.e., the probability that i client-like peers are simultaneously using a server-like peer that has a bandwidth equal to B when the tagged customer starts its download. To compute the probability distribution $P(i|B)$ we rely on an additional support model that is a Generalized Stochastic Petri Nets (GSPN) representation of a M/H/∞/K queue. This support model is described in Section IV-B

Furthermore, to obtain the distribution of the file transfer time regardless the server-like peer offered bandwidth, we must determine the probability $P(B|N(p))$, i.e., the probability that the tagged client finds the resource (whose popularity is p) on a peer whose offered bandwidth is B given that the number of server-like peers holding a copy of the resource is equal to $N(p)$. These probabilities depend on the selection policy the tagged client adopts to choose a server-like peer from which starting to download the requested resource. To compute these probabilities we derive a combinatorial expressions taking into account a particular selection policy. These derivations are described in Section IV-C.

We can then summarize the steps required to compute the distribution of the transfer time using the methodology we propose:

- compute $F_t(t|B, i, s, b)$ for each possible i and B for fixed s and b.
- compute $P(i|B)$ for each possible initial number of competing clients when the tagged customer starts its download given a server-like peer bandwidth equal to B;
- compute $P(B|N(p))$ for each possible value of B given the popularity of the requested resource (a fixed p) and

the selection policy;

We can then compute the cumulative distribution for the transfer time as

$$F_d(t|p,b,s) = \sum_i \sum_B F_t(t|B,i,s,b)P(i|B)P(B|N(p)).$$

The interaction among the different sub-models is illustrated by the block diagram presented in Figure 1

A. The FSPN model

The nature of peer-to-peer systems involves the simultaneous downloads from several client-like peers on the same server-like peer. To model this aspect we model the server-like peer as a finite capacity queue with infinite server policy. Using this particular representation, the load of the server-like peer is represented by the number of client-like peers that are downloading resources from the same server-like peer. The customers served by this queueing system represent client-like peers that are using this server-like peer while the tagged client is downloading its file.

We chose to represent the server-like peer activity as a M/H/$\infty/Max\_uploads - 1$ queue assuming that:

- the tagged client-like peer is being served (it is one among the maximum allowed concurrent downloads), therefore we set the queue capacity to $Max\_uploads - 1$. We set the maximum number of simultaneous downloads operations allowed by the server-like peer as function of its bandwidth according the following rule: Max_uploads=1 for 14.4 and 28.8 Kbps, Max_uploads=2 for 33.6 and 56 Kbps, Max_uploads=3 for 64 Kbps, Max_uploads=4 for 128 Kbps, Max_uploads=5 for DSL, Max_uploads=6 for Cable and T1, and Max_uploads=7 for T3.
- The service policy is infinite-server since we assume that the server-like peer available bandwidth is equally shared among the client-like peers that are downloading.
- The service time distribution is approximated by a two stage hyper-exponential distribution. The parameters of the hyper-exponential distribution have been chosen to match the 50th and 90th percentiles of the session length distribution reported in [11]. In particular, we set $\mu_1 = 0.001$, $\mu_2 = 0.1$, $\alpha_1 = 0.6$, and $\alpha_2 = 1 - \alpha_1 = 0.4$.
- The service requests to the server-like peer arrive according to a Poisson process whose rate depends on the server-like peer bandwidth. This assumption is derived from the results presented in [23]. In particular, we set the arrival rate equal to $\lambda(B) = \lambda w(B)$ as the product of the upload rate λ and a bandwidth dependent weight $w(B)$. The weight $w(B)$ has been determined referring to the measures reported in [23] regarding the number of uploads versus the bandwidth of the server peer.
  From these consideration we obtained the following distribution: w(modems, ISDN) $= 0.21$, w(dual ISDN, Cable, DSL) $= 0.72$, w(T1 + T3) $= 0.07$. The upload rate λ is derived from the results presented in [21], in particular we set $\lambda \approx 7$ queries per seconds.

The resulting M/H/$\infty/Max\_uploads - 1$ queue is represented by the FSPN model depicted in Figure 2. Timed transition A models the arrival of a client and its transition rate is equal to $\lambda(B)$. The sub-net composed by places ch, q_1, q_2 and e and transitions t_1, t_2, S_1, S_2 and t_3 models the two stage hyper-exponential service (exponential transitions S_1 and S_2).

The maximum number of downloads minus 1 is represented by the sum of the tokens in places c, q_1, and q_2. The number of client-like peers different from the tagged client-like peer that are downloading resources from the server-like peer is equal to the sum of tokens in places q_1, and q_2.

Fluid place x represents the bytes transferred by the tagged client. Transition D models the file transfer; its flow rate is a function of the number of client-like peers in the system and is defined as

$$f(\#q_1 + \#q_2) = \min\left(\frac{B}{\#q_1 + \#q_2 + 1}, b\right),$$

where B is the server-like bandwidth, and b is the client-like bandwidth.

The min function keeps track that the transfer rate is limited by the lowest bandwidth; in this way the actual flow rate depends on the client-like bandwidth (with the assumption that its entire bandwidth is available and dedicated to the file transfer) and on the available bandwidth of the server (which, in turn, depends on the instantaneous number of peers using the server).

Note that in the definition of f the tagged client is considered by adding one to the number of client-like peers in the system. In this way we consider only the first and the last mile bandwidths, neglecting the underlying network, i.e., we assume that the network is never the bottleneck during the file transfer.

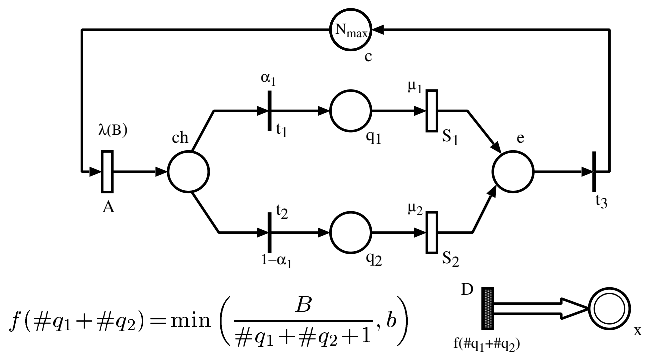

Fig. 2. FSPN model representation of a M/H/$\infty/Max\_uploads - 1$ queue for the computation of $F_t(t|B,i,s,b)$.

1) The analysis of the FSPN model: The FSPN represented in Figure 2 is analyzed using the techniques described in [10]. These techniques consider the discrete and the continuous part of the model separately. In particular the underlying Markov chain describing the discrete component of the model is obtained from the FSPN. Since the transition rates of the timed transitions that compose the model are constant, this underlying Markov chain can be characterized by a single

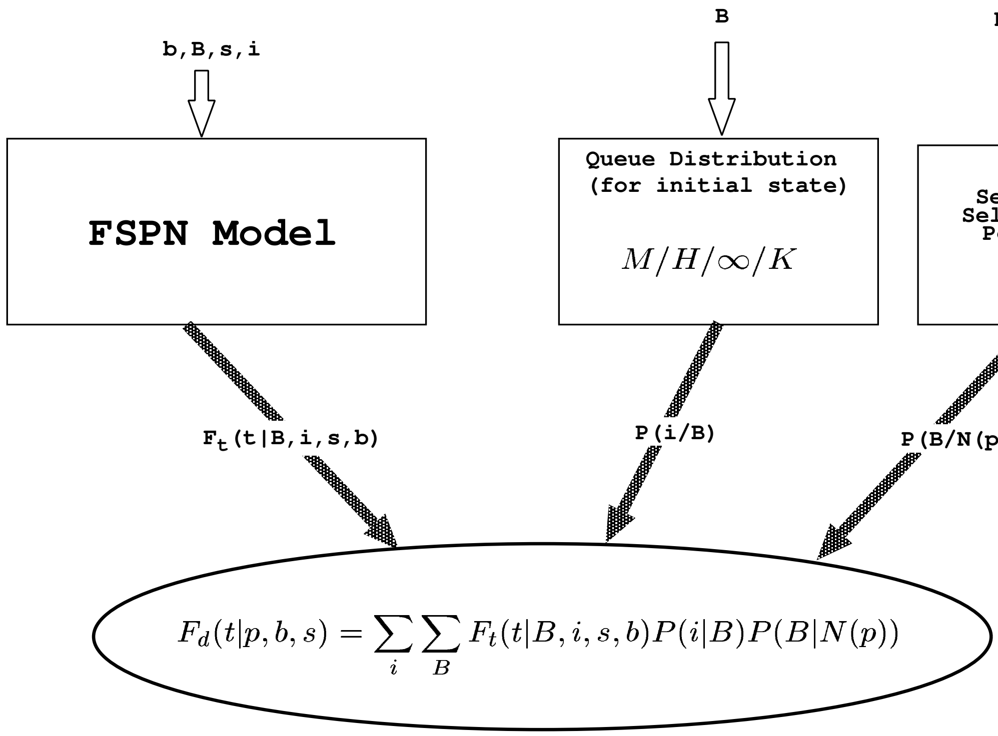

Fig. 1. Interaction among sub-models.

constant matrix $\boldsymbol{Q}$. The fluid interaction is taken into account in a diagonal matrix $\boldsymbol{R}$ whose elements represent the actual flow rate in each discrete state. Since discrete states correspond to different load configurations, each element r_j of matrix $\boldsymbol{R}$ represent the actual transfer rate with a particular load, that is: $r_j = f(q(j))$ where $q(j)$ corresponds to the load in discrete state j.

If we define with $\pi_j(\tau, x)$ the probability density of having x unit of fluid at time τ in discrete state j, i.e., the probability that x bytes of a resource have been download at time τ in state j, then according to the results presented in [10], we have that:

$$\frac{\partial \boldsymbol{\pi}(\tau, x)}{\partial \tau} + \frac{\partial \boldsymbol{\pi}(\tau, x)}{\partial x}\boldsymbol{R} = \boldsymbol{\pi}\boldsymbol{Q}, \tag{1}$$

where $\boldsymbol{\pi}(\tau, x)$ is a vector whose components correspond to $\pi_j(\tau, x)$. Since $r_j > 0$ for any state j, and since the fluid place is unbounded, we do not need any boundary condition while the initial condition is:

$$\boldsymbol{\pi}(0, x) = \delta(x)\boldsymbol{\pi}_0,$$

where $\boldsymbol{\pi}_0$ is the probability distribution of the initial state of the discrete part of the model.

When considering the complete server model, the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, and the vector $\boldsymbol{\pi}_0$ depend on the model parameters. In particular, matrix $\boldsymbol{Q}(B)$ depends only on the selected server bandwidth B, matrix $\boldsymbol{R}(B, b)$ depends on both the server and the client bandwidth, and the initial probability vector $\boldsymbol{\pi}_0(i)$ depends on the initial number of competing clients when the tagged customer starts its download (that we denoted as i).

With these assumption, we denote as $\boldsymbol{\pi}(\tau, x, B, b, i)$ the solution of Equation 1, for a given combination of parameters B, b and i. From this we can derive $F_t(t|B, i, s, b)$. The probability of having downloaded s bytes in less than t seconds is equal to the probability of having downloaded at least s bytes at time $\tau = t$, that is:

$$F_t(t|B, i, s, b) = \int_s^\infty \bar{\pi}(\tau, x, B, b, i)dx\bigg|_{\tau=t}, \tag{2}$$

where $\bar{\pi}(\tau, x, B, b, i) = \boldsymbol{\pi}(\tau, x, B, b, i)\mathbf{1}$, that is, $\bar{\pi}(\tau, x, B, b, i)$ is the probability density of the fluid level regardless of the discrete state, and $\mathbf{1}$ is a unit vector with a number of components equal to the number of discrete states of the model.

B. Computing $P(i|B)$

In Equation 2, $F_t(t|B, i, s, b)$ depends on the initial state of the server model, i.e., the initial number of competing clients when the tagged customer starts its download. This dependency is crucial because the initial state can have a significative impact on the overall download time distribution, especially when considering short files. For instance, consider the time required to download a 112 KByte JPEG image from a DSL server-like peer, using a DSL connection (that is $B = b = 640$ Kbps). Figure 3 represents the distribution of the transfer time as a function of the initial load of the server-like peer when the tagged client starts the file transfer. It is easily noted that the mean downloading time when no other peers are interfering with the file transfer is more than five times shorter than when there are 4 other peers accessing the server.

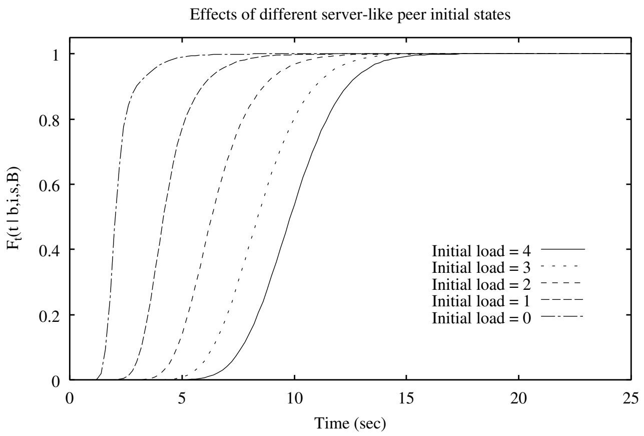

Fig. 3. Distribution of the transfer time as a function of the number of competing client-like peers when the tagged client starts the file transfer.

In order to obtain the distribution of the transfer time regardless the number of client-like peers accessing the server when the tagged client starts to download, we must correctly characterize the distribution $P(i|B)$. Our analysis is based on a GSPN model that is obtained by the FSPN model of Figure 2 by removing the fluid place and the fluid transition. For the computation of $P(i|B)$ we consider a queue with $n = Max\_upload(B)$ customers, i.e., this time we do not distinguish the tagged client from the other clients, and we approximate the whole system as a finite capacity queue. We compute the steady state distribution of this queue and use it to determine $P(i|B)$.

Let us denote $\hat{\pi}_n(i|B)$ the stationary distribution of the queue with $n = Max\_upload(B)$. When considering the complete model, the tagged client can be accepted only if there is at least a free position in the queue. We use this assumption to compute $P(i|B)$ by normalizing $\hat{\pi}_n(i|B)$, excluding the case where the server-like peer would reject the tagged client request (that is, when $i = Max\_upload(B)$). In this way we can compute:

$$P(i|B) = \frac{\hat{\pi}_n(i|B)}{\sum\limits_{j<Max\_upload(B)} \hat{\pi}_n(j|B)} \quad (3)$$

Note that this derivation has been possible thanks to the assumption that the tagged client-like peer can only choose a server-like peer with an available queue position.

C. Computing $P(B|N(p))$

To transfer a file a p2p user has to decide from which peer to get the resource according to different criteria. For instance, the selection could be based on the server-like peer with the fastest connection, the server-like peer with the lowest load, a random selection, etc.

The available bandwidth of the selected server-like peer plays an important role in the distribution of the transfer time, especially when the tagged client-like peer has a fast connection. To show how the performance can be affected by the server-like peer selection, we consider the behavior of different file downloads versus the selected server-like peer bandwidth. We perform two experiments by considering two different values for the tagged client-like peer bandwidths: modem 56 Kbps and DSL. For each value, we compute the distribution of the file transfer time for different values of the server-like peer bandwidths: 33 Kbps, 56 Kbps, DSL, Cable and T3.

Figure 4 shows the results of this investigation. In particular, when the client-like peer has a 56 Kbps bandwidth the distribution of the file transfer time is heavily conditioned by the bottleneck of the client-like peer connection. In fact, when the server-like peer uses a faster connection (DSL, Cable and T3) the performance cannot improve because the client-like peer bandwidth limits the flow of the data to be transferred. When the server-like peer uses the same (or lower) bandwidth (56 Kbps), the performance gets worse because the server-like peer becomes the bottleneck due to the simultaneous presence of other peers that waste his bandwidth. Even worse when the server-like peer has a 33 Kbps bandwidth.

In the case of a DSL bandwidth for the client-like peer, the performance are always influenced by the server-like peer bandwidth except for the T3 case. In this case, the client-like peer is the bottleneck since for any number of competing downloads the bandwidth the server-like peer can assign to the tagged client-like peer is always greater than the DSL bandwidth.

If the server-like peer has the Cable or DSL bandwidth instead, the role of bottleneck depends on the probability of having other peers that are downloading simultaneously from the same server-like peer. A high number of concurrent peers (limited to the maximum number of allowed uploads) means that most of the bandwidth is wasted, making the probability that the server-like peer becomes the bottleneck get higher.

In the case of lower bandwidths (56 Kbps or 33 Kbps) the server-like peer is always the bottleneck and the performance depends on its bandwidth.

We assume that after the research phase, the client-like peer selects the server-like peer with the largest bandwidth. To model this policy we refer to the bandwidth that Napster users voluntarily reported whose distribution is described in Table I, according to this distribution, we define a combinatorial manipulation that models the selection of the peer with the highest bandwidth. We assume that the number of peers holding a copy of the requested resource are a function of its popularity p and that the higher the popularity of a resource the greater is the number of peers holding a copy. Our model assumes that when the resource is available on only one server-like peer, the probability this peer has a given bandwidth is exactly the one reported in Table I. When the number of available resources is greater than one, the probability to select a peer with higher bandwidth grows according to the bandwidth distribution of the peers that are present in the system. The greater the number of available resources, the greater the probability to

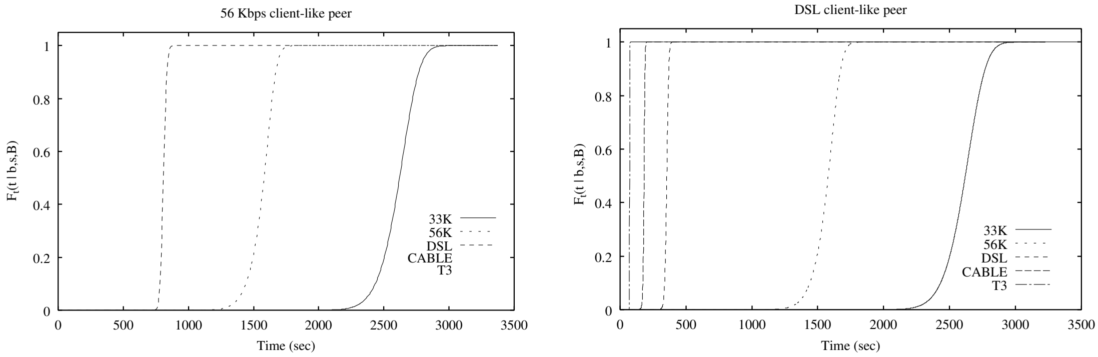

Fig. 4. The effect of server-like peer bandwidth on the distribution of the transfer time for a modem 56 Kbps client-like peer (left graph) and for a DSL client-like peer (right graph).

select a peer with the highest bandwidth. When the number of available resources is large enough, the probability of finding the resource on the highest bandwidth peers tends to 1. It is interesting to note that in the case of a resource available on only one peer, the selection reduces to a random choice policy where the probability distribution is the one reported in Table I.

The probability of finding a resource on a peer with a given bandwidth, using the fastest-connection policy, can be computed using the following recurrence relation:

$$\begin{aligned} P(B|N(p)) \;&=\; P(b_i \leq B|N(p)-1)P(B) + \\ &\quad +P(B, N(p)-1)P(b_i < B), \end{aligned} \qquad (4)$$

where

$$P(b_i \leq B|N(p)-1) \;=\begin{cases} \displaystyle\sum_{i|b_i \leq B} P(b_i|N(p)-1) & \text{if } N(p)>1, \\ 1 & \text{if } N(p)=1 \end{cases}$$

and

$$P(b_i < B) \;= \sum_{i|b_i<B} P(b_i).$$

The meaning of this formula is the following: the probability of selecting a peer with bandwidth B given that there are $N(p)$ available resources is equal to the probability to have $N(p)-1$ peers with bandwidth less than or equal to B and to find a peer with bandwidth B, or it is equal to the probability of finding a peer with bandwidth less than B but to have a peer with bandwidth B in the previous $N(p)-1$ resources.

Figure 5 depicts the probability $P(B/N(p))$ as function of B and $N(p)$. Each vertical slice parallel to the bandwidth axis of this 3-d plot identifies the probability distribution of selecting a given bandwidth B according to the fastest selection bandwidth policy. We note that the probability of selecting the server-like peer with fastest bandwidth increases with the number of server-like peers holding a copy of the resource.

V. Numerical Experiments

In this section we first discuss aspects related with the FSPN model solution (Section V-A) and then we present some numerical results that allows us to illustrate some of the potentialities of the proposed methodology.

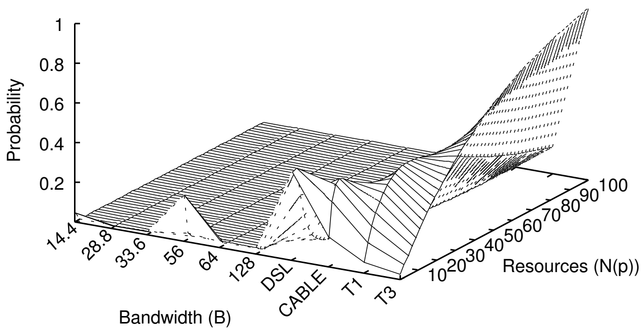

Fig. 5. $P(B/N(p))$ distribution as function of B, and $N(p)$.

Extensive validation of the model results for our modeling technique shares the same difficulty of previous work on analytical models for p2p systems [9], [14]. It is a difficult task since existing measurement studies on realistic file sharing p2p applications have not focused on characterizing the duration of the transfer phase. Although it might be possible to validate our model through detailed simulations of realistic p2p file sharing applications, the programming and computational cost would be prohibitive.

A. Model Solution

The most computationally expensive step of the proposed technique is the solution of the FSPN for the computation of $F_t(t|B,i,s,b)$. According the procedure defined in Section IV, we have to solve $\sum_{j \in B}(Max\_upload(j)-1)$ fluid models. We can reduce this number of transient solutions by using the following consideration: we observe that for a given bandwidth matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ do not depend on the initial number of competing clients when the tagged customer starts its download (see Section IV-A.1). In this manner we can solve, for each value of B, one FSPN model by using

$$\boldsymbol{\pi}_0 = \{P_n(0|B), P_n(1|B), \ldots\},$$

as initial distribution. In this manner we only have to solve $|B|$ FSPN models.

Client-like Bandwidth	# Avail. Resources	50%	90%
56 Kbps	1	13m 48s	24m 13s
DSL	1	3m 03s	24m 2s
T3	1	1m 31s	12m 30s
56 Kbps	30	13m 41s	14m 11s
DSL	30	1m 30s	3m 1s
T3	30	1m 29s	2m 57s
56 Kbps	90	13m 41s	14m 34s
DSL	90	1m 12s	2m 20s
T3	90	0m 45s	2m 16s

TABLE II

MODEL RESULTS FOR DIFFERENT CLIENT-LIKE PEER BANDWIDTHS AND DIFFERENT VALUES OF POPULARITY

The solution of the partial differential equations governing the fluid model is obtained by simple up-wind semi-discretization and Euler formula (see [10] for details). This technique requires two discretization processes: with respect to time (with step Δt), and with respect to fluid level (step Δx). The technique converges only if $\Delta t < \beta \Delta x$, where β is determined by the maximum flow rate. In our experiments we tuned Δx and Δt to ensure the solution algorithm convergency. One of the possible future improvements of this solution method could be derivation of the discretization steps according the system input parameters (i.e., p, b, s).

All the experiments have been performed by using a Pentium IV (2.4 Ghz) computer, with 1.5 GB of RAM, under Linux OS. In all the cases the model solution (steps presented in Section IV) required few teens of minutes (from 10 min to 60 min).

B. Numerical Results

We perform a "qualitative" validation by comparing the model results in particular cases: for instance, we compare the model results with the ideal case where there is no competition for the server-like peer bandwidth and the transfer is only conditioned by the minimum bandwidth between server-like and client-like peers.
We also verified that the model results obtained for the transfer of typical MP3 file agree with the common experience of p2p users. We fix a maximum file size to 4 MByte, typical size of a MP3 file, and consider three different values for the popularity of the requested file (1, 30, and 90 server-like peers are available to supply the file) and then we compute the distribution of the transfer time for three different client-like peer bandwidths: modem 56 Kbps, DSL (with 640 Kbps), T3 ($6,000$ Kbps). Table II and Figure 6 summarize the results we derive: the table reports the 50% and the 90% quantiles, while the figure depicts the cumulative distributions of the transfer time.

In the first case we consider a rare file, i.e., there is only one server-like peer having the resources available. This case corresponds to a random choice for the bandwidth of the server-like peer, the probability of finding a given server-like peer bandwidth is the one presented in Table I. The second set of experiments corresponds to the case of a file with higher popularity (30 server-like peers have the resource). In this case, the choice for the bandwidth of the server-like peer is more addressed towards faster bandwidths, according to the distribution obtained by using the recurrence Equation 4. Finally we considered a file with a even higher popularity (90 server-like peers have the resource). In this case, according to the selection criterion, the probability that the client-like peer chooses the server-like peer having the fastest bandwidth is rather high (almost equal to 1) and then, as we can see from the right plot of Figure 6 the resource transfer time is mainly dominated by the client-like peer bandwidth.

If we look at these results in terms of number of server-like peers holding the resource, we can see that this number modulates the probability of choosing a server-like peer with higher available bandwidth and hence the quantiles consequently decrease.

VI. CONCLUSIONS AND FURTHER DEVELOPMENTS

In this paper we propose an analytical modeling technique for the evaluation of transfer times in file sharing p2p applications. The technique we present allows us to provide QoS user-perceived measures related to the transfer phase for a particular resource, i.e., the cumulative distribution of the transfer time.

The technique is based on the combined use of a FSPN model, queueing system, and combinatorial manipulations allowing the derivation of the distribution of the file transfer time for p2p file sharing applications. Our approach is general and flexible and allows to characterize different p2p systems and peer characteristics. We have numerically analyzed a few different scenarios obtaining preliminary results on the effects of different system parameters on the distribution of the file transfer time.

Although the technique has been developed under several simplifying assumptions the results we derive for the analyzed scenarios highlight interesting issues that involve the relations among the parameters governing the duration of the transfer phase.

Several different extensions are currently underway. These extensions and improvements can be classified into three different categories:

1) extensions to remove some of the simplifying assumptions in order to obtain a modeling technique more adherent to the studied systems;
2) developments of solution techniques to improve the efficiency and/or the accuracy of the numerical solution of the FSPN model;
3) modeling of more complex p2p file sharing issues as well as extensions of our technique to address additional p2p based applications besides the classical file sharing.

For the first category we are currently working

- to obtain a more sophisticated representation of issues related with the popularity of a given resource. In fact, in this paper we consider the number of peers holding a copy of the requested resource as an input parameter $N(p)$ that can be arbitrarily varied to perform a what-if analysis of the transfer times distribution. We are

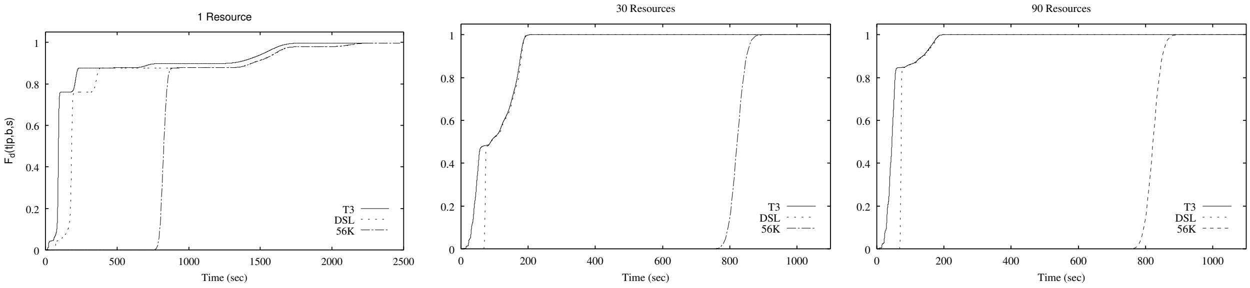

Fig. 6. Cumulative distributions of the transfer time for different client-like peer bandwidths and different values of popularity (1 resource left plot, 30 resources middle plot, and 90 resources right plot.

currently investigating the use of epidemic spreading of diseases in complex networks to obtain an estimation of the final number of peers holding a copy of a resource as a function of the content popularity as well as of topological properties of the network of peers [19], [20].

- to capture peer availability issues: in particular, we will try to model peers (server-like as well as client-like peers) that exhibit transient behaviors. In this case we are also interested in modeling transfers that can be interrupted on a server-like peer and that can continue on a different server-like peer;
- to represent more detailed peer behaviors: in this case we are trying to model server-like peers that may also behave as a client-peer during the transfer phase. In this manner we can have a more realistic representation of the bandwidth sharing among downloads and uploads. The same effort is devoted to the representation of the client-like peer behaviors;
- to account for p2p applications that allow parallel downloads of smaller resource chunks from different server-like peers;
- to model of selection strategies different from the criterion of choosing the server-like peer having the largest bandwidth connections, e.g., selection criteria based on the load of the server-like peer.

In the second category of extensions, fall all the improvements related with the transient solution of the FSPN model. In particular, we are investigating the use of different transient solution methods, e.g., [25].

The last category of extensions includes the use of the modeling technique to derive an optimization model that can be employed for designing and evaluating strategies to incentive cooperation in p2p file sharing systems. In particular, we are deriving an optimization technique that allows to evaluate server-like peer bandwidth sharing policies to incentive the peers' cooperation. These policies should account system parameters such as bandwidth characteristics of the peers, bandwidth maximum number of allowed uploads, peer behaviors, server-like peer bandwidth sharing policy. Furthermore we are extending our technique to address additional p2p based applications besides the classical file sharing. Our technique allows the derivation of QoS user-perceived measures and hence it would be interesting to apply and/or extend it to model p2p applications where the QoS in transfer phase is a stronger requirement. In particular, we are addressing streaming application on p2p architectures (see for instance [7], [12]), and p2p based information management in vehicular traffic applications [26].

References

[1] Napster protocol specification. Technical report, http://opennap.sourceforge.net/napster.txt, Mar 2001.

[2] WinMx. Homepage http://www.winmx.com. Technical report, Oct 2002.

[3] Kazaa. Homepage http://www.kazaa.com. Technical report, Jul 2003.

[4] E. Adar and B. Huberman. Free Riding in Gnutella. *First Monday*, 5(10), Oct 2000. http://www.firstmonday.dk/issues/issue5_10/adar/.

[5] I. Clarke, B. Sandberg, B. Wiley, and T. Hong. Frenet: A distributed anonymous infornation storage and retrival system. In *Proc. of the ICSI Workshop on Design Issue in Anonymity and Unobservability*, Berkeley, CA, USA, 2000.

[6] Clip2. The gnutella protocol specification v0.4. Technical report, http://www.clip2.com/GnutellaProtocol0.4.pdf, 2000.

[7] H. Deshpande, M. Bawa, and H. Garcia-Molina. Streaming live media over a peer-to-peer network. Technical report, CS Stanford University, 2002. Submitted for publication.

[8] M. Feldman, L. Kevin, J. Chuang, and I. Stoica. Quantifying Disincentives in Peer-to-Peer Networks. In *Proc. of the first workshop on economics of p2p systems*, Berkeley, CA, USA, Jun 2003.

[9] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley. Modeling Peer-Peer File Sharing System. In *Proceedings of INFOCOM 2003*, Santa Fe, NM, USA, Oct 2003.

[10] M. Gribaudo, M. Sereno, A. Bobbio, and A. Horvath. Fluid Stochastic Petri Nets augmented with Flush-out arcs: Modelling and Analysis. *Discrete Event Dynamic Systems*, 11(1 & 2), 2001.

[11] P. K. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. In *In Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19)*, Bolton Landing, NY, USA, Oct 2003.

[12] Y. Guo, K. Suh, J. Kurose, and D. Towsley. A Peer-to-Peer On-Demand Streaming Service and Its Performance Evaluation. In *Proc. of 2003 IEEE International Conference on Multimedia & Expo (ICME 2003)*, Baltimora, MD, USA, Jul 2003.

[13] Q. He, M. Ammar, G. Riley, H. Raj, and R. Fujimoto. Mapping Peer Behavior to Packet-level Details: A Framework for Packet-level Simulation of Peer-to-Peer Systems. In *In Proceeding of 11th IEEE/ACM*

International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications System (MASCOT 2003), Orlando, Florida, USA, Oct 2003.

[14] K. Kant. An Analytic Model for Peer to Peer File Sharing Networks. In *In proceedings International Communications Conference*, Anchorage, AL, USA, May 2003.

[15] K. Kant and R. Iyer. Modeling and simulation of ad-hoc/P2P file-sharing networks. In *13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation (tool presentation)*, Sep 2003.

[16] K. Kant, R. Iyer, and V. Tewari. A performance model for peer to peer file-sharing services. In *WWW-11, Poster Session*, May 2002.

[17] N. Lleibowitz, A. Bergman, R. Ben Shaul, and A. Shavit. Are file swapping networks cacheable? Characterizing P2P traffic. In *In Proceeding of 7-st Int. WWW Caching Workshop*, Aug 2002.

[18] S. Merugu, S. Srinivasan, and E. Zegura. p-sim: A Simulator for Peer-to-Peer Networks. In *In Proceeding of 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications System (MASCOT 2003)*, Orlando, Florida, USA, Oct 2003.

[19] Y. Moreno, R. Pastor-Satorras, and A Vespignani. Epidemic outbreaks in complex heterogenous networks. *European Physical Journal B*, 26:521–529, 2002.

[20] M. E. J. Newman. The spread of epidemic disease on networks. *Physical Review E, 016128*, 66, 2002.

[21] M. Ripeanu. Peer-to-Peer Architecture Case Study: Gnutella Network. In *In proceedings of IEEE 1st International Conference on Peer-to-peer Computing (P2P2001)*, Linkoping Sweden, Aug 2001.

[22] S. Saroiu, P. K. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An Analysis of Internet Content Delivery Systems. In *In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA, USA, Dec 2002.

[23] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A Mesaurement Study of Peer-to-Peer File Sharing System. In *In Proceedings Of Multimedia Computing and Networking (MMCN) 2002*, Jan 2002.

[24] S. Sen and J. Wang. Analyzing peer-to-peer traffic across large networks. In *In Proceedings of the Second SIGCOMM Internet Measurement Workshop (IMW 2002)*, Marseille, France, Nov 2002.

[25] B. Sericola. Transient Analysis of Stochastic Fluid Models. *Performance Evaluation*, 32(4):245–263, 1998.

[26] J. Tian and Rothermel K. Building Large Peer-to-Peer Systems in Highly Mobile Ad Hoc Networks: New Challenges? Technical report, University of Stuttgart, Germany, 2002.